



**Modellbasierter Entwurf und Implementierung eines  
robusten Echtzeit-Regelungssystems zur dynamischen  
Stabilisierung einer Kugel auf einem rotierenden Rad  
unter Verwendung von MATLAB/Simulink und  
Mikrocontrollertechnik**

**Masterarbeit**

im Studiengang Business and Systems Engineering

vorgelegt von

**Ken Marvin Hilz**

Matrikelnummer: 2180475

E-Mail: ken.hilz@stud.hshl.de

am 26.05.2025

an der Hochschule Hamm-Lippstadt

Erstprüfer: Prof. Dr.-Ing. Mirek Göbel

Zweitprüfer: Prof. Dr.-Ing. Dmitrij Tikhomirov



### **Gender-Hinweis**

Zur besseren Lesbarkeit wird auf die gleichzeitige Verwendung der Sprachformen männlich, weiblich, sowie divers verzichtet. Sämtliche Personenbezeichnungen gelten gleichermaßen für alle Geschlechter.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung.....</b>	<b>1</b>
1.1	Motivation und Zielsetzung .....	1
1.2	Struktur der Arbeit.....	1
<b>2</b>	<b>Theoretische Grundlagen.....</b>	<b>2</b>
2.1	Stand der Technik .....	2
2.2	Das Inverse Pendel.....	4
2.3	Mathematisches Modell .....	5
2.4	Berechnung des Zustandsreglers.....	9
2.5	Systemidentifikation .....	11
2.6	Identifikation der Systemzeit.....	13
2.7	VL53L0X Time-of-Flight Laser Abstandssensor .....	14
<b>3</b>	<b>Systemdesign und Entwicklung .....</b>	<b>16</b>
3.1	Systemdesign.....	16
3.1.1	Skizze des Systems.....	16
3.1.2	Regelkreis .....	17
3.1.3	Morphologischer Kasten.....	18
3.2	Hardware- und Software-Planung .....	18
3.2.1	Software-Planung .....	19
3.2.2	Komponentenwahl und Verdrahtung .....	20
3.2.3	Sensorauswahl .....	22
3.3	Simulation der Regelung mit Simulink .....	24
3.3.1	Ermittlung der Filterfrequenz.....	25
3.3.2	Simulation der Regelung.....	26
3.3.3	Simulationsergebnisse.....	30
3.4	Hard- und Software: Entwicklung, Umsetzung und Tests.....	31
3.4.1	Rahmenkonstruktion .....	31
3.4.2	Komponentenentwicklung .....	32
3.4.2.1	Sensorik .....	32
3.4.2.2	Taster .....	35
3.4.2.3	Signalverarbeitung.....	37
3.4.2.4	Richtungsansteuerung .....	38
3.4.2.5	Antrieb.....	39

3.4.2.6	Radhalterung.....	41
3.4.2.7	Rad.....	42
3.4.2.8	Kugel .....	43
<b>3.4.3</b>	<b>Integrationstest.....</b>	<b>44</b>
<b>3.5</b>	<b>Zusammenführung von Software und Hardware .....</b>	<b>44</b>
<b>4</b>	<b>Systemtest und Ergebnisse .....</b>	<b>45</b>
4.1	Systemtest .....	46
4.2	Vergleich von Simulation und realem System .....	52
4.3	Ergebnisanalyse.....	53
<b>5</b>	<b>Fazit und Ausblick.....</b>	<b>54</b>
5.1	Zusammenfassung der Ergebnisse .....	54
5.2	Potenzielle Verbesserungen .....	56
	<b>Literaturverzeichnis .....</b>	<b>I</b>
<b>Anhang A</b>	<b>SVN-Ordner Link.....</b>	<b>IV</b>
<b>Anhang B</b>	<b>Aluminiumrahmen .....</b>	<b>IV</b>
<b>Anhang C</b>	<b>Komponentenhalterung .....</b>	<b>VI</b>
<b>Anhang D</b>	<b>Parameterdatei .....</b>	<b>VII</b>
<b>Anhang E</b>	<b>Winkelberechnung MATLAB Function.....</b>	<b>IX</b>
<b>Anhang F</b>	<b>Verlötung der Kondensatoren am Motor .....</b>	<b>IX</b>
<b>Anhang G</b>	<b>Verworfen Radhalterung .....</b>	<b>X</b>
<b>Anhang H</b>	<b>Halterungen .....</b>	<b>XI</b>
<b>Anhang I</b>	<b>Software.....</b>	<b>XIII</b>
<b>Anhang J</b>	<b>Bill of Materials .....</b>	<b>XX</b>
<b>Anhang K</b>	<b>VL53L0X Datenblatt .....</b>	<b>XXII</b>
<b>Anhang L</b>	<b>Ball Bearing DC-Motor.....</b>	<b>XXVIII</b>
<b>Anhang M</b>	<b>Not-Aus-Knopf .....</b>	<b>XXXI</b>
<b>Anhang N</b>	<b>VL53L0X Sensormodul .....</b>	<b>XXXII</b>

## Abkürzungsverzeichnis

µC	Mikrocontroller
BSE	<b>B</b> usiness and <b>S</b> ystems <b>E</b> ngineering
DC	<b>D</b> irect <b>C</b> urrent (Gleichstrom)
FDM	<b>F</b> used <b>D</b> eposition <b>M</b> odeling
GND	<b>G</b> round (Masse)
HSHL	<b>H</b> och <b>s</b> chule <b>H</b> amm- <b>L</b> ippstadt
NaN	<b>N</b> ot <b>a</b> <b>N</b> umber
NC	normally <b>c</b> losed
PWM	<b>P</b> uls <b>w</b> eiten <b>m</b> odulation
RPS	<b>R</b> evolutions <b>p</b> er <b>S</b> econd (Umdrehungen pro Sekunde)
SCL	<b>S</b> erial <b>C</b> lock
SDA	<b>S</b> erial <b>D</b> ata
ToF	<b>T</b> ime- <b>o</b> f- <b>F</b> light

# 1 Einleitung

In diesem Kapitel werden die Motivation und Zielsetzung dieser Arbeit erläutert und die Strukturierung dargestellt. Das Vorgehen bei der Umsetzung des Projekts richtet sich nach dem V-Modell. Die einzelnen Entwicklungsschritte sind im SVN-Ordner hinterlegt und einsehbar. Der Link ist im Anhang A aufgeführt.

## 1.1 Motivation und Zielsetzung

Die Motivation für dieses Projekt entstand durch die Lehrveranstaltung *Regelungstechnische Systeme* im ersten Semester des Masterstudiengangs *Business and Systems Engineering (BSE)*. Die Lehrveranstaltung vermittelte dabei Grundlagen und tiefere Einblicke in die Theorie und Praxis von regelungstechnischen Systemen.

Beispiel eines solchen Systems war das Kugelbalanciersystem aus der BMW-Welt, wobei eine Kugel mithilfe von Sensoren und Aktoren auf einem Rad balanciert wird. Dieses System soll in dieser Masterarbeit umgesetzt werden.

Die Software und Hardware des Kugelbalanciersystems sollen eigenständig entwickelt und implementiert werden. Die gelernten Inhalte aus der zuvor genannten Lehrveranstaltung können direkt praxisorientiert auf dieses Projekt angewendet werden.

Das Ziel dieses Projekts ist es, im Rahmen der Masterarbeit ein funktionsfähiges System zu entwickeln, zu programmieren und zu bauen, sodass dieses System als Ausstellungsstück für die *Hochschule Hamm-Lippstadt (HSHL)* verwendet werden kann.

## 1.2 Struktur der Arbeit

Die vorliegende Arbeit ist in fünf Kapitel unterteilt.

Das folgende Kapitel bietet einen Überblick über bereits umgesetzte Balanciersysteme. Anschließend erfolgt die Beschreibung des mathematischen Modells und die Berechnung des hier verwendeten Reglers. Neben der Identifikation des Systems und der Systemzeit wird auch der verwendete Sensor beschrieben.

Das dritte Kapitel befasst sich mit dem Systemdesign, der Entwicklung, der Umsetzung und den Tests der Komponenten. Dafür werden zuerst die Skizze und der Regelkreis des Systems beschrieben. Mithilfe eines morphologischen Kastens erfolgt die Auswahl der Komponenten. Die mechanische Konstruktion und die dahinterliegende elektronische Schaltung werden beschrieben. Das Kapitel wird mit der Zusammenführung von Software und Hardware abgeschlossen.

Im vierten und vorletzten Kapitel erfolgt die Beschreibung der Systemtests. Auch werden die Simulationsergebnisse des Systems mit den realen Ergebnissen verglichen. Zum Schluss werden die gewonnenen Ergebnisse analysiert.

Abschließend werden die Ergebnisse dieser Arbeit zusammengefasst und es wird ein Fazit gezogen. Ein Ausblick zeigt zudem, welche weiteren Untersuchungen und Verbesserungen das Projekt in zukünftigen Arbeiten optimieren könnten.

## **2 Theoretische Grundlagen**

In diesem Kapitel erfolgt ein Überblick über bereits umgesetzte Balanciersysteme und die mathematische Beschreibung sowie Berechnung des Systems.

### **2.1 Stand der Technik**

Ballbalanciersysteme wurden bereits von verschiedenen Universitäten, sowie von Studenten im Rahmen von Abschlussarbeiten als Projekte umgesetzt.

Diese Umsetzungen weisen häufig ähnliche Strukturen auf: Ein Rad, das von einem Motor angetrieben wird. Die Position des Balls auf dem Rad wird durch einen Sensor ermittelt. Häufig verwendete Sensortechnologien sind Ultraschall- und Lasersensortechnik. Mithilfe eines Mikrocontrollers werden die ermittelten Daten verarbeitet. Die Software zur Datenauswertung und Aktor Ansteuerung kann mit verschiedenen Programmen erfolgreich programmiert werden.

Nachfolgend werden zwei Umsetzungen von Ballbalanciersystemen vorgestellt.

Abbildung 1 zeigt die Umsetzung der *Universität Stuttgart*. Das System besteht aus einem Rad (hellblauer Pfeil), das von einem Motor (blauer Pfeil) über einen Zahnriemen angetrieben wird. Der sich im oberen Bereich befindende Ultraschallsensor (lila Pfeil) wird zur Positionserfassung des Balls (roter Kreis)



eingesetzt, während der darunter liegende optische Sensor (gelber Pfeil) die Geschwindigkeit der Felge erfasst. Ein Arduino verarbeitet die erfassten Daten und steuert den Motor, welcher das Rad antreibt. Die Software dafür wurde mit MATLAB/Simulink erstellt, siehe [23].

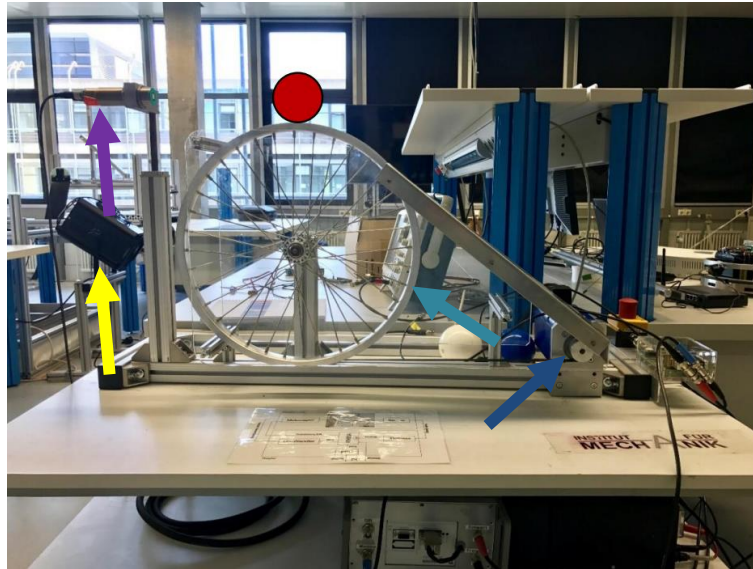


Abbildung 1: Ballbalanciersystem Universität Stuttgart  
Quelle: [23]

Im Vergleich dazu wird bei der *Johannes Kepler Universität* eine andere Sensortechnologie zur Erfassung des Balls verwendet. Der Aufbau ist in Abbildung 2 dargestellt.

Die Positionserfassung des Balls (roter Pfeil) erfolgt über einen Lasersensor (lila Pfeil), welcher eine höhere Genauigkeit bietet. Zudem wird das Rad (hellblauer Pfeil) nicht über einen Zahnriemen, sondern direkt an der Radnabe über einen DC-Motor (Direct Current, z. Dt. Gleichstrom) (blauer Pfeil) angetrieben. Auf beiden Seiten des Rads befinden sich Auffangarme (grüne Kästen), die verhindern, dass der Ball herunterfällt. Als Mikrocontroller wird hier ein Raspberry PI (schwarzer Kasten) verwendet, der mit Open-Source-Software programmiert wurde, siehe [21, S. 287].

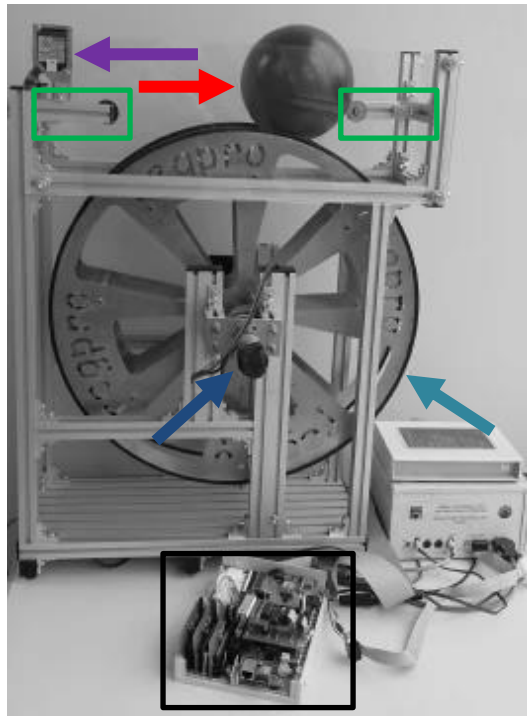


Abbildung 2: Ballbalanciersystem Johannes Kepler Universität  
Quelle: [21, S. 287]

Neben den beiden zuvor beschriebenen Ballbalanciersystemen existieren weitere Projekte, die angesichts des Umfangs dieser Masterarbeit nicht weiter dargestellt werden.

Zusammenfassend lässt sich festhalten, dass es unterschiedliche Ansätze zur Realisierung des Systems gibt, die durch verschiedene Technologien umgesetzt werden können.

## 2.2 Das Inverse Pendel

Wie im oberen Kapitel dargestellt, gibt es mehrere Möglichkeiten das Ballbalanciersystem umzusetzen. In einem Punkt sind alle Projektdurchführungen identisch: Das System, welches geregelt werden soll, ist ein instabiles System. Dabei bedeutet Instabilität, dass das System von seiner Ruhelage (Gleichgewichtslage) abweicht, wenn keine Regelung erfolgt.

Das inverse Pendel stellt ein klassisches instabiles System dar. Wie der Namen bereits andeutet, handelt es sich dabei um ein Pendel, das aufrecht auf einem Stab befestigt und balanciert wird. Dieser Stab ist an einem fahrbaren Wagen befestigt, welcher beschleunigt wird, um das Pendel im Gleichgewicht halten zu können. Die Ruhelage des Pendels ist das aufrechte Stehen. Bei einer Störung wird die Ruhelage sofort verlassen, wodurch das System als instabil gilt, siehe

[16, S. 3 f.]. Die untere Abbildung 3 zeigt ein Modell des inversen Pendels, bestehend aus einem fahrbaren Wagen ( $M$ ), auf dem das inverse Pendel frei auf einem Stab gelagert ist. Die Masse ( $m$ ) ist auf dem Stab befestigt und weicht schon bei kleinen Störungen von der Ruhelage ab, was zu einem Kippen des Pendels führt. Durch eine gezielte Beschleunigung des Wagens in Kipprichtung im Rahmen einer Regelung kann das Pendel aufrecht gehalten werden. Kippt das Pendel beispielsweise nach links muss auch der Wagen nach links beschleunigen, um das Pendel stabilisieren zu können, siehe [16, S. 29 f.].

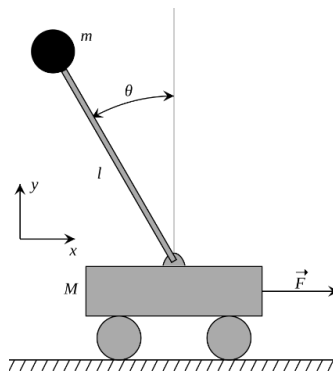


Abbildung 3: Inverses Pendel  
Quelle: [25]

Um instabile Systeme, wie das inverse Pendel oder das Ballbalanciersystem erfolgreich zu regeln, ist eine präzise mathematische Modellierung erforderlich. Das mathematische Modell bildet die physikalischen Eigenschaften und Dynamiken des Systems ab und dient als Grundlage für die Entwicklung geeigneter Regelungsstrategien. In Kapitel 2.3 wird das mathematische Modell des Kugelbalanciersystems beschrieben, um dessen Dynamik genauer zu analysieren. Auf Basis dessen kann das System anschließend simuliert werden.

## 2.3 Mathematisches Modell

Um das System analysieren zu können, müssen zuerst alle Bewegungsgleichungen des Systems aufgeführt werden.

Abbildung 4 stellt das Rad und die darauf liegende Kugel dar. Es werden somit zwei Bewegungsgleichungen benötigt, die die Körper beschreiben.

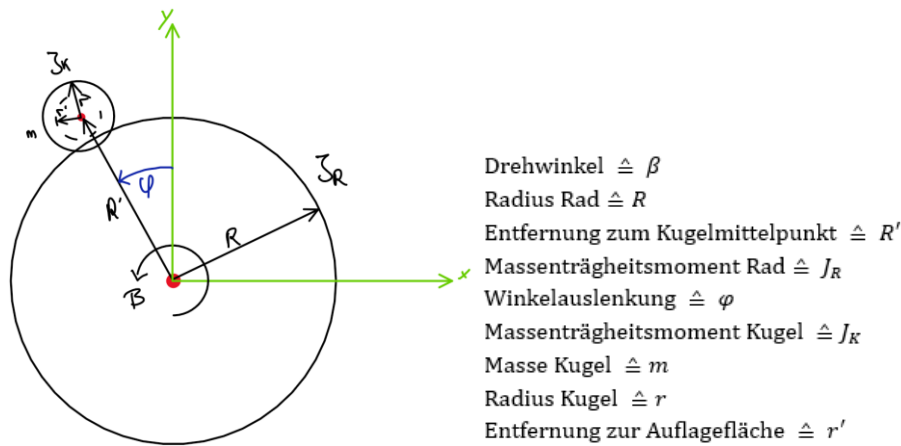


Abbildung 4: Modell Kugel auf Rad

Quelle: In Anlehnung an [9, S. 3]

Alle hier aufgeführten Bewegungsgleichungen des Systems *Kugel auf Rad* wurden anhand der Bewegungsgleichungen des Praktikumsskripts aufgebaut. Die ursprünglichen Gleichungen sind in Quelle [9] ab Seite 4 vorzufinden.

