



**Fakultät für Ingenieurwissenschaften**

Studiengang

Angewandte Forschung und Entwicklung

**Geregelter Feder-Masse-Schwinger:  
Technische Aufrüstung  
und Verbesserung der Störkompensation**

Projektarbeit I

von

**Eduard Ober**

Datum der Abgabe: 09.03.2024

Erstprüfer: Prof. Dr.-Ing. Peter Zentgraf

## Kurzfassung

Ein Feder-Masse-Schwinger, ein fundamentales mechanisches System, ist aufgrund seines nützlichen theoretischen Verhaltens von großem Interesse. Diese Projektarbeit konzentriert sich auf die technische Erweiterung eines geregelten Feder-Masse-Schwinger-Aufbaus und der Implementierung einer geeigneten Regelkreisstruktur, mit der Störungen durch manuelle Auslenkung besser kompensiert werden können.

Ein zentraler Aspekt ist die genaue Bestimmung der Kraft, die der Propeller als Aktor auf die Masse ausübt. Es wurde ein Drehzahlsensor implementiert, mit der die Drehzahl des bürstenlosen Gleichstrom-Motors gemessen und die für die Aktorik wichtige Stellkraft berechnet werden kann. Die Integration des Drehzahlsensors bietet eine präzisere Erfassung des Systemverhaltens. Was das Bilden eines besseren Streckenmodells und auch die Umsetzung eines geeigneten linearen Zustandsreglers ermöglicht. Zusätzlich wurde ein Drehzahlregler implementiert und eine präzisere Vorgabe der Stellkraft von der Aktorik gewährleistet. Der in dieser Projektarbeit entworfene Zustandsregler verändert, das dynamische Verhalten des Systems, was sich durch eine erhöhte Dämpfung und damit einer verbesserten Störkompensation äußert. Die Effektivität des gebildeten Zustandsreglers wird durch Vergleich mit anderen Regelkreisstrukturen herausgestellt.

Es wurde eine Fernbedienung implementiert, um den Aufbau als Stand-Alone-Variante zu betreiben. Diese Fernbedienung ermöglicht es, die Regelung aus der Ferne zu steuern und die Elektronik sicher herunterzufahren.

**Schlagwörter:** Geregelter Feder-Masse-Schwinger, Zustandsregelung, Mobiler Demonstrationsversuch, pzMove, MATLAB/ Simulink, Raspberry Pi Zero, Bürstenloser Gleichstrommotor, Störkompensation

## Abstract

A spring-mass oscillator, a fundamental mechanical system, is of great interest due to its useful theoretical behavior. This project focuses on the technical enhancement of a controlled spring-mass oscillator setup and the implementation of an appropriate control loop structure to better compensate for disturbances caused by manual displacement.

A central aspect is the precise determination of the force exerted by the propeller as an actuator on the mass. A speed sensor has been implemented to measure the speed of the brushless DC motor and calculate the force crucial for the actuation. The integration of the speed sensor provides a more accurate capture of the system's behavior, enabling the creation of a better plant model and the implementation of a suitable linear state controller. Additionally, a speed controller has been implemented, ensuring a more precise specification of the actuation force. The state controller designed in this project alters the dynamic behavior of the system, resulting in increased damping and thus improved disturbance compensation. The effectiveness of the formed state controller is highlighted through comparison with other control loop structures.

A remote control has been implemented to operate the setup as a standalone version. This remote control allows for remote control of the regulation and safe shutdown of the electronics.

**Keywords:** Controlled spring-mass oscillator, State Feedback Controller, Mobile Demonstration experiment, pzMove, MATLAB/ Simulink, Raspberry Pi Zero, Brushless DC motor, Disturbance Compensation

# Inhaltsverzeichnis

<b>Kurzfassung.....</b>	<b>II</b>
<b>Abstract.....</b>	<b>III</b>
<b>Inhaltsverzeichnis .....</b>	<b>IV</b>
<b>Abbildungsverzeichnis.....</b>	<b>VI</b>
<b>Abkürzungsverzeichnis .....</b>	<b>VIII</b>
<b>1 Einleitung .....</b>	<b>1</b>
1.1 Motivation .....	1
1.2 Zielsetzung und Randbedingungen .....	1
<b>2 Stand des Wissens und der Technik .....</b>	<b>3</b>
2.1 Beschreibung des Aufbaus .....	3
2.1.1 Die Masse .....	3
2.1.2 Der Electronic Speed Controller des bürstenlosen Gleichstrommotors .....	3
2.1.3 Der Positionssensor: .....	3
2.1.4 Schlittenführung .....	4
2.1.5 Plexiglasschutz .....	4
2.2 Auflistung bisheriger Ergebnisse .....	5
2.2.1 Verwendete Software .....	5
2.2.2 Fernbedienung .....	5
2.2.3 Erstellung des Streckenmodells.....	5
2.2.4 Erstellter Regler.....	6
2.2.5 Reaktion des Systems auf die Auslenkung von Hand.....	6
2.3 Für die Arbeit anzuwendendes Wissen .....	8
2.3.1 Überhitzungsschutz des ESC.....	8
2.3.2 Armierung des ESC.....	8
2.3.3 Berechnung der Varianz.....	9
2.3.4 Bestimmung der Propellerstellkraft.....	10
2.3.5 Software für Regelungstechnik pzMove .....	10
2.3.6 Bestimmung eines Zustandsreglers .....	10
2.3.7 Bestimmung eines Beobachters.....	11
<b>3 Durchführung und eigene Untersuchungen.....</b>	<b>12</b>
3.1 Technische Aufrüstung.....	12
3.1.1 Implementieren der Fernbedienung.....	12
3.1.2 Implementieren des Drehzahlsensors .....	14
3.2 Implementieren eines Drehzahlreglers .....	15
3.2.1 Systemantworten zur Identifikation des Motordrehzahlverhaltens aufnehmen ..	16

---

3.2.2	Identifizieren des Streckenmodells.....	19
3.2.3	Auslegung des Motordrehzahlreglers.....	19
3.2.4	Simulation des Motordrehzahlreglers.....	20
3.2.5	Implementierung des realen Motordrehzahlreglers.....	21
3.2.6	Ermitteln der tatsächlichen Stellkraft aus der Motordrehzahl.....	21
3.3	Auslegung eines neuen Positionsreglers .....	22
3.3.1	Mögliche Erklärungsansätze für die schlechte Störkompensation.....	22
3.3.2	Physikalische Herleitung des Feder-Massen-Schwinger-Modells .....	23
3.3.3	Untersuchen des Feder-Massen-Schwinger-Systemverhaltens.....	24
3.3.4	Umsetzen eines digitalen Zwillings der Strecke .....	25
3.3.5	Umsetzen des gewünschten Regelkreisverhaltens durch Polzuweisung.....	26
3.4	Austesten der verschiedenen Regler hinsichtlich ihres Führungs- und Störkompensationsverhaltens .....	27
3.5	Weitere Untersuchungen .....	28
3.5.1	Der Einfluss des Plexiglasschutzes .....	28
3.5.2	Überprüfen der Stellkraft abhängig von der Position.....	30
<b>4</b>	<b>Ergebnis der technischen Aufrüstung -Fernbedienung.....</b>	<b>32</b>
<b>5</b>	<b>Ergebnis der technischen Aufrüstung - Motordrehzahlregelung.....</b>	<b>35</b>
5.1.1	Der implementierte Drehzahlsensor.....	35
5.1.2	Identifizieren des Motordrehzahlmodells.....	36
5.1.3	Verifizieren des Motordrehzahlmodells .....	36
5.1.4	Auslegen und Testen des Motordrehzahlreglers bei verschiedenen Sollwertvorgaben.....	39
5.1.5	Bestimmen der Look-Up-Table für die Berechnung der Stellkraft aus der Drehzahl .....	41
<b>6</b>	<b>Ergebnisse der Verbesserung der Störkompensation.....</b>	<b>43</b>
6.1.1	Identifizieren des Feder-Masse-Schwinger-Modells.....	43
6.1.2	Verifizieren des Feder-Masse-Schwinger-Modells.....	43
6.1.3	Auslegen eines dynamisch langsameren PID-Reglers .....	46
6.1.4	Auslegen des Zustandsreglers .....	46
6.1.5	Testen der verschiedenen Positionsregler auf Führungsverhalten .....	48
6.1.6	Testen der verschiedenen Positionsregler auf Störkompensation .....	52
<b>7</b>	<b>Fazit und Ausblick.....</b>	<b>56</b>
	<b>Literaturverzeichnis .....</b>	<b>58</b>
	<b>Eidesstattliche Versicherung .....</b>	<b>60</b>
	Auszug aus dem Strafgesetzbuch (StGB).....	60

## Abbildungsverzeichnis

Abbildung 1: Geregelter Feder-Masse-Schwinger – Versuchsaufbau [1, Seite 26].....	4
Abbildung 2: Erster Prototyp der Fernbedienung [1, Seite 54] .....	5
Abbildung 3: Reaktion des Systems auf die Auslenkung von Hand .....	6
Abbildung 4: Instabile Reaktion des geregelten Systems auf eine große Auslenkung ....	7
Abbildung 5: Reaktion des unregulierten Systems auf eine große Auslenkung .....	8
Abbildung 6: Messrauschen im Positionssignal um die 0 cm-Marke herum .....	9
Abbildung 7: Arduino Nano IoT 33 [5].....	12
Abbildung 8: OLED-Display mit I2C-Schnittstelle .....	13
Abbildung 9: UDP Receive -Block aus Simulink .....	13
Abbildung 10: Verdrahtungsbild für den RPM-Sensor [11] .....	14
Abbildung 11: Richtungsabhängiges Motordrehzahlverhalten .....	16
Abbildung 12: Drehzahl-Zeit-Verlauf einer sehr langsam ansteigenden Drehzahl- Rampe .....	17
Abbildung 13: Drehzahl-Tastgrad-Kennlinie .....	18
Abbildung 14: Regelkreisstruktur zur Regelung der Motordrehzahl .....	19
Abbildung 15: Simulink-Modell - BLDCMotorControlLoop_Sim.slx .....	20
Abbildung 16: Simulink-Modell - BLDCMotorControlLoop.slx .....	21
Abbildung 17: Simulink-Modell - OL_Test_w_m.slx .....	25
Abbildung 18: Kaskadenregelkreisstruktur .....	26
Abbildung 19: In Simulink umgesetzter Beobachter.....	27
Abbildung 20: Beispielhafter Verlauf der Propellerstellkraft durch die Auslenksimulation .....	28
Abbildung 21: Stufenweiser Anstieg der Propellerstellkraft.....	29
Abbildung 22: Auf 0 cm-normierte Positionswerte bei unterschiedlichen Gewichtskräften .....	30
Abbildung 23: Vollständige Fernbedienung zusammengebaut .....	32
Abbildung 24: Schaltplan für Fernbedienung.....	32
Abbildung 25: Display der Fernbedienung.....	33
Abbildung 26: Simulink-Modell - Auswertung UDP-Daten.....	33
Abbildung 27: Simulink-Modell – Initialisierungsblock für Fernbedienung.....	34
Abbildung 28: Drehzahlsensor zusammengebaut.....	35
Abbildung 29: Simulink-Modell - Drehzahl-Auswertung.....	35
Abbildung 30: Sprungantwort der Motordrehzahl.....	36
Abbildung 31: Systemantwort der Motordrehzahl bei einem sinusförmigen Eingangssignal .....	37
Abbildung 32: Systemantwort der Motordrehzahl, bei einer langsamen Eingangsrampe.....	37
Abbildung 33: Drehzahlregler mit einem Sprung hin und zurück für die Sollwertvorgabe .....	39
Abbildung 34: Drehzahlregler mit einer Rampe hin und zurück für die Sollwertvorgabe .....	40
Abbildung 35: Drehzahlregler mit einer sinusförmigen Sollwertvorgabe.....	40

---

Abbildung 36: Look-Up-Table - Stellkraft zu Motordrehzahl .....	41
Abbildung 37: Look-Up-Table - Motordrehzahl zu Stellkraft .....	42
Abbildung 38: Grundlegende Beschreibung des Feder-Masse-Schwinger-Systems .....	43
Abbildung 39: Sprungantwort des Feder-Masse-Schwingers .....	44
Abbildung 40: Rampe hin und zurück -Systemantwort des Feder-Masse-Schwingers .....	45
Abbildung 41: Sinus -Systemantwort des Feder-Masse-Schwingers .....	45
Abbildung 42: Simulink-Modell - SS_ControlLoopComplete.slx .....	47
Abbildung 43: Führungsverhalten des offenen Regelkreises (OL) .....	49
Abbildung 44: Führungsverhalten des Reglers 1 (R1) .....	49
Abbildung 45: Führungsverhalten des Reglers 2 (R2) .....	50
Abbildung 46: Führungsverhalten des Zustandsreglers (SS) .....	50
Abbildung 47: Bodediagramm von OL, R1, R2, SS .....	51
Abbildung 48: Störgrößenkompensation des offenen Regelkreises (OL) .....	53
Abbildung 49: Störgrößenkompensation des Reglers 1 (R1) .....	53
Abbildung 50: Störgrößenkompensation des Reglers 2 (R2) .....	54
Abbildung 51: Störgrößenkompensation des Zustandsreglers (SS) .....	54

## Abkürzungsverzeichnis

LiPo-Akku	Lithium-Polymer-Akku
PWM	Pulsweitenmodulation
BLDC-Motor	Bürstenloser Gleichstrommotor
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 (Revolutions per minute)
BLDC-Motor	Bürstenloser Gleichstrommotor
PLA	Polylactid
UDP	User Datagram Protocol



# 1 Einleitung

## 1.1 Motivation

Die initiale Version eines geregelten Feder-Masse-Schwingers wurde im Rahmen der Bachelorarbeit „Entwurf, Simulation und Bau eines geregelten Feder-Masse-Schwingers“ entworfen und implementiert [1]. Dabei wurde ein Regelkreis zur Regelung der Position der Masse implementiert.

Ziel des Aufbaus ist das Erfahren der Regelung mit allen Sinnen [1, Seite 1]. Dazu gehört das haptische Erfahren des Versuchsaufbaus und dies muss auf sichere Weise geschehen.

In der Bachelorarbeit wurde der Versuch unternommen, eine Fernbedienung zu entwickeln, mit der entweder der Sollwert oder die Stellgröße des Systems mithilfe eines Schiebepotentiometers vorgegeben werden sollte. Dieser Ansatz konnte jedoch nicht erfolgreich abgeschlossen werden, da keine Lösung gefunden werden konnte, um eine Verbindung zwischen der Fernbedienung und dem Mikrocontroller herzustellen [1, Seite 53].

Wie in Kapitel 2.2.5 gezeigt wird, zeigt die Regelung eine schlechte Kompensation der Auslenkung durch Hand. Das geregelte System verhielt sich in dieser Hinsicht schlechter als das unregelte System. Dies hat auch zu Folge, dass das System instabil wird und der Aufbau beschädigt wird.

Basierend auf den Aspekten der technischen Ausrüstung und der Störkompensation soll in dieser Projektarbeit eine Verbesserung erarbeitet werden.

## 1.2 Zielsetzung und Randbedingungen

Das Hauptziel dieser Arbeit ist es, den Feder-Masse-Schwinger technisch durch die Integration einer Fernbedienung zu erweitern und gleichzeitig die Störkompensation zu verbessern.

Die Fernbedienung wird auf Basis des vorhandenen Designs um neue Hardware erweitert, um drahtlose Kommunikation zwischen Fernbedienung und Versuchsaufbau zu ermöglichen. Dabei müssen sowohl Software-Erweiterungen am Mikrocontroller als auch an der Hardware der Fernbedienung berücksichtigt werden.

Ziel ist es auch, mögliche Probleme bei der Kommunikation zwischen Fernbedienung und Mikrocontroller zu dokumentieren und bestmöglich zu lösen. Die Fernbedienung soll die Vorgabe von Werten wie Stellgröße und Sollwert ermöglichen sowie ein sicheres Ausschalten aus der Entfernung ermöglichen.

Die Verbesserung der Störkompensation erfordert die Analyse über das Regelkreisverhalten. Es sollen verschiedene Regelkreisstrukturen getestet werden, um die Effizienz der Kompensation zu maximieren. Zur Umsetzung einer besseren Störkompensation wird ein Drehzahlsensor integriert, um die tatsächliche Propellerdrehzahl zu erfassen. Diese technische Erweiterung des Aufbaus ist erforderlich, um präzise Daten für die Regelung zu liefern.

Zusätzlich zu diesen Hauptzielen sollen weitere Untersuchungen durchgeführt werden, um negative Einflüsse auf den Aufbau zu identifizieren und zu bewerten.

Erstens wird untersucht, wie und ob sich Turbulenzen auf das Systemverhalten auswirken und wie diese Effekte minimiert werden können. Zweitens wird nach Nichtlinearitäten im Regelkreis gesucht, die unerwünschte Effekte verursachen könnten, und es wird untersucht, wie diese Einflüsse gemildert werden können. Dies kann auch technische Erweiterungen des Systems selbst umfassen, die das Systemverhalten beeinflussen können.

## 2 Stand des Wissens und der Technik

### 2.1 Beschreibung des Aufbaus

#### 2.1.1 Die Masse

Der Versuchsaufbau besteht aus einem 3D-gedrucktem aus Polylactid (PLA) Schlitten, an dem ein BLDC-Motor an der Unterseite fest geschraubt ist. Dieser Schlitten fungiert als Masse des Feder-Masse-Schwingers. Der bürstenlose Gleichstrommotor (BLDC-Motor) erzeugt mithilfe seines Propellers eine Kraft, die den Schlitten nach oben drückt. Im Inneren des Schlittens befinden sich ein Raspberry Pi Zero W, ein ESC und ein Lithium-Polymer-Akku (LiPo-Akku). Der Schlitten wird durch Zugfedern am Rahmen aufgehängt, die an zwei Haken an der Oberseite vom Holzrahmen befestigt sind.

#### 2.1.2 Der Electronic Speed Controller des bürstenlosen Gleichstrommotors

Der Electronic Speed Controller (ESC) „EMAX BLHeli -12A“ [2] steuert die Drehzahl des Motors und versorgt den Mikrocontroller, den Raspberry Pi Zero W, mit den benötigten 5 Volt mithilfe eines Battery Eliminator Circuit (BEC). Der ESC selbst wird von einem 3-Zellen-LiPo-Akku gespeist.

Der Electronic Speed Controller (ESC) dient der Drehzahlregelung des Motors. Dieser ermöglicht die Vorgabe der Motordrehzahl durch den Raspberry Pi Zero W. Dabei wird der ESC durch PWM angesteuert. Das verwendete PWM-Signal hat eine Frequenz von 500 Hz, wobei der ESC Pulsweiten zwischen 1 ... 2ms verwendet. Dies entspricht einem Tastgrad von 50 ... 100%. Der verwendete bürstenlose Gleichstrommotor Graupner Ultra 2408 2300KV [14] besitzt drei Phasen, die abwechselnd 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 [18, Seite 110].

Die Parameter des Emax 12a ESC können mithilfe von PWM programmiert werden, wobei die Standardwerte beibehalten werden.

#### 2.1.3 Der Positionssensor:

Auf dem Schlitten ist ein Laser-Abstandssensor angebracht [1, Seite 7], der den Abstand zur Oberseite des Rahmengestells des Feder-Masse-Schwinger-Aufbaus misst.

Der Sensor hat verschiedene Abtastmodi und wird im Default-Modus betrieben. Dies entspricht einer Abtastrate von 30 Hz, wobei nach einem Positionswert alle 0.03 Sekunden abgefragt wird. Dies bedeutet, es wird tatsächlich mit einer Rate von  $33.\bar{3}$  Hz abgetastet.

### 2.1.4 Schlittenführung

Der Schlitten wird entlang zweier Führungsrundstäbe geführt. Diese Führungen wurden mit einem gewissen Bewegungsspiel konstruiert, um ein Verklemmen des Schlittens zu verhindern.

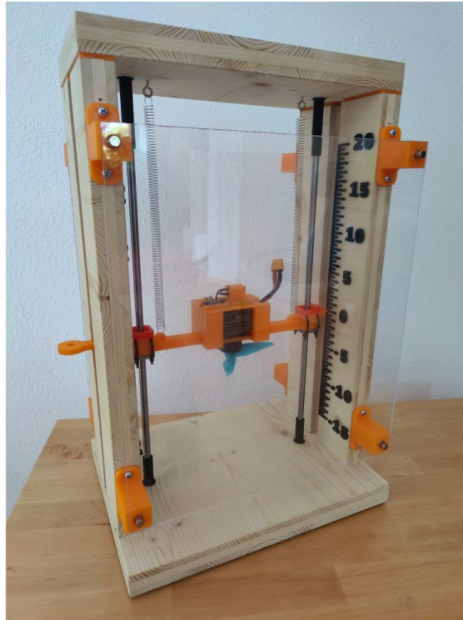


Abbildung 1: Geregelter Feder-Masse-Schwinger – Versuchsaufbau [1, Seite 26]

An beiden Enden der Führungsrundstäbe sind 3D-gedruckte Adapter angebracht, die nicht nur eine Anpassung ermöglichen, sondern auch das Risiko von Verklemmungen minimieren. Der Holzrahmen, wie in Abbildung 1 zu sehen ist, erfüllt mehrere Funktionen: Er bietet einen stabilen Stand, ermöglicht Befestigungsmöglichkeiten und dient als Referenz für den Laser-Abstandssensor. Da es schwierig ist, die Führungen spielfrei und dennoch verklemmsicher zu gestalten, wurden die Gleitlager beweglich am Schlitten befestigt.

### 2.1.5 Plexiglasschutz

Um das versehentliche Berühren des sich schnell drehenden Propellers zu verhindern, werden Plexiglasscheiben als Schutz angebracht. Der Motor dreht sich zusammen mit dem Propeller mit einer Drehzahl von bis zu 253 Umdrehungen pro Sekunde.

Die Plexiglasscheiben werden mithilfe von Neodym-Magneten festgehalten. Die Idee dahinter ist der schnelle An- und Abbau der Plexiglasscheibe. Dieser ermöglicht eine einfache Handhabung und reduziert die mentale Hürde den Versuch ohne Plexiglasschutz laufen zu lassen, was ein Sicherheitsrisiko darstellen kann.

Diese Plexiglasscheiben haben einen Einfluss auf das Systemverhalten, also dem Verhalten des geregelten Feder-Masse-Schwingers. Nach dem Autor der Bachelorarbeit könnten ungünstige

aerodynamische Effekte dafür verantwortlich sein. Um diesen Effekt zu minimieren, wurden Abstandshalter verwendet [1, Seite 25].

## 2.2 Auflistung bisheriger Ergebnisse

### 2.2.1 Verwendete Software

Als Software des geregelten Feder-Masse-Schwingers wurden hauptsächlich Matlab und Simulink [3] Version R2022b verwendet.

### 2.2.2 Fernbedienung

Für die Sollwertvorgabe wurde vom Autor der Bachelorarbeit eine Fernbedienung entwickelt. Diese Fernbedienung sollte es ermöglichen, Sollwerte außerhalb des Eingabefeldes des Rechners vorzugeben. Die Abbildung 2 zeigt diese Fernbedienung.



Abbildung 2: Erster Prototyp der Fernbedienung [1, Seite 54]

„Kernelement war hierbei ein Schiebereglerpotentiometer, dessen Position in eine Sollwertvorgabe umgewandelt werden kann. Die Position des Potentiometers wird über einen Arduino ausgelesen und anschließend über eine serielle Schnittstelle an den Computer weitergeleitet. Zusätzlich können über ein OLED-Display Informationen wie die aktuelle Position des Systems oder der Sollwert angezeigt werden“ [1, Seite 53]. Es konnte allerdings keine sinnvolle Lösung für die Kommunikation zwischen Raspberry Pi Zero W und Arduino erarbeitet werden.

### 2.2.3 Erstellung des Streckenmodells

In der Bachelorarbeit wird eine Federkonstante  $c = 16.56 \text{ kg/s}^2$  durch die resultierende Elongation aufgrund der Masse ermittelt [1, Seite 30]. Des Weiteren wurde abhängig vom

Tastgrad eine Kennlinie der resultierenden Stauchung der Zugfedern erarbeitet. Diese Kennlinie wurde in einem Look-Up-Table implementiert, um die Kraft, die durch den Propeller entstehen soll, zu kommandieren [1, Seite 32,33]. Ab einem Tastgrad von 0.85 stößt die Masse an der Oberseite des Aufbaus an [1, Seite 31]. Aus diesem Grund wird der Tastgrad in allen Simulink-Modellen auf diesen Wert begrenzt.