Teilsystem Kugel:

$$\ddot{\varphi} = \frac{R \cdot J_K \cdot T + m \cdot g \cdot \sin(\varphi) \cdot (R^2 \cdot J_K + r'^2 \cdot J_R)}{R' \cdot L}. \quad (2.1)$$

Teilsystem Rad:

$$\ddot{\beta} = \frac{(J_K + m \cdot r'^2) \cdot T + m \cdot g \cdot R \cdot J_K \cdot \sin(\varphi)}{L}. \quad (2.2)$$

Wobei  $L$  das Gesamt-Massenträgheitsmoment um den Mittelpunkt des Rads darstellt und wie folgt definiert ist:

$$L = J_K \cdot J_R + m \cdot (R^2 \cdot J_K + r'^2 \cdot J_R). \quad (2.3)$$

Aufgrund der vorhandenen Sinusfunktionen in beiden Bewegungsgleichungen müssen diese für weitere Berechnungen linearisiert werden. Da die Kugel in der Ruhelage auf dem Rad balanciert werden soll, wird vorausgesetzt, dass nur sehr kleine Winkel auftreten. Für  $\varphi \ll 1$  gilt:  $\sin(\varphi) = \varphi$ . Somit können folgende linearisierte Gleichungen (2.4) und (2.5) aufgestellt werden:

Teilsystem Kugel:

$$\ddot{\varphi} = \frac{R \cdot J_K \cdot T + m \cdot g \cdot \varphi \cdot (R^2 \cdot J_K + r'^2 \cdot J_R)}{R' \cdot L}. \quad (2.4)$$

Teilsystem Rad:

$$\ddot{\beta} = \frac{(J_K + m \cdot r'^2) \cdot T + m \cdot g \cdot R \cdot J_K \cdot \varphi}{L}. \quad (2.5)$$

Für die weitere Verwendung dieser Gleichungen wird die Zustandsraumdarstellung benötigt.

Mithilfe der Zustandsraumdarstellung ist es möglich, Systeme zu beschreiben, die aus mehreren Differentialgleichungen bestehen und mehrere Ein- und Ausgangssignale besitzen. Die Signale werden in Vektoren und das Systemverhalten in Matrizen aufgeführt. Die allgemeine Formel für die Zustandsraumdarstellung lautet:

$$\dot{\underline{x}}(t) = \underline{A} \cdot \underline{x}(t) + \underline{B} \cdot \underline{u}(t), \quad (2.6)$$

$$\underline{y}(t) = \underline{C} \cdot \underline{x}(t) + \underline{D} \cdot \underline{u}(t). \quad (2.7)$$

Die in den Gleichungen (2.6) und (2.7) verwendeten Bezeichnungen  $x, u$  und  $y$  sind Vektoren, wobei  $x$  der Zustandsvektor,  $u$  der Eingangsvektor und  $y$  (die Bezeichnung kann auch  $v$  sein) der Ausgangsvektor ist. Die weiteren Bezeichnungen stellen die Matrizen dar.  $A$  ist die Systemmatrix,  $B$  die Eingangsmatrix,  $C$  die Ausgangsmatrix und  $D$  die Durchgriffsmatrix. Die Durchgriffsmatrix wird nur bei sprunghafigen Systemen verwendet und ist daher meist 0, siehe [20, S. 24].

Die Matrizen und Vektoren können nun auf Basis des mathematischen Modells und der Differentialgleichungen gebildet werden.

Die Zustände des Systems setzen sich zusammen aus:

- der Winkeländerung  $\varphi$  und Winkelgeschwindigkeit  $\dot{\varphi}$  der Kugel sowie
- der Winkeländerung  $\beta$  und Winkelgeschwindigkeit  $\dot{\beta}$  des Rads.

Als Eingang wirkt ein Drehmoment  $T$  des Motors. Als Ausgänge werden die Winkelgeschwindigkeit  $\dot{\beta}$  des Rads und die Winkeländerung  $\varphi$  der Kugel betrachtet.

Die demnach aufgestellten Zustands-, Eingangs- und Ausgangsvektoren lauten:

$$x = \begin{bmatrix} \beta \\ \dot{\beta} \\ \varphi \\ \dot{\varphi} \end{bmatrix}, u = T \text{ und } v = \begin{bmatrix} \dot{\beta} \\ \dot{\varphi} \end{bmatrix}. \quad (2.8)$$

Die Systemmatrix  $A$ , Eingangsmatrix  $B$  und Ausgangsmatrix  $C$  lauten:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{m \cdot g \cdot R \cdot J_K}{L} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{m \cdot g \cdot (R^2 \cdot J_K + r'^2 \cdot J_R)}{R' \cdot L} & 0 \end{bmatrix}, \quad (2.9)$$

$$B = \begin{bmatrix} 0 \\ \frac{J_K + m \cdot r'^2}{L} \\ 0 \\ \frac{R \cdot J_K}{R' \cdot L} \end{bmatrix}, \quad (2.10)$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (2.11)$$

Die Durchgriffsmatrix wird für dieses System mit 0 angenommen.

Eingesetzt in den Formeln (2.6) und (2.7) ergeben sich folgende Systemmatrizen:

$$\begin{bmatrix} \dot{\beta} \\ \ddot{\beta} \\ \dot{\varphi} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{m \cdot g \cdot R \cdot J_K}{L} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{m \cdot g \cdot (R^2 \cdot J_K + r'^2 \cdot J_R)}{R' \cdot L} & 0 \end{bmatrix} \cdot \begin{bmatrix} \beta \\ \dot{\beta} \\ \varphi \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{J_K + m \cdot r'^2}{L} \\ 0 \\ \frac{R \cdot J_K}{R' \cdot L} \end{bmatrix} \cdot T, \quad (2.12)$$

$$\begin{bmatrix} \dot{\beta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \beta \\ \dot{\beta} \\ \varphi \\ \dot{\varphi} \end{bmatrix}. \quad (2.13)$$

Für dieses Projekt werden nur die Zustände  $\varphi$  und  $\dot{\varphi}$  betrachtet, da nur die Kugelposition sensorisch erfasst werden soll. Aus diesem Grund erfolgt eine Reduktion der Matrizen (2.9) bis (2.11) auf ein Zustandsmodell mit einer Dimension von 2x2.

Die reduzierten Matrizen können wie folgt aufgestellt werden:

$$A = \begin{bmatrix} 0 & 1 \\ \frac{m \cdot g \cdot (R^2 \cdot J_K + r'^2 \cdot J_R)}{R' \cdot L} & 0 \end{bmatrix}, \quad (2.14)$$

$$B = \begin{bmatrix} 0 \\ \frac{R \cdot J_K}{R' \cdot L} \end{bmatrix}, \quad (2.15)$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (2.16)$$

Daraus ergibt sich die folgende Zustandsraumdarstellung:

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\dot{\varphi}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{m \cdot g \cdot (R^2 \cdot J_K + r'^2 \cdot J_R)}{R' \cdot L} & 0 \end{bmatrix} \cdot \begin{bmatrix} \varphi \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{R \cdot J_K}{R' \cdot L} \end{bmatrix} \cdot T, \quad (2.17)$$

$$\begin{bmatrix} \varphi \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \varphi \\ \dot{\varphi} \end{bmatrix}. \quad (2.18)$$

## 2.4 Berechnung des Zustandsreglers

Aufgrund der Mehrgrößenstruktur des zu regelnden Systems sowie der Darstellung des Systemverhaltens mittels Matrizen wurde ein Zustandsregler gewählt, siehe [8, S. 420 f.].

Die gewünschte Dynamik des Systems wird dabei durch das Verfahren der Polvorgabe realisiert. Die Verschiebung der aktuellen Pole in die linke s-Halbebene erfolgt über eine Rückführmatrix, deren Elemente gezielt berechnet werden. Eine direkte Bestimmung dieser Rückführwerte ist nur möglich, wenn das System in Regelungsnormalform vorliegt, da in diesem Fall ein Koeffizientenvergleich zur Berechnung herangezogen werden kann. Voraussetzung für die Umsetzbarkeit des Verfahrens ist außerdem die Steuerbarkeit des Systems.

Die Wahl der Zielpole ist auf die linke s-Halbebene beschränkt, da sich nur dort stabile Systeme realisieren lassen. Eine zu nahe Platzierung an der imaginären Achse kann jedoch zu verstärktem Schwingungsverhalten führen. Eine zu weit links gelegene Polposition bewirkt große Stellgrößenanforderungen, die vom realen System unter Umständen nicht erfüllt werden können, siehe [10, S. 77 f.].

Auf Basis der reduzierten Matrizen (2.14) bis (2.16) wird im weiteren Verlauf die Rückführmatrix  $K$  berechnet. Das Vorgehen orientiert sich an der Formelsammlung aus der Lehrveranstaltung Mess- und Regelungstechnik, siehe [19, S. 12 f.].

Zur Überprüfung wird die Steuerbarkeitsmatrix  $P_S$  bestimmt:

$$P_S = [B \ AB]. \quad (2.19)$$

Einsetzen der Matrizen (2.14) und (2.15) in (2.19) ergibt:

$$P_S = \left[ \begin{bmatrix} 0 \\ \frac{R \cdot J_K}{R' \cdot L} \end{bmatrix} \left( \begin{bmatrix} 0 & 1 \\ \frac{m \cdot g \cdot (R^2 \cdot J_K + r'^2 \cdot J_R)}{R' \cdot L} & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \frac{R \cdot J_K}{R' \cdot L} \end{bmatrix} \right) \right], \quad (2.20)$$

$$P_S = \begin{bmatrix} 0 & \frac{R \cdot J_K}{R' \cdot L} \\ \frac{R \cdot J_K}{R' \cdot L} & 0 \end{bmatrix} \neq 0. \quad (2.21)$$

Da  $\det(P_S) \neq 0$  ergibt sich ein  $\text{Rang}(P_S) = 2$ . Das bedeutet, dass beide Zustände des Systems steuerbar sind. Als nächstes erfolgt die Berechnung der Rückführmatrix  $K$ .

Für die Zustandsrückführung wird die neue Systemmatrix  $W = A - BK$  benötigt. Die Matrix  $BK$  ergibt sich aus:

$$BK = \begin{bmatrix} 0 \\ \frac{R \cdot J_K}{R' \cdot L} \end{bmatrix} \cdot [-K_1 \quad -K_2] = \begin{bmatrix} 0 & 0 \\ -\frac{R \cdot J_K}{R' \cdot L} \cdot K_1 & -\frac{R \cdot J_K}{R' \cdot L} \cdot K_2 \end{bmatrix}. \quad (2.22)$$

Somit ergibt sich die Matrix  $W$  zu:

$$W = A - BK = \begin{bmatrix} 0 & 1 \\ \frac{m \cdot g \cdot (R^2 \cdot J_K + r'^2 \cdot J_R)}{R' \cdot L} & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ -\frac{R \cdot J_K}{R' \cdot L} \cdot K_1 & -\frac{R \cdot J_K}{R' \cdot L} \cdot K_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{m \cdot g \cdot (R^2 \cdot J_K + r'^2 \cdot J_R)}{R' \cdot L} + \frac{R \cdot J_K}{R' \cdot L} \cdot K_1 & \frac{R \cdot J_K}{R' \cdot L} \cdot K_2 \end{bmatrix}. \quad (2.23)$$

Zur Berechnung der Polstellen wird die Determinante  $\det(\lambda E - W)$  gebildet. Zunächst:

$$\lambda E = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}. \quad (2.24)$$

$$\lambda E - W = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ \frac{m \cdot g \cdot (R^2 \cdot J_K + r'^2 \cdot J_R)}{R' \cdot L} + \frac{R \cdot J_K}{R' \cdot L} \cdot K_1 & \frac{R \cdot J_K}{R' \cdot L} \cdot K_2 \end{bmatrix} = \begin{bmatrix} \lambda & -1 \\ -\frac{m \cdot g \cdot (R^2 \cdot J_K + r'^2 \cdot J_R)}{R' \cdot L} - \frac{R \cdot J_K}{R' \cdot L} \cdot K_1 & \lambda - \frac{R \cdot J_K}{R' \cdot L} \cdot K_2 \end{bmatrix}. \quad (2.25)$$



Die Determinante lautet:

$$\det \left( \begin{bmatrix} \lambda & -1 \\ -\frac{m \cdot g \cdot (R^2 \cdot J_K + r'^2 \cdot J_R)}{R' \cdot L} - \frac{R \cdot J_K}{R' \cdot L} \cdot K_1 & \lambda - \frac{R \cdot J_K}{R' \cdot L} \cdot K_2 \end{bmatrix} \right) = \lambda \cdot \left( \lambda - \frac{R \cdot J_K}{R' \cdot L} \cdot K_2 \right) - (-1) \cdot \left( -\frac{m \cdot g \cdot (R^2 \cdot J_K + r'^2 \cdot J_R)}{R' \cdot L} - \frac{R \cdot J_K}{R' \cdot L} \cdot K_1 \right). \quad (2.26)$$

Sortiert nach der höchsten Ableitung ergibt sich die finale Gleichung:

$$\lambda^2 - \lambda \cdot \frac{R \cdot J_K}{R' \cdot L} \cdot K_2 - \frac{m \cdot g \cdot (R^2 \cdot J_K + r'^2 \cdot J_R)}{R' \cdot L} - \frac{R \cdot J_K}{R' \cdot L} \cdot K_1 = 0. \quad (2.27)$$

Die Gleichung (2.27) entspricht formal der eines PT2-Systems, siehe [19, S. 10]. Durch Koeffizientenvergleich werden die Rückführwerte  $K_1$  und  $K_2$  bestimmt.

Für die Bestimmung von  $K_1$  und  $K_2$  werden folgende Gleichungen aufgestellt:

$$\omega_0^2 = -\frac{m \cdot g \cdot (R^2 \cdot J_K + r'^2 \cdot J_R)}{R' \cdot L} - \frac{R \cdot J_K}{R' \cdot L} \cdot K_1 \quad \text{und} \quad (2.28)$$

$$2D\omega_0 = -\frac{R \cdot J_K}{R' \cdot L} \cdot K_2. \quad (2.29)$$

Die Gleichungen (2.28) und (2.29) werden nach  $K_1$  und  $K_2$  umgestellt. Die Werte für  $D$  und  $\omega_0$  können frei nach gewünschtem Systemverhalten gewählt werden.

$$K_1 = \frac{-\omega_0^2 \cdot R' \cdot L - m \cdot g \cdot (R^2 \cdot J_K + r'^2 \cdot J_R)}{R \cdot J_K}. \quad (2.30)$$

$$K_2 = -\frac{2D\omega_0 \cdot R' \cdot L}{R \cdot J_K}. \quad (2.31)$$

## 2.5 Systemidentifikation

Zur Analyse des Systemverhaltens werden die zuvor bestimmten Systemmatrizen (2.14) bis (2.16) in Simulink implementiert und die Reaktion des Systems bei einem Sprung nach 1 s simuliert. Mit dieser Analyse sollen die Stabilitätseigenschaften des Systems und die Verhaltensweise der Kugel dargestellt werden. Ebenfalls wird das System in der Realität identisch angeregt.

In der unteren Abbildung 5 sind drei Graphen zu sehen. Die Informationen der Graphen lassen sich wie folgt beschreiben:

- Roter Graph: Dargestellt wird die Winkeländerung der Kugel zum Ruhezpunkt, welche exponentiell ansteigt.
- Blauer Graph: Die Winkelgeschwindigkeit von der Mitte der Kugel zum Ruhezpunkt steigt nach der Einleitung des Sprungs ebenfalls exponentiell.
- Lila Graph: Stellt den eingeleiteten Sprung auf 26 dar.

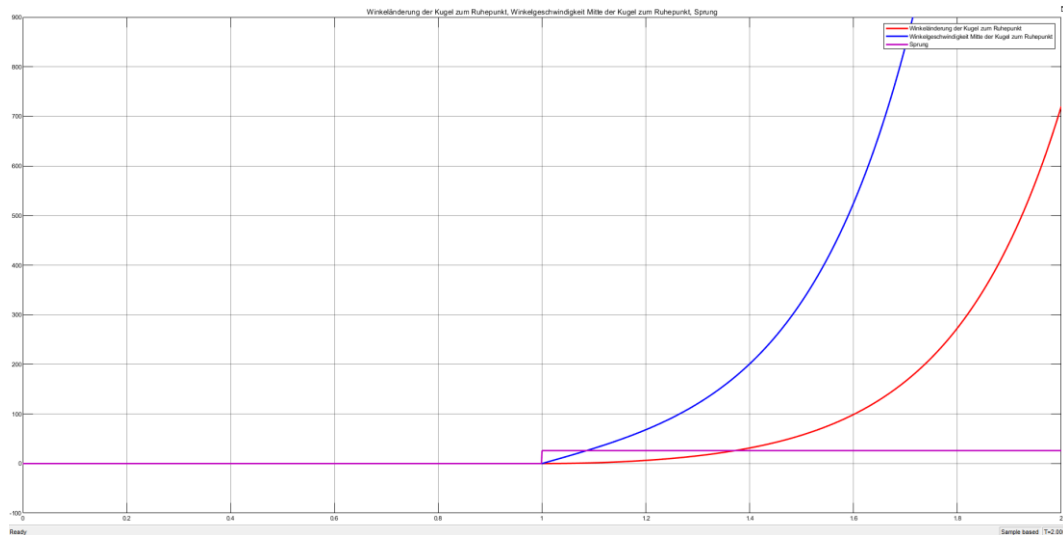


Abbildung 5: Sprungantwort Simulation

Die Simulation bestätigt, dass das System ohne eine Regelung nicht stabil ist. Sobald eine äußere Anregung erfolgt, wird die Kugel zunehmend instabil und verlässt schließlich das Rad. Dieses Verhalten entspricht den Erwartungen an das System. Ohne Rückkopplung wird es nicht möglich sein, dieses System zu stabilisieren.

Als Vergleich dazu wurde das dynamische Verhalten der Kugel erfasst. Dabei lag die Kugel auf dem Rad und wurde nach einer Sekunde durch einen Sprung in Höhe von 26 PWM (Pulsweitenmodulation) durch den Motor angeregt. Dieser Wert wurde in den Komponententests ermittelt, siehe Kapitel 3.4.2.5. Die folgende Abbildung 6 zeigt die entsprechenden Messwerte.

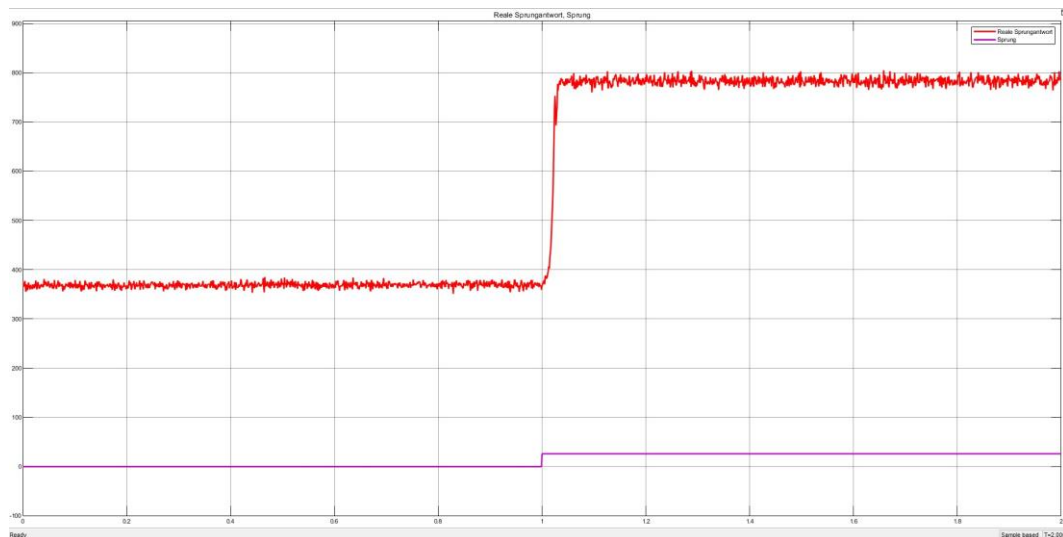


Abbildung 6: Sprungantwort reales System

Die gemessene Kugel befand sich in einer Entfernung von  $360\text{ mm}$ . Deutlich erkennbar ist, dass die Messkurve exponentiell ansteigt, sobald die Kugel angeregt wird, wie bereits in Abbildung 5. Das reale Bewegungsverhalten der Kugel stimmt weitgehend mit den Simulationsergebnissen überein. Dies bestätigt, dass die mathematischen Gleichungen des Systems das reale Verhalten zuverlässig abbilden. Sowohl in der Simulation als auch in der Realität zeigt sich, dass das System instabil ist und eine Regelung erforderlich wird.

## 2.6 Identifikation der Systemzeit

Die Bestimmung der Systemzeit erfolgt anhand der Simulation des geschlossenen Regelkreises, welcher eine Zustandsrückführung unter Verwendung der Systemmatrizen beinhaltet, siehe Abbildung 17. Die Identifikation ist zur Auslegung der Filterfrequenz notwendig. Die Simulation erfolgt mit einer Sprunganregung des Eingangssignals auf den Wert  $0,1\text{ Nm}$  bei  $10\text{ s}$  mit einer Dämpfung  $D = 0,1$  und Eigenkreisfrequenz  $\omega_0 = 12$ , um die dynamische Reaktion des Systems zu analysieren.

Abbildung 7 zeigt die zeitliche Entwicklung der Winkeländerung (rote Kurve) und der Winkelgeschwindigkeit der Kugel (blaue Kurve).

Das Systemverhalten stimmt mit dem mathematischen Modell überein und zeigt ein gedämpftes PT2-System. Die Bestimmung der Systemzeit lässt sich laut [19, S. 10] anhand der Zeit einer Periode festlegen. Wie aus der folgenden Abbildung 7 ersichtlich, beträgt die Systemzeit  $1,598\text{ s}$ . Das entspricht einer Frequenz von  $0,63\text{ Hz}$ .

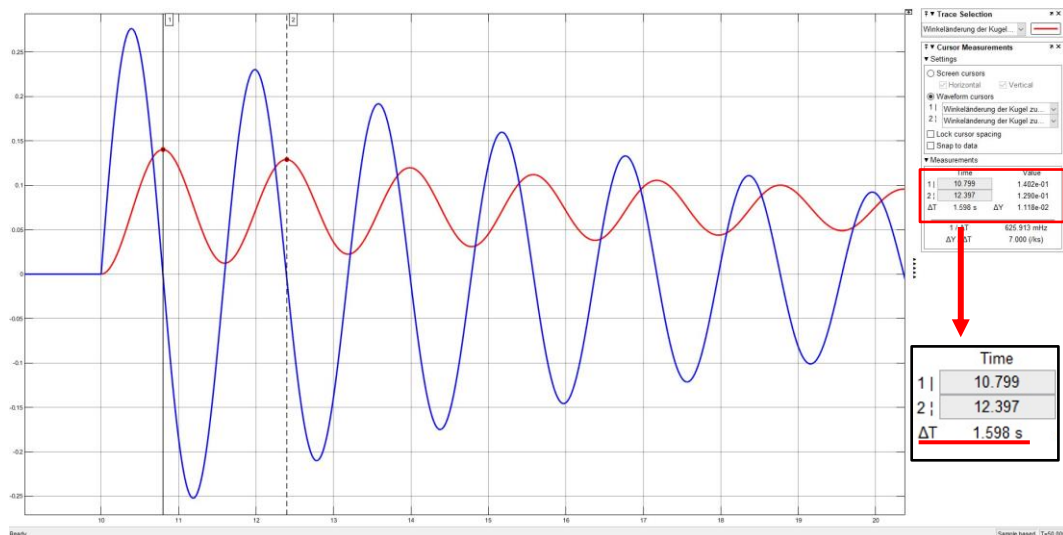


Abbildung 7: Ermittlung der Systemzeit

## 2.7 VL53L0X Time-of-Flight Laser Abstandssensor

Abhängig von der Funktionsweise lassen sich Lasersensoren in kontinuierliche Wellen-Phasensensoren und Puls-Phasensensoren unterteilen. Die gepulste Lasermessung verwendet die „time-of-flight“ (ToF) Differenz (z. Dt. Flugzeit) des Sensorsignals, um die Entfernung zu ermitteln. Der Sensor sendet regelmäßig Lichtstrahlen pulsförmig vom Laseremitter aus. Diese Lichtstrahlen werden von dem zu messenden Objekt reflektiert und an den Sensor zurückgeworfen. Über die Zeitdifferenz, der Emission und Detektion des Lichtstrahls, ermittelt der Sensor die Entfernung. Neben der Laufzeit des Lichts wird die Lichtgeschwindigkeit benötigt, wodurch die Entfernung des Objekts ermittelt werden kann, siehe [26, S. 2]. Die beschriebene Funktionsweise eines ToF-Sensors ist in Abbildung 8 dargestellt.

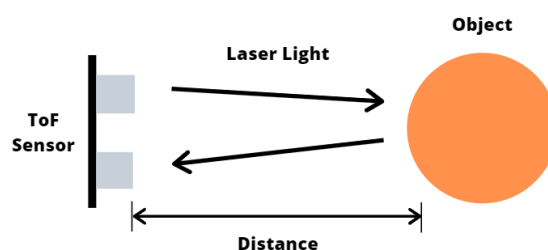


Abbildung 8: Funktionsweise ToF-Sensor

Quelle: [17]

Der VL53L0X ToF-Sensor verwendet einen Emitter des Typen VCSEL (Vertical Cavity Surface-Emitting Laser). Das bedeutet, dass der Sensor einen Oberflächenemitter verwendet, der senkrecht vom Board abstrahlt. Der Laser ist für das menschliche Auge unsichtbar. Diese technische Gegebenheit sorgt dafür,

dass der Sensor unempfindlicher gegenüber Umgebungslicht ist, siehe [15, S. 301].

Darüber hinaus bietet der Sensor die Möglichkeit, zwischen drei verschiedenen *Ranging Modes* zu wählen: *High Accuracy*, *Long Range* und *High Speed*. Diese Modi ermöglichen es, entweder besonders präzise Messungen durchzuführen, größere Entfernungen abzudecken oder eine höhere Messfrequenz zu erzielen, siehe [15, S. 575].

Der Lasersensor wird über 5 V und GND betrieben, hat eine maximale Reichweite von 2 m und verwendet eine Wellenlänge von 940 nm (Nanometer). Aufgrund der kompakten Abmessungen von 32 mm x 24 mm x 8 mm lässt sich der Sensor problemlos integrieren, siehe [4]. Im *High Speed* Modus kann eine maximale Reaktionszeit von 20 ms erzielt werden. Das Sichtfeld (Field of View) des Sensors deckt einen Bereich von 25 ° ab, mit einer Messgenauigkeit von  $\pm 5 \%$ , siehe [22, S. 28 ff.]. Der Lichtstrahl ist für das menschliche Auge nicht sichtbar, da lediglich im Bereich von 400 nm bis 700 nm Wellenlängen durch das menschliche Auge erfasst werden können, siehe [5, S. 38]. Die Kommunikation erfolgt via I<sup>2</sup>C Schnittstelle. Diese Schnittstelle findet häufig Verwendung bei der Verbindung zwischen integrierten Schaltungen und Sensoren. Die Datenübertragung erfolgt über die Ports Serial Data Line (SDA) und Serial Clock Line (SCL). Die SCL-Leitung dient als Taktgeber, während die SDA-Leitung für die Übertragung der Daten zuständig ist, siehe [7, S. 144].

Mit der folgenden Formel wird die Entfernung des Objekts berechnet werden:

$$d = \frac{c_o \cdot t}{2}. \quad (2.32)$$

Der Abstand  $d$  wird dabei berechnet, indem die Lichtgeschwindigkeit  $c_o$  ( $300000 \frac{m}{s}$ ) mit der Zeit  $t$  multipliziert wird, die benötigt wird, bis die gesendeten Lichtstrahlen zurück reflektiert werden. Anschließend wird dieser Wert durch 2 dividiert, damit lediglich die einmalige Entfernung zum Objekt bestimmt wird, siehe [11, S. 172].

Der ToF-Sensor wird aufgrund seiner kompakten Bauweise, seines breiten Einsatzspektrums und seiner hohen Messgenauigkeit, die für diese Arbeit erforderlich ist, eingesetzt. Für die Umsetzung des Projekts wird der Sensor im

Modus *High Speed* verwendet. Die Abbildung 9 zeigt den VL53L0X ToF-Sensor, das Sensormodul ist rot umrandet.

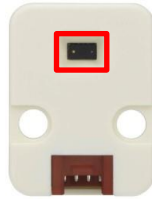


Abbildung 9: VL53L0X Time-of-Flight Laser Abstandssensor  
Quelle: [3]

### 3 Systemdesign und Entwicklung

In diesem Kapitel werden die Software, Hardware und deren Zusammenführung beschrieben. Damit die Projektumsetzung erfolgreich verläuft, wurde zu Beginn eine Anforderungsliste erstellt. Diese enthält alle funktionalen und technischen Anforderungen des Systems.

#### 3.1 Systemdesign

Im Folgenden wird der funktionale Systementwurf vorgestellt.

##### 3.1.1 Skizze des Systems

Basierend auf den Anforderungsdefinitionen wurde ein erster Entwurf des Systems skizziert, der in Abbildung 10 zu sehen ist.

Die Basis stellt ein stabiler Aluminiumprofilrahmen (gelb) dar, auf dem alle weiteren Komponenten befestigt werden.

Geplant ist, das Rad (hellblau) mithilfe einer 3D-gedruckten Halterung mit Bodenplatte zu montieren.

Links und rechts neben dem Rad ist die Anbringung von Auffangarmen vorgesehen, die mit Mikrotastern (grün) ausgestattet sind. Diese dienen zur Erkennung der Kugelposition.

Auf der rechten Seite ist die Anbringung einer Sensorhalterung (lila) geplant. Die Ausrichtung erfolgt so, dass der Sensor die Kugelposition präzise erfassen kann.

Für den Antrieb des Systems ist ein DC-Motor mit Riemenantrieb (blau) vorgesehen. Die Montage soll so erfolgen, dass das Rad zuverlässig angetrieben und beschleunigt werden kann.

Die zu balancierende Kugel (rot) muss zuverlässig erfasst werden. Zudem ist darauf zu achten, dass ausreichend Reibung zwischen Kugel und Rad gewährleistet ist, um eine stabile Regelung zu ermöglichen.

Für die Regelung kommt ein Mikrocontroller ( $\mu C$ , schwarz) zum Einsatz. Dieser dient der Verarbeitung der Sensordaten und zur Ansteuerung des Motors, sodass die Kugel stabil auf dem Rad gehalten wird.

Das System soll durch eine Scheibe abgegrenzt werden, um Verletzungen durch das drehende Rad zu verhindern.

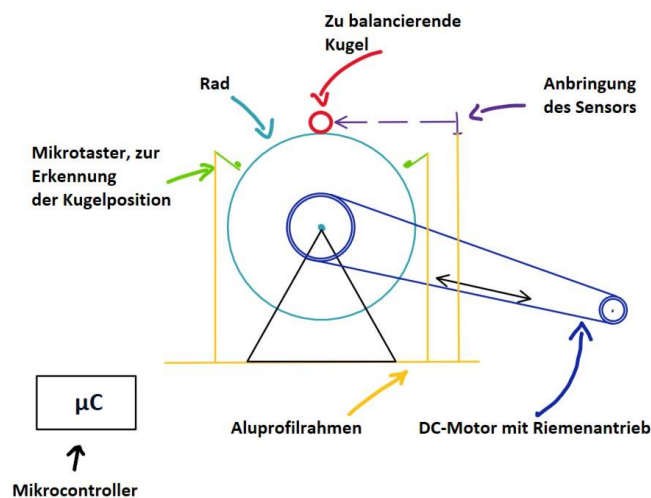


Abbildung 10: Systemskizze

### 3.1.2 Regelkreis

Nachdem die Grundstruktur für die Umsetzung des Projekts erstellt wurde, kann darauf aufbauend der Regelkreis für das System definiert werden. Da das System autonom, also ohne direkte menschliche Interaktion, funktionieren soll, wird die Sollposition der Kugel einmalig durch eine Benutzereingabe in der Software festgelegt.

Die eingestellte Sollposition  $d_{soll}$  wird mit der gemessenen Position  $d_{mess}$  verrechnet. Die daraus resultierende Regelabweichung  $\Delta_d$  wird von einem Regler verarbeitet, der eine Regelgröße in Form einer PWM-Signalsteuerung, die gleichzeitig auch als Drehrichtungsansteuerung dient, ausgibt. Diese Signale werden an das Stellglied (DC-Motor) weitergegeben.

Der Motor erzeugt daraufhin ein Drehmoment als Stellgröße, das an die Regelstrecke (Kugel auf Rad) weitergeleitet wird. Die dadurch veränderte Ist-Position  $d_{ist}$  der Kugel wird erneut erfasst und zurückgeführt. Durch die

kontinuierliche Erfassung und Regelung wird die Kugel stabil an ihrer Sollposition gehalten. Der beschriebene Regelkreis ist in Abbildung 11 dargestellt und wurde anhand der Formelsammlung aufgestellt, siehe [19, S. 5].

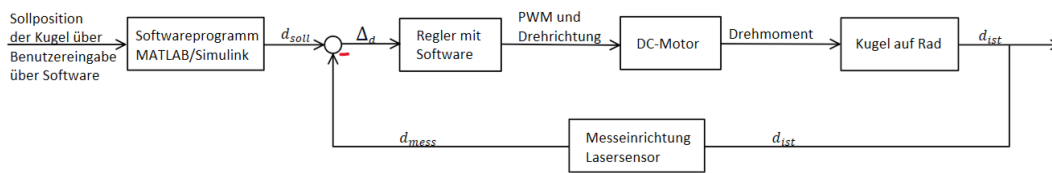


Abbildung 11: Regelkreis des Systems

### 3.1.3 Morphologischer Kasten

Zur systematischen Auswahl geeigneter Komponenten wurde ein morphologischer Kasten erstellt, zu sehen in Abbildung 12.

Durch die vorherige Recherche, wie solche Systeme umgesetzt wurden, konnte der Einsatz eines Aluminiumrahmens bereits zuvor als notwendig bestimmt werden. Andere Parameter, wie zum Beispiel der Motortyp oder die Lagerung der Radnabe, wurden anhand der verschiedenen Optionen systematisch bewertet.

Die grün markierten Felder zeigen die Ausprägungen, die im weiteren Verlauf umgesetzt wurden. Ausschlaggebend für ihre Auswahl waren insbesondere Verfügbarkeit, Implementierbarkeit und Kosten. Eine detaillierte Begründung der Auswahl erfolgt in den nachfolgenden Kapiteln.

Parameter	Ausprägungen			
Systemvariante	Regelung (Closed-Loop)	Steuerung (Open-Loop)		
Software	Arduino IDE	MATLAB/Simulink	Visual Studio	Python
Mikrocontroller	Raspberry Pi 4	Arduino UNO R3	Arduino UNO R4	Arduino MEGA 2560
Radtyp	3D-gedrucktes Rad	Eigenbau	Fahrradfelge	
Radgröße (Durchmesser)	0-20 cm	20-30 cm	30-40 cm	40-50 cm
Radmaterial	Holz	Kunststoff	Aluminum	Edelstahl
Radflanken Überzieher	Kein Überzieher	Gummiüberzieher	Schaumstoffüberzieher	
Kugelmateriale	Metall	Kunststoff	Holz	Gummi
Sensorik	Ultraschallsensor	Kamera	Infrarotsensor	
Endlagen Erkennung	Kapazitiver Sensor	Mikrotaster	Lichtschränke	
Motortyp	DC-Motor	BLDC-Motor	Schrittmotor	Servomotor
Motoransteuerung	Integrierte Treiber-ICs	Doppel-H Brücke	Relais	BLDC-Motortreiber
Motordrehgeschwindigkeit Ansteuerung	konstante Geschwindigkeit	PWM Ansteuerung	spannungsgeregelt	
Auffangmechanismus	Federelagerte Arme	Statische Arme	Netz	Kein Auffangmechanismus
Radhalterung	Holzgestell	3D-gedrucktes Gestell	Metallgestell	Aluminiumprofile
Sensorhalterung	Holzgestell	3D-gedrucktes Gestell	Metallgestell	Aluminiumprofile
Rahmenkonstruktion	Stahl	Holz	Aluminiumprofile	Kunststoff
Lagerung Radachse	Löcher im Gestell	Rollenlager	Stehlager	Kugellager
3D Modellierung	Fusion360	SolidWorks	FreeCAD	
Energieversorgung Mikrocontroller	Solarzelle	9V Batterie	Netzteil 12 V	Powerbank
Energieversorgung Motor	Solarzelle	9V Batterie	Netzteil 12 V	Powerbank
Radantrieb	Antrieb an der Radnabe	Antrieb über Kette	Antrieb über Keilriemen	Reibradantrieb
Systemabgrenzung	Plexiglasscheibe	Glasscheibe	Folie	

Abbildung 12: Morphologischer Kasten

## 3.2 Hardware- und Software-Planung

Im Folgenden wird der technische Systementwurf dargestellt.



### 3.2.1 Software-Planung

Vorab wurde der technische Systementwurf der Softwarekomponenten erstellt, um darzustellen, welche Ein- und Ausgänge die jeweiligen Komponenten benötigen. In der nachfolgenden Abbildung 13 ist der Systementwurf visualisiert. Die einzelnen Subsysteme sind dabei mit Punkten farblich markiert. Die Sensorik (lila Punkt) speist das Signal in die Komponente Signalverarbeitung (schwarzer Punkt) ein. Die Komponente Taster (grüner Punkt) übermittelt die Signale an die Komponente Richtungssteuerung (gelber Punkt) und Antrieb (blauer Punkt). Im Folgenden werden diese Komponenten genauer beschrieben.