Das herausgefundene Streckenmodell lautet

$$G_M(s) = \frac{416.32 * s + 1388.3}{s^3 + 4.5779 * s^2 + 81.497 * s + 232.19}$$

## 2.2.4 Erstellter Regler

Es wurde für den Feder-Masse-Schwinger ein PID-Regler implementiert. Dabei wurden die folgenden Regelparameter iterativ ermittelt.

$$K_p = 0.008$$

$$K_I = 0.56$$

$$K_D = 0.0092$$

$$T_1 = 0.03s$$

Der Integralblock des Reglers hat eine Stellgrößenbegrenzung im Bereich von 0...3.1023. Der Regler wurde mit Simulink implementiert.

## 2.2.5 Reaktion des Systems auf die Auslenkung von Hand

In der Bachelorarbeit wird gezeigt, wie in Abbildung 3 zu sehen ist, dass abhängig von der Position der Masse das Einschwingverhalten des geregelten Systems variiert [1, Seite 51].

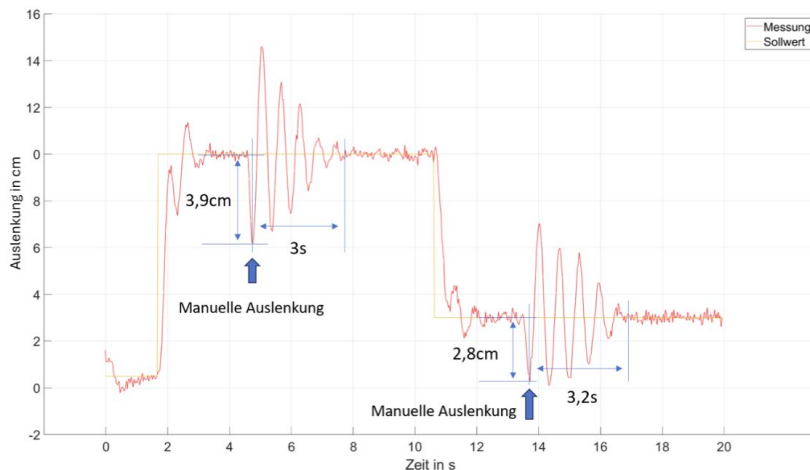


Abbildung 3: Reaktion des Systems auf die Auslenkung von Hand

Dabei wurde festgestellt, dass obwohl die Auslenkung der Masse beim ersten Fall größer war als die des zweiten Falls, ist die Einschwingdauer kürzer. Im ersten Fall wurde die Auslenkung von der Position  $10\text{ cm}$  aus um  $3.9\text{ cm}$  nach unten ausgelenkt. Dabei schwingt die Masse ca.  $3\text{ Sekunden}$  lang. Im zweiten Fall wurde die Auslenkung von der Position  $3\text{ cm}$  aus um  $2.8\text{ cm}$  nach unten ausgelenkt. Dabei schwingt die Masse ca.  $3.2\text{ Sekunden}$  lang.

Des Weiteren wird in beiden Fällen die Masse maximal um  $3.9\text{ cm}$  ausgelenkt. Das Verhalten des Systems kann sich drastisch verändern, sobald die Auslenkung in den Bereich von  $10\text{ cm}$  gerät. Wie in der Abbildung 4 zu sehen, gerät das System in einen Dauerschwingzustand. Es wurden verschiedene Sollwertpositionen getestet und versucht, die Masse um ca.  $10\text{ cm}$  mit der Hand auszulenken. Dabei wurden die Sollwertpositionen  $0 \dots 14\text{ cm}$  in  $2\text{ cm}$ -Schritten erhöht. Dabei wurde so schnell wie möglich die Masse losgelassen, um ein Aufbauen des Integral-Anteils des Positionsreglers kleinzuhalten. Dabei kam als Ergebnis raus, dass sich bei einer ungefähren Position von  $3 \dots 6\text{ cm}$ , die Auslenkung, wie oben beschrieben, eine Dauerschwingung erzeugt.

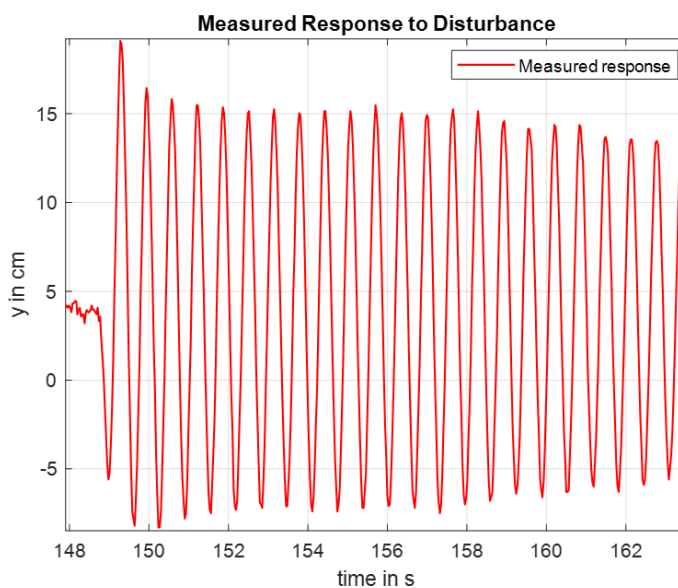


Abbildung 4: Instabile Reaktion des geregelten Systems auf eine große Auslenkung

Um eine Einschätzung für das unregelmäßige Verhalten zu bekommen, wurde beispielhaft eine Auslenkung, wie in der Abbildung 5 zu sehen, vorgenommen.

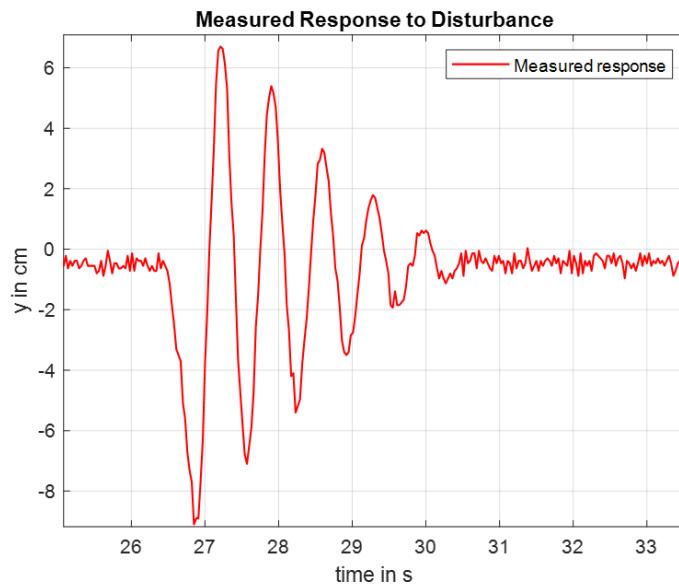


Abbildung 5: Reaktion des unregulierten Systems auf eine große Auslenkung

Der resultierende Plot zeigt, dass das unregulierte System, wie physikalisch angenommen, sich wie ein gedämpftes Feder-Masse-Schwinger-System verhält.

## 2.3 Für die Arbeit anzuwendendes Wissen

### 2.3.1 Überhitzungsschutz des ESC

Die Overheat-Protection kann nicht ausgeschaltet bzw. beeinflusst werden.

Unter dem Punkt „Overheat-Protection“ steht geschrieben: „When the temperature of the ESC MOSFETS exceeds 100 Celsius degree, power will be lowered gradually and will resume when the temperature decreases.“[2]

### 2.3.2 Armierung des ESC

Der ESC muss jedes Mal beim Start eines Versuchs armiert werden. Dabei muss zuerst das niedrigste Signal, also ein Tastgrad von 0.50, für wenige Sekunden gehalten werden, bevor der Motor anfangen kann, sich zu drehen [2].



### 2.3.3 Berechnung der Varianz

"Die Stichprobenvarianz  $s^2$  ist als mittlerer quadratischer Abstand zum Mittelwert definiert." [17, Seite 67]. Die Gleichung, die dafür verwendet wird lautet:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

Dabei gilt für die Gleichung:

- $n$  ist die Anzahl der Messwerte
- $\bar{x}$  ist der Mittelwert der Messwerte
- $x_i$  ist der  $i$ -te Messwert

In Abbildung 6 wird beispielhaft das Messrauschen des Positionssensors über einen Zeitbereich von 30 Sekunden gezeigt, bei einer gleich bleibenden Position von 0 cm.

In Matlab wird zur Berechnung der Varianz die Funktion `var(A)` verwendet. Wobei `A` ein Array der Messwerte ist [19].

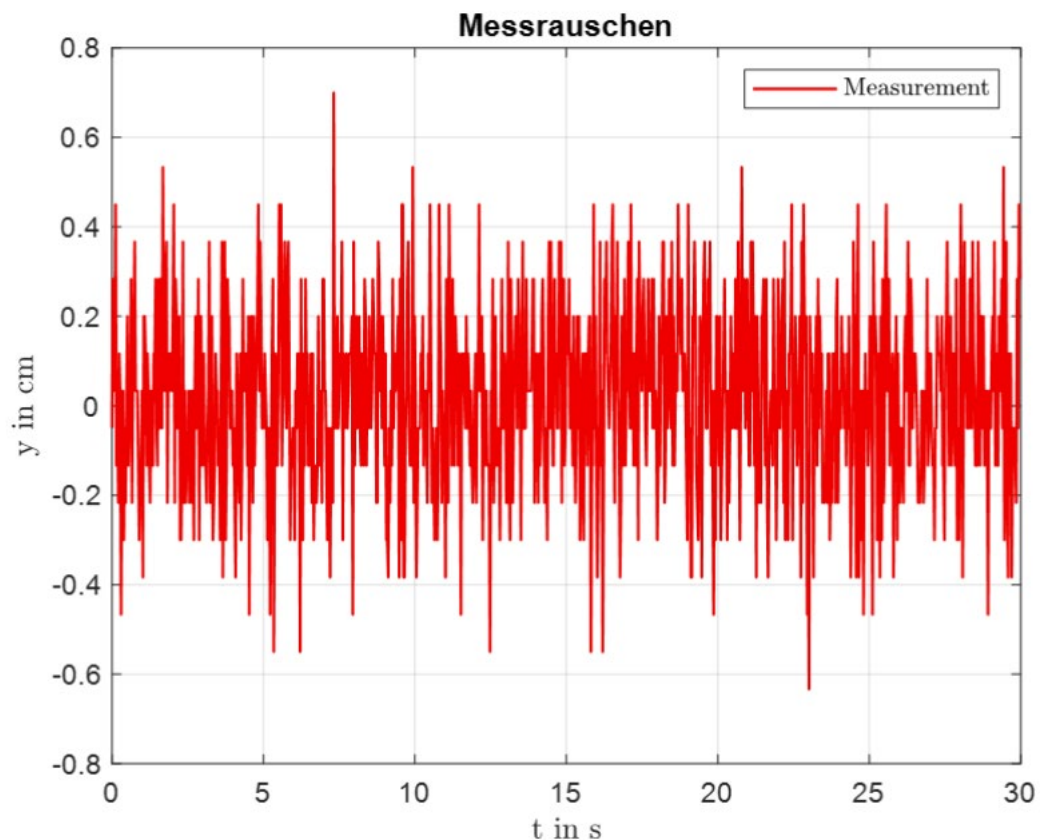


Abbildung 6: Messrauschen im Positionssignal um die 0 cm-Marke herum

### 2.3.4 Bestimmung der Propellerstellkraft

Laut [20, Seite 20] kann die Kraft  $T$ , die der Propeller erzeugt, durch folgende Formel berechnet werden:

$$T = (2 * \pi * R^2 * \rho * (c_p * n^p)^2)^{\frac{1}{3}}$$

Dabei gilt für die Formel:

- $R$  ist der Propeller-Radius
- $\rho$  ist die Luftdichte
- $n$  ist die Drehzahl in Umdrehungen pro Minute (rpm)
- $c_p$  ist die Propellerkonstante
- $p$  ist der Leistungsfaktor

### 2.3.5 Software für Regelungstechnik pzMove

Für die Identifikation der Streckenmodelle wird das Lerntool pzMove [15] verwendet.

Die in dem Tool verwendete Methode zur Identifikation basiert auf dem Verfahren zur experimentellen Modellbildung von Herrn Zentgraf [16].

### 2.3.6 Bestimmung eines Zustandsreglers

Zur Bestimmung des Zustandsreglers werden die Systemmatrizen  $A$ ,  $B$ ,  $C$  verwendet. Es wird davon ausgegangen, dass die Durchgangsmatrix  $\underline{D} = \underline{0}$  ist. In der Projektarbeit wird ein Reglerentwurf zur Polzuweisung durchgeführt [21, Kapitel 6].

Zur Berechnung der Reglermatrix  $R$  werden die für das geregelte System gewünschten Pole „desPoles“ benötigt. In Matlab gibt es die Funktion `place()` [21, Kapitel 6.5]:

$$R = \text{place}(A, B, \text{desPoles})$$

Dabei gilt für die Funktion:

- $\underline{A}$  ist die Zustandsmatrix
- $\underline{B}$  ist die Eingangsmatrix

Für die Bestimmung des Vorfilters  $V$  wird die folgende Gleichung verwendet nach [21, Kapitel 4.4.2]:

$$V = \frac{1}{\underline{C} * (\underline{B} * \underline{R} - \underline{A})^{-1} * \underline{B}}$$

Dabei gilt für diese Gleichung:

- $A$  ist die Zustandsmatrix
- $B$  ist die Eingangsmatrix
- $C$  ist die Ausgangsmatrix
- $R$  ist die Reglermatrix

### 2.3.7 Bestimmung eines Beobachters

Zur Berechnung der Beobachterrückführung  $K$  werden die gewünschten Pole „desPolesObsv“ benötigt. In Matlab kann dies mit folgendem Code [21, Kapitel 8.4]:

$$K = (\text{place}(A', C', \text{desPolesObsv}))'$$

Dabei gilt für diesen Code:

- $A$  ist die Zustandsmatrix
- $C$  ist die Ausgangsmatrix
- $()'$  entspricht der Transposition

## 3 Durchführung und eigene Untersuchungen

In dieser Arbeit wurde zur Bearbeitung der Software des Mikrocontrollers, Matlab und Simulink Version R2023b verwendet [3]. Das Programmieren der Software der Fernbedienung wird in der Arduino Entwicklungsumgebung durchgeführt [7].

### 3.1 Technische Aufrüstung

Die technische Ausrüstung wird sowohl um eine Fernbedienung als auch um einen Drehzahlsensor erweitert.

#### 3.1.1 Implementieren der Fernbedienung

Um eine drahtlose Verbindung zum verwendeten Mikrocontroller, dem Raspberry Pi Zero W, herzustellen, wird programmierbare Hardware benötigt, die drahtlos kommunizieren kann. Der in der Abbildung 7 dargestellte Arduino Nano IoT 33 bietet sich als kompakte Lösung an. Aufgrund seiner geringen Abmessungen von 43.13mm x 17.77 mm [5, Seite 12] kann ein Arduino Nano IoT 33 problemlos in eine speziell dafür entworfene Fernbedienung integriert werden.



Abbildung 7: Arduino Nano IoT 33 [5]

Der Arduino Nano IoT 33 besitzt ein Wi-Fi®+Bluetooth®-Modul basierend auf ESP32 [5, Seite 1]. Dieses Modul ermöglicht die Kommunikation mit dem Raspberry Pi Zero W über Wi-Fi.

Darüber hinaus besitzt der Arduino bis zu 7 10-Bit-Analog-Pins [5, Seite 10], mit denen Werte mithilfe eines Schiebepotentiometers vorgegeben werden können. Dies ermöglicht die in der Zielsetzung vorgegebene Bestimmung des Sollwerts bzw. der Stellgröße.

Der Arduino Nano IoT 33 verfügt auch über eine I2C-Schnittstelle, die für das verwendete Display [6] zur Benutzung gebraucht wird, siehe die Abbildung 8.

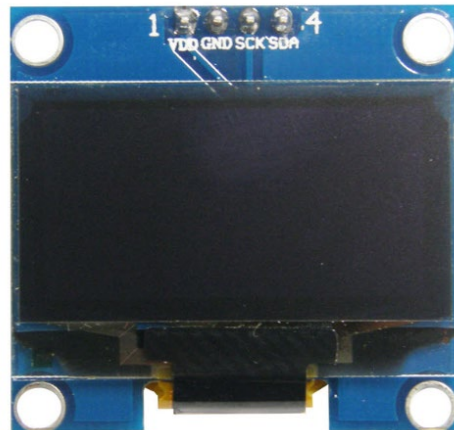


Abbildung 8: OLED-Display mit I2C-Schnittstelle

In Verbindung zur Hardware braucht es aber auch die passende Software. Hierfür wird aufseiten des Arduino Nano IoT 33 die Arduino Entwicklungsumgebung verwendet [7], die verschiedene Open-Source-Bibliotheken bietet, um das Display und das Wi-Fi®+Bluetooth®-Modul anzusteuern [8, 9].

Aufseiten des Raspberry Pi Zero W sind weitere Softwareänderungen erforderlich. Um die Kommunikation zwischen dem Raspberry Pi Zero W und der Fernbedienung zu ermöglichen, muss die IP-Adresse des Raspberry Pi Zero W statisch gemacht werden. Das ermöglicht, dass der Arduino-Code nicht ständig an die sich dynamisch ändernde IP-Adresse durch DHCP geändert werden muss. Momentan befindet sich auf dem Raspberry Pi Zero W ein Systemfile namens config.txt, der so erweitert wurde, dass dieser ständig im Hintergrund auf die Betätigung des Taster, der sich auf dem Schlitten befindet, lauscht und dann ein Herunterfahren des Raspberry Pi Zero W durchführen lässt [10]. Durch Erweiterung des Codes kann auf einen weiteren Pin gelauscht werden, der durch das Simulink-Programm, was gleich beschrieben wird, angesteuert wird. Auf diese Weise kann ein Shutdown-Vorgang mithilfe der Fernbedienung implementiert werden. Dabei soll das Simulink-Programm gleichzeitig die Regelung und die Funktionen der Fernbedienung umsetzen.

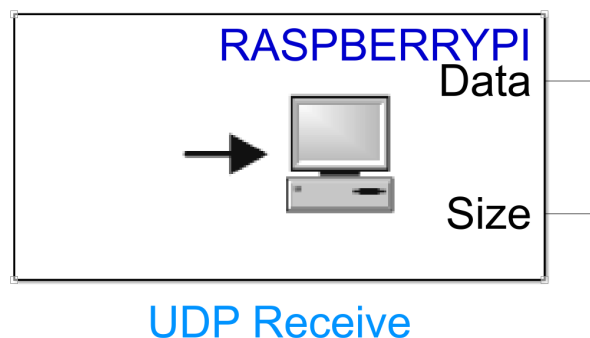


Abbildung 9: UDP Receive -Block aus Simulink

Mithilfe des „UDP Receive“-Blocks (siehe Abbildung 9) können vom Raspberry Pi Zero W die notwendigen Daten der Fernbedienung erhalten werden. Es wird also für eine beschleunigte Kommunikation, das User Datagram Protocol (UDP) verwendet.

Für die Implementierung des vollständigen Simulink-Programmes, wird zuerst ein vereinfachtes Programm erstellt, welches die verwendeten Daten interpretieren und für die Stellgrößen- bzw. Sollwertvorgabe nutzbar macht.

Die Verbindung zwischen Fernbedienung und Raspberry Pi Zero W wird alle *0.03 Sekunden* durchgeführt. Nachdem die Tests im vereinfachten Programm durchgeführt wurden, soll das komplettierte Programm, mit allen Funktionen getestet werden.

### 3.1.2 Implementieren des Drehzahlsensors

Aus der Motordrehzahl ließe sich theoretisch die Stellkraft ermitteln. Es ergibt sich die Möglichkeit, die tatsächliche Stellgröße zu messen. Das Problem an dem ESC ist, dass die Drehzahl des Propellers nicht nur vom kommandierten Eingangssignal abhängt, sondern auch die Akkuspannung einen Einfluss auf die resultierende tatsächlich Drehzahl des Propellers hat.

#### Standard-Anschluss/Standard Connection:

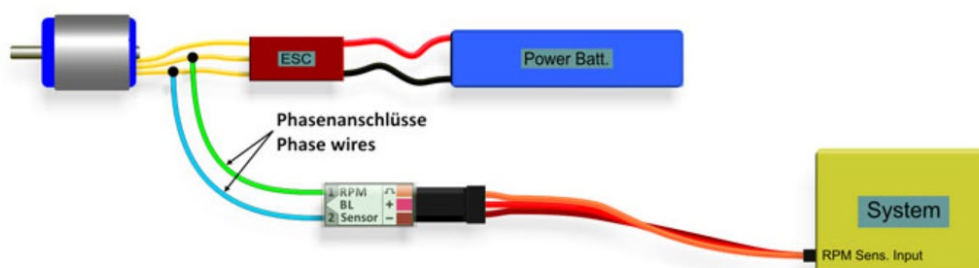


Abbildung 10: Verdrahtungsbild für den RPM-Sensor [11]

Um die Drehzahl zu messen, wurde ein universeller RPM-Sensor [11] verwendet. Dieser wurde verwendet, da er mit seinen Abmessungen von  $22 \times 8 \times 6,5 \text{ mm}$  und einem Gewicht von  $2 \text{ g}$ , in den Aufbau leicht hinzugefügt werden kann, ohne, dass das Systemverhalten aufgrund der zusätzlichen Masse sich stark verändert. Er verstärkt die Pulse, die durch die Back-EMF (Back Electromotive Force) des BLDC-Motors erzeugt werden [13]. Dadurch werden  $3.3 \text{ Volt}$ -Pulse ausgegeben, die mithilfe eines Zählerbausteines gezählt werden können.

In der Abbildung 10 wird gezeigt, wie der universelle RPM-Sensor angeschlossen wird. In der Abbildung 10 stellt das „System“ mit „RPM Sensor Input“ die Auswerteelektronik für die  $3.3 \text{ Volt}$ -Pulse da. Diese Auswerteelektronik ist im bestehenden Aufbau der Zählerbaustein und der Raspberry Pi Zero W. Es wurde ein Zählerbaustein S-35770 verwendet, der über I2C mit dem Raspberry Pi Zero W kommuniziert. Der Zählerbaustein ist ein 24-Bit

Hochzählerbaustein und kann maximal mit einer Frequenz von  $1000\text{ kHz}$  zählen [12]. Es wurde sich für diesen entschieden, da dieser ausreichend viele Zustände zählen kann und mit seinen Ausmaßen von  $4.0\text{ mm} \times 2.9\text{ mm}$  wenig Platz beansprucht. Zusätzlich muss der Zählerstand des Zählerbausteins lediglich mit zwei Kabeln übertragen werden.

Der Propeller dreht sich maximal mit 253 Umdrehungen pro Sekunde. Da der BLDC-Motor 7 Polpaare besitzt, werden pro Umdrehung 7 Pulse durch Back-EMF erzeugt und vom Zählerbaustein gezählt. Dies bedeutet, dass der Zählerbaustein dazu in der Lage sein muss in 0.03 Sekunden bis zu  $[7 \times 253\text{ 1/s} \times 0.03\text{ s}] = 54\text{ Pulse}$  zählen zu können. Die Auflösung für die Drehzahlmessung beträgt  $4.762\text{ 1/s}$ .

Die Abtastzeit von 0.03 Sekunden des I2C-Busses wurde gewählt, da das etablierte Positionsmessgerät schon mit dieser Abtastzeit arbeitet. Da die Drehzahlmessauflösung ausreichend genau zu sein scheint, wird diese Abtastzeit weiter verwendet. An diesem I2C-Bus befinden sich 3 Geräte, einmal der Raspberry Pi Zero W als Master und die beiden Slave-Geräte sind der Positionssensor und der Zählerbaustein.

## 3.2 Implementieren eines Drehzahlreglers

Um ein besseres Verhalten der Aktorik zu bekommen, wurde zuerst versucht den ESC umzuprogrammieren. Dabei sollten die folgenden Einstellungen angepasst werden.

Einmal wurde versucht, den Überhitzungsschutz (Overheat-Protection in Kapitel 2.3.1) auszuschalten, da dies zu unvorhersehbaren Veränderungen der Leistung am BLDC-Motor führt, die das Verhalten der Motordrehzahl beeinflusst.

Mit der anderen Einstellung, dem sogenannten „Damped Light Modus“, soll aktives Bremsen des BLDC-Motors ermöglicht werden. Dies würde den BLDC-Motor schneller auf eine niedrigere Drehzahl bringen, wenn das kommandierte Eingangssignal es erfordert. Ohne aktive Bremsung wird der BLDC-Motor bzw. Propeller nur aufgrund des Luftwiderstandes abgebremst [4].