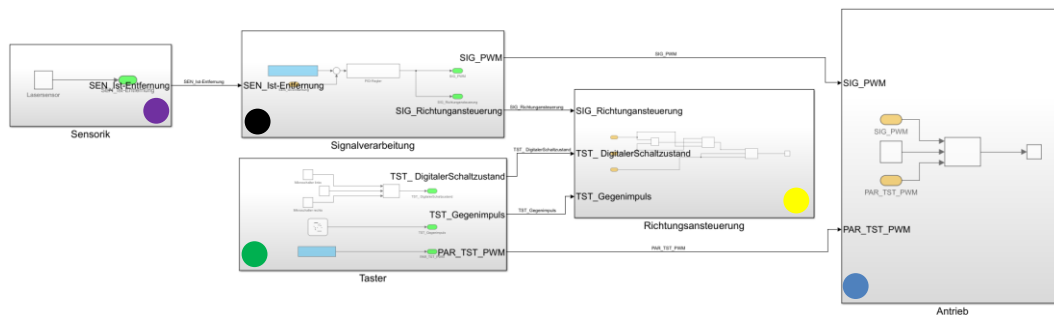


Abbildung 13: Technischer Systementwurf

Das Sensorsignal *SEN\_Ist-Entfernung* ist die primäre Eingangsgröße für die Regelung. Es wird in der Signalverarbeitung verarbeitet und dient zur Erzeugung der Signale *SIG\_PWM* und *SIG\_Richtungsansteuerung*. Das Signal *SIG\_PWM* steuert direkt die Komponente Antrieb basierend auf der ermittelten Regelgröße, während *SIG\_Richtungsansteuerung* an die Richtungsansteuerung übergeben wird und die Drehrichtung des Motors festlegt.

Die Komponente Taster liefert, ebenso wie die Sensorik, ausschließlich Signale an andere Komponenten, ohne selbst Signale zu empfangen. Bei Betätigung eines Tasters werden die Signale *TST\_DigitalerSchaltzustand* und *TST\_Gegenimpuls* an die Richtungsansteuerung weitergegeben. Zusätzlich wird der Parameter *PAR\_TST\_PWM* an den Antrieb ausgegeben.

Die Richtungsansteuerung bestimmt anhand der Eingangssignale die Drehrichtung des Motors, um die Kugel auszubalancieren. Die Vorgabe für die PWM stammt dabei aus der Signalverarbeitung, während die Komponente Taster bei Betätigung zusätzlich den Parameter *PAR\_TST\_PWM* an den Antrieb übergibt.

Die Komponente Antrieb erhält zwei Eingangssignale zur Steuerung der PWM. Da *PAR\_TST\_PWM* nur dann übermittelt wird, wenn ein Taster betätigt wird, erfolgt die PWM-Regelung ansonsten ausschließlich durch die Signalverarbeitung.

### **3.2.2 Komponentenwahl und Verdrahtung**

Die Planung der Verdrahtung der einzelnen Komponenten wurde mit *Fritzing* durchgeführt. Die erstellte Steckplatten-Ansicht zeigt die Verbindungen aller, der für dieses System relevanten Komponenten und ist in Abbildung 14 dargestellt. Ebenfalls wird der Auswahlgrund der Komponenten beschrieben.

#### **Mikrocontroller**

Als zentrale Steuereinheit wird der Arduino Mega 2560 gewählt. Dieser bietet zahlreiche digitale und analoge Ein- und Ausgänge, die eine flexible Anbindung von Sensoren und Aktoren ermöglichen. Dank der vielen Schnittstellen werden eingehende Signale effizient verarbeitet und der Motor präzise über die Doppel H-Brücke angesteuert. Der Mikrocontroller wird mit 12 V und 2 A betrieben, siehe [2, S. 6 ff.].

#### **Sensor**

Für die Auswahl des Sensors wurden mehrere Varianten getestet. Die Ergebnisse sind im nachfolgenden Kapitel 3.2.3 dargestellt.

#### **Mikrotaster**

Zur Erfassung der Schaltzustände werden Mikrotaster eingesetzt, die als Öffner oder Schließer konfiguriert werden können. Sie ermöglichen eine zuverlässige Bestimmung der Kugelposition und sind jeweils über einen digitalen Pin und *GND* mit dem Mikrocontroller verbunden.

#### **Antrieb**

Für den Antrieb wird ein 775 Kugellager DC-Motor verwendet, der mit einer variablen Spannung von 12 V bis 36 V betrieben werden kann. Eine maximale Drehzahl von  $9000 \frac{U}{min}$  ist bei 36 V erreichbar. Ein DC-Motor wird gewählt, da er sich durch seine präzise Steuerung, hohe Regelbarkeit und kompakte Bauweise besonders gut für schnelle Richtungswechsel eignet. Eine präzise Drehzahlsteuerung über PWM ist möglich. Der Motor wird für dieses Projekt mit

einer Spannung von 12 V betrieben. Dabei wird ein Nennstrom von 1,2 A benötigt, siehe [12]. Die Spannungsversorgung und PWM-Ansteuerung erfolgen über die Doppel H-Brücke.

### Doppel H-Brücke

Die Doppel H-Brücke wird aufgrund ihrer Fähigkeit, hohe Spannungen und Ströme (bis 12 V und 7 A) sicher zu verarbeiten, gewählt. Sie steuert den Gleichstrommotor, ermöglicht die Änderung der Drehrichtung und regelt die Geschwindigkeit.

Die Doppel H-Brücke wird mit 5 V und GND vom Mikrocontroller versorgt, während die Steuersignale für Drehrichtung und Geschwindigkeit über digitale Pins und einen PWM-Pin übertragen werden. Über ein externes Netzteil erfolgt die Spannungsversorgung mit 12 V und 7 A, siehe [1].

### Not-Aus-Knopf

Zwischen der Spannungsversorgung der Doppel H-Brücke und dem Netzteil wird ein Not-Aus-Knopf geschaltet. Im Bedarfsfall kann dieser gedrückt werden, wodurch die Stromzufuhr zum Motor unterbrochen wird. Der Knopf wird aufgrund seiner Belastbarkeit von 660 V und 10 A für diese Anwendung ausgewählt, siehe [13].

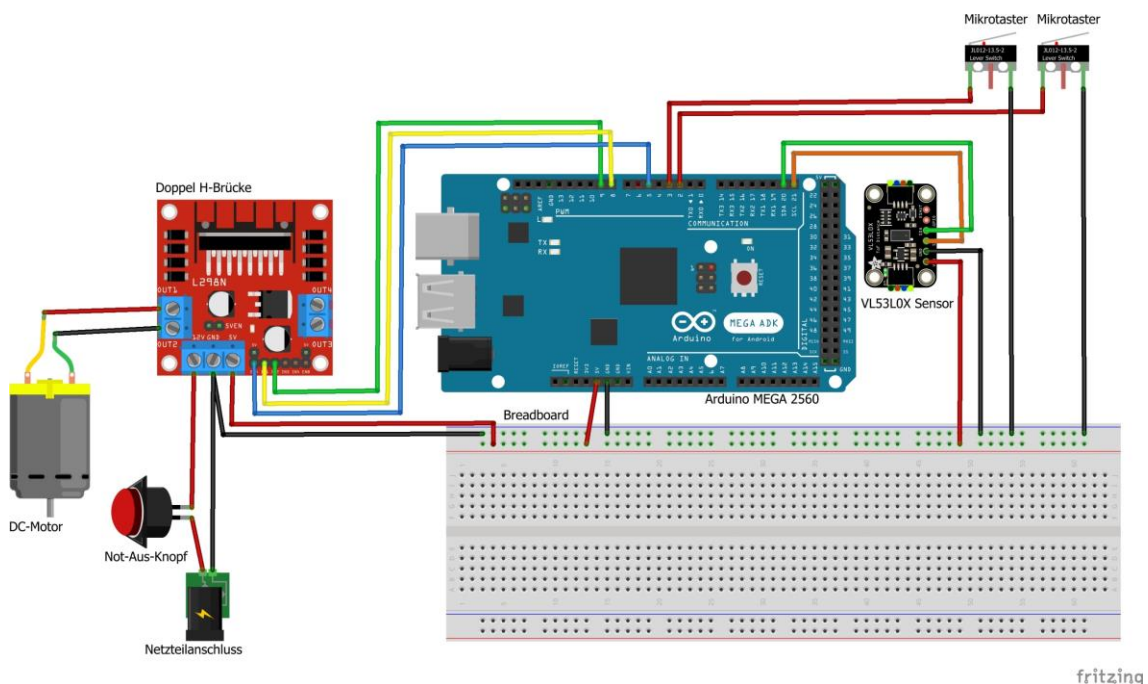


Abbildung 14: Verdrahtungsplan mit Fritzing  
Quelle: Fritzing

## **Radhalterung**

Die Radhalterung dient der drehbaren Lagerung des Rads. Für die Entwicklung wird die Variante mit Rillenkugellagern des Typs 6001 2RS gewählt, da diese eine zuverlässige und effiziente Lagerung ermöglichen.

## **Rad**

Die Wahl fiel auf eine Vorderrad-Fahrradfelge mit einem Durchmesser von 32 *cm*, da die Gesamtkonstruktion mobil bleiben soll.

## **Kugel**

Für die Auswahl der richtigen Kugel wurden verschiedene Materialarten und Größen in Betracht gezogen, darunter Edelstahl-, Holz-, Vollmetall- und Gummikugeln. Die Auswahl erfolgte im Rahmen der Komponententests und wird in Kapitel 3.4.2.8 vorgestellt.

### **3.2.3 Sensorauswahl**

Um den richtigen Sensor bestimmen zu können, wurden der VL53L0X ToF-Sensor (einmal von AZ-Delivery und einmal von Conrad) sowie der GP2Y0A02YK0F Sharp IR-Abstandssensor ausgewählt und getestet. Die Auswahlkriterien waren eine kompakte Bauweise, hohe Messgenauigkeit, gute Implementierbarkeit, Kompatibilität mit Arduino und MATLAB/Simulink sowie eine gefahrlose Anwendung, insbesondere im Hinblick auf potenziell schädliche Laserstrahlung.

Die VL53L0X Sensoren nutzen den  $I^2C$ -Bus (*SDA*, *SCL*), während der GP2Y0A02YK0F Sensor analog Daten überträgt.

Die Tests umfassten Messungen auf eine helle Fläche (300 *mm* Entfernung) sowie auf eine Kugel (240 *mm* Entfernung), um sowohl die allgemeine Messgenauigkeit als auch die Robustheit gegenüber verschiedenen Oberflächen zu bewerten. Alle Sensoren konnten die gerade helle Fläche zuverlässig erfassen. Dennoch zeigten sich Unterschiede in der Messgenauigkeit. Die Messergebnisse sind in Abbildung 15 dargestellt.

- Der VL53L0X von AZ-Delivery zeigte einen Mittelwert von 308,60 *mm*  $\pm$  1,90 *mm* auf (blaue Messkurve).

- Der VL53L0X von Conrad erzielte einen durchschnittlichen Messwert von  $306,36 \text{ mm} \pm 1,90 \text{ mm}$  (rote Messkurve).
- Der Sharp IR-Sensor lieferte einen Mittelwert von  $304,10 \text{ mm} \pm 2,36 \text{ mm}$  (grüne Messkurve).

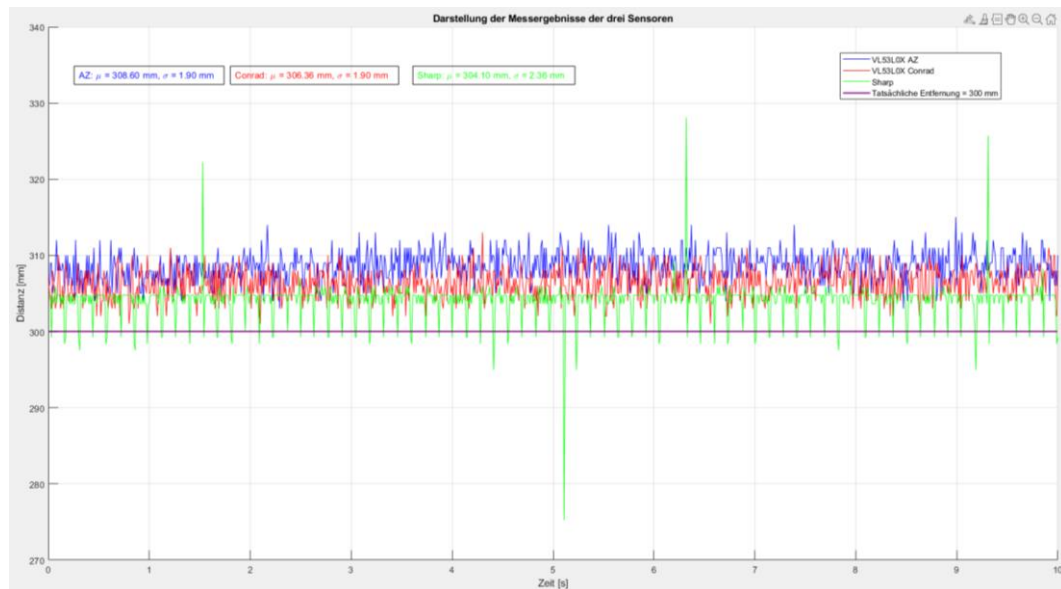


Abbildung 15: Vergleichsmessung glatte Oberfläche

Bei diesem Test erwies sich der VL53L0X von Conrad als der präziseste Sensor, da er die geringste Messabweichung vom tatsächlichen Wert aufwies. Beide VL53L0X Sensoren wiesen eine Messunsicherheit von  $\pm 1,90 \text{ mm}$  auf.

Die Messungen an einer Kugel führten bei allen Sensoren zu größeren Messunsicherheiten, die in Abbildung 16 dargestellt sind.

- Der VL53L0X ToF-Sensor von AZ-Delivery lieferte Messwerte von  $227,90 \text{ mm} \pm 2,88 \text{ mm}$  (blaue Messkurve).
- Der VL53L0X ToF-Sensor von Conrad erzielte Messwerte von  $242,79 \text{ mm} \pm 2,20 \text{ mm}$  (rote Messkurve).
- Der Sharp IR-Sensor erfasste die Kugel in einer Entfernung von  $257,51 \text{ mm} \pm 2,23 \text{ mm}$  (grüne Messkurve).

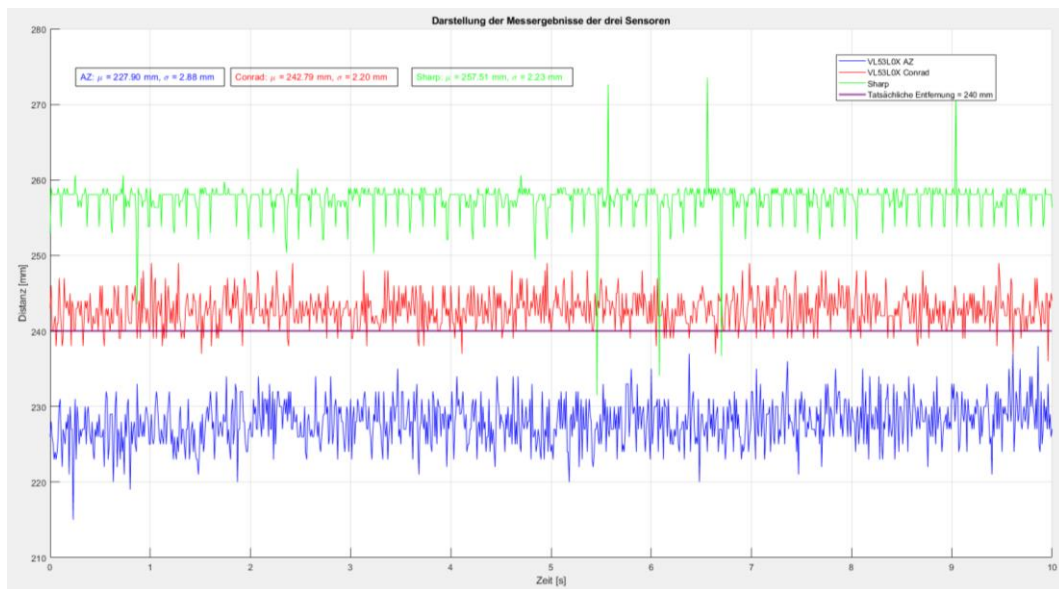


Abbildung 16: Vergleichsmessung Kugel

In der zweiten Testmessung mit einer Kugel wiesen alle Sensoren höhere Abweichungen auf. Der VL53L0X von Conrad war trotz einer Messdifferenz von  $+2,79\text{ mm}$  die beste Wahl, da er mit  $\pm 2,20\text{ mm}$  die geringste Unsicherheit aufwies und trotz der gewölbten Oberfläche stabile Werte lieferte.

Basierend auf beiden Testmessungen wurde der VL53L0X ToF-Sensor von Conrad als beste Wahl für die weitere Projektdurchführung bestimmt. Dieser zeigte die geringsten Messunsicherheiten und insgesamt die stabilsten Werte.

### 3.3 Simulation der Regelung mit Simulink

Die in Kapitel 2.3 aufgestellten Matrizen (2.14), (2.15) und (2.16) werden nun für die Simulation weiterverwendet.

Das Modell besteht aus zwei parallelen PT1-Filtern, welche die Eingangssignale vorfiltern. Diese Filterung dient der Glättung der Sensorsignale und reduziert das Rauschen.

Die gefilterten Signale werden anschließend über einen Mux-Block gebündelt. Dieser Vektor wird über die Zustandsrückführung  $K$  geführt, wodurch die Regelgrößen berechnet werden. Die Regelgrößen sind auf einen Bereich von  $-255$  bis  $+255$  begrenzt. Danach erfolgt eine Skalierung mittels des Verstärkungsfaktors  $PAR\_RPS$  von  $0,227$ . Dieser Wert entspricht der Anzahl an Motorumdrehungen pro Sekunde (RPS) bei einem PWM-Signal von 1.

Der in Simulink aufgestellte Regelkreis ist in der unteren Abbildung 17 zu sehen.