Dies äußert sich durch ein langsames Erreichen der niedrigeren Motordrehzahl als der BLDC-Motor die schnellere Drehzahl erreichen könnte. Siehe dazu folgende Abbildung 11. In dieser wird, basierend auf der Bachelorarbeit [1], sprunghaft eine Kraft kommandiert und die normalisierte Position der Masse gemessen. Dabei stellt die blaue Kurve die Sprungantwort durch einen positiven Sprung um  $1.4\text{ N}$  da. Die rote Kurve wiederum zeigt das Verhalten von dem Sprung zurück, also  $-1.4\text{ N}$ .

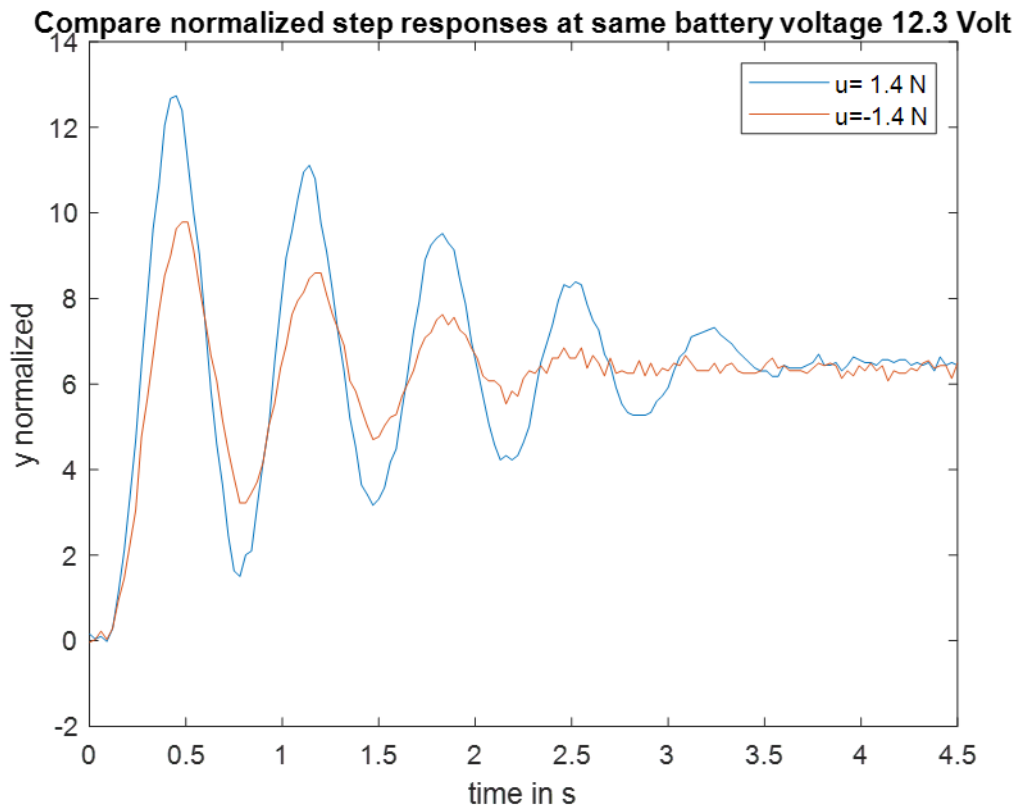


Abbildung 11: Richtungsabhängiges Motordrehzahlverhalten

Die Abbildung 11 zeigt, dass der BLDC-Motor aufgrund des fehlenden aktiven Bremsens richtungsabhängige Verzögerungen beim Erreichen der Sollgeschwindigkeit hat. Dies resultiert in ein langsames "Fallenlassen" der Masse, da der Propeller eine Kraft in die entgegengesetzte Richtung verursacht. Der Überschwinger der negativen Sprungantwort wird reduziert und das System wird stärker gedämpft.

Aktives Bremsen könnte zu einer vergleichbaren Dynamik zwischen positiver und negativer Stellgrößenänderung führen. Der „Damped Light Modus“ kann bei diesem ESC nicht eingestellt werden, womit kein aktives Bremsen des BLDC-Motors erfolgt. Eine größere Verzögerung bei negativer Drehzahländerung muss sich nicht negativ auf das Regelkreisverhalten auswirken, da bei einer ausreichend langsamen Änderung des Stellsignals, die Auswirkung klein bleibt. Dies wird sich bei der Auswertung der verschiedenen Regelkonzepte zeigen.

### 3.2.1 Systemantworten zur Identifikation des Motordrehzahlverhaltens aufnehmen

Nachdem der Drehzahlsensor implementiert wurde, kann eine Möglichkeit zur Regelung der Drehzahl in Betracht gezogen werden. Um einen geeigneten Regler zu implementieren, sollte zuerst analysiert werden, bei welchem kommandierten PWM-Tastgrad, man welche stationäre Drehzahl erhält. Um dies zu testen, wurde eine sehr langsame Rampe als Tastgrad-Verlauf vorgegeben. Gleichzeitig wurde versucht, den Test möglichst schnell ablaufen zu lassen, um



ungünstige Effekte zu vermeiden. Diese Effekte sind zum einen der Einfluss, der sich reduzierenden Akkuspannung und zum anderen die Leistungsrosselung durch den Überhitzungsschutz, siehe dazu Kapitel 2.3.1. Die Messung wurde bei einem Akku-Ladezustand von  $11.9\text{ Volt}$  durchgeführt und hatte nach der Messung  $0.1\text{ Volt}$  verloren. Da die Rampe  $10\text{ Sekunden}$  braucht um von dem niedrigsten Tastgrad von  $0.50$  den maximalen Tastgrad von  $0.85$  zu erreichen, wird angenommen, dass der BLDC-Motor schnell genug ist um zu jedem Zeitpunkt seine stationäre Drehzahl zu erreichen. Die Rampe, die in der Abbildung 12 gezeigt wird, zeigt den Drehzahl-Zeit-Verlauf des BLDC-Motors im unteren Sub-Plot.

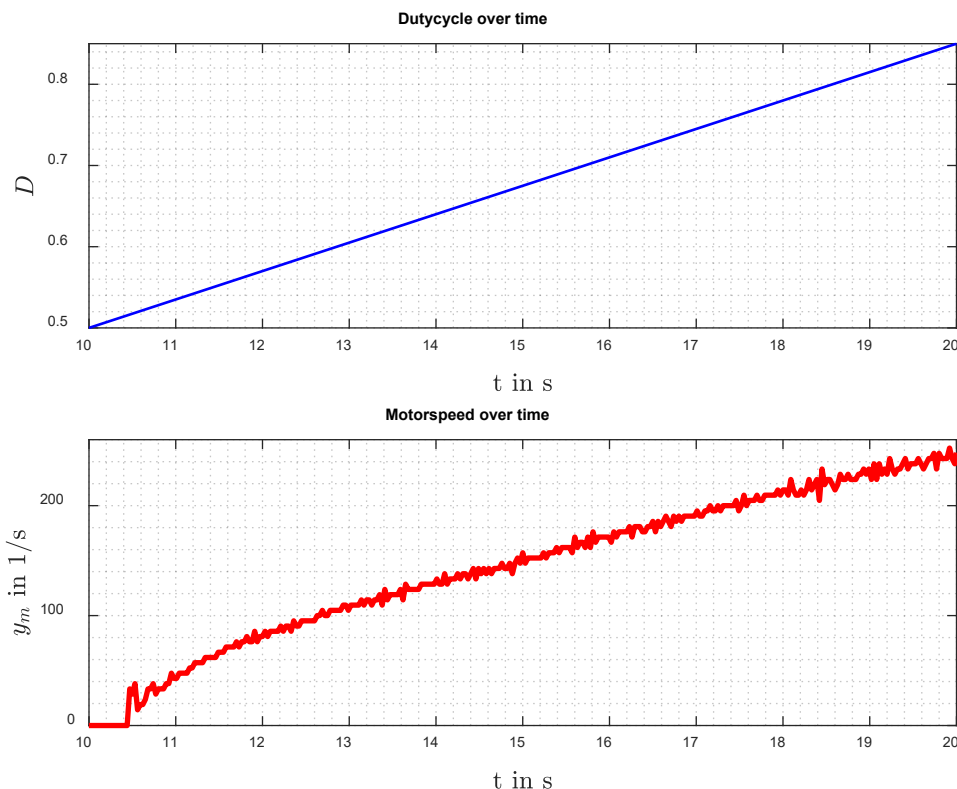


Abbildung 12: Drehzahl-Zeit-Verlauf einer sehr langsam ansteigenden Drehzahl-Rampe

Der obere Subplot zeigt den Tastgrad-Zeitverlauf als Eingangssignal. Aus der Abbildung 12 kann durch Abgleich der Rampe vom kommandierten Tastgrad und resultierender Drehzahl ein Drehzahl-Tastgrad-Diagramm erzeugt werden, siehe die Abbildung 13.

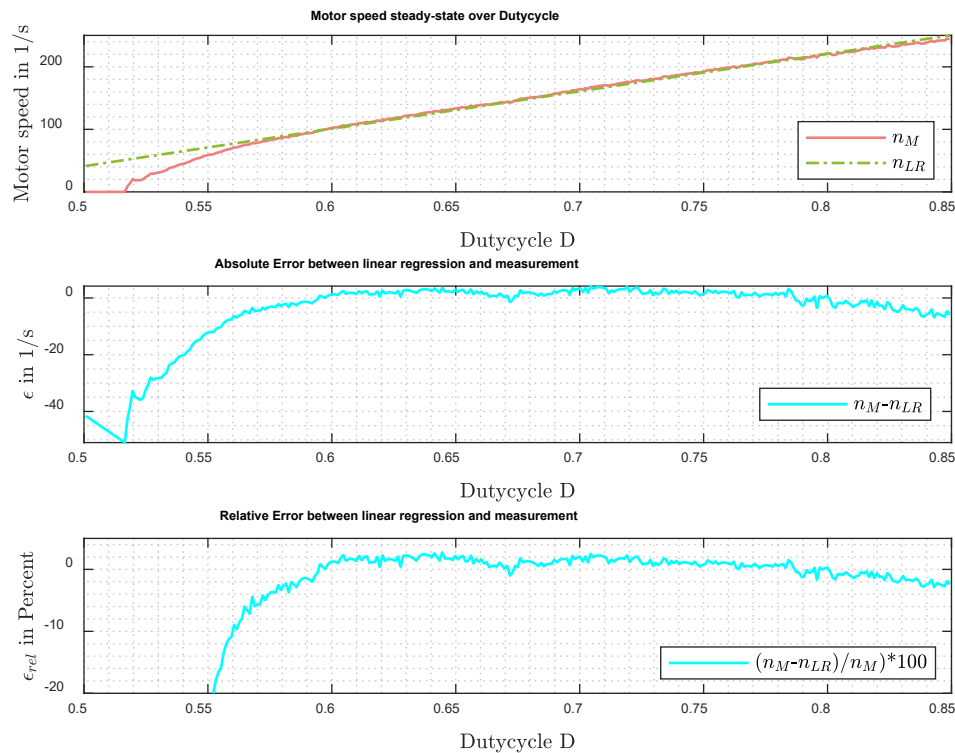


Abbildung 13: Drehzahl-Tastgrad-Kennlinie

Die in der Abbildung 13, Subplot 1, gezeigte Kennlinie, zeigt gutes lineares Verhalten im Tastgrad-Bereich von 0.6 bis zum Schluss. Darunter wird, wie in den unteren Subplots gezeigt wird, die Motordrehzahl immer weniger linear zum Tastgrad. Der relative Fehler liegt beim Tastgrad 0.55 schon bei  $-20\%$ .

Ziel der Kennlinienanalyse ist es herauszufinden, von welchem Betriebspunkt aus eine Streckensystemidentifikation für das Motordrehzahlverhalten stattfindet.

Für die Bestimmung des Betriebspunktes wurden drei Faktoren berücksichtigt. Erstens verhält sich der BLDC-Motor unter einem Tastgrad von 0.6 nicht linear. Zweitens wird versucht der Abstand zwischen den Stellwertbegrenzungen möglichst groß zu halten und drittens wird darauf geachtet, dass bei höheren Drehzahlen, der ESC sich schneller erhitzt und die in Kapitel 2.3.1 angesprochene Overheat-Protection sich aktiviert. Aus diesen drei Gründen wird als Betriebspunkt für den BLDC-Motor ein Tastgrad von 0.65 mit einer daraus resultierenden Betriebs-Drehzahl von 128.34 1/s gewählt. Dieser eignet sich, da für die Streckenidentifikation sich ein „sicheres“ Band von 0.65 bis 0.8 ergibt, in der der Einfluss der Störfaktoren klein bleibt.

Die Systemantworten wurden möglichst bei gleichem Akku-Ladestand von 11.9 Volt durchgeführt, um ein gleichbleibendes Verhalten zu erhalten.

### 3.2.2 Identifizieren des Streckenmodells

Die Identifikation wird nach den in Kapitel 2.3.5 angebrachten Verfahren in pzMove aus dem Messdatensatz generiert. Das resultierende Motordrehzahlmodell wird vereinfacht als PT1-System gesehen und beschrieben durch:

$$G_{Mm} = \frac{K_m}{T_m * s + 1}$$

Das Motordrehzahlmodell soll dann mit andersartigen Eingangssignalen verglichen und verifiziert werden.

### 3.2.3 Auslegung des Motordrehzahlreglers

In der Abbildung 14 wird gezeigt, wie die Regelkreisstruktur des Motordrehzahlreglers aussieht. Für diese Struktur wurde sich entschieden, da auf diese Weise die schnelle Reaktion des ESC kombiniert wird mit der ausgleichenden Wirkung des Integral-Anteils. Der Integralanteil soll die sich reduzierende Akkuspannung kompensieren.

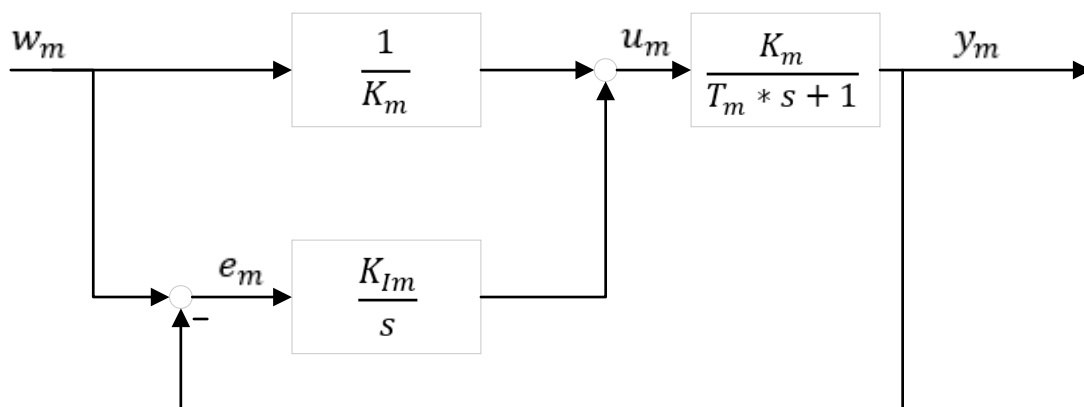


Abbildung 14: Regelkreisstruktur zur Regelung der Motordrehzahl

Es wurde sich gegen eine klassische PI-Regler-Struktur entschieden, da es sich gezeigt hat, dass ein starker impulsartiger Drehzahlmessfehler sich zu sehr wegen den Proportional-Anteil auswirkt. Dies ist aufgrund des trägen Charakters des Integral-Anteils nicht der Fall. Der Integralblock des Reglers gibt Werte im Bereich von  $-0.1 \dots 0.1$  aus und wurden willkürlich gewählt.

Die Sollwert-Proportional-Verstärkung  $K_{WPM}$  ist die Inverse der stationären Verstärkung  $K_m$  des BLDC-Motors:

$$K_{WPM} = \frac{1}{K_m}$$

Aus der Reglerstruktur kann folgende Referenz-Übertragungsfunktion für die Motordrehzahl gebildet werden:

$$G_{wm} = \frac{\frac{1}{K_m * K_{Im}} * s + 1}{\frac{T_m}{K_m * K_{Im}} * s^2 + \frac{1}{K_m * K_{Im}} * s + 1}$$

Mithilfe dieser kann durch Probieren eine geeignete Integralverstärkung simuliert werden.

### 3.2.4 Simulation des Motordrehzahlreglers

In Simulink wurde eine Simulation des Motordrehzahlreglers entworfen. Ziel dessen ist es eine Einschätzung für das reale Regelkreisverhalten zu bekommen. Dies hilft bei der Erweiterung des Reglers zu einem Kaskadenregler, um das Positionsverhalten zu simulieren.

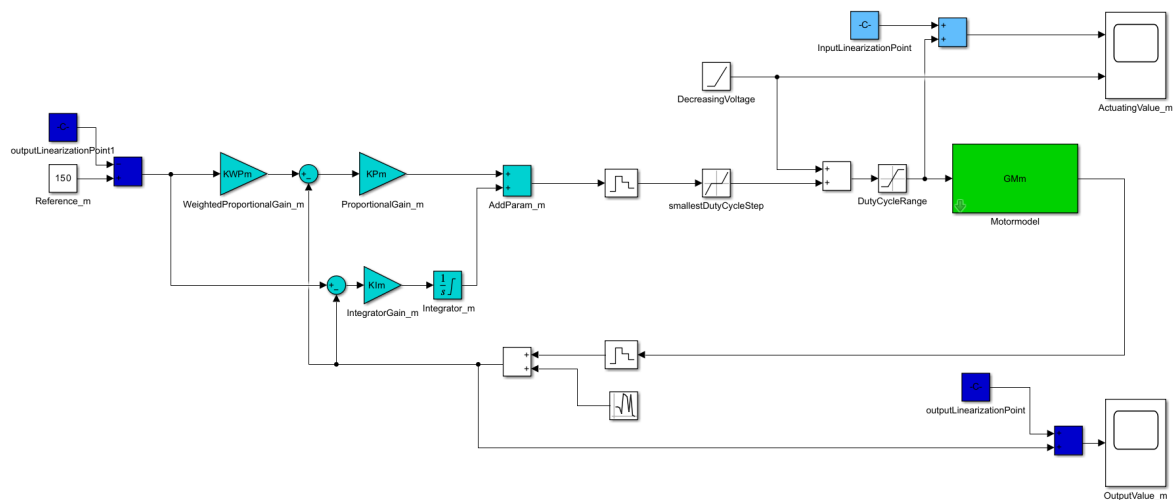


Abbildung 15: Simulink-Modell - BLDCMotorControlLoop\_Sim.slx

Das Simulink-Modell in der Abbildung 15 läuft mit einer Abtastzeit von 0.001 Sekunden und mithilfe von Zero-order-hold-Blöcken wird versucht, die Abtastung des Digitalreglers nachzuahmen. Es wurde eine negative Rampe „DecreasingVoltage“ implementiert um die langsame Abnahme der Akkuspannung zu simulieren. In das Simulink-Modell wurden auch die Betriebspunkte, aus Kapitel 3.2.1, „InputLinearizationPoint“ und „outputLinearizationPoint“ eingeführt.

Simuliert wird das Messrauschen mit dem "random number"-Block in Simulink. Wobei ein Seed und ein Mittelwert von null gewählt wird. Die Varianz wird aus einem Messdatensatz nach der in Kapitel 2.3.3 Matlab-Funktion `var()` bestimmt.

### 3.2.5 Implementierung des realen Motordrehzahlreglers

Für den realen Test wird das Simulink-Modell aus Abbildung 15 entsprechend erweitert. In der Abbildung 16 sieht man die neue Struktur.

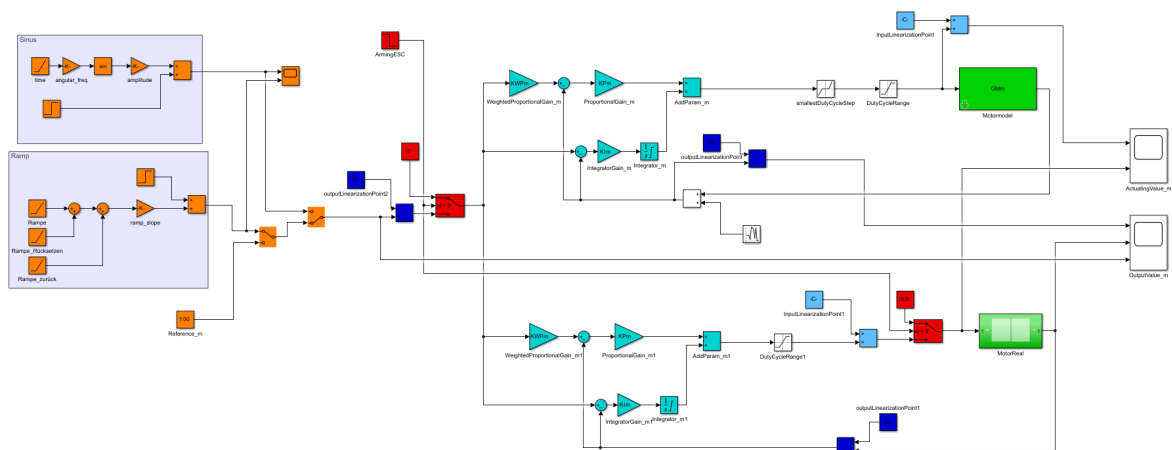


Abbildung 16: Simulink-Modell - BLDCMotorControlLoop.slx

Für die Armierung des ESC werden „Multiport-Switch“-Blöcke (in rot) verwendet, siehe Kapitel 2.3.2. Diese sorgen dafür, dass zuerst für 5 Sekunden der Tastgrad auf 0.5 gebracht wird und gleichzeitig wird der Eingang für den Integralanteil des Reglers auf 0 gesetzt, um ein Aufbauen des Integral-Anteils zu verhindern. Dies soll ein Aufschnellen des BLDC-Motors verhindern. Diese Methode den ESC zu armieren wird auch in allen weiteren Real-Anwendungen dieser Projektarbeit verwendet.

Auf der linken Seite wurden Blöcke zur Generierung verschiedenartiger Sollwertsignale implementiert. Auf diese Weise können als Sollwertverhalten ein Sinus-, ein Rampen- und ein Sprungsignal vorgegeben werden.

### 3.2.6 Ermitteln der tatsächlichen Stellkraft aus der Motordrehzahl

Da nun ein Drehzahlregler implementiert wurde, kann abhängig von der Drehzahl die Stellkraft bestimmt werden. Dabei wird davon ausgegangen, dass die Feder eine lineare Kennlinie hat. Es wird also abhängig von der Verkürzung der Feder die Stellkraft bestimmt. Es entsteht eine Kennlinie, die als Look-Up-Table umgesetzt eine Möglichkeit gibt die gewünschte Stellkraft vorzugeben.

Die Umkehrfunktion der Kennlinie bietet zusätzlich die Möglichkeit, die tatsächlich am systemwirkende Stellkraft, durch Messung der Drehzahl zu ermitteln.

### 3.3 Auslegung eines neuen Positionsreglers

#### 3.3.1 Mögliche Erklärungsansätze für die schlechte Störkompensation

Es wurden sich mehrere Überlegungen gemacht, warum die Störkompensation des geregelten Systems schlechter als die Störkompensation des unregulierten Systems ist. Also eine längere Einschwingzeit bei einer Auslenkung von Hand aufweist. Es werden im Folgenden mehrere Erklärungsmöglichkeiten angeführt:

##### 3.3.1.1 Erklärungsansatz 1: Eine zu große Abtastzeit

Diese Erklärung konnte durch Simulation der Strecke ausgeschlossen werden. Des Weiteren ist die Abtastzeit deutlich kürzer als die Zeitkonstante des dominanten Polpaars. Hierzu wurde das Modell aus dem Kapitel 2.2.3 in die Zeitkonstantendarstellung überführt:

$$G_M(s) = \frac{5.9792 * (1 + 0.2999 * s)}{(1 + 0.3308 * s) * (1 + 2 * 0.0887 * (0.1141 * s) + (0.1141 * s)^2)}$$

Die Zeitkonstante in diesem Modell beträgt  $T_M = 0.1141$  s. Rechnet man die Abtastzeit in eine Periodendauer um, erhält man  $T_{Abt} = \frac{0.03}{2 * \pi} = 0.48 * 10^{-2}$  s. Daraus folgt, dass  $T_M$  23.78 x größer ist als  $T_{Abt}$ . Dies bedeutet, dass die Abtastzeit ausreichend schnell ist.