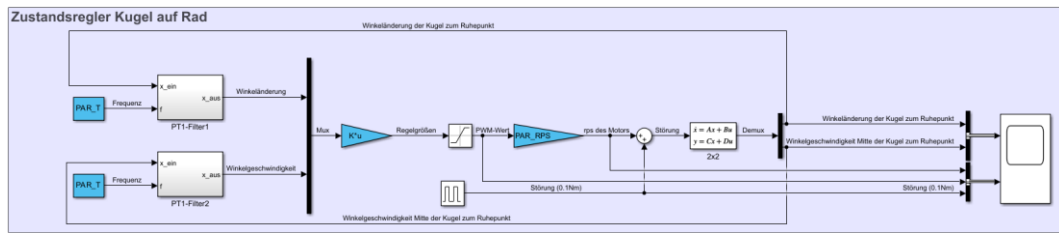


Abbildung 17: Simulationsaufbau des Systems Kugel auf Rad mit Zustandsregler

### 3.3.1 Ermittlung der Filterfrequenz

Da eine Filterung mit der in Kapitel 2.6 ermittelten Zeitkonstante zu einer starken Verzögerung der Systemantwort führt, wird in diesem Kapitel eine alternative Frequenz vorgestellt, welche schnellere Reaktionen ermöglicht. In den Komponententests der Sensorik konnte eine deutlich bessere Filterfrequenz von  $15\text{ Hz}$  identifiziert werden, siehe Abbildung 26. Die untere Abbildung 18 zeigt die Simulation des Systems mit beiden Filterfrequenzen. Im oberen Plot sind die Winkeländerungen und im unteren die Winkelgeschwindigkeiten dargestellt. Auffällig sind die hohen Ausschläge in den Messkurven bei  $0,63\text{ Hz}$  (rot und grün), die zu einem instabilen und aufschwingenden Verhalten führen. Im Gegensatz dazu, zeigen die mit  $15\text{ Hz}$  gefilterten Kurven (blau und lila) ein deutlich stabileres und schnelleres Regelverhalten. Die Regelung reagiert auf die eingeleitete Störung und bringt die Kugel schnell wieder in eine stabile Lage zurück.

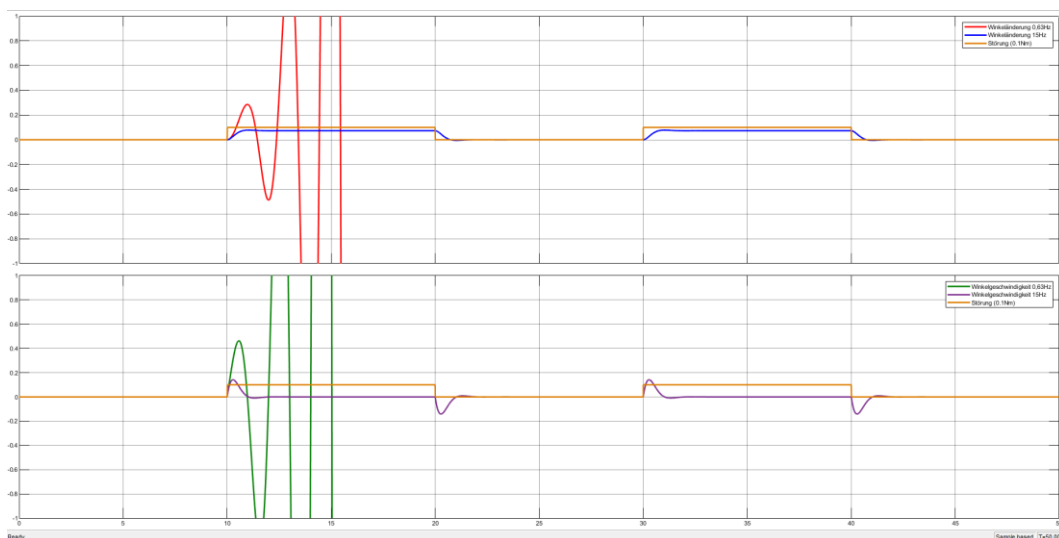


Abbildung 18: Ermittlung der Filterfrequenz

Laut [18, FolienID 186.0] muss die Abtastrate des Systems mindestens zehnmals schneller als die kleinste Zeitkonstante des Systems sein. Bei einer Frequenz von  $15\text{ Hz}$  (entspricht  $\sim 0,067\text{ s}$ ) wird aus diesem Grund die Abtastrate auf  $0,001\text{ s}$  festgelegt, um eine deutlich schnellere Abtastung zu gewährleisten.

### 3.3.2 Simulation der Regelung

Um die Zustandsregelung in der Simulation zu testen, wird eine Störung eingegeben. Der Störimpuls wird bei  $t = 10\text{ s}$  und  $t = 30\text{ s}$  für jeweils  $10\text{ s}$  in Höhe von  $0,1\text{ Nm}$  eingelegt. Da die Störung positiv ist, werden negative Regelgrößen und RPS-Werte erwartet. Insgesamt wurden drei Durchläufe mit den Dämpfungsfaktoren  $D = 0,5$ ,  $D = 1$  und  $D = 1,5$  und Eigenkreisfrequenzen  $\omega_0 = 12$  und  $\omega_0 = 20$  durchgeführt. Alle drei Durchläufe erzeugen die folgenden fünf Signalverläufe:

- Rot: Winkeländerung der Kugel zum Ruhezustand
- Blau: Winkelgeschwindigkeit der Kugel zum Ruhezustand
- Grün: Umdrehungen pro Sekunde des Motors (RPS)
- Lila: PWM-Regelgröße
- Orange: Störsignal

Der erste Simulationsdurchlauf wurde mit der Dämpfung  $D = 0,5$  gestartet. In der Tabelle 1 sind die maximalen und dauerhaften Kurvenverläufe dargestellt.

Tabelle 1: Messwerte bei fester Dämpfung und variierender Eigenkreisfrequenz 1

	$D = 0,5 \quad \omega_0 = 12$	$D = 0,5 \quad \omega_0 = 20$
Winkeländerung	Maximal: $0,100\text{ rad}$ Dauerhaft: $0,073\text{ rad}$	Maximal: $0,023\text{ rad}$ Dauerhaft: $0,015\text{ rad}$
Winkelgeschwindigkeit	Maximal: $0,197 \frac{\text{rad}}{\text{s}}$	Maximal: $0,101 \frac{\text{rad}}{\text{s}}$
RPS	Maximal: $-0,338\text{ rps}$ Dauerhaft: $-0,243\text{ rps}$	Maximal: $-0,201\text{ rps}$ Dauerhaft: $-0,130\text{ rps}$
PWM-Wert	Maximal: $-1,493$ Dauerhaft: $-1,071$	Maximal: $-0,897$ Dauerhaft: $-0,573$

Im Vergleich der beiden Durchläufe zeigt sich deutlich, dass bei gleicher Dämpfung, aber höherer Eigenkreisfrequenz, alle betrachteten Werte signifikant niedriger ausfallen. Sowohl die maximale als auch die dauerhafte (bleibende) Winkeländerung während der Störungseinwirkung sind stark reduziert, was auf eine geringere Auslenkung des Systems hinweist. Die Winkelgeschwindigkeit und die Regelgröße zeigen ebenfalls bei höherer Eigenkreisfrequenz deutlich



geringere Spitzen- und Dauerwerte. Besonders auffällig ist zudem der geringere PWM-Wert, was auf einen reduzierten Regelaufwand hindeutet. Insgesamt reagiert das System mit höherer Eigenkreisfrequenz ruhiger und effizienter, wodurch eine stabilere Regelung erreicht wird. Aufgrund der geringen Dämpfung benötigt das System etwas Zeit, bis es auf den neuen Wert eingeschwungen ist. Ein Überschwingen ist somit vorhanden, was jedoch nicht gewünscht ist. Die Messverläufe sind in der folgenden Abbildung 19 dargestellt.

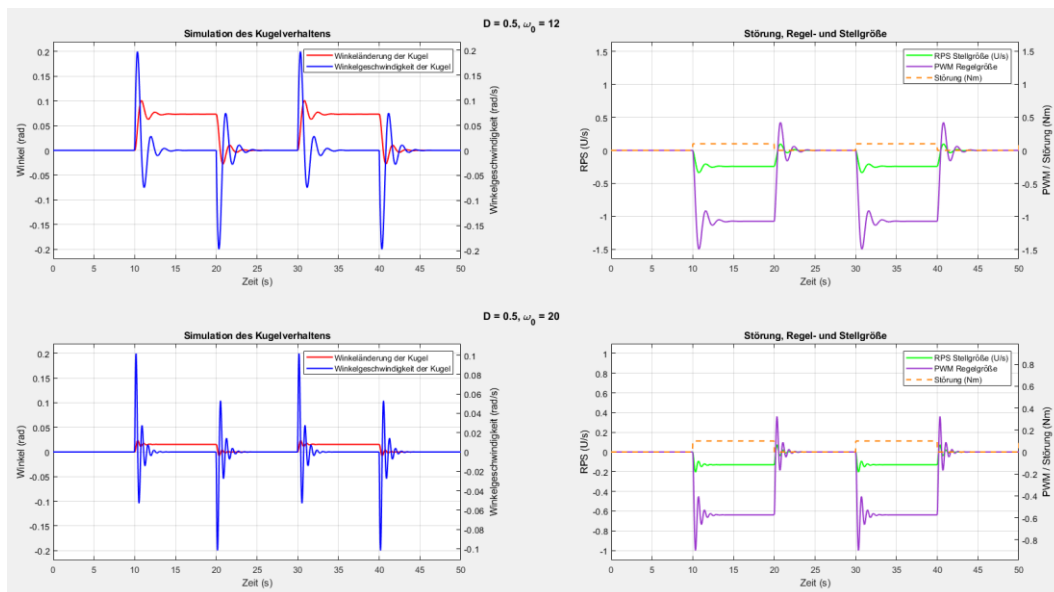


Abbildung 19: Simulationsdurchführung 1

Der zweite Simulationsdurchlauf wurde mit der Dämpfung  $D = 1$  durchgeführt. In der Tabelle 2 sind die maximalen und dauerhaften Kurvenverläufe dargestellt

Tabelle 2: Messwerte bei fester Dämpfung und variierender Eigenkreisfrequenz 2

	$D = 1 \quad \omega_0 = 12$	$D = 1 \quad \omega_0 = 20$
Winkeländerung	Maximal: 0,078 rad Dauerhaft: 0,073 rad	Maximal: 0,018 rad Dauerhaft: 0,015 rad
Winkelgeschwindigkeit	Maximal: 0,141 $\frac{rad}{s}$	Maximal: 0,077 $\frac{rad}{s}$
RPS	Maximal: -0,263 rps Dauerhaft: -0,243 rps	Maximal: -0,165 rps Dauerhaft: -0,130 rps
PWM-Wert	Maximal: -1,161 Dauerhaft: -1,071	Maximal: -0,729 Dauerhaft: -0,573

Auch bei einer Dämpfung von  $D = 1$  zeigt sich ein ähnliches Bild wie zuvor. Mit zunehmender Eigenkreisfrequenz verringern sich sowohl die maximalen als auch die dauerhaften Werte aller betrachteten Größen. Die Winkeländerung fällt bei höherer Eigenkreisfrequenz erneut deutlich geringer aus, was auf ein stabileres Systemverhalten schließen lässt. Die Winkelgeschwindigkeit nimmt ebenfalls ab, was auf eine ruhigere Bewegung hinweist. Ähnlich verhält es sich mit der Regelgröße, deren Werte bei  $\omega_0 = 20$  deutlich niedriger liegen. Auch der PWM-Wert ist reduziert. Insgesamt bewirkt die Erhöhung der Eigenkreisfrequenz bei gleicher Dämpfung eine effizientere und dynamisch ruhigere Systemantwort. Das Verhalten bei einer Dämpfung von 1 führt dazu, dass das System kurzzeitig leicht überschwingt, dann aber direkt den neuen Wert erreicht. Ein stark überschwingendes System ist somit nicht mehr gegeben. Die erfassten Messverläufe sind in der unteren Abbildung 20 zu sehen.

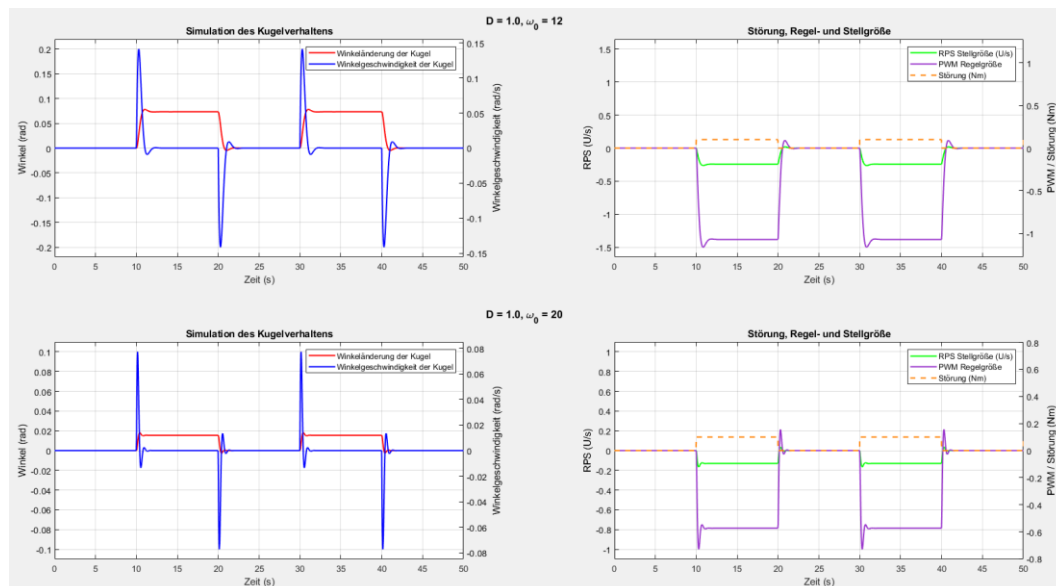


Abbildung 20: Simulationsdurchführung 2

Der dritte Simulationsdurchlauf erfolgte mit einer Dämpfung  $D = 1,5$ . In der Tabelle 3 sind die maximalen und dauerhaften Kurvenverläufe dargestellt

Tabelle 3: Messwerte bei fester Dämpfung und variierender Eigenkreisfrequenz 3

	$D = 1,5 \quad \omega_0 = 12$	$D = 1,5 \quad \omega_0 = 20$
Winkeländerung	Maximal: $0,073 \text{ rad}$ Dauerhaft: $0,073 \text{ rad}$	Maximal: $0,015 \text{ rad}$ Dauerhaft: $0,015 \text{ rad}$
Winkelgeschwindigkeit	Maximal: $0,109 \frac{\text{rad}}{\text{s}}$	Maximal: $0,061 \frac{\text{rad}}{\text{s}}$
RPS	Maximal: $-0,243 \text{ rps}$ Dauerhaft: $-0,243 \text{ rps}$	Maximal: $-0,145 \text{ rps}$ Dauerhaft: $-0,130 \text{ rps}$
PWM-Wert	Maximal: $-1,071$ Dauerhaft: $-1,071$	Maximal: $-0,640$ Dauerhaft: $-0,573$

Bei einer Dämpfung von  $D = 1,5$  zeigt sich erneut eine Verbesserung des Systemverhaltens mit erhöhter Eigenkreisfrequenz. Im Testdurchlauf mit  $\omega_0 = 12$  entsprechen die Maximalwerte den dauerhaften Werten, was auf ein schwingungsfreies Verhalten des Systems hinweist. Im Vergleich dazu liegen beim Durchlauf mit  $\omega_0 = 20$  sämtliche Werte zwar unterhalb der zuvor betrachteten Werte, jedoch ist ein leichtes Überspringen insbesondere beim PWM-Signal erkennbar. Dies deutet auf eine kurzfristig stärkere Reaktion des Reglers hin, bevor sich das System stabilisiert. Die Messverläufe sind in Abbildung 21 dargestellt.

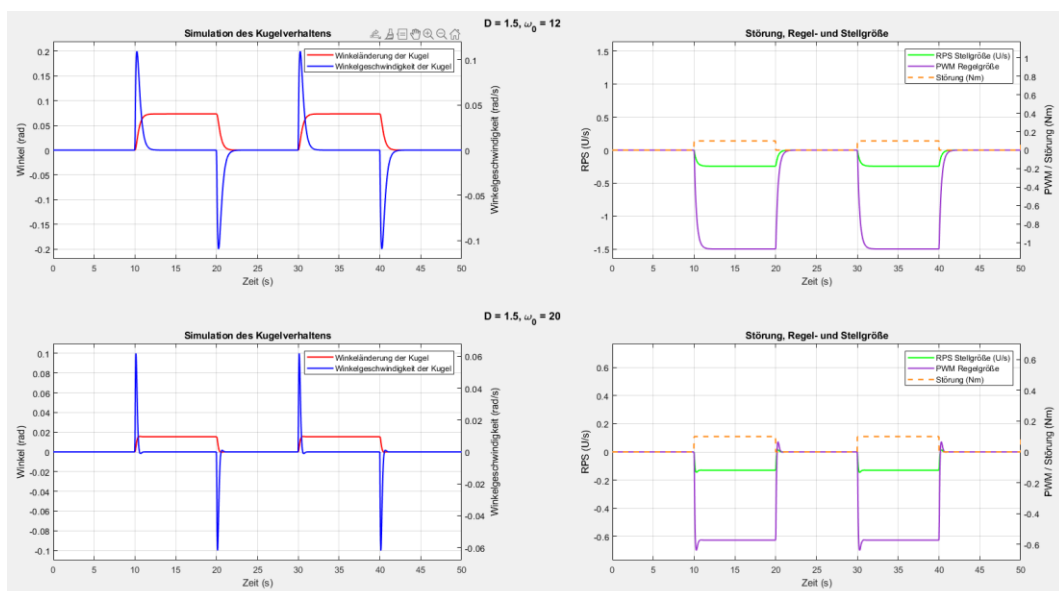


Abbildung 21: Simulationsdurchführung 3

Die durchgeführten Simulationen zeigen, dass sowohl die Erhöhung der Dämpfung  $D$  als auch der Eigenkreisfrequenz  $\omega_0$  einen maßgeblichen Einfluss auf das Systemverhalten haben. Insbesondere eine Erhöhung von  $\omega_0$  führt in allen betrachteten Fällen zu deutlich geringeren Ausschlägen bei Winkeländerung, Winkelgeschwindigkeit, RPS- und PWM-Werten. Dies bedeutet, dass das System ruhiger, stabiler und effizienter regelt.

Für dieses System ist es entscheidend, einen Dämpfungswert  $\geq 1$  zu wählen, um Schwingungen zu vermeiden. Damit das System schnell auf mögliche Störungen reagiert, ist ebenfalls eine höhere Eigenkreisfrequenz notwendig.

### **3.3.3 Simulationsergebnisse**

Die in der Simulation gewonnenen Ergebnisse zeigen, dass die Regelung des Systems mit dem berechneten Zustandsregler mathematisch möglich ist. Innerhalb der Simulation konnten verschiedene Reglerparameter getestet werden, um das gewünschte Systemverhalten zu erzielen. Das System reagiert aktiv auf Störungen und kompensiert diese, solange sie anliegen. Nach dem Ende der Störung kehrt die Kugel zuverlässig in ihre Ausgangslage zurück. Die erwarteten negativen Regelgrößen- und RPS-Werte wurden bei einer positiven Störung erfolgreich erzielt. Entsprechend führte eine negative Störung zu positiven Werten dieser Größen.

Ziel der Simulation war es, das Verhalten des Systems in einer virtuellen Umgebung zu visualisieren und zu analysieren, ohne dabei auf die reale Hardware angewiesen zu sein. Darüber hinaus sollte durch die Simulation das Systemverhalten so abgestimmt werden, dass die ermittelten Reglerwerte im weiteren Verlauf auf das reale System übertragen werden können.