##### 3.3.1.2 Erklärungsansatz 2: Integral Wind Up

Lenkt man die Masse von Hand aus, erhöht sich der Integral-Anteil des Positionsreglers enorm und kann für ein starkes Aufschwingen sorgen.

##### 3.3.1.3 Erklärungsansatz 3: Der nicht lineare Einfluss der Aktorik

Wie man an der Drehzahl-Tastgrad-Kennlinie aus Kapitel 3.2.1 sehen kann, verhält sich die Motordrehzahl deutlich nicht linear unter einem Tastgrad von 0.6. Jedoch wird beim Auslenken der Masse das geregelte System dazu gezwungen sich in diesen nicht linearen Bereich zu bewegen. Dies erklärt auch den Umstand, warum es nicht zur Instabilität an höheren Positionen kommt. Bei diesen hat der BLDC-Motor eine höhere Grunddrehzahl und dies führt bei einer Auslenkung nicht in den niedrigen Drehzahlbereich bei einem Tastgrad von unter 0.6. Um den nicht linearen Einfluss durch die Aktorik kleinzuhalten, kann versucht werden, die Dynamik der geregelten Strecke zu reduzieren. Auf diese Weise sollte sich dieser Regler dem unregulierten System mehr ähneln.

Im Weiteren soll die Erarbeitung des neuen Reglers zur besseren Kompensation der Störauslenkung unternommen werden. Dies würde zum Beispiel durch die Veränderung der Dynamik

des Systems mit einer höheren Dämpfung erreicht werden. Eine geeignete Möglichkeit dafür ist ein Zustandsregler. Dieser kommt nämlich ohne Integral-Anteil aus. Das einzige Integral wäre dann in der Motordrehzahlregelung. Auf diese Weise kann ein Wind-Up und damit eine Überreaktion des Systems verhindert werden. Erklärungsansatz 2 wäre damit eliminiert.

### 3.3.2 Physikalische Herleitung des Feder-Massen-Schwinger-Modells

Zuerst wird die physikalische Differentialgleichung betrachtet, basierend auf [1, Seite 40]. Da das System linearisiert wird, werden Variablen wie die Schwerkraft oder die Anfangslänge der unbelasteten Zugfeder ignoriert. Es resultiert die Gleichung (I):

$$m * \ddot{y}(t) = F_{prop}(t) - c * y(t) - d * \dot{y}(t)$$

Dabei gilt für Gleichung (I):

- $y$  entspricht dem Wert für die Position der Masse in  $m$
- $m$  entspricht dem Wert für das Gewicht der Masse in  $kg$
- $d$  entspricht dem Wert für die Dämpfung im System in  $\frac{kg}{s}$
- $c$  entspricht dem Wert für die Federsteifigkeit in  $\frac{kg}{s^2}$

Aus der Differentialgleichung (I) kann eine Laplace-Transformation in den Bildbereich vorgenommen werden, in der die Anfangsbedingung alle gleich 0 gesetzt werden. Es resultiert daraus die Gleichung (II):

$$m * Y(s) * s^2 = U(s) - c * Y(s) - d * Y(s) * s$$

Dabei gilt für Gleichung (II):

- $U(s)$  entspricht der Bildfunktion von  $F_{prop}(t)$

Aus Gleichung (II) kann man nun durch Umformung die Übertragungsfunktion  $G_{Mp}$  bestimmen als Gleichung (III):

$$G_{Mp}(s) = \frac{Y(s)}{U(s)} = \frac{100}{m * s^2 + d * s + c}$$

Dabei gilt für die Übertragungsfunktion  $G_{Mp}$ :

- Für die Umrechnung auf cm wurde der Zähler mit 100 multipliziert
- $[G_{Mp}(s)] = \frac{cm}{N}$

Die Übertragungsfunktion  $G_{Mp}$  ist zweiter Ordnung und durch Umformulierung des Nenners lässt sich die Zeitkonstanten-Darstellung bilden als Gleichung (IV):

$$G_{Mp}(s) = \frac{K_p}{\frac{1}{\omega_{0p}^2} * s^2 + 2 * \frac{D_p}{\omega_{0p}} * s + 1}$$

Dabei gilt für Gleichung (IV):

- Die stationäre Verstärkung  $K_p = \frac{100}{c}$  in  $\frac{cm}{N}$
- Die Eigenkreisfrequenz  $\omega_{0p} = \sqrt{\frac{c}{m}}$  in  $\frac{rad}{s}$
- Der Dämpfungsgrad  $D_p = \frac{d}{2\sqrt{c*m}}$ , *Einheitenlos*

Als Nächstes soll für einen Zustandsregler die Zustandsraumdarstellung erarbeitet werden. Basierend auf Gleichung (I) werden folgende Zustände definiert:

$$x_1(t) = y(t)$$

$$x_2(t) = \dot{y}(t)$$

Für die bessere Übertragbarkeit aus Gleichung (IV) wird Gleichung (I) so umgeformt, sodass diese durch die Eigenkreisfrequenz, dem Dämpfungsgrad und der stationären Verstärkung ausgedrückt wird. Es resultiert Gleichung (V):

$$\ddot{y}(t) = \omega_{0p}^2 * K_p * u(t) - \omega_{0p}^2 * y(t) - 2 * D_p * \omega_{0p} * \dot{y}(t)$$

Dabei gilt für Gleichung (V):

- $u(t) = F_{prop}(t)$
- $y(t)$  in  $cm$

Nun kann basierend auf den definierten Zuständen und Gleichung (V), die Zustandsgleichung gebildet werden, als Gleichung (VI):

$$\dot{\underline{x}}(t) = \begin{bmatrix} 0 & 1 \\ -\omega_{0p}^2 & -2 * D_p * \omega_{0p} \end{bmatrix} * \underline{x}(t) + \begin{bmatrix} 0 \\ \omega_{0p}^2 * K_p \end{bmatrix} * u(t)$$

Und die Ausgangsgleichung gebildet werden, Gleichung (VII):

$$y(t) = [1 \quad 0] * \underline{x}(t)$$

Dabei gilt für die Gleichungen (VI) und (VII):

- $\underline{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$

### 3.3.3 Untersuchen des Feder-Massen-Schwinger-Systemverhaltens

Für den Betriebspunkt wird ein Wert von 0.6 N verwendet. Dieser Wert wurde verwendet, da dies ungefähr die resultierende Stellkraft um den Motordrehzahl-Betriebspunkt  $128.34 \frac{1}{s}$  ist.



Bei dieser Stellkraft wird durch Messung der Positions-Betriebspunkt von  $3.53 \text{ cm}$  bestimmt. Es sei zu erwähnen, dass die Masse bedingt durch den Aufbau einen möglichen Bewegungsbereich von ca.  $-18 \dots 18 \text{ cm}$  hat und die Masse im eingeschwungenen Zustand und sich nicht drehenden Propeller bei ca.  $-1 \text{ cm}$  befindet.

Wie bei der Motordrehzahl, im Kapitel 3.2.1, werden verschiedene Systemantworten mit verschiedenartigen Eingangssignalen aufgenommen. Nun wird aber für die Identifikation des Systems die aus der Drehzahlmessung bestimmte Kraft als Eingang verwendet. Auf diese Weise können Effekte, wie die langsamere Erreichung der Drehzahl, wie in Kapitel 3.2 angesprochen wurde, mit berücksichtigt werden. Ein weiterer Vorteil ist, dass zusätzliche Verzögerungen durch beispielsweise der Aktorik umgangen werden. Zusätzlich wird darauf geachtet, dass die verwendeten Eingangssignale möglichst die nicht linearen Effekte, die durch Haftreibung oder Verkeilen des Schlittens entstehen, vermeidet werden. Das heißt, es wird darauf geachtet, dass die resultierende Dynamik möglichst groß im Vergleich zu den Nichtlinearen Effekten wird. Dies hat auch den Vorteil, dass man auf diese Weise den schlechtesten Fall hinsichtlich der Dynamik erzeugt. Dieses Verhalten tritt nämlich beim geregelten System bei großen Sollwertveränderungen auf. Bei kleinen Sollwertveränderungen dürfte physikalisch gesehen sich der richtige stationäre Endwert einstellen.

### 3.3.4 Umsetzen eines digitalen Zwillings der Strecke

Basierend auf dem Streckenmodell  $G_{Mp}$  kann nun das Drehzahlregler-Modell aus Kapitel 3.2.4 erweitert werden.

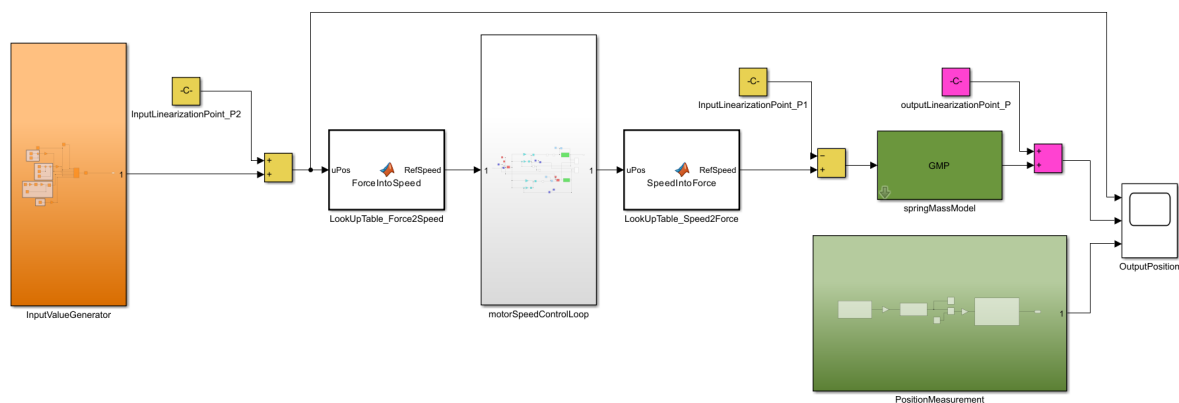


Abbildung 17: Simulink-Modell - OL\_Test\_w\_m.slx

Dabei dient der Drehzahlregler als Stellglied und liefert für das Streckenmodell  $G_{Mp}$  die erforderliche Eingangskraft. Für die Umrechnung werden die Look-Up-Table für die Umwandlung von Kraft zu Drehzahl „LookUpTable\_Force2Speed“ und umgekehrt „LookUpTable\_Speed2Force“ verwendet. Das in der Abbildung 17 dargestellte Simulink-Modell

verwendet für eine bessere Übersichtlichkeit Subsysteme für die Eingangssignalvorgabe „InputValueGenerator“ und der Motordrehzahlregelung „motorSpeedControlLoop“. In das Simulink-Modell wurden auch die Betriebspunkte, aus Kapitel 3.3.3, „InputLinearizationPoint\_P“ (in gelb) und „outputLinearizationPoint\_P“ (in pink) eingeführt.

### 3.3.5 Umsetzen des gewünschten Regelkreisverhaltens durch Polzuweisung

Es wird also ein Kaskadenregler gebildet. Das Stellglied des Systems ist die Motordrehzahlregelung, siehe Abbildung 18.

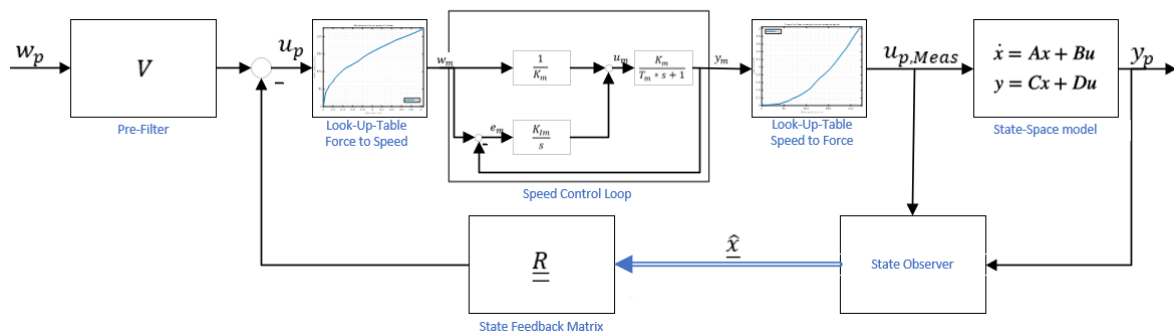


Abbildung 18: Kaskadenregelkreisstruktur

Für die Umsetzung wird nun erforderlich, geeignete Pole für das geschlossene System zu bestimmen.

Schlussendlich soll der geschlossene Regelkreis im Optimal-Fall wie in der folgenden Übertragungsfunktion aussehen als Gleichung (VIII)

$$G_{Wp}(s) = \frac{1}{\frac{1}{\omega_{0w}^2} * s^2 + 2 * \frac{D_w}{\omega_{0w}} * s + 1}$$

Das resultierende Polpaar aus dieser Übertragungsfunktion sieht wie folgt aus:

$$s_{p1,2} = -D_w * \omega_{0w} \pm \sqrt{1 - D_w^2} * \omega_{0w}$$

Um die zur Regelung verwendete Energie kleinzuhalten, wird versucht, die bestehenden Pole im Streckenmodell  $G_{Mp}$  möglichst wenig zu verändern. Dies ließe sich beispielsweise durch das alleinige Anpassen des Dämpfungsgrades erreichen. Dabei bleibt die Eigenkreisfrequenz des Systems gleich. Dies bedeutet in diesem Fall:

$$\omega_{0w} = \omega_0$$

Der Dämpfungsgrad wird so gewählt, dass sich das System den aperiodischen Grenzfallverhalten ähnelt. Somit gilt für den Dämpfungsgrad:

$$D_w = \frac{1}{\sqrt{2}}$$

Damit der Zustandsregler funktioniert, muss der zweite Zustand  $x_2$ , also die Geschwindigkeit der Masse, geschätzt werden.

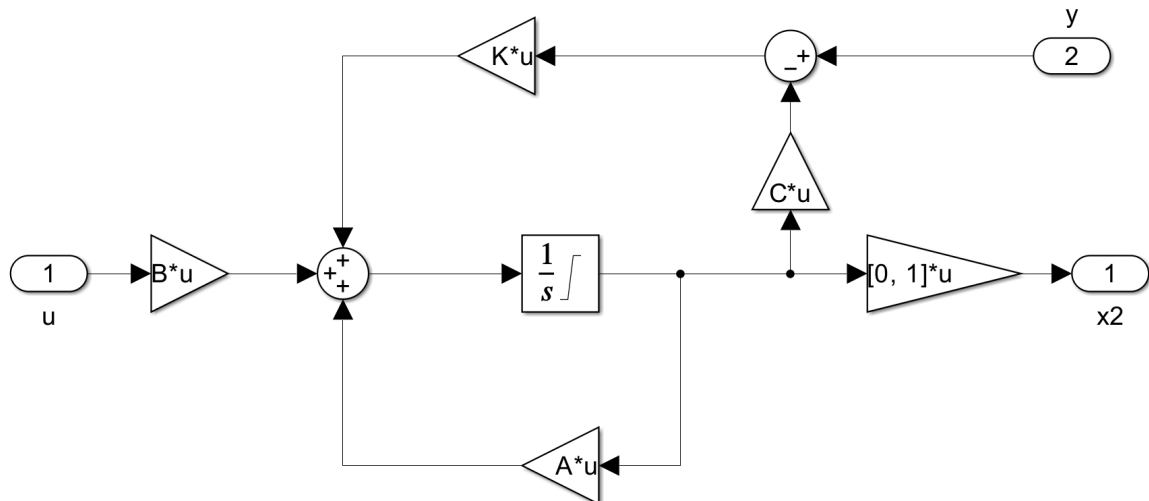


Abbildung 19: In Simulink umgesetzter Beobachter

Dies wird am besten mit einem Beobachter gemacht, siehe Abbildung 19, da dieser das starke Messrauschen unterdrückt.

### 3.4 Austesten der verschiedenen Regler hinsichtlich ihres Führungs- und Störkompensationsverhaltens

Es sollen folgende Regelkreisstrukturen getestet werden:

1. Offener Regelkreis
2. Vorgänger-PID-Regler aus Kapitel 2.2.4
3. PID-Regler mit niedrigerer Dynamik
4. Zustandsregler

Beim Testen der Regelkreisstrukturen wird folgendes Skript durchgegangen:

- Sekunde 0: Armierung des ESC, nach Kapitel 2.3.2
- Sekunde 5: Erster Sollwert von 1 cm vorgeben
- Sekunde 10: Ermitteln des Führungsverhaltens. Sollwert von 5 cm vorgeben
- Sekunde 20: Ermitteln des Störkompensationsverhaltens: Auslenkung simulieren

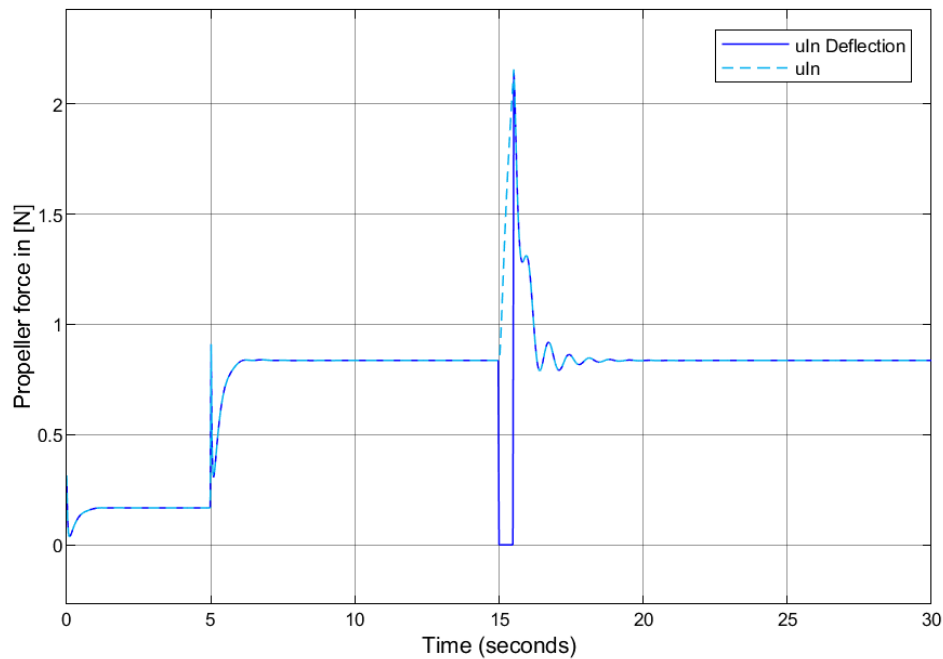


Abbildung 20: Beispielhafter Verlauf der Propellerstellkraft durch die Auslenksimulation

Für die Definition der Auslenkung wird davon ausgegangen, dass der Propeller nichts gegen die Auslenkung unternehmen kann. Um dies nachzuahmen, wird der BLDC-Motor für *0.5 Sekunden* ausgeschaltet. Des Weiteren erlaubt diese Vorgehensweise die Auslenkung zu simulieren. In Abbildung 20 wird beispielhaft der Verlauf des Stellsignals bei der beschriebenen Auslenksimulation gezeigt. Dabei beschreibt die türkis gestrichelte Kurve das Stellgrößenverhalten, was der Regler vorgeben möchte. Die blaue Volllinie wiederum zeigt das Stellsignal, dass wirklich kommandiert wird. Man kann sehen, dass in diesem Beispiel die Auslenksimulation ab Sekunde 15 beginnt.

Damit die verschiedenen Regler unter gleichen Bedingungen getestet werden können, sollen diese zuerst auf die Position 5 cm gebracht werden. Der Versuch kann auch beim offenen Regelkreis durchgeführt werden. Dafür wird der reine Drehzahlregler verwendet und mithilfe des Look-Up Tables die benötigte Kraft für die 5 cm-Position vorgegeben.

### 3.5 Weitere Untersuchungen

In Rahmen dieser Projektarbeit wurde versucht, mögliche Ursachen für das schlechte Störgrößenkompensationsverhalten zu analysieren.

#### 3.5.1 Der Einfluss des Plexiglasschutzes

Aufgrund des Plexiglasschutzes wird, wie in Kapitel 2.1.5 beschrieben, ein negativer Einfluss auf das geregelte System befürchtet. Es wird dazu folgende Hypothese aufgestellt:

Das Plexiglas verursacht durch beispielsweise Turbulenzen das Instabil werden des geregelten Systems bei einer Auslenkung von Hand.

Testen der Hypothese:

Wie in Kapitel 2.2.5 beschrieben, wurde der Sollwert auf  $5\text{ cm}$  vorgegeben. Es wurde mit und ohne Plexiglasschutz versucht durch Auslenkung das System instabil werden zu lassen.

Ergebnis:

Egal ob mit oder ohne Plexiglasscheiben, das System wird instabil bei einer Auslenkung von ca.  $10\text{ cm}$ .

Es konnte auch dokumentiert werden, dass die Stellkraft stark von der kommandierten Stellkraft abweicht, sobald eine Schwelle zu lange überschritten wird. In der Abbildung 21 wird gezeigt, wie die Stellkraft im Subplot 1 ansteigt. Im Subplot 2 sieht man, wie das identifizierte Streckenmodell aus Kapitel 2.2.4 das reale System maximal um ungefähr  $1\text{ cm}$  im Subplot 3 abweicht.

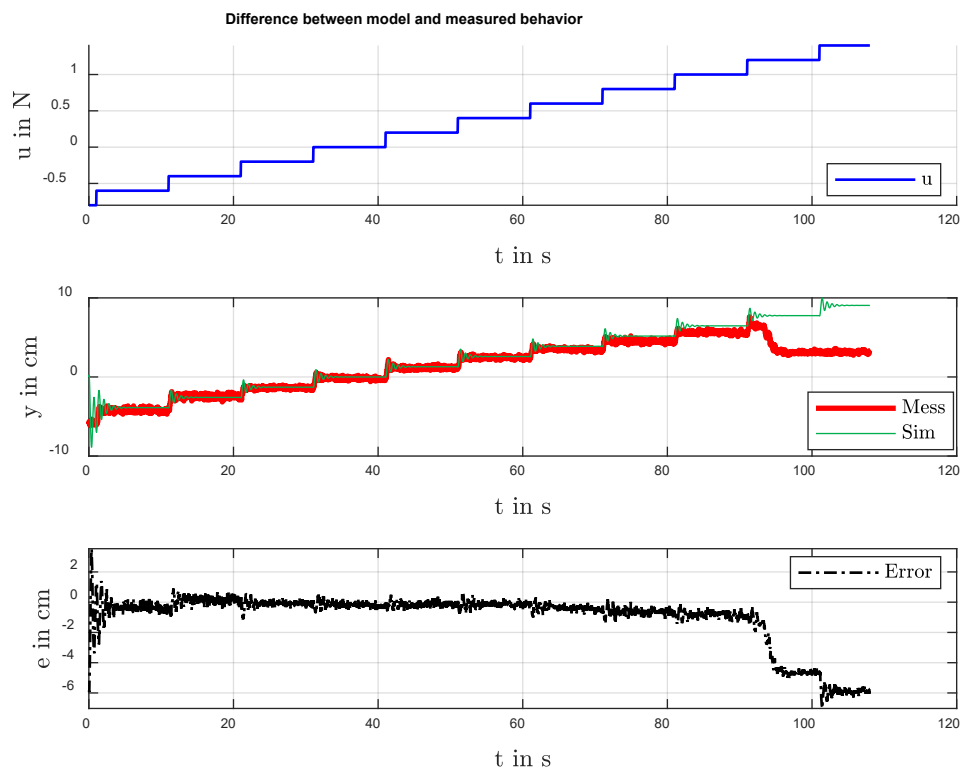


Abbildung 21: Stufenweiser Anstieg der Propellerstellkraft

Es sei erwähnt, dass bei diesen Untersuchungen in, Abbildung 21, andere Betriebspunkte, bei  $u_0 = 1\text{ N}$  und  $y_0 = 7\text{ cm}$  gewählt wurden und deshalb negative Stellkräfte und Positionswerte in der Abbildung 21 gezeigt werden. Dies bedeutet umgerechnet, dass ab einer vorgegebenen

Stellkraft von 2 N (alte Messweise aus [1]), die Leistung des BLDC-Motors stark reduziert wird und es zu einer Abweichung um 6 cm zum erwarteten Wert kommt. Dies führt beim geregelten System zu einem starken Schwingen.

### 3.5.2 Überprüfen der Stellkraft abhängig von der Position

Aufgrund der Befürchtung, dass der Plexiglasschutz einen Einfluss auf die Stellkraft abhängig von der Schlittenposition hat, werden Tests mit Gewichten unternommen. Dabei werden mit Gewichten, die Masse nach unten gedrückt. Auf diese Weise soll untersucht werden, ob bei gleicher vorgegebener Stellkraft, aufgrund des Zusammenziehens der Zugfedern, der zurückgelegte Weg gleich bleibt

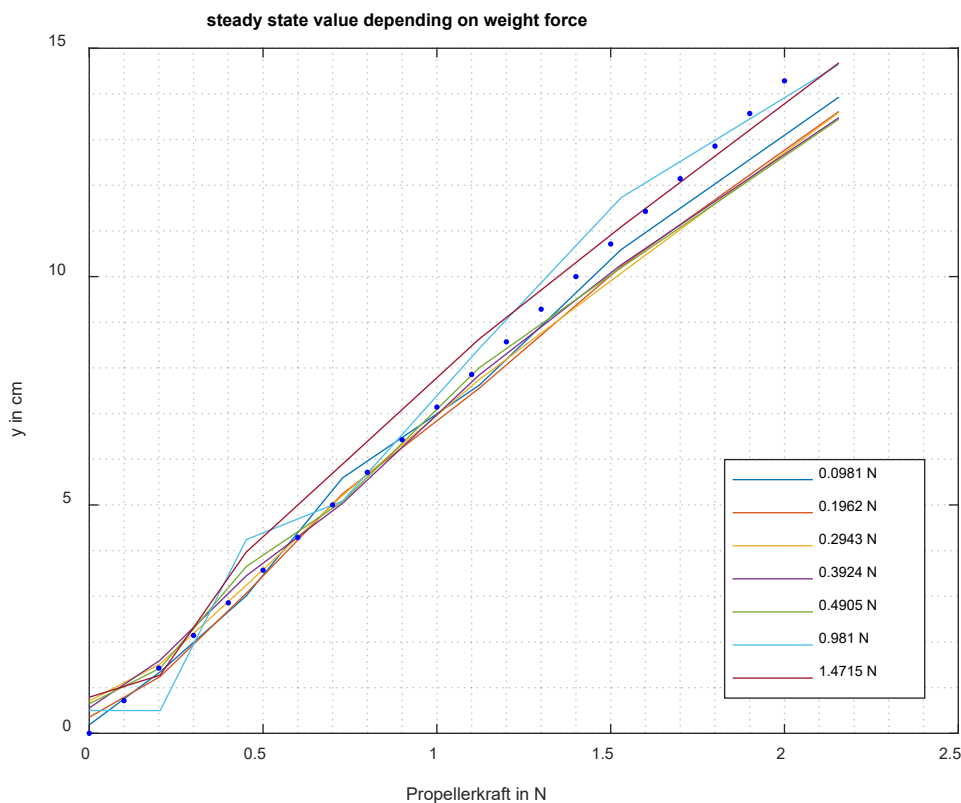


Abbildung 22: Auf 0 cm-normierte Positionswerte bei unterschiedlichen Gewichtskräften

In Abbildung 22 sieht man verschiedene Kurven geplottet, die das Anstiegsverhalten abhängig von der vorgegebenen (Propeller)Stellkraft zeigen. Bei jeder Kurve wurde die Masse um zusätzliches Gewicht erhöht. In der Legende sieht man die verschiedenen auf die Masse wirkenden Gewichtskräfte, die hinzukamen. Die Kurven wurden zur besseren Übersichtlichkeit auf die 0 cm-Position normiert. Verwendet wurden Gewichte im Bereich von 10 ... 150 g. Man kann sehen, dass die Abweichung zwischen den Geraden ungefähr 1 cm ergibt. Die gepunktete Linie soll die theoretische Federwegabnahme bei den vom Hersteller vorgegebenen

Federsteifigkeitswert von  $14 \frac{N}{m}$  darstellen. Da die Abweichung als gering eingeschätzt wird, wird die Propeller und Federkraft als ausreichen linear angenommen.

## 4 Ergebnis der technischen Aufrüstung -Fernbedienung

Die resultierende Fernbedienung wurde mithilfe eines FDM-Printers gefertigt, siehe Abbildung 23. Es wurde, wie in Abbildung 24 zu sehen ist, für die Analogwertvorgabe ein Potentiometer in Kombination mit einem Kondensator verwendet. Der Kondensator dient der Glättung des Analog-Signals. Es wurden zwei Taster hinzugefügt, für die Bedienung der Fernbedienung. Die Fernbedienung benötigt als Stromversorgung lediglich einen USB-Anschluss.



Abbildung 23: Vollständige Fernbedienung zusammengebaut

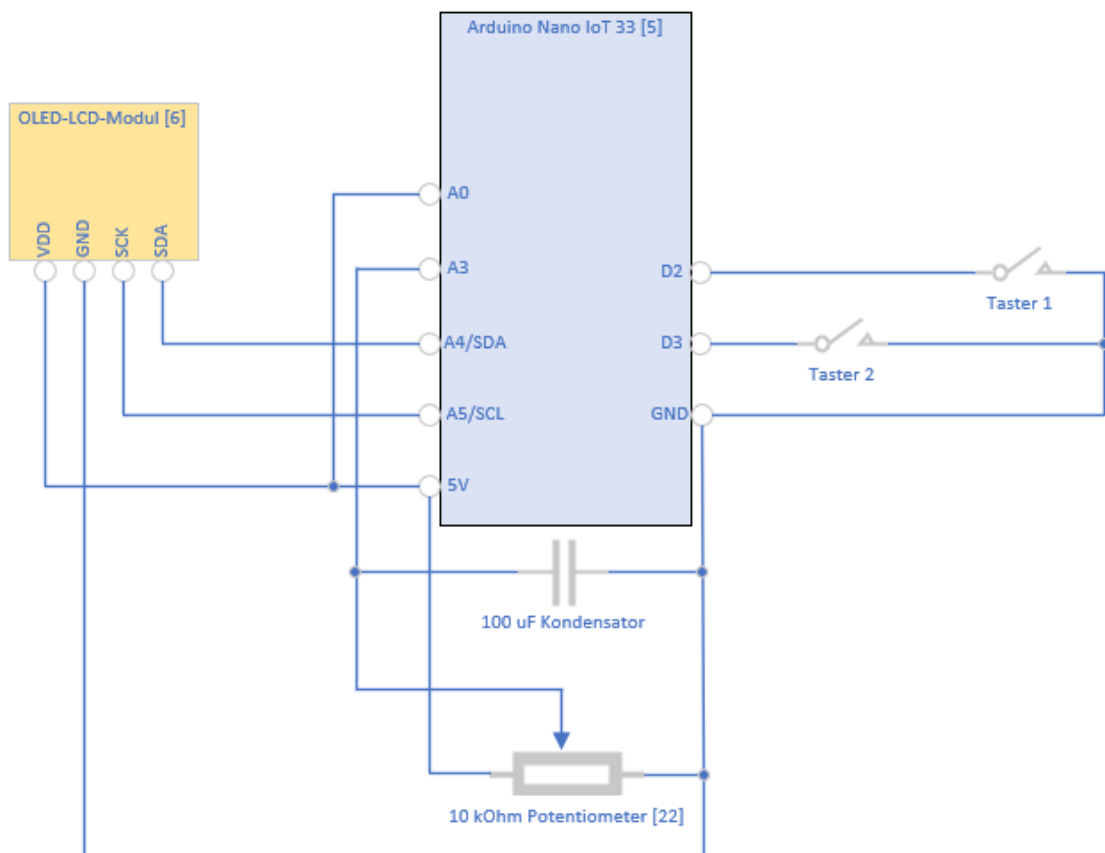


Abbildung 24: Schaltplan für Fernbedienung



Es wurde ein Arduino-Code für den in der Fernbedienung befindlichen Arduino Nano IOT 33 geschrieben. Der Arduino-Code ermöglicht es auf dem Display, siehe Abbildung 25, der Fernbedienung die Menü-Punkte „Open Loop“, „Closed Loop“ und „Shutdown“ auszuwählen.

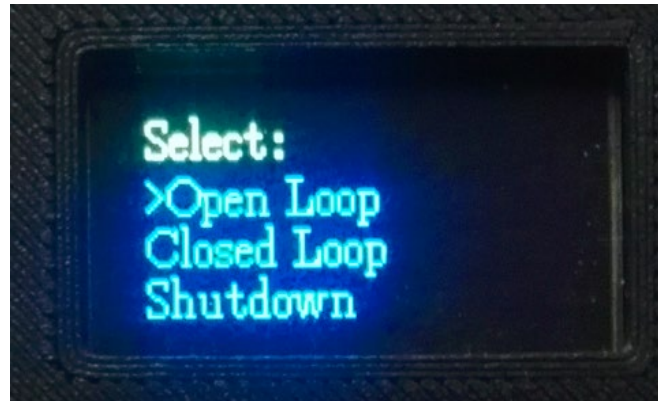


Abbildung 25: Display der Fernbedienung

Zusätzlich wurde eine UDP-Verbindungs-Schnittstelle implementiert, mit der sich die Fernbedienung mit dem Raspberry Pi Zero W verbinden kann. Bei der Auswahl „Open Loop“-Modus wird der Regelkreis geöffnet und die durch den Propeller erzeugte Kraft kann vorgegeben werden.

Der „Closed Loop“-Modus schließt den Regelkreis und man kann den Sollwert vorgeben. Um eine sprunghafte Änderung des Sollwerts zu vermeiden, wird der vorgegebene Sollwert mit  $0.01 * c$  multipliziert. Dies bedeutet eine Umrechnung von Position in Kraft. Die Masse bleibt deswegen ungefähr auf der Position von vorher.

Der „Shutdown“-Modus ermöglicht das Herunterfahren des Raspberry Pi Zero W von außerhalb. Für die Umsetzung des „Shutdown“-Modus wurde am Raspberry Pi Zero W ein 230 Ohm-Widerstand zwischen Pin 6 und Pin 26 verlötet. Auf diese Weise kann über Simulink programmiert werden, dass Pin 6 auf High geschaltet wird und damit im config.txt-File des Raspberry Pi Zero W, der Shutdown wegen der Schaltung auf High beim Pin 26 durchgeführt wird.

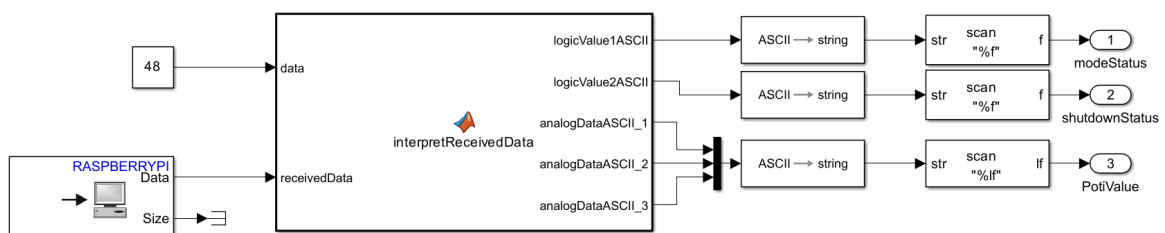


Abbildung 26: Simulink-Modell - Auswertung UDP-Daten

Es sei zu erwähnen, dass auf diese Weise momentan nur diese Möglichkeit bei der Stand-Alone Variante existiert, um den Raspberry Pi Zero W herunterzufahren. Der Shutdown-Knopf am Schlitten hat momentan keine Funktion mehr.

Über die UDP-Verbindung werden im ASCII-Format Daten transferiert, die in Simulink übersetzt werden müssen. Die in der Abbildung 26 gezeigte Zusammenstellung aus Blöcken, zeigt die Umsetzung. Es wird 1 Wert für die Bestimmung des Modus, offener oder geschlossener Regelkreis ausgegeben. Ein weiterer Wert wird für den Shutdown genutzt.

Der dritte Wert beschreibt, den Wert, den das Potentiometer ausgibt. Dieser Wert wird entweder in einen Positions-Sollwert oder in eine Stellkraft umgerechnet.

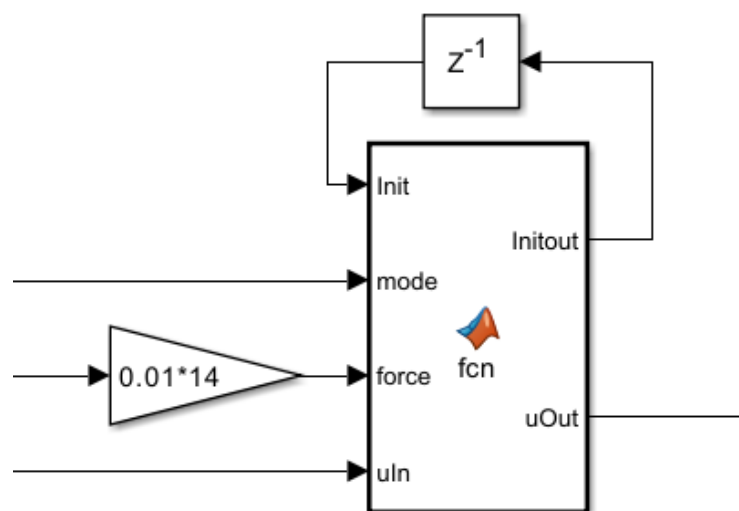


Abbildung 27: Simulink-Modell – Initialisierungsblock für Fernbedienung

Um einen sicheren Betrieb zu ermöglichen, wurde ein Initialisierungsphase für die Fernbedienung implementiert, siehe Abbildung 27. Erst nachdem das Schiebe-Potentiometer auf die unterste Position gebracht wird, startet das System. Momentan muss beachtet werden, dass diese Initialisierungsphase übersprungen wird, falls die Fernbedienung beim Start des Systems noch nicht in Betrieb war. Das bedeutet, dass die Position des Schiebe-Potentiometers sich direkt auswirkt.

## 5 Ergebnis der technischen Aufrüstung - Motordrehzahlregelung

### 5.1.1 Der implementierte Drehzahlsensor

Der resultierende Drehzahlsensor ist in der Abbildung 28 dargestellt. Baut man den Drehzahlsensor in den Schlitten ein, erhöht sich die Masse von 319 g auf 323 g, was einer Gewichtszunahme um 1.25% entspricht. Der Drehzahlsensor wurde aus dem RPM-Sensor [11], den Zählerbaustein [12], einem 10.5 kOhm-Widerstand und einem Breakout-Board zusammgebaut.

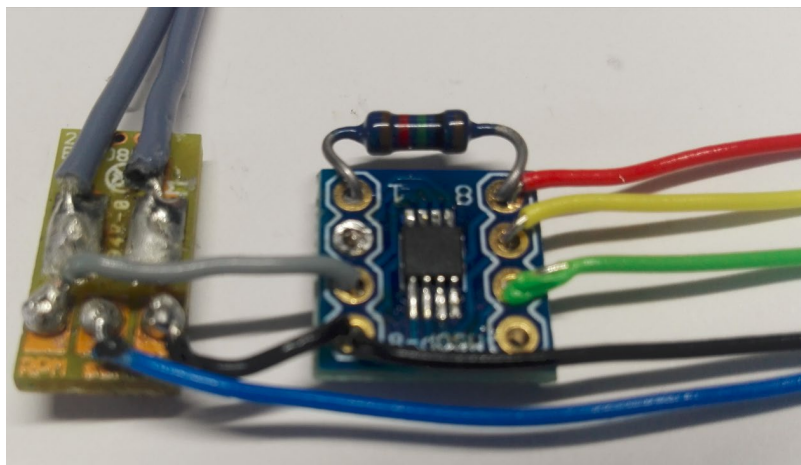


Abbildung 28: Drehzahlsensor zusammgebaut

In Simulink wurde ein Programm zur Auswertung der Motordrehzahl implementiert, siehe Abbildung 29. Um fehlerhafte Zählerwerte herauszufiltern, laufen zwei Filter im Hintergrund. Einmal wird überprüft, ob die maximale Geschwindigkeit überschritten wurde und beim anderen wird überprüft, ob die maximale Beschleunigung überschritten wird.

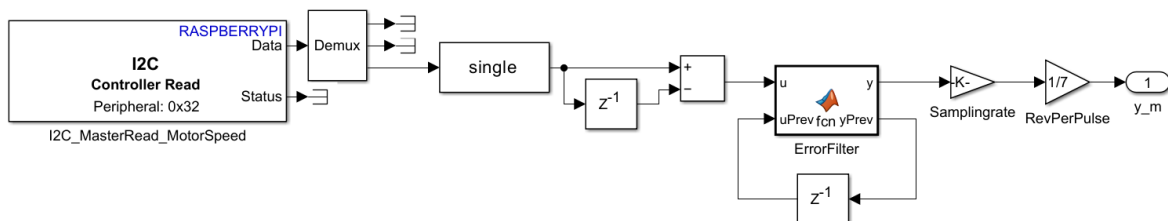


Abbildung 29: Simulink-Modell - Drehzahl-Auswertung

Für die maximale Geschwindigkeit werden Fehler herausgefiltert, die in einen Zählerstand resultieren, der, wie im Kapitel 3.1.2 beschrieben, um mehr als 54 höher geworden ist. Für die maximale Beschleunigung wiederum, werden Fehler herausgefiltert, die nach zwei Zählungen eine Zähler-Erhöungsänderung von 25 überschreitet. Rechnerisch bedeutet das, dass keine

Beschleunigung höher als  $3968 \frac{1}{s^2}$  gemessen werden darf, ohne dass ein Fehler aufgetreten ist. Dieser Wert basiert auf eine gemessene Sprungantwort der Motordrehzahl.

Die Filter geben im Falle einer fehlerhaften Messung, den letzten richtig gemessenen Wert aus.

### 5.1.2 Identifizieren des Motordrehzahlmodells

Es wurde eine Sprungantwort aufgenommen. Mit dieser wird das Streckenmodell identifiziert. Die Abbildung 30 zeigt das Sprung-Eingangssignal im oberen Subplot und die Sprungantwort im unteren Subplot. Von dem Eingangssignal wird 0.65 abgezogen und vom Ausgangssignal wird 128.34 1/s abgezogen, um den Betriebspunkt zu berücksichtigen. Auf diese Weise sind Ein- und Ausgang zuerst genullt. Es wird vereinfacht ein PT1-Strecken-Modell identifiziert:

$$G_{Mm} = \frac{K_m}{T_m * s + 1} = \frac{534.42}{0.0662 * s + 1}$$

### 5.1.3 Verifizieren des Motordrehzahlmodells

Das identifizierte Motordrehzahlmodell muss nun durch Abgleich mit andersartigen Eingangssignalen verifiziert werden.

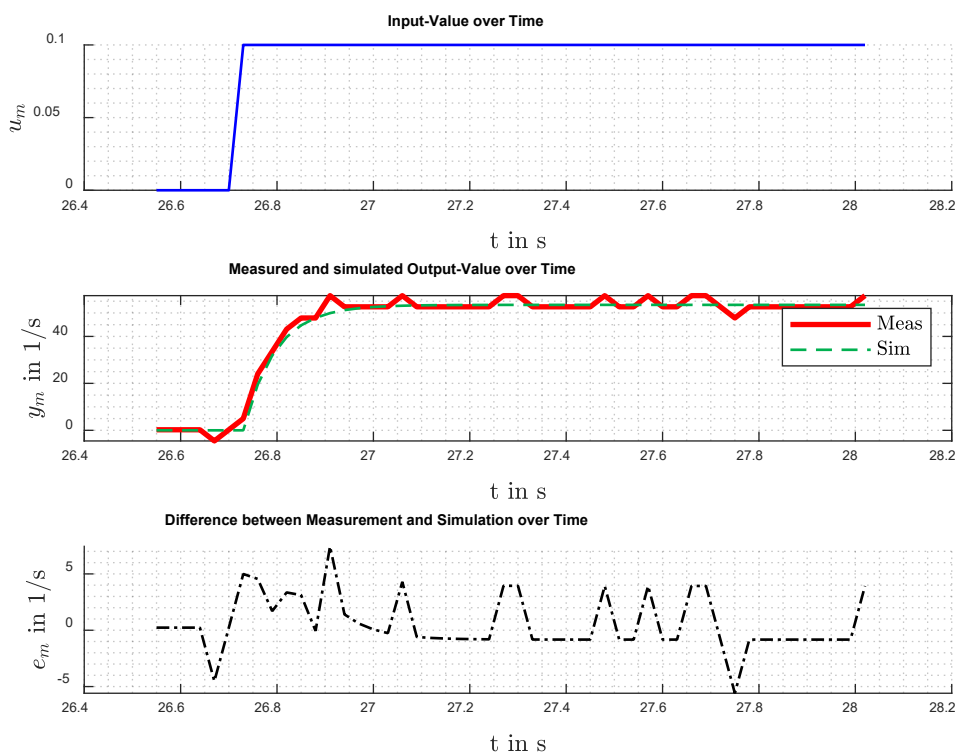


Abbildung 30: Sprungantwort der Motordrehzahl

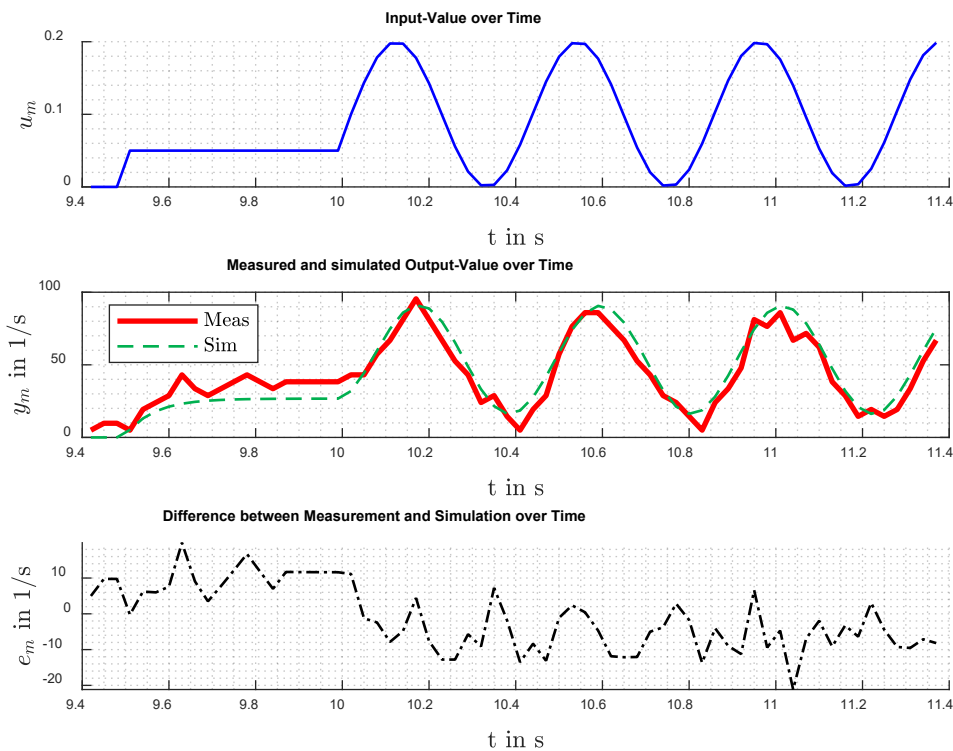


Abbildung 31: Systemantwort der Motordrehzahl bei einem sinusförmigen Eingangssignal

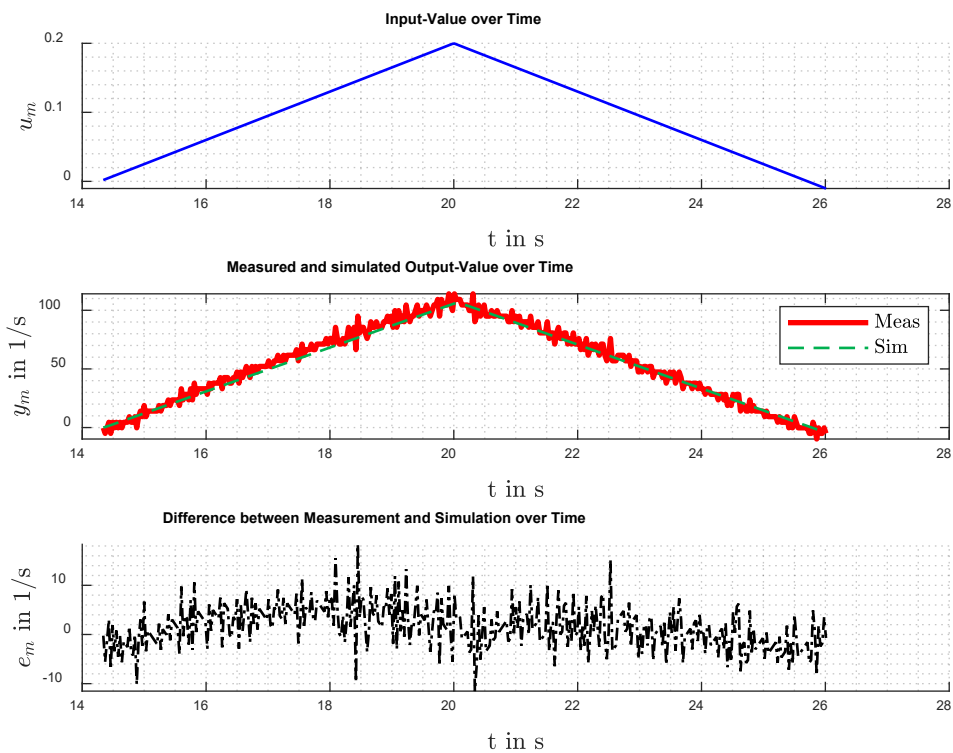


Abbildung 32: Systemantwort der Motordrehzahl, bei einer langsamen Eingangsrampe

Abbildungen 30 bis 32 zeigen jeweils im ersten Subplot das Eingangssignal, im zweiten Subplot die gemessene und simulierte Systemantwort und im dritten Subplot wird der absolute Fehler zwischen Messung und Simulation im Verlauf der Zeit dargestellt.