Die Dämpfung wurde auf  $D \geq 1$  festgelegt, da dieser Wert ein gedämpftes System beschreibt. Dadurch wird ein Überschwingen weitgehend vermieden. Dämpfungswerte  $< 1$  führen zu einem unterdämpften, schwingungsfähigen System, während Werte  $> 1$  ein überdämpftes Verhalten zur Folge haben, bei dem das System zwar nicht überschwingt, jedoch langsamer reagiert. Die Simulationsergebnisse mit diesen Dämpfungswerten sind in der unteren Abbildung 22 visualisiert. Dabei entspricht der Dämpfungswert  $D = 0,5$  der roten,  $D = 1$  der blauen und  $D = 1,5$  der grünen Messkurve.

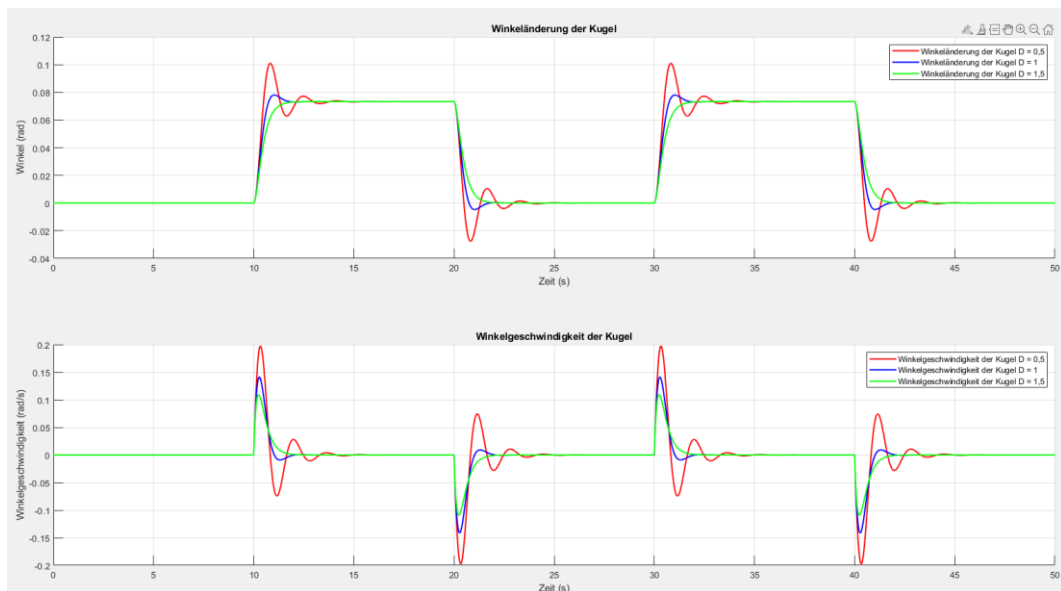


Abbildung 22: Dämpfungsvergleich

Die Eigenkreisfrequenz kann dabei individuell an die Dynamik des Systems angepasst werden. Aufgrund der während der Simulation gesammelten Daten, wird der Systemtest mit  $D = 1$  und  $\omega_0 = 12$  gestartet.

### 3.4 Hard- und Software: Entwicklung, Umsetzung und Tests

Dieses Kapitel beschreibt die Umsetzung der Komponenten anhand der Komponentenspezifikationen.

#### 3.4.1 Rahmenkonstruktion

Die Aluminium-Rahmenkonstruktion wurde mit dem online Engineeringtool der Firma *item* entworfen und anschließend in der Projektwerkstatt der *HSHL* gefertigt, zu sehen in Abbildung 23. Der Rahmen bildet die Basis zur Befestigung aller Komponenten und sorgt für ein kompaktes Gesamtsystem. Im Anhang B sind die technischen Zeichnungen aufgeführt.

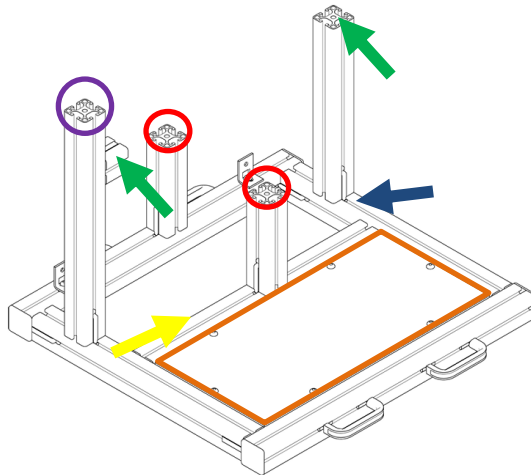
Die Radhalterungen werden mit Nutensteinen auf den beiden rot hervorgehobenen Profilen montiert. An den grün markierten Positionen ist die Befestigung der Auffangarme vorgesehen. Die Sensorhalterung wird auf das linke, senkrechte Aluprofil aufgeschoben (lila Kreis). Die Nachspannvorrichtung für den Riemen mit Motor befindet sich neben der Radhalterung (gelber Pfeil).

Elektronische Komponenten wie Arduino und Doppel H-Brücke werden auf der Holzplatte (orangener Rahmen, hinten) montiert. Eine Kombination aus Bodenplatte und Abdeckung schützt sie vor der Kugel, siehe Anhang C. Der Not-

Aus-Knopf ist hinter dem rechten, senkrechten Profil angebracht (blauer Pfeil) und eine Plexiglasscheibe an der Vorderseite schirmt das System ab.

Offene Schnittkanten der Aluprofile sind mit Abdeckungen versehen. Der Rahmen hat zudem vier Tragegriffe.

(a) Aluminiumrahmen Modell



(b) Gefertigter Aluminiumrahmen

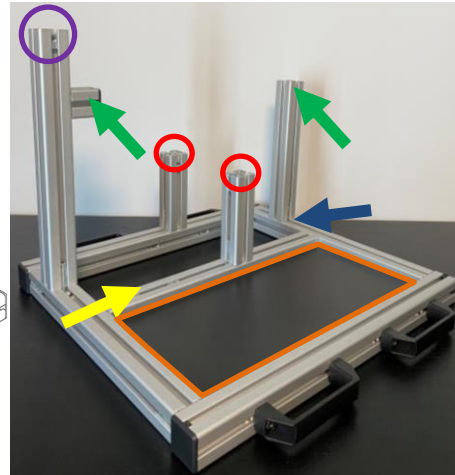


Abbildung 23: Aluminiumrahmen Aufbau  
Quelle: [14]

### 3.4.2 Komponentenentwicklung

Die Komponenten werden, sofern möglich, in den Bereichen Hardware, Software und elektrische Verbindung beschrieben. Alle 3D-Modelle wurden eigenständig modelliert und mittels FDM-Druckverfahren (Fused Deposition Modeling z. Dt. Düsensmelzverfahren) hergestellt. Des Weiteren werden in den einzelnen Komponenten die Ergebnisse der Komponententests dargestellt. Eine Parameterdatei wird in MATLAB erstellt, um alle Parameter zentral anzuzeigen und eine einfache Anpassung zu ermöglichen, siehe Anhang D.

#### 3.4.2.1 Sensorik

##### Hardware

Für den Sensor wurde ein Gehäuse entworfen, in dem der Sensor fest verschraubt werden kann (blauer Pfeil). Dieses Gehäuse umschließt den Sensor und schützt vor unerwünschtem Umgebungslicht. Des Weiteren wurde ein Ausgang für die Kabel (lila Pfeil) berücksichtigt, sodass diese nicht direkt sichtbar sind. Das Gehäuse wird auf das Aluminiumprofil gesteckt und durch zwei Aussparungen mit Nutensteinen in dem Profil verschraubt (grüner Pfeil). In Abbildung 24 sind das Modell und die 3D gedruckte Halterung dargestellt.

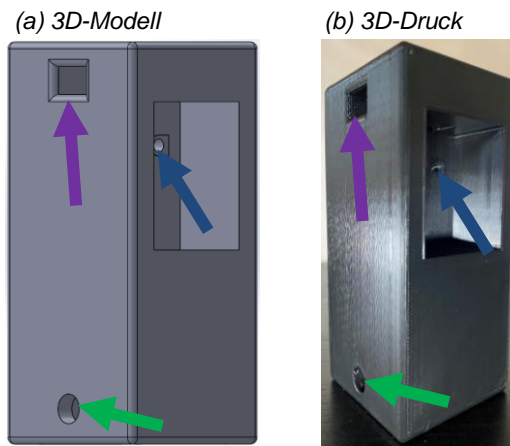


Abbildung 24: Sensorgehäuse

## Software

Die Softwareumsetzung der Komponente Sensorik erfolgt in mehreren Schritten. Zunächst gibt der ToF-Sensor die gemessene Entfernung in Millimetern als Signal *SEN\_Ist-Entfernung\_mm* aus. Dieses Signal wird in einen double-Wert konvertiert und anschließend über einen Saturation-Block auf den Bereich von  $0\text{ mm}$  bis  $400\text{ mm}$  begrenzt. Danach erfolgt die Umrechnung in Meter durch Division mit dem Parameter *PAR\_SEN\_Umrechnung*.

Zur Umrechnung der Entfernung in einen Winkel wird der Kosinussatz verwendet:

$$\cos(\gamma) = \frac{a^2 + b^2 - c^2}{2 \cdot a \cdot b}. \quad (3.1)$$

Um den Winkel  $\gamma$  zu berechnen, wird der Arkuskosinus angewendet:

$$\gamma = \cos^{-1} \left( \frac{a^2 + b^2 - c^2}{2 \cdot a \cdot b} \right), \text{ siehe [24, S. 57]}. \quad (3.2)$$

Übertragen auf dieses System ergibt sich:

$$\varphi = \cos^{-1} \left( \frac{x^2 + r^2 - d^2}{2 \cdot x \cdot r} \right). \quad (3.3)$$

Dabei steht  $x$  für den Abstand vom Sensor zur Radachse,  $r$  für den Abstand der Kugelmitte zur Radachse und  $d$  für die aktuell gemessene Entfernung. Zur Bestimmung der Position relativ zur Ruhelage wird die Differenz zum Winkel im Ruhezustand gebildet, sodass dieser immer  $0\text{ rad}$  ( $0^\circ$ ) entspricht. Das resultierende Signal *SEN\_Winkel* wird anschließend an die nachfolgende

Komponente weitergeleitet. Der MATLAB Function Code zur Winkelberechnung ist im Anhang E aufgeführt.

Abbildung 25 stellt die Umsetzung der Software dar.

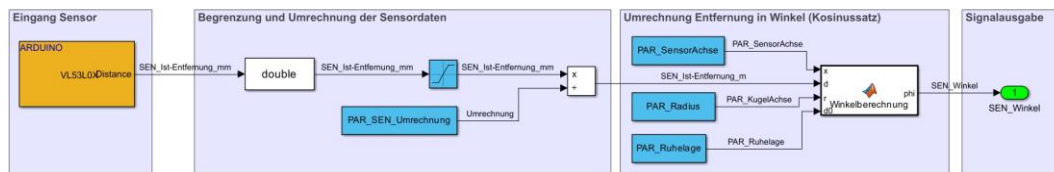


Abbildung 25: Softwaretechnische Umsetzung der Komponente Sensorik

## Elektrische Verbindung

Die Spannungsversorgung des Sensors erfolgt direkt über den Mikrocontroller. Die Signalleitungen *SDA* und *SCL* werden an die entsprechenden *I<sup>2</sup>C*-Pins angeschlossen (vgl. Kapitel 3.2.2).

## Komponententest

Durch die Komponententests konnte die Reproduzierbarkeit von Messergebnissen unter identischen Bedingungen bestätigt werden. Gleichzeitig zeigte sich, dass die Eingangssignale Schwankungen aufweisen, die eine Glättung erfordern. Zur Verbesserung der Signalqualität wurde daher ein PT1-Filter eingesetzt, der in der Komponente Signalverarbeitung implementiert wurde. Die folgende Abbildung 26 zeigt den Vergleich der Originalmessung (rot) mit gefilterten Signalen bei unterschiedlichen Filterfrequenzen (0,63 Hz bis 30 Hz). Dabei lässt sich erkennen, dass niedrigere Frequenzen (z. B. 0,63 Hz in blau) das Signal stark glätten und Details verlieren, während höhere Frequenzen (z. B. 30 Hz in hellblau) das Rauschen weniger unterdrücken, aber die Signalstruktur weitgehend erhalten bleibt. Der Kompromiss zwischen Glättung und Signaltreue liegt bei mittleren Frequenzen wie 10 Hz oder 15 Hz, die ein ausgewogenes Bild der Originaldaten liefern. Aus diesem Grund wurde sich für eine Filterung mit 15 Hz entschieden.



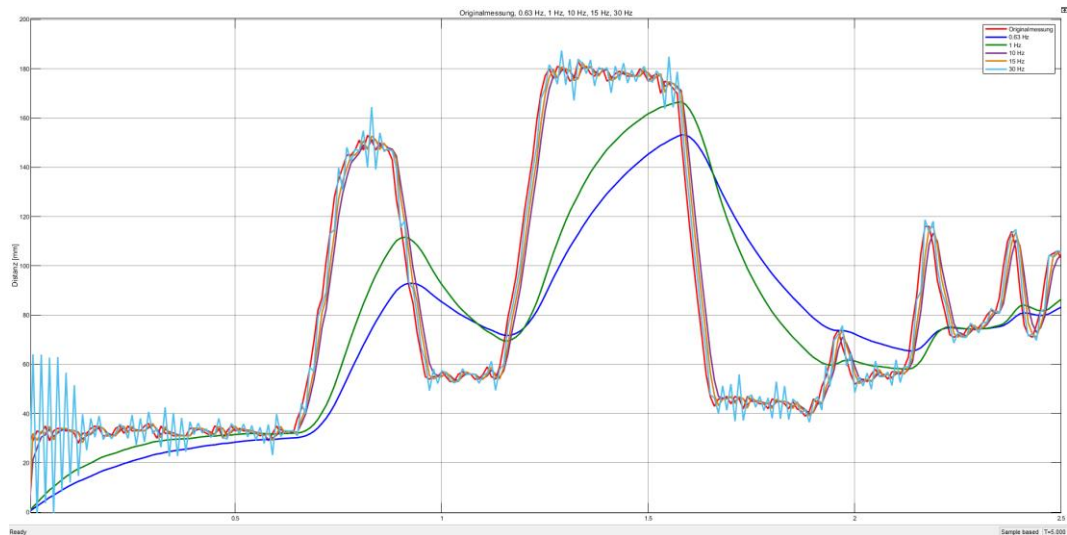


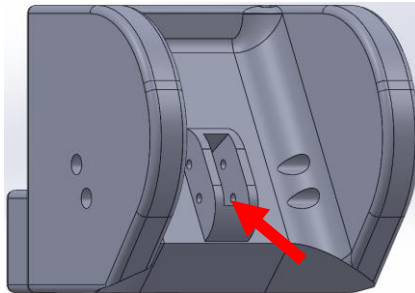
Abbildung 26: PT1-Filter Vergleich

### 3.4.2.2 Taster

#### Hardware

Damit die Kugel nicht vollständig vom Rad fallen kann, wurden Auffangarme modelliert, die links und rechts neben dem Rad angebracht werden. Diese sind in Abbildung 27 zu sehen. Um die Kugel zu erfassen, sobald diese in den Auffangarmen liegt, werden Mikrotaster integriert (roter Pfeil). Diese geben die Lageposition an die Software weiter.

(a) 3D-Modell



(b) 3D-Druck



Abbildung 27: Auffangarm

#### Software

Im Bereich *Eingang Taster* werden die Signale *TST\_DigitalerSchaltzustandL* und *TST\_DigitalerSchaltzustandR* verarbeitet. Diese stammen von den Mikrotastern und dienen der Richtungssteuerung sowie weiteren internen Abläufen.

Zur Erzeugung eines Gegenimpulses werden beide Signale einem XOR-Block zugeführt. Das resultierende Signal *TST\_XORStatus* aktiviert bei ungleichen Zuständen ein Stateflow-Chart, das kurzzeitig eine Beschleunigung mit einem

PWM-Wert von 170 an den Antrieb ausgibt. Anschließend wird ein abrupter Stopp durch einen Gegenimpuls mit einer PWM von 255 eingeleitet. Die Signale *TST\_AnsteuerungSwitchBlock* und *TST\_Gegenimpuls* werden dabei an die Komponente Antrieb übergeben.

In der nachfolgenden Schaltungsebene werden die Signale mit logischen UND-Verknüpfungen ausgewertet, um abhängig vom betätigten Taster gezielt *TST\_SchaltungGegenimpulsL* oder *TST\_SchaltungGegenimpulsR* zu aktivieren.

Die Parameter *PAR\_TST\_PWM* und *PAR\_TST\_PWMGI* werden ebenfalls in dieser Komponente definiert und an den Antrieb übergeben.

Die Software der Taster ist in der unteren Abbildung 28 dargestellt.

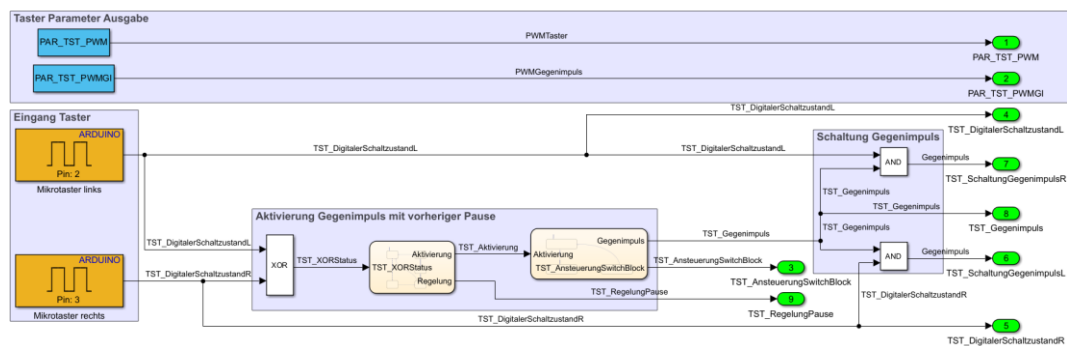


Abbildung 28: Softwaretechnische Umsetzung der Komponente Taster

## Elektrische Verbindung

Die Mikrotaster sind im NC-Modus (normally closed z. Dt. normalerweise geschlossen) geschaltet und mit *GND* verbunden. Der linke Taster ist an Pin 2 und der rechte Taster an Pin 3 des Arduino angeschlossen. Durch die Aktivierung der internen Pull-up-Widerstände liegt im Normalzustand ein High-Signal an, das bei Betätigung auf Low wechselt.

## Komponententest

Bei den Tests zeigte sich eine Instabilität des Signals, welche durch die Aktivierung interner Pull-up-Widerstände in Simulink gelöst werden konnte. Durch diese Aktivierung sind stabile Signalzustände gewährleistet.

Die erwarteten Signale *TST\_AnsteuerungSwitchBlock* und *TST\_Gegenimpuls* werden durch die Taster korrekt ausgelöst. Zusätzlich wurde ein vorgelagerter Stateflow-Chart implementiert, das bei Betätigung eines Tasters eine kurze Systempause einleitet, bevor der Gegenimpuls folgt. Auch diese Funktion wurde

erfolgreich getestet: Das Signal *TST\_RegelungPause* wurde korrekt ausgegeben und *TST\_Aktivierung* wechselte nach Ablauf der Pause zuverlässig auf 1.

### 3.4.2.3 Signalverarbeitung

#### Software

Im Bereich *Signaleingang und Filterung* wird das Eingangssignal mithilfe eines PT1-Filters mit einer Frequenz von 15 Hz geglättet. Während das ungefilterte Signal *SEN\_Winkel* im Bereich *Regelung pausieren* weiterverwendet wird, dient das gefilterte Signal *SEN\_WinkelPT1* als Eingang für den Zustandsregler.

Eine logische Schaltung stellt im Bereich *Regelung pausieren* sicher, dass die Regelung nur dann aktiv ist, wenn sich die Kugel innerhalb des Messbereichs befindet. Überschreitet der gemessene Wert den Schwellenwert, wird der Ausgang auf 0 gesetzt, wodurch die Regelung pausiert und ungewollte Reaktionen vermieden werden.

Im Bereich *Zustandsregler* erfolgt die Berechnung der Regelgröße auf Basis der Eingangssignale und der zuvor definierten Verstärkungswerte  $K_1$  und  $K_2$  (vgl. Kapitel 2.4). Als Eingangssignale dienen der Winkel und die Winkelgeschwindigkeit (1. Ableitung). Die Regelgröße wird anschließend an den Bereich *Richtungsansteuerung durch die Sensorwerte* übergeben, in dem auf Basis des Vorzeichens die Drehrichtung bestimmt wird. Positive Werte führen zur Linksdrehung, negative zur Rechtsdrehung.

Innerhalb des Bereichs *PWM Offset für den Motor* wird ein Offset von 26 berücksichtigt: Bei negativen Regelgrößen wird das Offset zunächst mit  $-1$  multipliziert und dann zur Regelgröße addiert, bei positiven Werten erfolgt die Addition direkt. Diese Anpassung wurde in den Komponententests des Antriebs ermittelt und ist in Kapitel 3.4.2.5 aufgeführt.

Die nachfolgende Abbildung 29 stellt die Software der Signalverarbeitung dar.

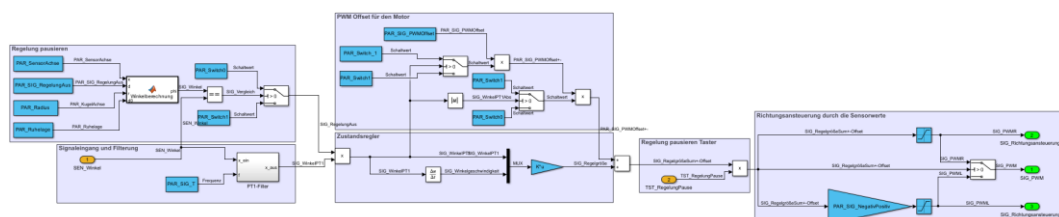


Abbildung 29: Softwaretechnische Umsetzung der Komponente Signalverarbeitung

## Komponententest

Die Pausierung der Regelung wurde erfolgreich ab einem Abstand von  $0,4\text{ m}$  ( $0,6108\text{ rad} \triangleq \sim 35^\circ$ ) verifiziert. Werte darunter wurden regulär verarbeitet. Der Zustandsregler lieferte bei Ruhelage der Kugel eine Regelgröße von 0, bei Abweichungen entsprechend positive oder negative Werte.

Die Richtungsansteuerung reagierte korrekt auf das Vorzeichen der Regelgröße. Ebenso wurde die korrekte Ausgabe der PWM über den Switch Block bestätigt.

Der Stopp des Systems über die Taster funktionierte wie vorgesehen. Das Signal *TST\_RegelungPause* setzte die Regelgröße temporär auf 0. Nach Ablauf der Pause wurde die Regelung wieder aktiviert.

### 3.4.2.4 Richtungsansteuerung

#### Software

In der Komponente Richtungsansteuerung werden die Eingangssignale zur Drehrichtungssteuerung verarbeitet. In den Bereichen *Schaltung Richtungsansteuerung links* und *Schaltung Richtungsansteuerung rechts* erzeugen die Signale *SIG\_RichtungsansteuerungL* und *SIG\_RichtungsansteuerungR* die entsprechende Ansteuerung des Motors.

Die Signale *TST\_DigitalerSchaltzustandL* und *TST\_DigitalerSchaltzustandR* werden im Bereich *Schaltung Tasterwerte* verarbeitet. Bei Betätigung eines Tasters wird unmittelbar die Richtungsansteuerung erzeugt.

Im Bereich *Ansteuerung des Gegenimpulses* wird bei gleichzeitigem Vorliegen eines Taster- und Gegenimpuls-Signals ein starker, kurzer Gegenimpuls ausgelöst. Dieser bewirkt ein abruptes Stoppen der Drehung und ein gezieltes Anheben der Kugel aus den Auffangarmen.

Der beschriebene Aufbau der Software ist in Abbildung 30 zu sehen.

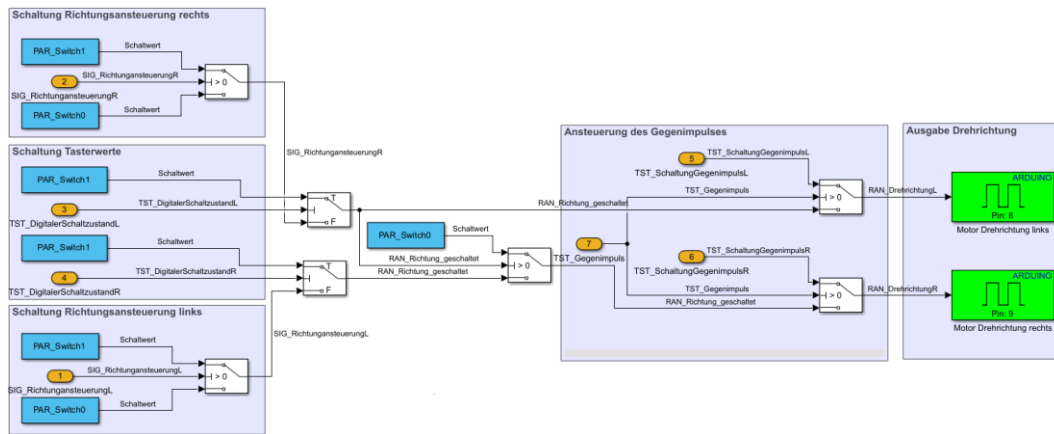


Abbildung 30: Softwaretechnische Umsetzung der Komponente Richtungsansteuerung

## Elektrische Verbindung

Die Richtungsansteuerung erfolgt über digitale Ausgangssignale (0 und 1). Eine Linksdrehung wird über Pin 8 und eine Rechtsdrehung über Pin 9 aktiviert. Die Signale werden direkt an die Doppel H-Brücke zur Motorsteuerung übermittelt.

## Komponententest

Die Funktionalität der Richtungsansteuerung konnte in allen Tests erfolgreich bestätigt werden. Sowohl linke als auch rechte Drehrichtungen wurden korrekt ausgeführt. Die Priorisierung der Taster-Signale sowie die Auslösung des Gegenimpulses funktionierten zuverlässig. Die Komponente reagierte erwartungsgemäß auf alle Eingangssignale.

### 3.4.2.5 Antrieb

#### Hardware

Für den Antrieb des Rads wurde ein Riemenantrieb mit 2GT-Zahnriemenscheiben gewählt, da dieser eine robuste Umsetzung sowie eine zuverlässige Kraftübertragung ermöglicht. Zum Einsatz kommen eine Zahnriemenscheibe mit 80 Zähnen an der Radachse und eine mit 20 Zähnen an der Motorwelle. Die resultierende Übersetzung wird mit folgender Formel berechnet:

$$u = \frac{z_{\text{Großrad}}}{z_{\text{Kleinrad}}} \geq 1, \text{ siehe [6, S. 161].} \quad (3.4)$$

Die Übersetzung des Systems steht somit im Verhältnis 4 : 1. Das bedeutet, dass sich das Rad viermal langsamer dreht, dafür aber mit vierfachem Drehmoment.

Verwendet wird ein geschlossener 2GT-Zahnriemen mit einer Länge von 0,5 m.

Zur Nachspannung des Riemens ist der Motor auf einem verschiebbaren Schlitten (blauer Punkt) montiert, der entlang zweier M4-Gewindestangen (rote Punkte) geführt wird, welche in einem Rahmen (grüner Punkt) fixiert sind. Rändelmutter ermöglichen die Fixierung des Motors in der gewünschten Position, wobei Gegenmutter zusätzlich für Sicherheit sorgen. Die Konstruktion ist in Abbildung 31 dargestellt.

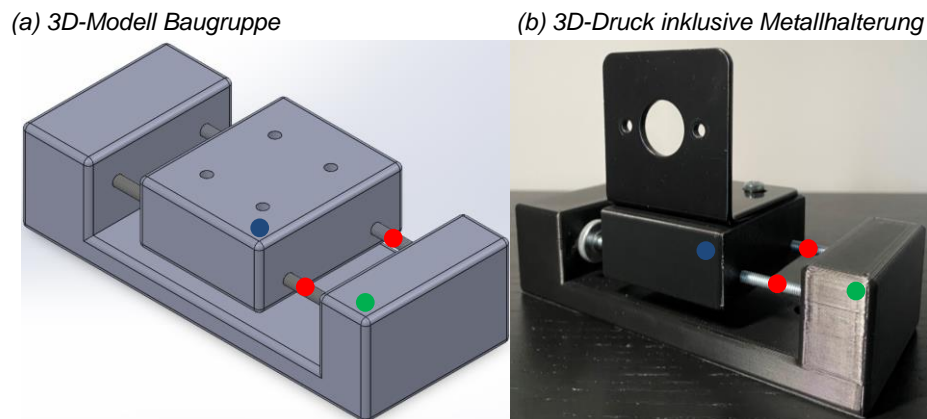


Abbildung 31: Nachspannvorrichtung

## Software

Im Bereich *Schaltung der PWM auf Basis der Komponente Taster* steuert das Signal *TST\_Gegenimpuls* die Ausgabe einer gewünschten PWM. Bei einem Signalwert von 0 wird der Parameter *PAR\_TST\_PWM* ausgegeben. Ist das Signal hingegen 1, wird *PAR\_TST\_PWMGI* aktiviert, wodurch ein Gegenimpuls mit einer PWM von 255 erzeugt wird, der zu einem abrupten Stopp des Rads führt.

Im Anschluss erfolgt die Weiterverarbeitung des PWM-Signals im Bereich *Schaltung der gesteuerten oder geregelten PWM*. Ein Switch-Block, gesteuert durch *TST\_AnsteuerungSwitchBlock*, entscheidet über die Signalquelle: Bei Signalwert 1 wird der zuvor ermittelte PWM-Wert ausgegeben, bei 0 die vom Regler berechnete PWM.

Das endgültige PWM-Signal wird im Bereich *Ausgabe Drehgeschwindigkeit* durch einen uint8-Block geleitet, wodurch Nachkommastellen aufgerundet und nur ganzzahlige Werte ausgegeben werden. Der resultierende Wert wird schließlich an den digitalen PWM-Ausgang (Pin 5) des Arduino übertragen.

Die Umsetzung der Software ist in der nachstehende Abbildung 32 zu sehen.

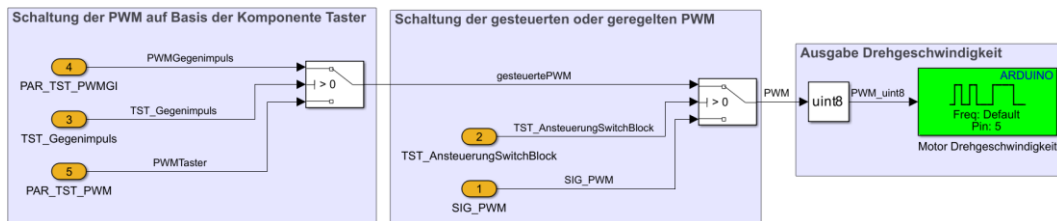


Abbildung 32: Softwaretechnische Umsetzung der Komponente Antrieb

## Elektrische Verbindung

Die Ansteuerung des Motors erfolgt über eine Doppel H-Brücke auf Steuerlogik der L298 Doppel H-Brücke. Diese wird mit 5 V und GND des Mikrocontrollers verbunden. Die Pins 8 und 9 steuern die Drehrichtung (IN1 und IN2), während der PWM-Pin 5 die Geschwindigkeitsregelung (ENA) übernimmt.

Das externe Netzteil *Mean Well OWA-200E-12*, mit einer Festspannung von 12 V, 15 A und 180 W, versorgt die Doppel H-Brücke über die Schraubklemmen P2 (Plus) und PGND (Minus). Der Motor wird über die Ausgänge Out1 (Plus) und Out2 (Minus) mit den entsprechenden Polen verbunden.

## Komponententest

Es zeigte sich, dass der Motor erst ab einer PWM von 26 anläuft. Dieses Offset wird in der Signalverarbeitung berücksichtigt, sowohl bei positiven als auch bei negativen Regelgrößen.

Bei schnellen Richtungswechseln kam es zu Funkenbildung am Motor. Um dies zu verhindern, wurden Entstörkondensatoren (22 nF und 47 nF Keramikkondensatoren) parallel zwischen Plus und Minus des Motors verlötet, um Spannungsspitzen abzufangen. Zwei weitere 22 nF Keramikkondensatoren dienen dazu, Störungen abzuleiten und wurden jeweils zwischen einem Pol und dem Gehäuse des Motors verbunden, siehe Anhang F.

### 3.4.2.6 Radhalterung

#### Hardware

Die Fixierung der Radachse erfolgt mithilfe von Kugellagern, um eine rotatorische Bewegung zu ermöglichen. Die Kugellager sind in die Aussparungen (grüne Kreise) der Halterungen eingepresst und das Bauteil auf den Aluminiumprofilen befestigt. Eine Verschraubung der Halterungen von oben und vorne (lila Kreise) sorgt für sicheren Halt und verhindert ein Lösen oder Herausdrehen der Achse.



Die Modelle und gedruckten Radhalterungen sind in der unteren Abbildung 33 dargestellt.

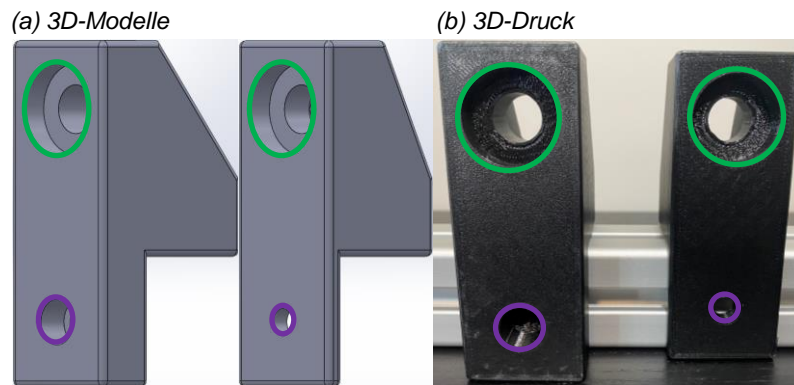


Abbildung 33: Radhalterungen

### Komponententest

Die Komponententests zeigten, dass Kugellager die beste Lösung für die drehbare Lagerung der Radachse sind. Sie ermöglichen eine nahezu reibungsfreie Rotation und lassen sich leicht in die Halterungen einpressen. Der geplante Aufbau aus Kapitel 3.1.1 konnte aufgrund der Instabilität der Halterung nicht umgesetzt werden. Durch die Rotation des Rads hatten sich die Seitenhalterungen nach außen gebogen, wodurch das Rad nicht mehr zentriert und fest gelagert war. Aus diesem Grund wurde diese Variante verworfen. Die Baugruppe ist im Anhang G aufgeführt. Zur Lösung wurden die in Abbildung 33 dargestellten stabileren Halterungen entwickelt, die auf dem Aluminiumprofil verschraubt werden und das Rad somit sicher fixieren.

#### 3.4.2.7 Rad

##### Hardware

Ein Verbindungsstück wurde modelliert, um die Riemenbewegung auf das Rad zu übertragen. Das Verbindungsstück ist innen hohl, sodass es einfach auf die Achse geschoben werden kann. An der Seite, die an der Radnabe anliegt, wurde eine Aussparung für die Mutter modelliert (blauer Pfeil), wodurch das Verbindungsstück auf die Mutter gesteckt werden kann, um die Kraft auf das Rad zu übertragen. Die Riemenscheibe wird auf das Verbindungsstück aufgeschoben (gelber Pfeil) und mit Gewindestiften durch die Löcher (lila Pfeile) fest an der Achse verschraubt. Der obere Teil des Verbindungsstücks (grüner Pfeil) ist so gestaltet, dass er in die Kugellager eingesetzt werden kann, um die rotatorische



Bewegung zu ermöglichen. Sowohl das Modell als auch das gedruckte Bauteil sind in der folgenden Abbildung 34 zu sehen.

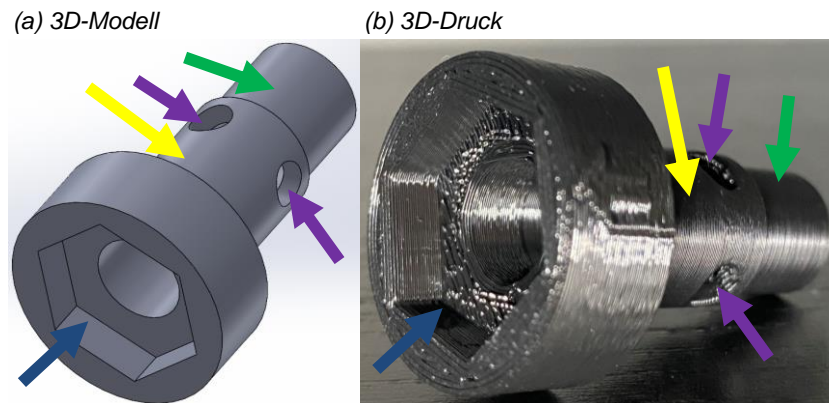


Abbildung 34: Verbindungsstück Antrieb

### Komponententest

Das Rad mit 32 cm Durchmesser wurde als geeignet ermittelt. Der Felgenreand muss jedoch mit einem Gummi-Kantenschutz versehen werden, um ausreichend Reibung zwischen Rad und Kugel zu gewährleisten. Die rotatorisch gelagerte Achse musste für eine feste Kraftübertragung mit dem Rad verschraubt werden, wobei Schraubenrosetten für die zentrierte Ausrichtung sorgen.