Die Systemantwort aus Abbildung 31 zeigt eine geringe Korrelation zwischen der Messung im Subplot 2 und des Simulationsfehlers im Subplot 3 im Zeitbereich, wo die Sinusschwingungen sich befinden. Dies weist auf einen geringen Phasenverschiebungsunterschied zwischen Modell und Messung hin. Es lässt sich also annehmen, dass die identifizierte Zeitkonstante ausreichend genau ist.

Die Systemantwort aus Abbildung 32 beschreibt das Verhalten des Systems bei einer langsamen Eingangsrampe hin und zurück, wie die in Kapitel 3.2.1. Die Zeitkonstante hat deswegen nur einen kleinen Einfluss auf das Systemverhalten. Auf diese Weise lässt sich das stationäre Verhalten des Modells verifizieren, denn der Unterschied zwischen Messung und Simulation im Subplot 3 befindet sich bis auf wenige Ausnahmen bei weniger als  $10 \text{ 1/s}$  und scheint Effekt des Messrauschens zu sein.

### 5.1.4 Auslegen und Testen des Motordrehzahlreglers bei verschiedenen Sollwertvorgaben

Mithilfe der Simulation aus Kapitel 3.2.4 lässt sich ein geeigneter Regelungsparameter  $K_{Im} = 0.005$  finden, mit dem die reale Regelung ausgetestet wird. Die Abbildungen 33 bis 35 zeigen die Reaktion des simulierten und des real gemessenen Reglers bei verschiedenen Sollwertvorgaben. Hierbei beschreibt der erste Subplot das simulierte und das in der Realität verwendete Stellsignal. Der zweite Subplot zeigt den Verlauf des Sollwerts, der Simulation und der Messung. Der dritte Subplot zeigt jeweils Simulationsfehler (Simulation Error) und den Unterschied zwischen Messung und Sollwert (Reference Error).

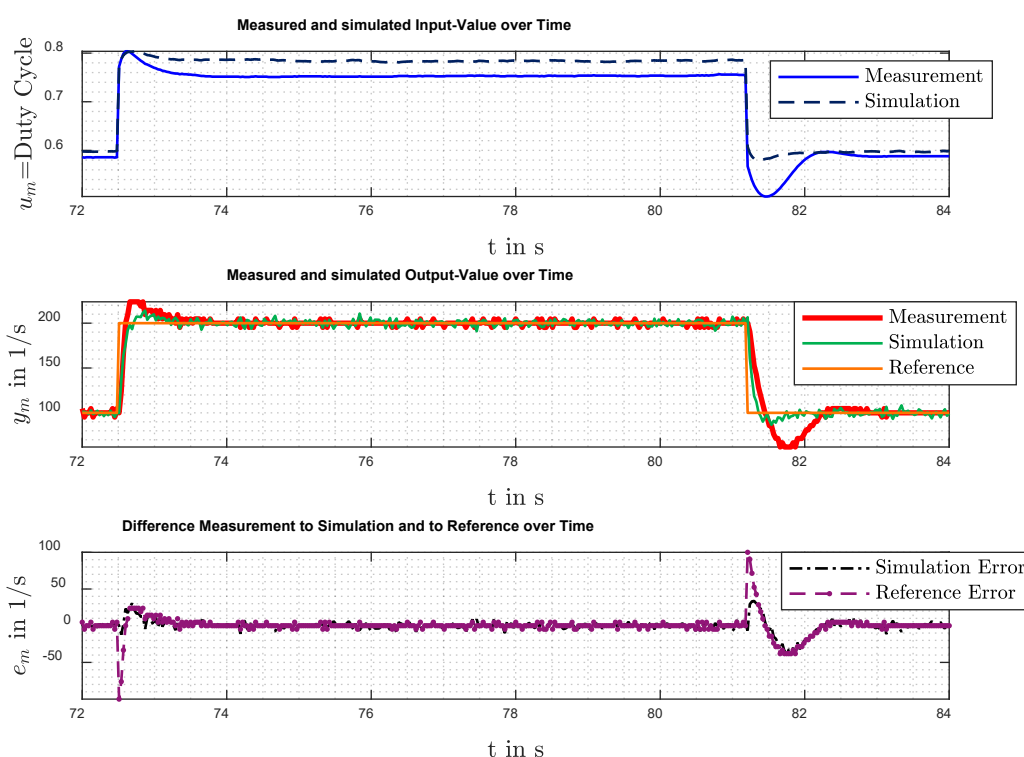


Abbildung 33: Drehzahlregler mit einem Sprung hin und zurück für die Sollwertvorgabe

In Abbildung 33 wird der gewünschte Sollwert sprunghaft auf eine Drehzahl von 200 1/s und wieder zurück auf 100 1/s gebracht. In beiden Fällen erreicht der Drehzahlregler seinen Sollwert. In der Messung schwingt das System weiter über als die Simulation. Beim Sollwertsprung zurück ist der Überschwinger fast doppelt so hoch wie beim Sollwertaufsprung. Dies lässt sich wegen der fehlenden aktiven Bremsung des ESC, wie in Kapitel 3.2 beschrieben, erklären. Daraus folgt, dass der Integral-Anteil aufgrund der trägeren Dynamik in die negative Drehzahlrichtung länger Zeit hat zu wirken. Dieser Effekt zeigt sich auch in Abbildung 34.

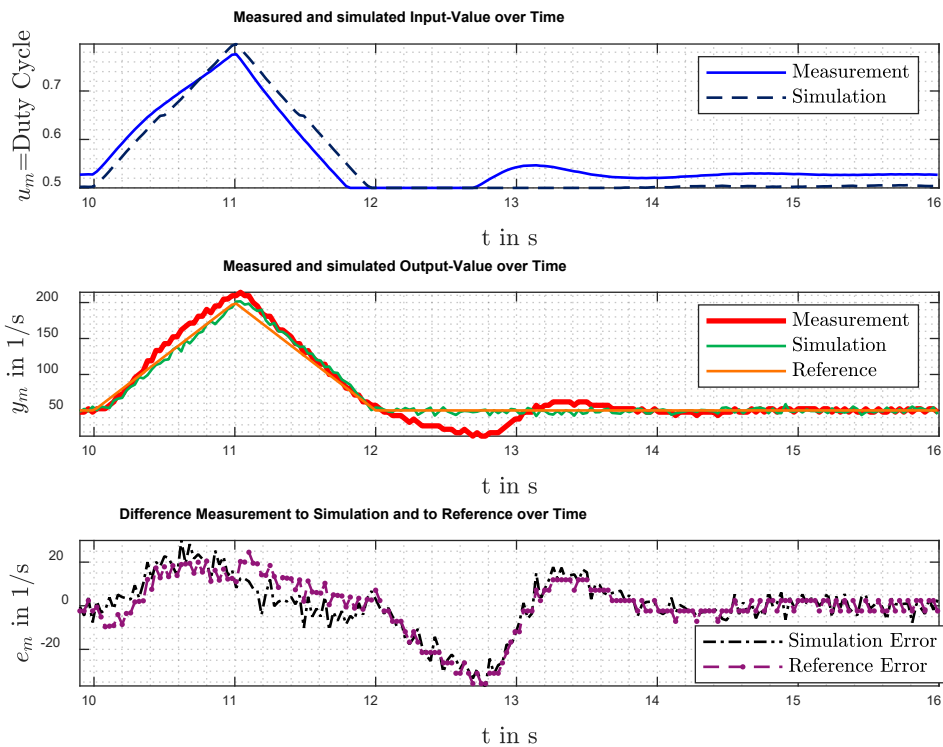


Abbildung 34: Drehzahlregler mit einer Rampe hin und zurück für die Sollwertvorgabe

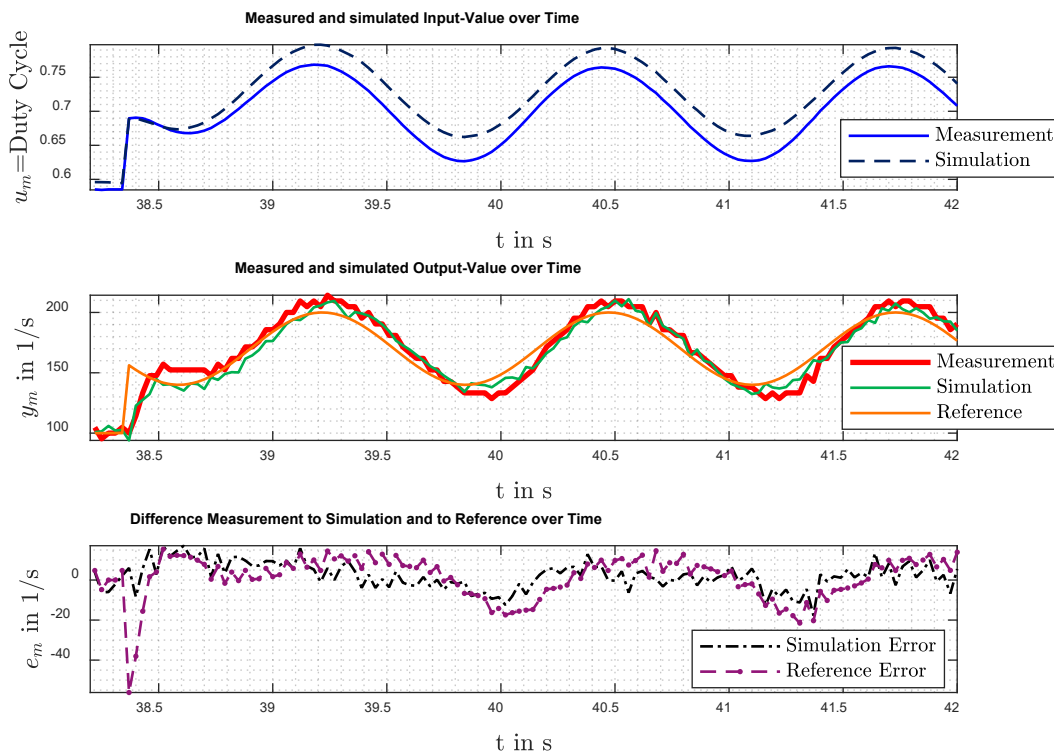


Abbildung 35: Drehzahlregler mit einer sinusförmigen Sollwertvorgabe



Hier muss aber erwähnt werden, dass hier in einem Drehzahlbereich gearbeitet wird, in der das System nicht linear wird. Schaut man sich im Subplot 1 den real verwendeten Tastgrad als Eingangssignal an, geratet dieses in die Stellgrößenbegrenzung von 0.5.

In Abbildung 35 wird ein Sinussignal als Sollwert vorgegeben. Dabei wurde darauf geachtet, dass das Stellsignal, wie in Kapitel 3.2.1 beschrieben, im linearen Bereich bleibt. Es lässt sich beobachten, dass der Istwert maximal um 20 1/s vom Sollwert abweicht.

Es wird davon ausgegangen, dass der hier implementierte Drehzahlregler für die weiteren Untersuchungen verwendbar ist.

### 5.1.5 Bestimmen der Look-Up-Table für die Berechnung der Stellkraft aus der Drehzahl

Es wird angenommen, dass bei konstanten Motor- bzw. Propellerdrehzahl, die erzeugte Stellkraft gleich bleibt. Basierend auf dieser Annahme wird ein Look-Up-Table durch schrittweise Erhöhen der Motordrehzahl um 10 1/s bestimmt. Abbildungen 36 und 37 zeigen grafisch die für die Motordrehzahlvorgabe und für die Stellkraft-Messung verwendete Look-Up-Table.

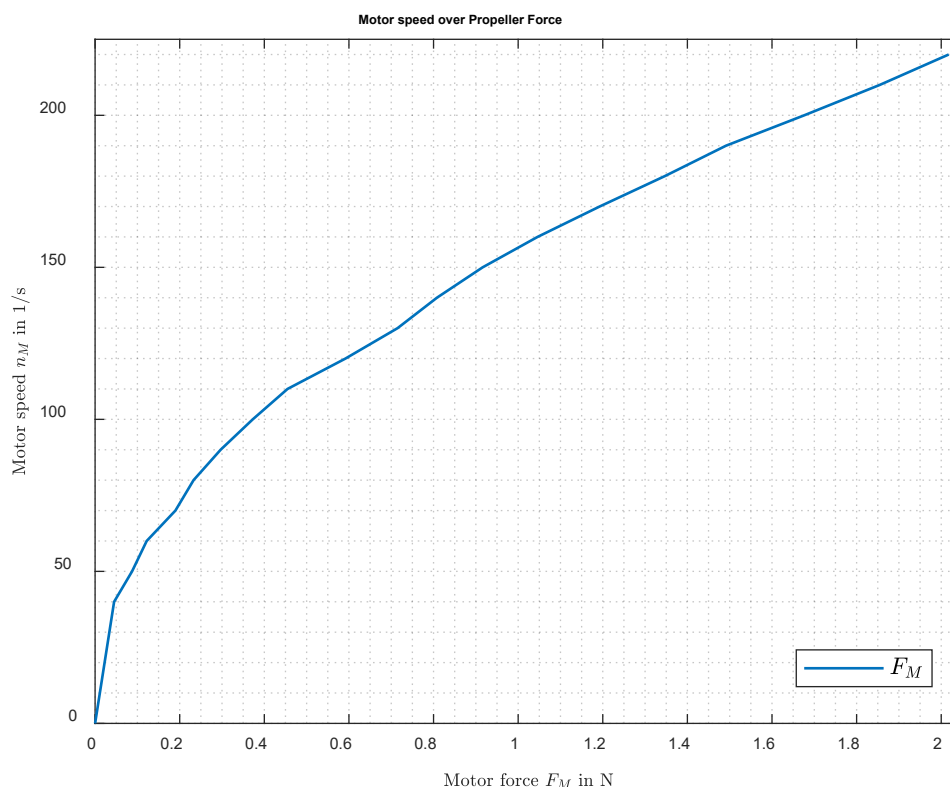


Abbildung 36: Look-Up-Table - Stellkraft zu Motordrehzahl

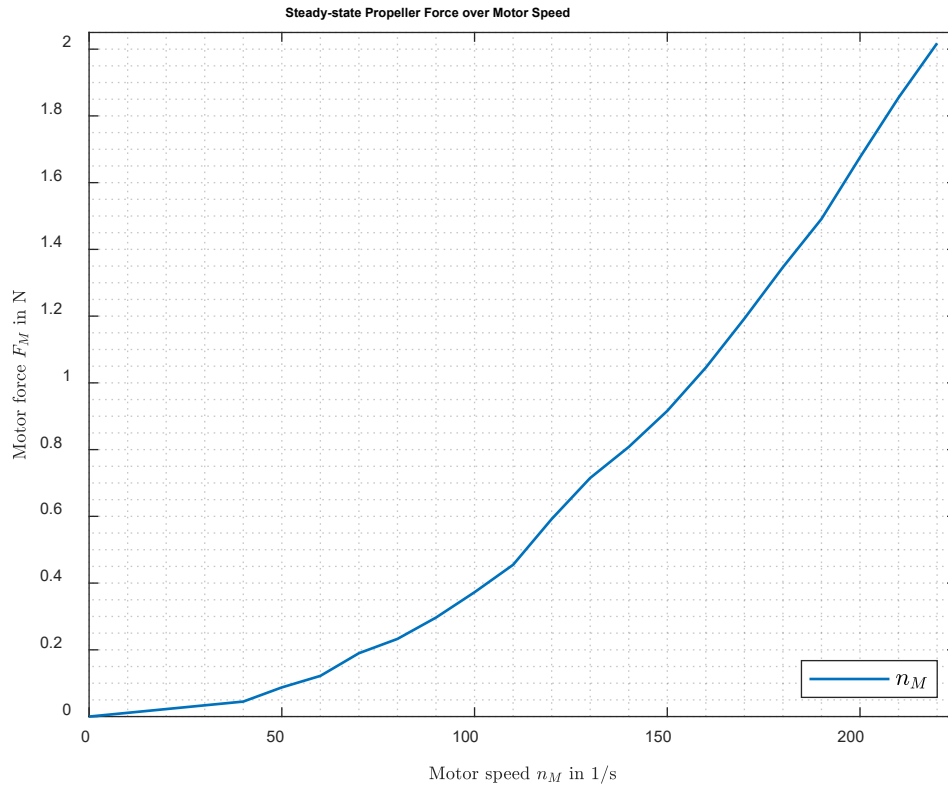


Abbildung 37: Look-Up-Table - Motordrehzahl zu Stellkraft

## 6 Ergebnisse der Verbesserung der Störkompensation

### 6.1.1 Identifizieren des Feder-Masse-Schwinger-Modells

Für die Identifikation des Feder-Masse-Modells wurde der Drehzahlregler als Stellglied verwendet. Es wird also, wie man in Abbildung 38 sehen kann, die gewünschte Kraft in eine gewünschte Motordrehzahl umgerechnet und so das physikalische System gesteuert.

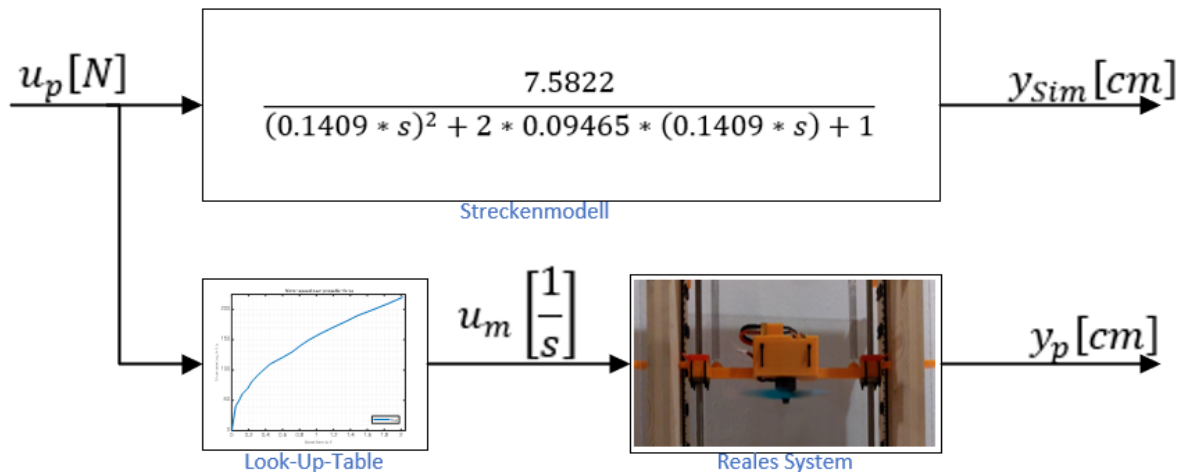


Abbildung 38: Grundlegende Beschreibung des Feder-Masse-Schwinger-Systems

Aus der gemessenen Drehzahl wird die tatsächlich an der Masse anliegende Stellkraft errechnet und als Eingangssignal verwendet. Auf diese Weise können die Effekte, wie in Kapitel 3.3.3 beschrieben, umgangen werden.

Der in Abbildung 39 gezeigte Verlauf wurde als Messdatensatz für die Identifikation verwendet. Es resultiert daraus folgendes Streckenmodell:

$$G_{Mp}(s) = \frac{7.5822}{(0.1409 * s)^2 + 2 * 0.09465 * (0.1409 * s) + 1}$$

### 6.1.2 Verifizieren des Feder-Masse-Schwinger-Modells

Die Abbildungen 39 bis 41 zeigen die Verläufe, der für die Verifikation gemessenen Messdatensätze. Es sei erwähnt, dass für die Identifikation die Betriebspunkte aus Kapitel 3.3.3 berücksichtigt werden und deswegen negative Kräfte und Positionen auftreten.

In den Beschriftungen der Abbildungen wird beschrieben, welche Kraft durch den Motordrehzahlregler erzeugt werden soll.

Subplot 1 zeigt jeweils die gemessene Stellkraft. Subplot 2 zeigt den Verlauf von der Systemantwort als Messung und Simulation. Subplot 3 zeigt den Unterschied zwischen Messung und Simulation.

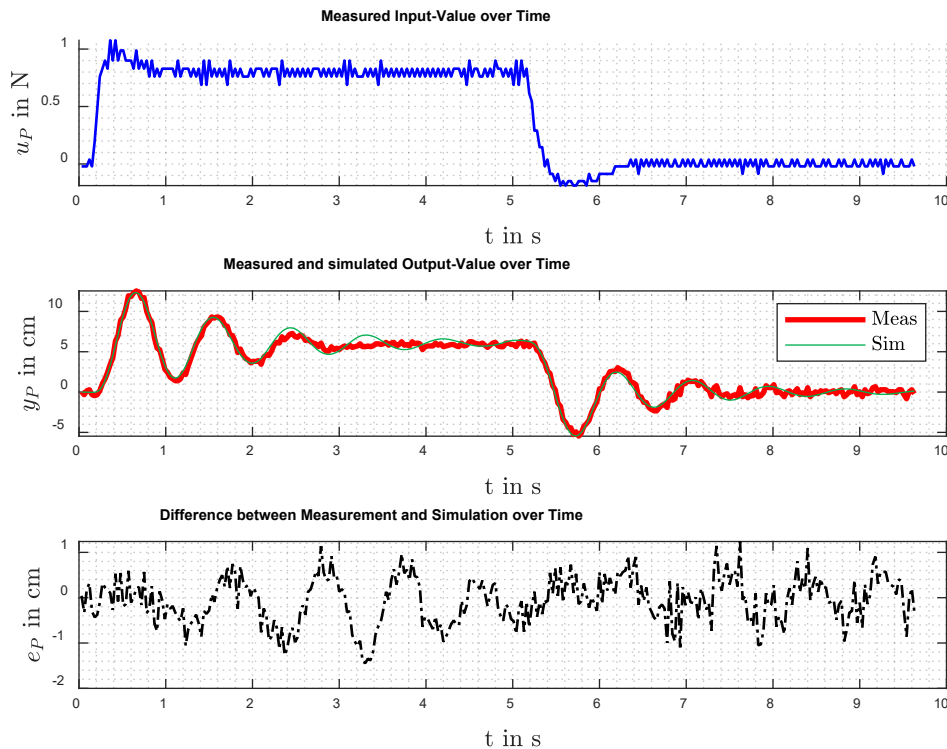


Abbildung 39: Sprungantwort des Feder-Masse-Schwingers

Die Verläufe der drei Abbildungen 39 bis 41 zeigen, dass sobald das gemessene reale System aus einer höheren in eine niedrigere Dynamik übergeht, nicht lineare Reibungskräfte das System stärker dämpfen als die Simulation berücksichtigt.

Das identifizierte System-Modell scheint das reale System ausreichend genau zu beschreiben. Durch Koeffizientenvergleich kann eine Zustandsraumbeschreibung erfolgen, gebildet aus Gleichung (VI) und (VII):

$$\dot{\underline{x}}(t) = \begin{bmatrix} 0 & 1 \\ -50.37 & -1.34 \end{bmatrix} * \underline{x}(t) + \begin{bmatrix} 0 \\ 381.92 \end{bmatrix} * u(t)$$

$$y(t) = [1 \quad 0] * \underline{x}(t)$$

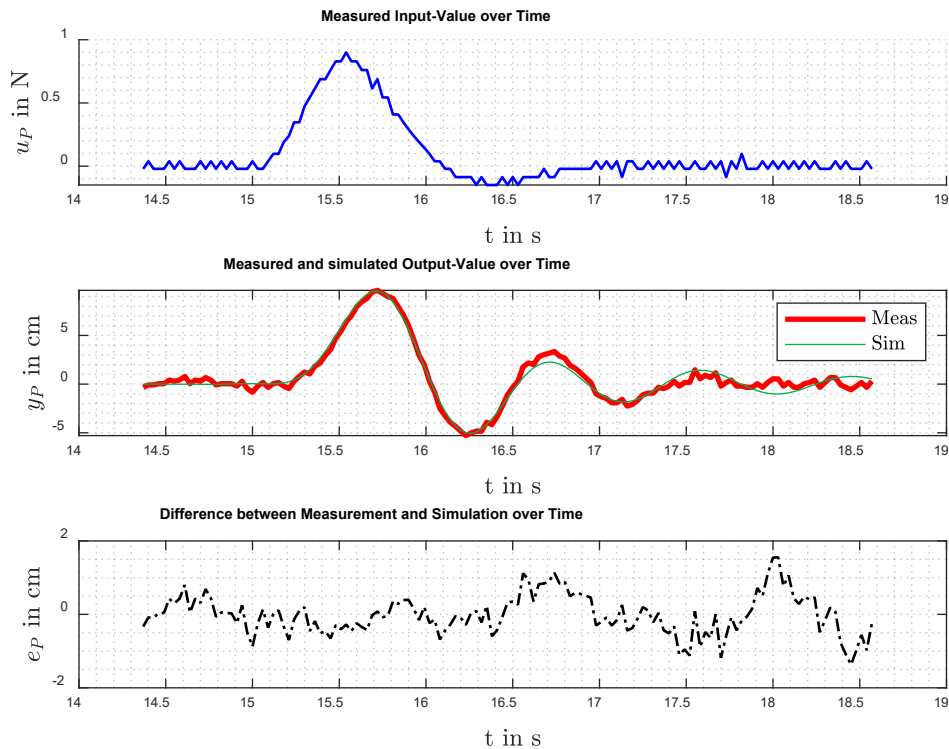


Abbildung 40: Rampe hin und zurück -Systemantwort des Feder-Masse-Schwingers

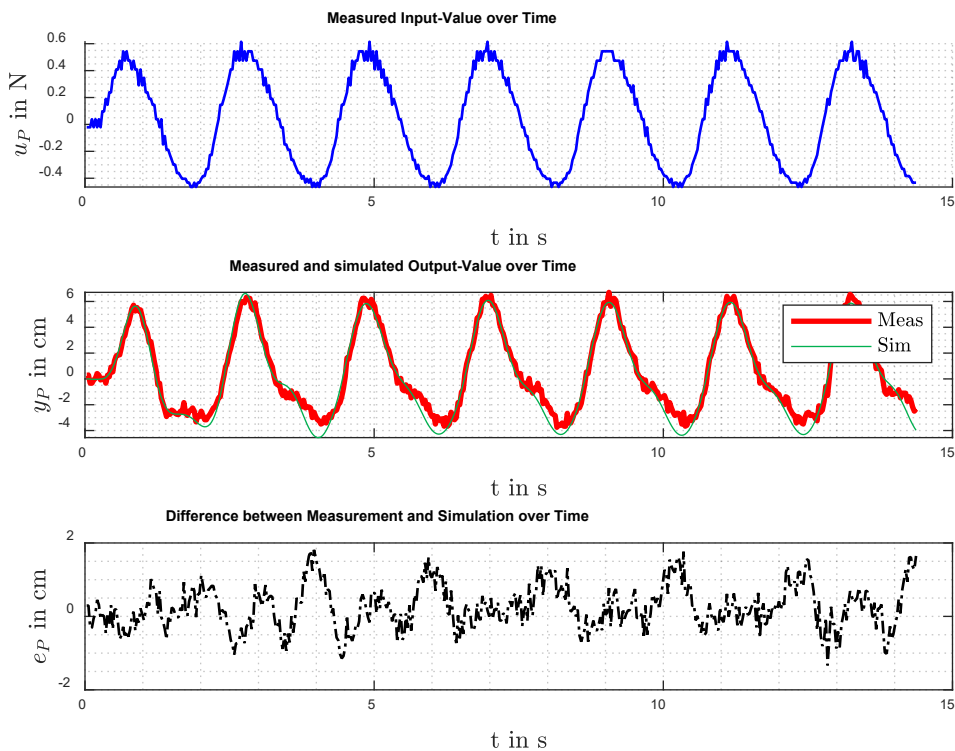


Abbildung 41: Sinus -Systemantwort des Feder-Masse-Schwingers

### 6.1.3 Auslegen eines dynamisch langsameren PID-Reglers

Nach dem Erklärungsansatz 3 in Kapitel 3.3.1 wurde ein PID-Regler gefunden.

Die Parameter dieses Reglers wurden iterativ gefunden:

$$K_P = 0.0001$$

$$K_I = 0.2$$

$$K_D = 0.002$$

$$T_1 = 0.03s$$

Das Finden dieser Parameter basiert auf der Annahme, dass der Proportional-Anteil des Reglers einen sehr geringen Einfluss hat. Basierend darauf wurden abwechselnd der Integral- und Differential-Anteil des Reglers in einer Simulation angepasst. Der Proportional-Anteil wurde auf einen sehr kleinen Wert reduziert. Dieser Regler wird in der Bewertung des Führungs- und Störkompensationsverhalten als „Regler 2“ bezeichnet.

### 6.1.4 Auslegen des Zustandsreglers

Mit den in Kapitel 3.3.5 definierten Polen kann eine Reglermatrix mithilfe der Matlab-Funktion `place()`, wie im Kapitel 2.3.6 gezeigt, bestimmt werden. Ebenso muss ein Vorfilter errechnet werden, dafür wird die Vorfilter-Gleichung aus Kapitel 2.3.6 herangezogen.

Es entsteht eine Reglermatrix:

$$\underline{\underline{R}} = [-0.0020 \quad 2.4159]$$

Und ein Vorfilter:

$$V = 13.998$$

Durch richtiges Auswählen der Beobachter-Pole kann das Messrauschen herausgefiltert werden. Durch Probieren wurden zwei gewünschte Pole bestimmt.

Mit jeweils den Wert:

$$s_{Obsv1,2} = -25.1326$$

Der Integrator-Block des Beobachters gibt Werte von  $[-1 \dots 1]$  aus. Simulationen haben gezeigt, dass der Integrator-Block nicht höhere Werte als 0.5 ausgibt.

Zur Berechnung der Beobachterrückführung  $K$  wird der in Kapitel 2.3.7 gezeigte Code verwendet.

$$K = \begin{bmatrix} 48.92 \\ 515.54 \end{bmatrix}$$

In der Abbildung 42 wird das komplette Kaskadenregler-Modell samt digitalen Zwillingen gezeigt.

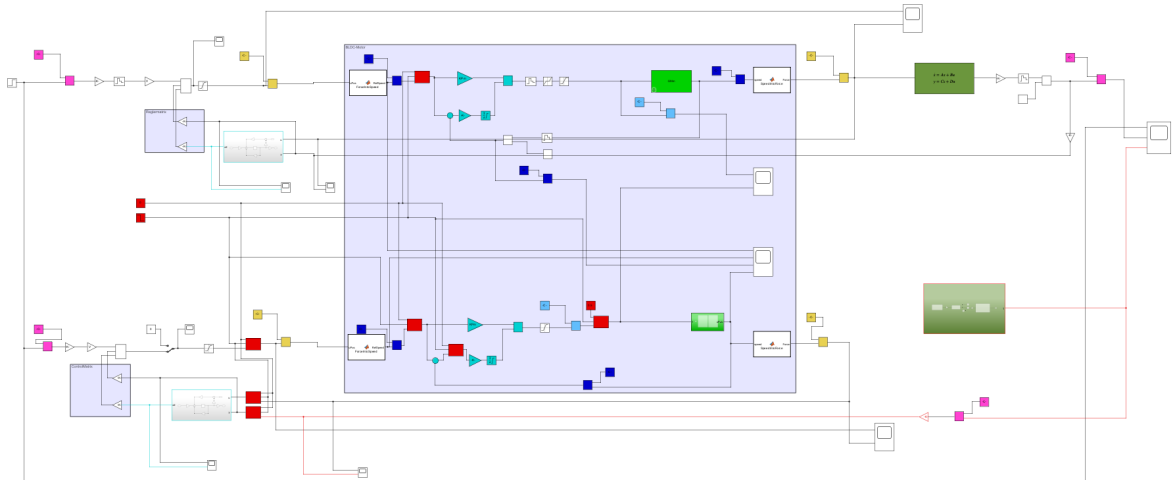


Abbildung 42: Simulink-Modell - SS\_ControlLoopComplete.slx

Um das Messrauschen zu simulieren, werden die Messdaten des unregulierten Systems mit einer Stellkraft von null, also bei einem ruhenden Propeller genutzt, um die Varianz auszurechnen. Abbildung 6 zeigt, wie sich das tatsächliche Messrauschen verhält. Die Varianz wird, wie in Kapitel 2.3 erwähnt, mit der Matlab-Funktion `var()` aus dem Messdatensatz berechnet. Als Ergebnis kommt eine Varianz von 0.04284 heraus. Simuliert wird das Messrauschen mit dem "random number"-Block in Simulink. Wobei ein Seed und ein Mittelwert von null gewählt wird.

### 6.1.5 Testen der verschiedenen Positionsregler auf Führungsverhalten

In den folgenden zwei Kapitel werden die Messdaten, die basierend auf Kapitel 3.4 gemessen wurden, bewertet. Getestet wurde der offene Regelkreis (OL), der Regler aus Kapitel 2.2.4 Regler 1 (R1), der Regler aus Kapitel 6.1.3 Regler 2 (R2) und der Zustandsregler aus Kapitel 6.1.4 (SS).

Es werden die 4 Regelkreisstrukturen nach den Bewertungskriterien aus Tabelle 1 bewertet.

Tabelle 1: Bewertungskriterien

Anstiegszeit ( $\epsilon \leq 0.4 \text{ cm}$ )	Zeit bis der Istwert in das Toleranzband von $0.4 \text{ cm}$ zum ersten Mal gerät
Überschwingweite %	Wie weit der Istwert beim Überschwingen den stationären Endwert überschritten hat in %
Einschwingzeit ( $\epsilon \leq 0.4 \text{ cm}$ )	Zeit bis Istwert dauerhaft in das Toleranzband von $0.4 \text{ cm}$ gerät
Einschwingzeit Simulation ( $\epsilon \leq 0.4 \text{ cm}$ )	Zeit bis Istwert dauerhaft in das Toleranzband von $0.4 \text{ cm}$ in der Simulation gerät
Bleibende Regelabweichung	Regelabweichung nach langer Zeit

In den Abbildungen 43 bis 46 werden die Führungsverhalten der 4 Regelkreisstrukturen gezeigt. Dabei starten alle von der Position  $1 \text{ cm}$ . Ab  $t = 10 \text{ s}$  wird ein Sollwertsprung auf  $5 \text{ cm}$  durchgeführt. Von diesem Zeitpunkt aus beginnt die Bewertung, wie in Tabelle 1 beschrieben.

Bei Abbildung 43 wird der Messdatenverlauf des offenen Regelkreises (OL) dargestellt. Hier zeigt der Subplot 1 den Stellgrößenverlauf. Subplot 2 zeigt den gemessenen Ausgangsverlauf. Bei den Abbildungen 44 bis 46 werden der Messdatenverlauf von R1, R2 und SS gezeigt. Dabei gilt bei allen, dass im Subplot 1 die kommandierte Stellkraft für das reale System und die Simulation zu sehen ist. In Subplot 2 kann das jeweilige Ausgangsverhalten der Simulation, der Messung und des Sollwerts gesehen werden. Der Subplot 3 zeigt jeweils die Differenz zwischen Sollwert und Istwert (Reference Error) und die Differenz zwischen Messung und Simulation (Simulation Error).



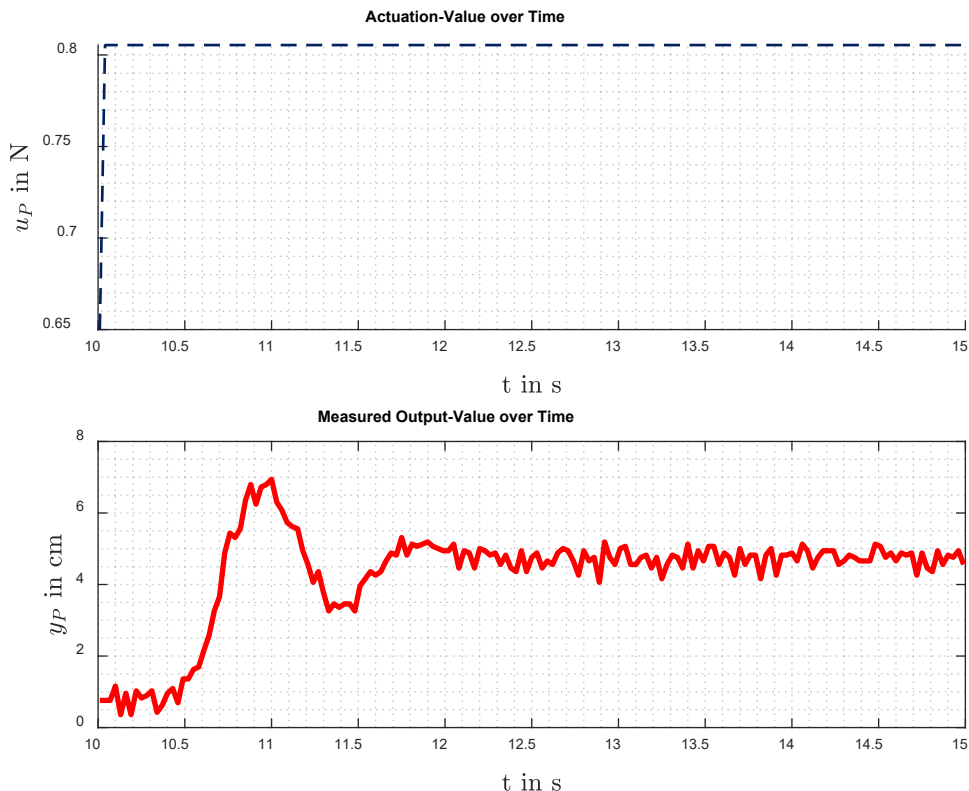


Abbildung 43: Führungsverhalten des offenen Regelkreises (OL)

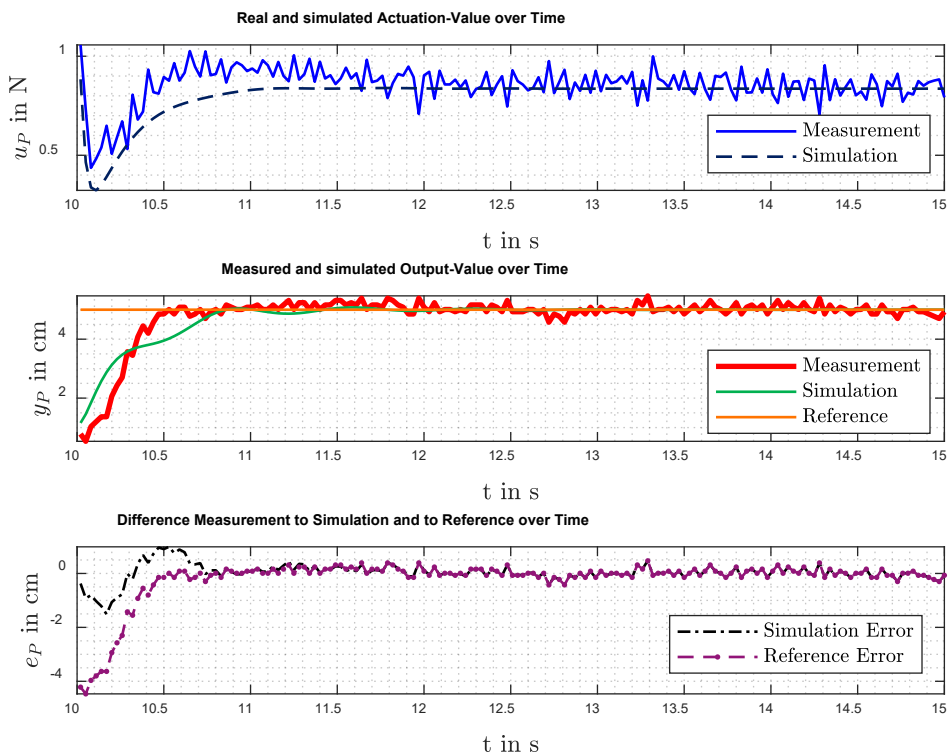


Abbildung 44: Führungsverhalten des Reglers 1 (R1)

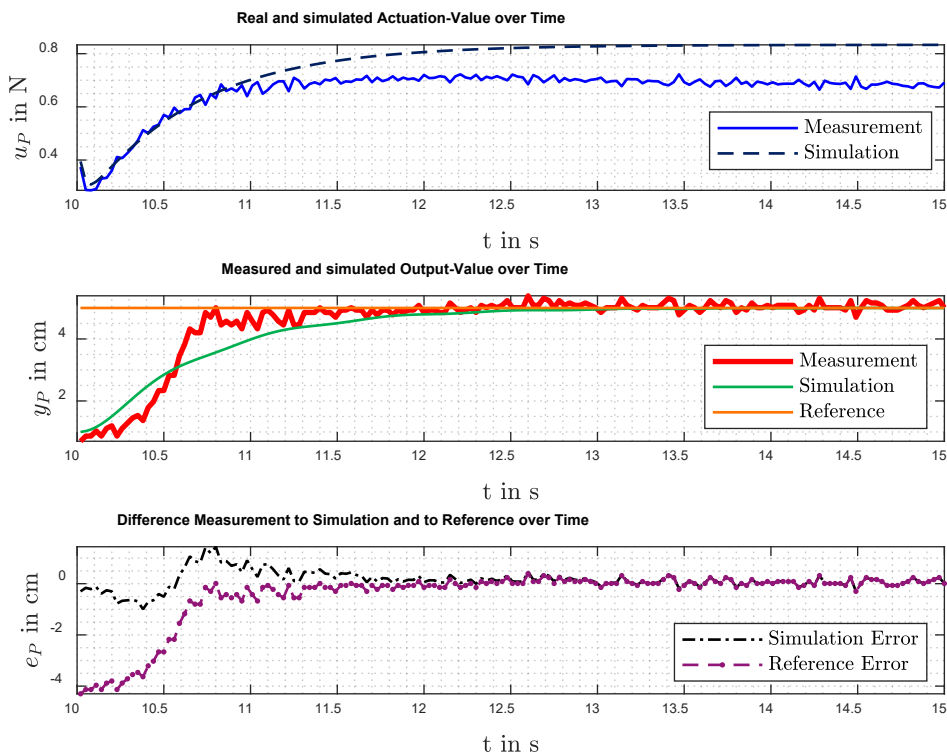


Abbildung 45: Führungsverhalten des Reglers 2 (R2)

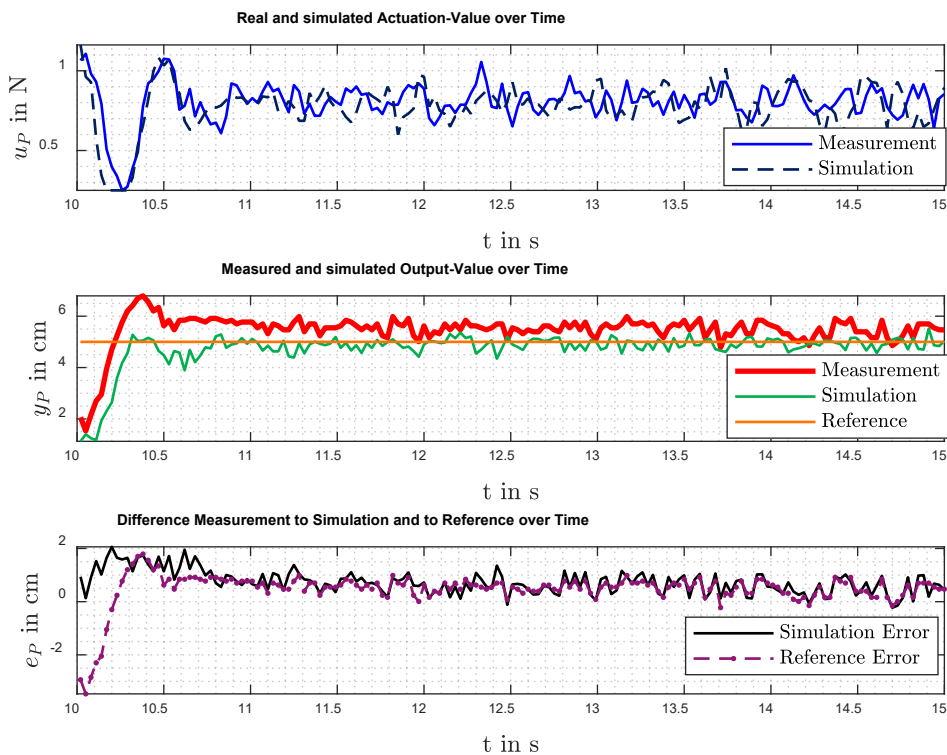


Abbildung 46: Führungsverhalten des Zustandsreglers (SS)

In Tabelle 2 können nun die 4 Regelkreisstrukturen anhand der Bewertungskriterien beurteilt werden. Dabei lassen sich verschiedene Beobachtungen machen.

- Die niedrigste Anstiegszeit und mehr als doppelt so schnell wie der zweitschnellste Regler ist der Zustandsregler (SS)
- R1 und R2 haben keine Überschwingweite
- Die niedrigste Einschwingzeit hat Regler 1 (R1)
- Der Regler SS hat eine bleibende Regelabweichung von  $0.51 \text{ cm}$ . Dies könnte von einer fehlerhaften Zustandsraumbeschreibung kommen

Tabelle 2: Vergleich zwischen der Positionsregler hinsichtlich des Führungsverhaltens

Regler	OL	R1	R2	SS
Anstiegszeit ( $\epsilon \leq 0.4 \text{ cm}$ )	0.73 s	0.44 s	0.71 s	0.20 s
Überschwingweite %	48.5 %	0 %	0 %	28.4 %
Einschwingzeit ( $\epsilon \leq 0.4 \text{ cm}$ )	1.66 s	0.44 s	0.71 s	0.50 s
Einschwingzeit Simulation ( $\epsilon \leq 0.4 \text{ cm}$ )	–	0.71 s	1.62 s	0.29 s
Bleibende Regelabweichung	0 cm	0 cm	0 cm	0.51 cm

In einem Bodediagramm kann im Frequenzbereich ein Vergleich zwischen offener Regelkreis und den geschlossenen Regelkreisen gezogen werden.

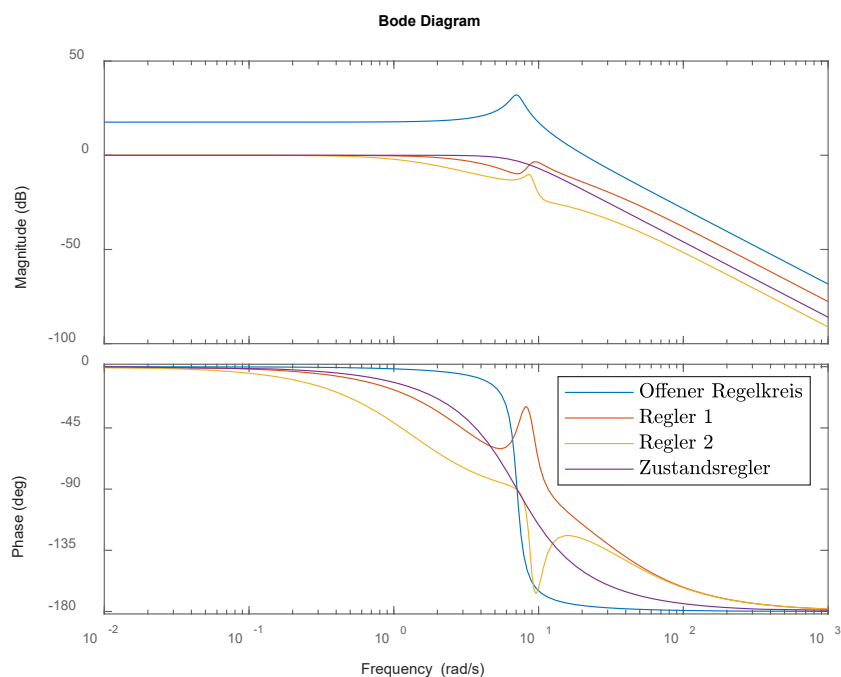


Abbildung 47: Bodediagramm von OL, R1, R2, SS

In der Abbildung 47 kann man sehen, dass die Amplitude bei den geschlossenen Regelkreisen bis zur Eckfrequenz bei  $0\text{ dB}$  liegt, was eine stationäre Verstärkung von 1 bedeutet. Im offenen Regelkreis gibt es eine Amplitude  $> 0\text{ dB}$ , daraus folgt, dass das Eingangssignal so angepasst werden muss, um den gewünschten Wert zu erreichen. Im Phasendiagramm lässt sich feststellen, dass die Bandbreite der geschlossenen Regelkreise ungefähr  $10\times$  kleiner ist, als die des offenen Regelkreises. Die geschlossenen Regelkreise werden bei keiner Frequenz positiv verstärkt. Die Regler 2 wird bei höheren Frequenzen am schnellsten gedämpft und Phasenverschoben.

### 6.1.6 Testen der verschiedenen Positionsregler auf Störkompensation

In den Abbildungen 48 bis 51 werden die Störkompensationsverhalten der 4 Regelkreisstrukturen gezeigt. Dabei starten alle von der Position  $5\text{ cm}$ . Ab  $t = 20\text{ s}$  wird dem BLDC-Motor ein Tastgrad von 0.5 als Eingangssignal kommandiert, siehe Kapitel 3.4.

Von dem Zeitpunkt  $20.5\text{ s}$  aus beginnt die Bewertung wie in Tabelle 1 beschrieben.

Bei Abbildung 48 wird der Messdatenverlauf des offenen Regelkreis (OL) dargestellt. Hier zeigt der Subplot 1 den gemessenen Stellgrößenverlauf. Subplot 2 zeigt den gemessenen Ausgangsverlauf. Bei den Abbildungen 49 bis 51 werden der Messdatenverlauf von R1, R2 und SS gezeigt. Dabei gilt, außer in Abbildung 51, dass im Subplot 1 die kommandierte Stellkraft für das reale System und die Simulation zu sehen ist. In Abbildung 51 wird im Subplot 1 die gemessene Stellkraft gezeigt. In Subplot 2 kann das jeweilige Ausgangsverhalten der Simulation, der Messung und des Sollwerts gesehen werden. Der Subplot 3 zeigt jeweils die Differenz zwischen Sollwert und Istwert (Reference Error) und die Differenz zwischen Messung und Simulation (Simulation Error).

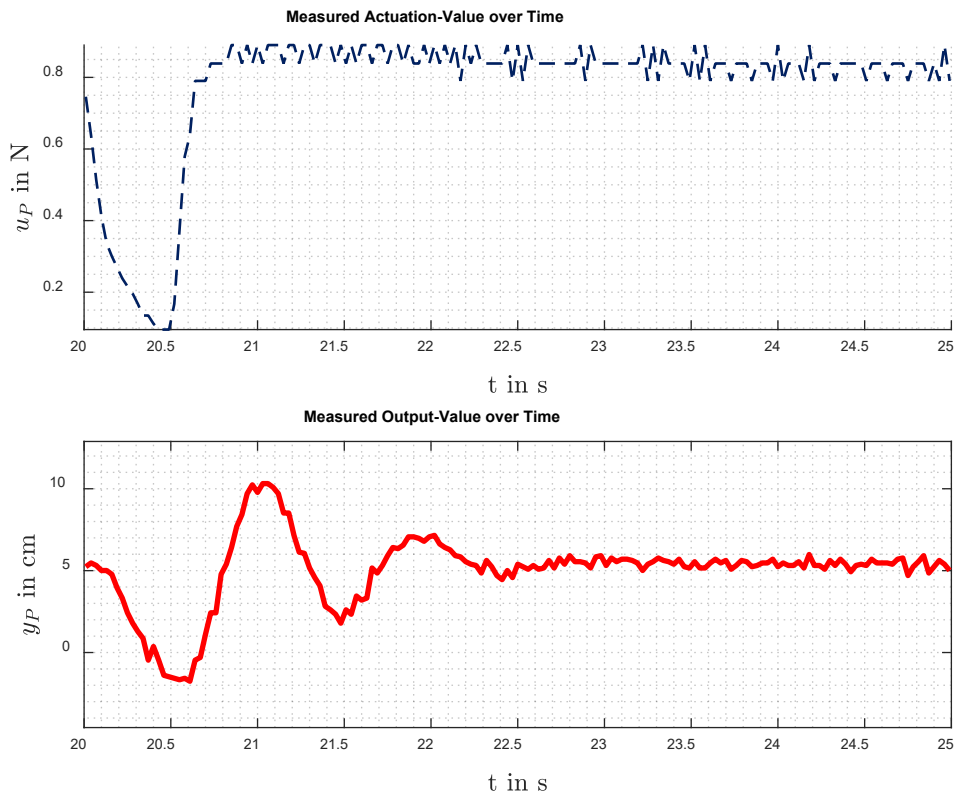


Abbildung 48: Störgrößenkompensation des offenen Regelkreises (OL)

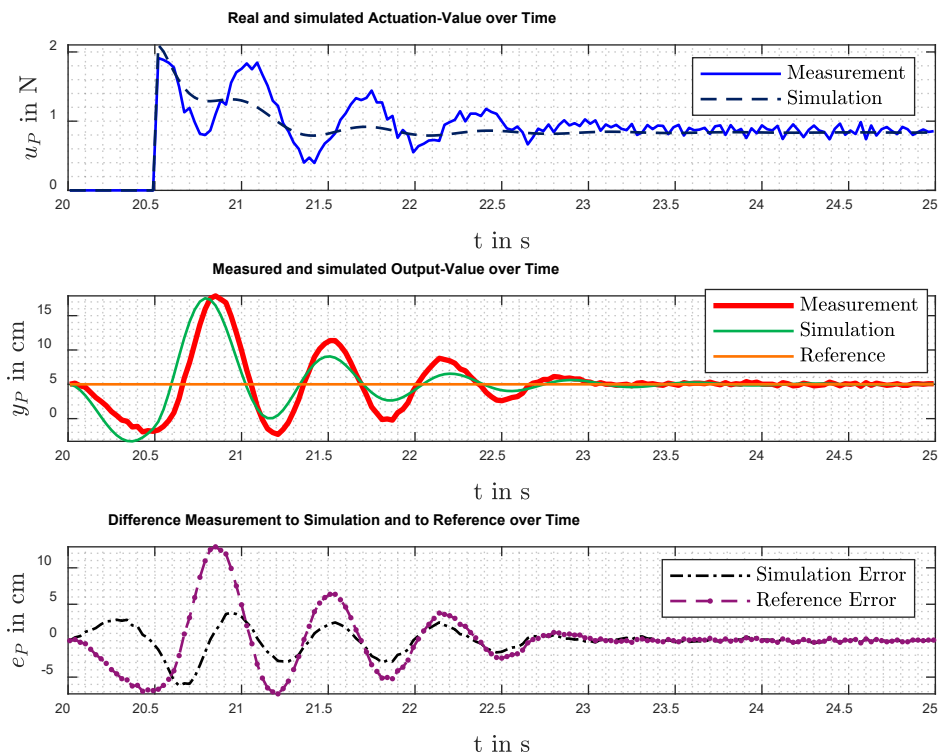


Abbildung 49: Störgrößenkompensation des Reglers 1 (R1)

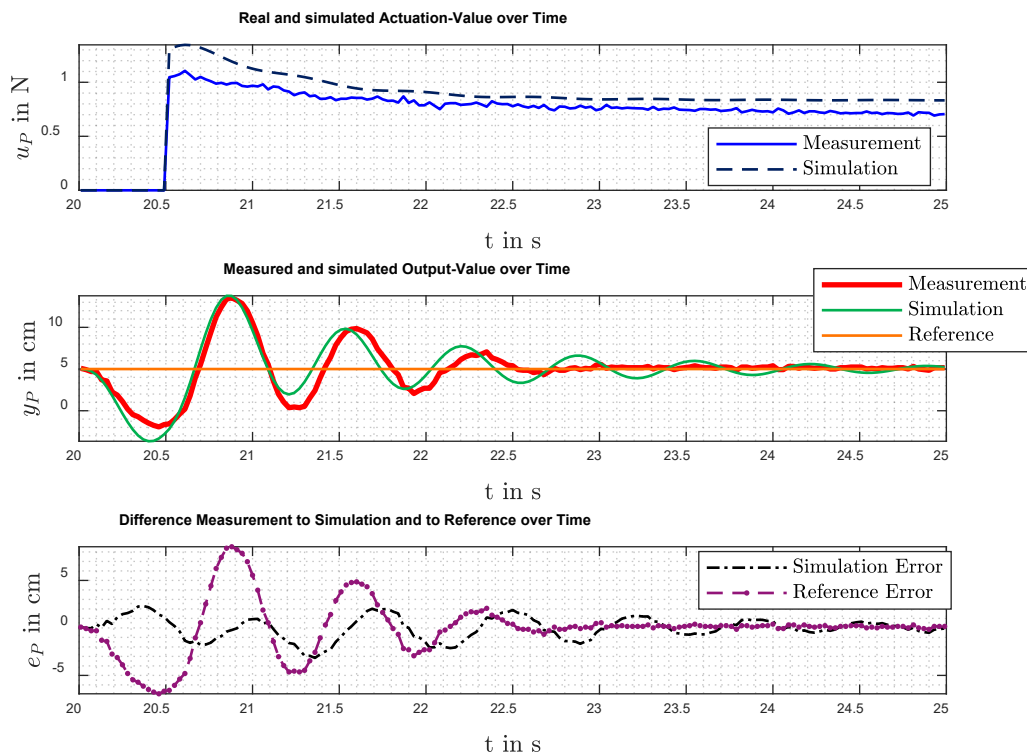


Abbildung 50: Störgrößenkompensation des Reglers 2 (R2)

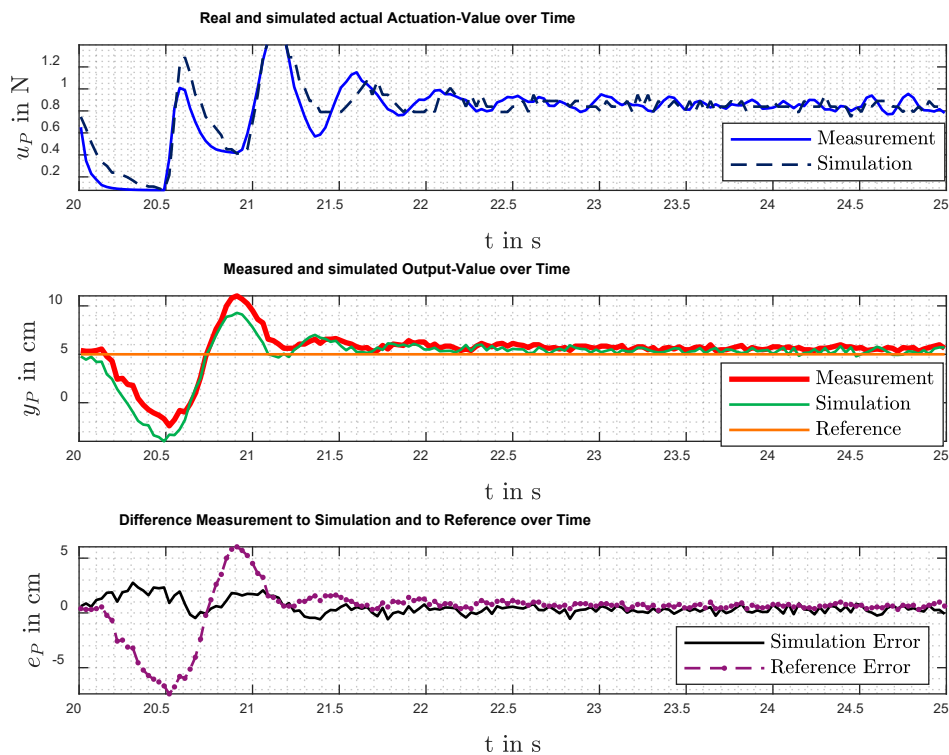


Abbildung 51: Störgrößenkompensation des Zustandsreglers (SS)

Tabelle 3: Vergleich zwischen der Positionsregler hinsichtlich der Störkompensation

<b>Regler</b>	<b>OL</b>	<b>R1</b>	<b>R2</b>	<b>SS</b>
Anstiegszeit ( $\epsilon \leq 0.4 \text{ cm}$ )	0.29 s	0.16 s	0.20 s	0.23 s
Überschwingweite %	106.4 %	257.6 %	170.9 %	94.1 %
Einschwingzeit ( $\epsilon \leq 0.4 \text{ cm}$ )	1.73 s	2.48 s	2.00 s	1.07 s
Einschwingzeit Simulation ( $\epsilon \leq 0.4 \text{ cm}$ )	–	2.51 s	4.13 s	1.01 s
Bleibende Regelabweichung	0 cm	0 cm	0 cm	0.68 cm

In Tabelle 3 können nun die 4 Regelkreisstrukturen anhand der Bewertungskriterien beurteilt werden. Dabei lassen sich verschiedene Beobachtungen machen.

- Alle Regelkreisstrukturen haben eine ähnliche Anstiegszeit
- R1 hat die höchste Überschwingweite, während SS die niedrigste Überschwingweite hat. R1 und R2 haben höhere Überschwingweiten als der offene Regelkreis.
- Der Zustandsregler SS hat eine deutlich schnellere Einschwingzeit als der offene Regelkreis. R1 und R2 benötigen doppelt so lange, um sich einzuschwingen und benötigen länger als der offene Regelkreis.
- Der Regler SS hat eine bleibende Regelabweichung von 0.68 cm.

Ein weiterer Vorteil des Zustandsreglers SS ist, dass dieser für eine längere Zeit von ein paar Sekunden ausgelenkt und gehalten werden kann, ohne dass sich die Stellkraft wie bei den Regelkreisen R1 und R2 immer weiter aufbaut.

Ein Nachteil des Zustandsreglers ist die bleibende Regelabweichung. Diese könnte durch Anpassung des Vorfilters minimiert werden.

## 7 Fazit und Ausblick

Es wurde in der Projektarbeit erfolgreich eine technische Aufrüstung durchgeführt. Diese wurde durch Hardware-Erweiterung am Versuch selber, durch Hinzufügen eines Drehzahlsensors erreicht. Des Weiteren wurde eine Fernbedienung implementiert, die über eine UDP-Wifi-Verbindung mit dem Mikrocontroller des Aufbaus kommuniziert und so es ermöglicht den Aufbau, als Stand-Alone zu betreiben. Dabei wurde die Fernbedienung so programmiert, dass diese es ermöglicht den Betriebsmodus des Feder-Masse-Schwingers einzustellen. Es kann zwischen offener Regelkreis, geschlossener Regelkreis und Herunterfahren gewählt werden.

In der Projektarbeit wurde ebenso eine verbesserte Störkompensation umgesetzt. Dabei wurde das physikalische Verhalten genauer untersucht, um möglich Einflüsse aufgrund von Nichtlinearitäten mitzubersichtigen. Durch die Verwendung des Drehzahlsensors konnte auch ein besseres Verständnis über die tatsächliche, vom Propeller erzeugte Kraft gewonnen werden. Dies hat dazu geführt, dass ein besseres, der physikalischen Beschreibung eines gedämpften Feder-Masse-Schwingers nahe kommenden, Modell erarbeitet werden konnte.

Des Weiteren wurde eine Drehzahlregelung für den BLDC-Motor umgesetzt, da auf diese Weise der Einfluss durch die Reduzierung der LiPo-Akku-Spannung entgegengewirkt werden kann. Es wurde ein Zustandsregler entworfen, indem durch Polvorgabe, die Dämpfung des Systems, als aperiodischer Grenzfall, erhöht wurde.

Das Ziel der verbesserten Störkompensation konnte so erreicht werden und ein Auslenken der Masse von Hand wird im besten Fall sehr schnell nach einem Überschwingen kompensiert.

Es können zahlreiche Verbesserungen und Möglichkeiten der Erforschung durch dieses Projekt genannt werden.

Man könnte eine vertiefte Untersuchung der Propellerstellkraft, wie in Kapitel 2.3.4 dargelegt, durchführen. Dies könnte das Verständnis über den BLDC-Motor und den Propeller verbessern.

Des Weiteren besteht die Möglichkeit, die Funktionalität der Fernbedienung zu erweitern. Dies könnte durch eine Benachrichtigungsfunktion auf dem Display geschehen, um den Benutzer darüber zu informieren, dass der Mikrocontroller erfolgreich heruntergefahren ist oder momentan eine Verbindung zum Mikrocontroller vorherrscht. Man könnte versuchen, die Fernbedienung so zu erweitern, dass eine Vorgabe der Dämpfung und der Federsteifigkeit, im laufenden Betrieb, ermöglicht wird. Dazu müssen in der Software 3 Regelparameter in Form des Vorfilters und der Reglermatrix berechnet werden.

Ein weiterer Aspekt ist die Untersuchung und Behebung numerischer Probleme, die bei der Sollwertvorgabe auftreten können. Es treten NaN-Werte im Programm. Dies führt dazu, dass



der Versuch neu gestartet werden muss. Durch eine Analyse und mögliche Umgestaltung des Simulink-Modells können diese Probleme behoben werden.

Momentan benötigt man für die Wifi-Verbindung der Fernbedienung einen WLAN-Router. Die Einrichtung eines mobilen Hotspots auf dem Mikrocontroller könnte eine elegante Lösung bieten, um diese Abhängigkeit zu beseitigen und die Portabilität des Systems zu verbessern.

Zusätzlich sei zu erwähnen, dass der Shutdown-Button am Schlitten nicht mehr funktioniert. Man könnte durch einer Hardware oder Software-Erweiterung dafür Sorge tragen, dass der Mikrocontroller über den Shutdown-Button und der Fernbedienung herunterfahrbar ist.

In Bezug auf den Zustandsregler ist eine Erweiterung zu einem PI-Zustandsregler eine Möglichkeit, um Probleme bei der Erreichung des Sollwerts anzugehen.

Zusätzlich sollten umfassende Tests über den gesamten Entladeverlauf des Akkus durchgeführt werden, um ein besseres Verständnis für das Verhalten des Systems zu erhalten.

## Literaturverzeichnis

- [1] ] T. Luft, Entwurf, Simulation und Bau eines geregelten Feder-Masse-Schwingers, Bachelor Thesis, Rosenheim, 2023
- [2] YIN YAN MODEL TECH. MFT., EMAX User Instruction for BLHeli Series ESC, Verfügbar unter:  
<https://www.tme.eu/Document/959503479e9fa7b0e61049c62875b2e2/BLHeli%20Instruction.pdf>, Zugriff am 08.03.2024
- [3] The MathWorks, Inc., PDF Documentation for Simulink,  
[https://de.mathworks.com/help/releases/R2022b/pdf\\_doc/simulink/index.html](https://de.mathworks.com/help/releases/R2022b/pdf_doc/simulink/index.html), Zugriff am 08.03.2024
- [4] Oscar Liang, What is Oneshot ESC Protocol - Active Braking, Verfügbar unter:  
<https://oscarliang.com/oneshot125-esc-quadcopter-fpv/>, Zugriff am 08.03.2024
- [5] Arduino, Datasheet Arduino Nano 33 IoT, Verfügbar unter:  
<https://docs.arduino.cc/resources/datasheets/ABX00027-datasheet.pdf>, Zugriff am 08.03.2024
- [6] ETT CO.,LTD, A guide of Module OLED Model. OLED 1.3, Verfügbar unter:  
<https://robu.in/wp-content/uploads/2019/12/1.3-Inch-I2C-IIC-OLED-LCD-Module-4pin-with-VCC-GND-Blue-1.pdf>, Zugriff am 08.03.2024
- [7] Arduino, Arduino IDE, Verfügbar unter: <https://www.arduino.cc/en/software>, Zugriff am 08.03.2024
- [8] U8glib library for monochrome displays, version 2, Verfügbar unter: <https://github.com/olikraus/u8g2>, Zugriff am 08.03.2024
- [9] Arduino, WiFiNINA , Verfügbar unter: <https://www.arduino.cc/reference/en/libraries/wifi-nina/>, Zugriff am 08.03.2024
- [10] Vilros, Raspberry Pi GPIO On Off Button, Verfügbar unter: <https://vilros.com/pages/on-off-button>, Zugriff am 08.03.2024
- [11] freakware GmbH, Beastx, Brushless RPM Sensor, universal, Verfügbar unter:  
<https://www.freakware.de/p/beastx-rpm-bl-sensor-universal-bxa760013-a108878.htm>  
, Zugriff am 08.03.2024
- [12] ABLIC, COUNTER IC WITH 2-WIRE (I2C-bus) INTERFACE, Verfügbar unter:  
[https://www.mouser.de/datasheet/2/360/S35770\\_I\\_E-1628617.pdf](https://www.mouser.de/datasheet/2/360/S35770_I_E-1628617.pdf), Zugriff am 08.03.2024
- [13] Zikodrive, What exactly is back-EMF?, Verfügbar unter:  
<https://zikodrive.com/de/ufaqs/rucken-emf-sensorlos-bursten-motoren-arbeit-arbeit-bursten-bork-kontrolle-i-need-einsatz/>, Zugriff am 08.03.2024

[14] Hobbydirekt Modellbau e.K., ULTRA 2804 2300KV brushless Motor LEFT Graupner S7048, Verfügbar unter:

<https://www.hobbydirekt.de/Neuheiten/Neuheiten-2020/Graupner/Elektromotoren/Brushless-Motoren/Brushless/ULTRA-2804-2300KV-brushless-Motor-LEFT-Graupner-S7048::290462.html>, Zugriff am 08.03.2024

[15] Technische Hochschule Rosenheim, pzMove, Verfügbar unter: <https://www.th-rosenheim.de/die-hochschule/labore/regelungstechnik/pzmove>, Zugriff am 08.03.2024

[16] P. Zentgraf. Ein neues Verfahren zur Modelierung linearer Systeme - Experimentelle Modellbildung mit Ein- und Ausgangsdaten. atp magazin, 2019.

[17] w. Kohn, R Öztürk, Statistik für Ökonomen, Datenanalyse mit R und SPSS. Springer Gabler, Berlin, Heidelberg, 2013.

[18] R. Hagl, Elektrische Antriebstechnik, 3., überarbeitete und erweiterte Auflage. München, Hanser, 2021.

[19] The MathWorks, Inc., Documentation für var(), Verfügbar unter:

<https://de.mathworks.com/help/matlab/ref/var.html>, Zugriff am 08.03.2024

[20] C. H. Shen, F. Y. C. Albert, C. K. Ang, D. J. Teck and K. P. Chan, "Theoretical development and study of takeoff constraint thrust equation for a drone," 2017 IEEE 15th Student Conference on Research and Development (SCORED), Wilayah Persekutuan Putrajaya, Malaysia, 2017, pp. 18-22, doi: 10.1109/SCORED.2017.8305428.

[21] J. Lunze, Regelungstechnik 2, Mehrgrößensysteme, Digitale Regelung, 9., überarbeitete und aktualisierte Auflage 2016, Springer, Berlin Heidelberg, 2016

[22] DigiKey, RS45111A900F -Potentiometer, Verfügbar unter: <https://www.digikey.de/de/products/detail/alps-alpine/RS45111A900F/19529172>, Zugriff am 08.03.2024

## Eidesstattliche Versicherung

Name: Ober Vorname: Eduard  
Matrikel-Nr.: 868176 Studiengang: AFE

Hiermit versichere ich, Eduard, Ober., an Eides statt, dass ich den vorliegenden Projektarbeitsbericht mit dem Titel „Geregelter Feder-Masse-Schwinger: Technische Aufrüstung und Verbesserung der Störkompensation“ selbständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Ich habe die Bedeutung der eidesstattlichen Versicherung und prüfungsrechtlichen Folgen (§ 26 Abs. 2 Bachelor-SPO bzw. § 19 Abs. 2 Master-SPO der Hochschule der Medien Stuttgart) sowie die strafrechtlichen Folgen (siehe unten) einer unrichtigen oder unvollständigen eidesstattlichen Versicherung zur Kenntnis genommen.

### Auszug aus dem Strafgesetzbuch (StGB)

#### § 156 StGB Falsche Versicherung an Eides Statt

Wer von einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

Surberg, 08.03.2024

Ort, Datum

Eduard Ober

Unterschrift