Das Verbindungsstück wurde entwickelt, um die Kraft- und Drehzahlübertragung auf Achse und Mutter zu ermöglichen. Da die Riemenscheibe nur zwei Gewindelöcher hatte, wurden zwei weitere gebohrt, um ein gleichmäßiges Anziehen zu gewährleisten. Auf der anderen Seite der Radnabe wird eine 3D-gedruckte Hülse verwendet, die das Ausdrehen aus dem Kugellager verhindert. Abschließend wurde das Rad manuell ausgewuchtet.

#### 3.4.2.8 Kugel

##### Hardware

Als Kugel wird eine graue Gummikugel mit 50 mm Durchmesser verwendet, zu sehen in Abbildung 35.



Abbildung 35: Gummikugel

## **Komponententest**

Der Komponententest ergab, dass ein Kugeldurchmesser von 50 *mm* die stabilste Lage auf dem Rad erzielt. Ein Durchmesser von 40 *mm* war zu klein, während 60 *mm* eine zu geringe Auflagefläche hatten. Die graue Gummikugel wurde aufgrund ihrer hohen Reibung, ausreichendem Gewicht und präziser Sensorerfassung als geeignet ausgewählt. Alle anderen Kugeln waren aufgrund von Gewicht, Reibung oder Reflexion ungeeignet.

### **3.4.3 Integrationstest**

Während des Integrationstests wurde überprüft, ob die einzelnen Softwarekomponenten korrekt miteinander interagieren. Die Komponenten arbeiten als Gesamtsystem reibungslos zusammen. Alle durchgeführten Tests wurden erfolgreich abgeschlossen.

## **3.5 Zusammenführung von Software und Hardware**

Mit den zuvor beschriebenen Komponenten konnte die Zusammenführung des Projekts erfolgen.

Zunächst wurden die Radhalterungen mit Kugellagern und Rad an der definierten Position montiert. Darauf aufbauend wurden Sensorhalterung samt Sensor sowie die Auffangarme mit Mikrotastern auf das Rad ausgerichtet. Danach folgte der Zusammenbau der Motorhalterung: Die Metallhalterung wurde verschraubt, der Motor eingesetzt und die gesamte Einheit neben der Radhalterung auf dem Aluminiumprofil befestigt, sodass der Riemen geradlinig geführt wird.

Arduino und Doppel H-Brücke wurden in der Komponentenhalterung verschraubt und auf das Holzbrett geklebt, wodurch eine Umpositionierung möglich ist. Auch der Not-Aus-Knopf wurde an der dafür vorgesehenen Position montiert. Nach vollständiger Montage und Ausrichtung aller Bauteile auf dem Aluminiumrahmen erfolgte die systematische Verkabelung.

Die Verkabelung orientierte sich am Verdrahtungsplan aus Kapitel 3.2.2. Der Sensor wurde mit einem neuen 4-poligen JST-XH-Stecker versehen und über 5 *V*, *GND*, *SDA* und *SCL* an den Mikrocontroller angeschlossen. Die Mikrotaster wurden, wie vorgesehen, mit *GND* und digitalen Eingängen verbunden. Die Doppel H-Brücke erhielt ihre Verbindung über 5 *V*, *GND* sowie die Pins 5, 8 und 9 des Mikrocontrollers.

Die externe Spannungsversorgung erfolgt über das genannte Netzteil, dessen 4-poliger Stecker durch einen XT60-Stecker ersetzt wurde. Die *GND*-Leitung führt direkt zur Doppel H-Brücke, während die 12 V-Leitung zunächst in den Not-Aus-Knopf (NC-Anschluss) und von dort zur Doppel H-Brücke weitergeführt wird. Bei Betätigung wird so die Stromzufuhr zum Motor unterbrochen. Der DC-Motor wurde wie geplant mit der Doppel H-Brücke verbunden.

Zur strukturierten Kabelführung wurden eigens modellierte und 3D-gedruckte Kabelhalter in die Aluminiumprofile eingesetzt. Als Halterungen für die Plexiglasscheibe wurden ebenfalls Halter modelliert, mit der die Scheibe verschraubt wird, zu sehen im Anhang H. Diese ist so zugeschnitten, geformt und montiert worden, dass das rotierende Rad nicht angefasst werden kann. Die Vorder- und Rückansicht des finalen Aufbaus sind in der untenstehenden Abbildung 36 dargestellt.

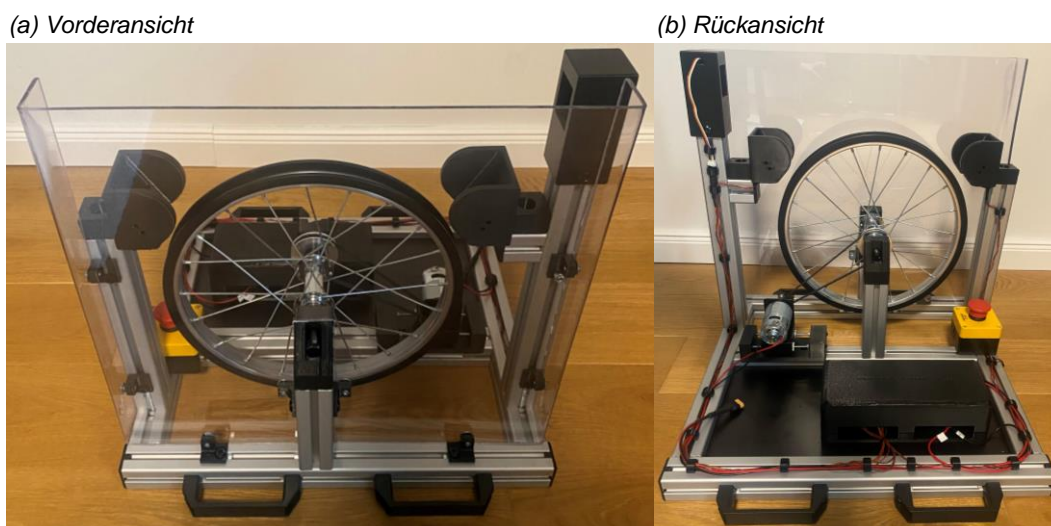


Abbildung 36: Finaler Systemaufbau

Abschließend wurde der Arduino per USB mit dem Computer verbunden, um die finale Software aufzuspielen. Die Software ist im Anhang I aufgeführt. Alle verwendeten Bauteile sind im Anhang J vorzufinden.

## 4 Systemtest und Ergebnisse

Dieses Kapitel dient der Beschreibung des Systemtests und der Betrachtung der gewonnenen Ergebnisse im Vergleich von Simulation und realer Umsetzung. Ebenfalls werden die Ergebnisse analysiert und der Zustandsregler bewertet.

## 4.1 Systemtest

Um das Systemverhalten analysieren zu können, wurden Messungen durchgeführt. Die Dämpfung und Eigenkreisfrequenz waren dabei definiert als:  $D = 1$  und  $\omega_0 = 12$ .

Der erste Systemtest wird mit der in Kapitel 2.6 berechneten Frequenz von  $0,63 \text{ Hz}$  durchgeführt, um das Systemverhalten bei Anwendung der ermittelten Filterung zu analysieren.

In Abbildung 37 ist im linken Plot die gemessene Entfernung zur Kugel in Metern dargestellt. Der rechte Plot zeigt die daraus berechnete Winkelgröße (blau) und Winkelgeschwindigkeit (grün) der Kugel sowie das gefilterte Signal (rot) der Winkelgröße.

Wie im rechten Plot ersichtlich ist, folgt das gefilterte Signal dem tatsächlichen Verlauf der Winkeländerung mit erheblicher Verzögerung. Die Filterung unterdrückt alle dynamischen Anteile, was sich auch in der nahezu konstanten Steigung der Kurve zeigt.

Die Winkelgeschwindigkeit verdeutlicht hingegen die vorhandenen Bewegungsänderungen, die im gefilterten Signal jedoch nicht mehr erkennbar sind. Damit ist das System in dieser Konfiguration für Anwendungen mit schnellen Reaktionen ungeeignet.

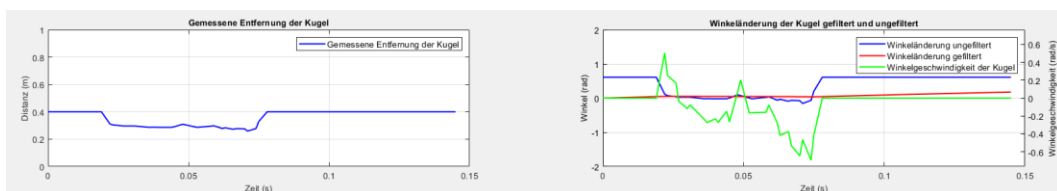


Abbildung 37: Systemtest Filterung mit  $0,63 \text{ Hz}$ ,  $D = 1$  und  $\omega_0 = 12$

Aus diesem Grund werden die weiteren Systemtests mit einer Filterfrequenz von  $15 \text{ Hz}$  durchgeführt. Abbildung 38 zeigt die erfassten Daten. Dabei sind sechs Plots zu sehen, die die Messergebnisse aus dem Systemtest visualisieren.

In dem Plot *Gemessene Entfernung der Kugel* ist zu sehen, dass der Wert  $0,4 \text{ m}$  ausgegeben wird, solange keine Kugel auf dem Rad platziert ist. Wird eine Kugel auf dem Rad platziert, erfasst der Sensor diese. In dem rechten Plot *Winkeländerung der Kugel gefiltert und ungefiltert* ist die Kugelposition in einen Winkel umgerechnet dargestellt. Deutlich zu sehen ist, dass die rote Linie, welche

das gefilterte Signal darstellt, zeitlich verzögert zu dem Ursprungssignal reagiert. Um die Winkelgeschwindigkeit (grüne Messkurve) zu erhalten, wird der Winkel abgeleitet. Dabei erzielen bereits kleine Änderungen große Winkelgeschwindigkeiten. Aufgrund der schwankenden Messwerte, werden auch bei gleichbleibender Position der Kugel vermeintliche Abweichungen wahrgenommen und dementsprechend Regelgrößen ermittelt.

Der Plot *Regelgröße* zeigt die Regelgröße und die Addition des Offsets des Motors. Die blaue Linie stellt dabei die ermittelte Regelgröße und die rote Linie die Regelgröße  $\pm$  Offset in Form einer PWM dar. Die Regelgröße überschreitet am Ende den maximal zu erreichenden Wert von 255.

Der Plot *PWM-Wert*, rechts daneben, stellt das ermittelte PWM-Signal dar, welches als Ausgabe der Komponente Signalverarbeitung fungiert. Die Ansteuerung der Drehrichtung nach links ist dabei in rot und die nach rechts in grün dargestellt. Die blaue Linie zeigt die PWM. Beim Betrachten des Plots *Drehrichtungsansteuerung* ist ersichtlich, dass bei einer Ansteuerung der Drehrichtung nach links (blau), auch der PWM-Wert für die Ansteuerung links Werte über 0 ausgibt. Für die Drehrichtung nach rechts dementsprechend auch.

Der rechte Plot *PWM-Ausgabe an den Motor* visualisiert die schlussendliche PWM, welche ausgegeben wird. Dieser PWM-Wert kann, anders als der Regelgrößenwert, keine Kommastellen annehmen, sondern wird immer auf ganzzahlige Werte aufgerundet.

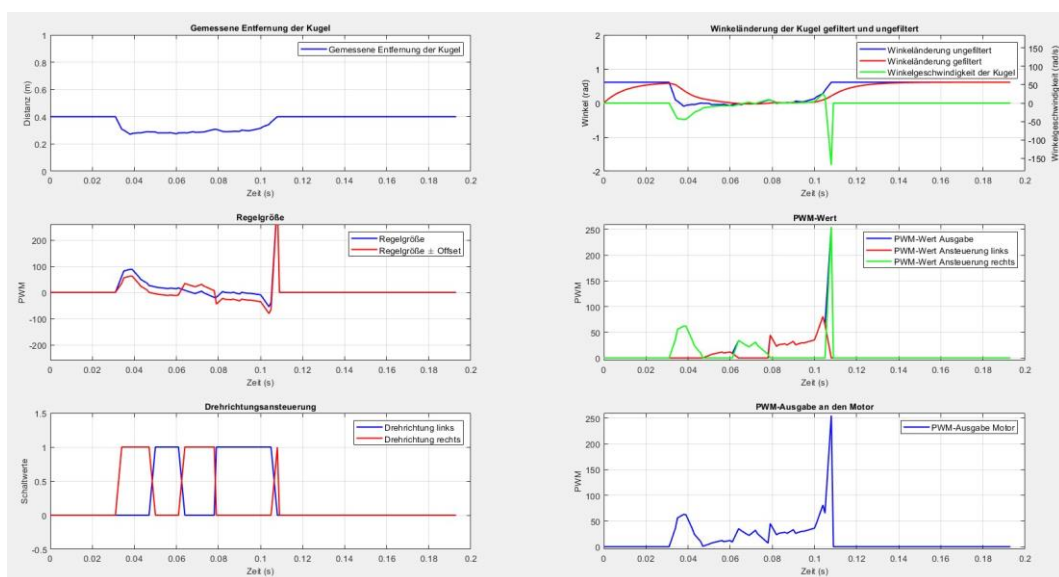


Abbildung 38: Systemtest 1 Filterung mit 15 Hz,  $D = 1$  und  $\omega_0 = 12$

Durch die Filterung war es möglich, die Schwankungen des Eingangssignals zu glätten, jedoch führte dies zu einer verzögerten Reaktion des Systems, wodurch keine akkurate Reaktion auf Änderungen erfolgte. Testläufe ohne Filterung führten wiederum zu einer direkten Reaktion des Systems, allerdings mit höheren Schwankungen, siehe Abbildung 39. Durch die schwankenden Eingangssignale erzeugt die Ableitung hohe Werte, wodurch die Zustandsregelung ebenfalls hohe Regelgrößen erzeugt, auch wenn die Kugel die Position nicht verlassen hat.

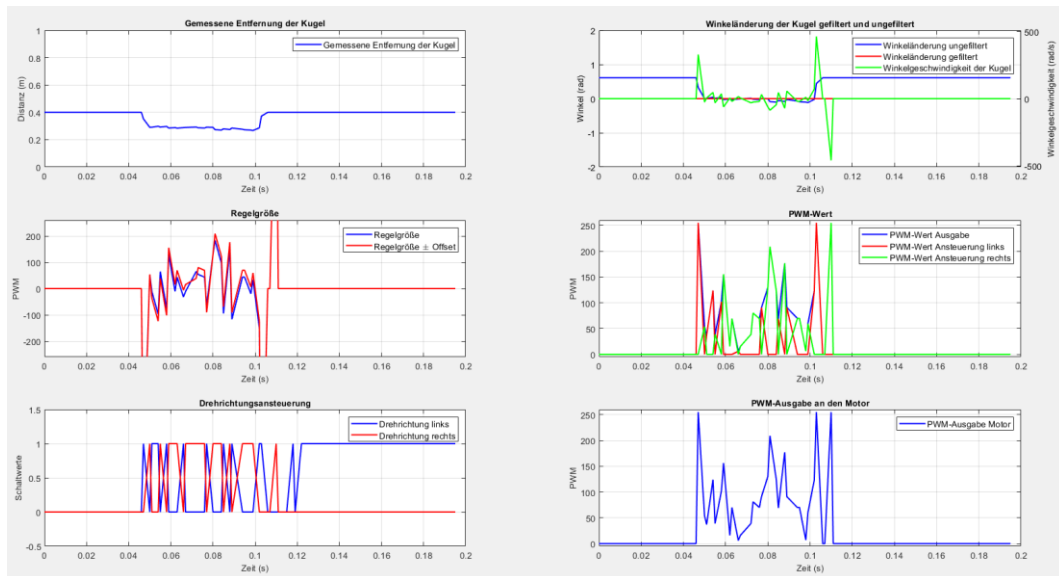


Abbildung 39: Systemtest 2 ohne Filterung mit  $D = 1$  und  $\omega_0 = 12$

Auch bei Testdurchläufen mit einer erhöhten Eigenkreisfrequenz von  $\omega_0 = 20$ , ohne Filterung, war es dem System nicht möglich, die Kugel zu balancieren, trotz schnellerer Reaktion auf Änderungen. Abbildung 40 zeigt die entsprechenden Messverläufe. Deutlich zu sehen ist auch, dass die Kugel nur wenige Zehntelsekunden auf dem Rad liegt, bevor sie herunterrollt. Die Regelgröße überschreitet mehrfach den Maximalwert von 255. Je höher die Eigenkreisfrequenz gewählt wird, desto ausschlaggebender sind die Ableitungen der Winkeländerungen. Während die Winkelgeschwindigkeit in Abbildung 38 nur wenig ins Gewicht fällt, zeigt sie in der unteren Abbildung 40 einen erheblichen Einfluss durch die starken Ausschläge.

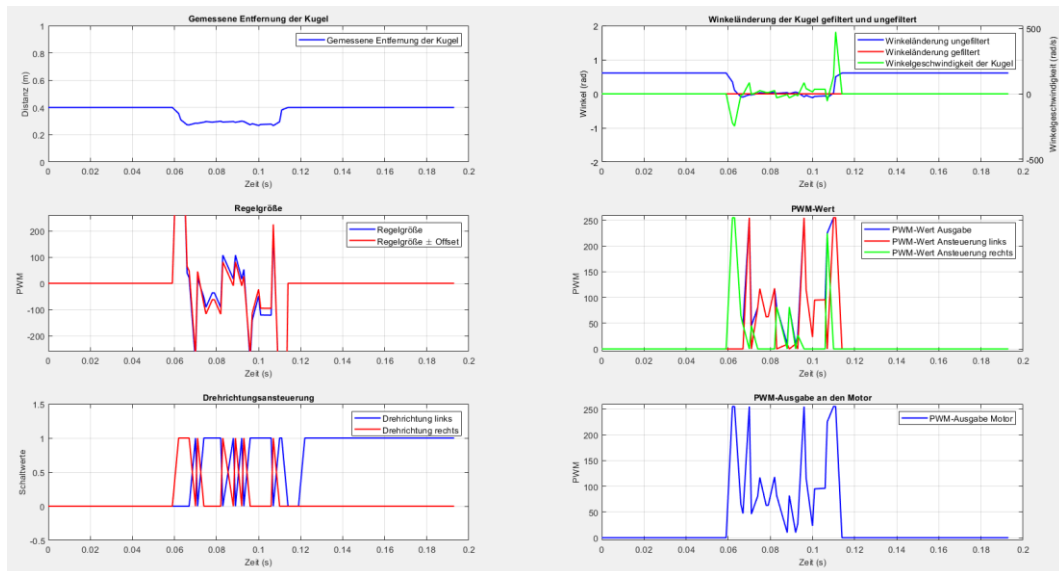


Abbildung 40: Systemtest 3 ohne Filterung mit  $D = 1$  und  $\omega_0 = 20$

Ebenfalls führte eine Variation der Dämpfung zu keiner Besserung des Systemverhaltens. Durch die höhere Gewichtung der Dämpfung werden größere Regelgrößen erzeugt. Die untere Abbildung 41 zeigt das Systemverhalten bei einer Dämpfung von  $D = 1,5$  und  $\omega_0 = 20$ , ohne Filterung.

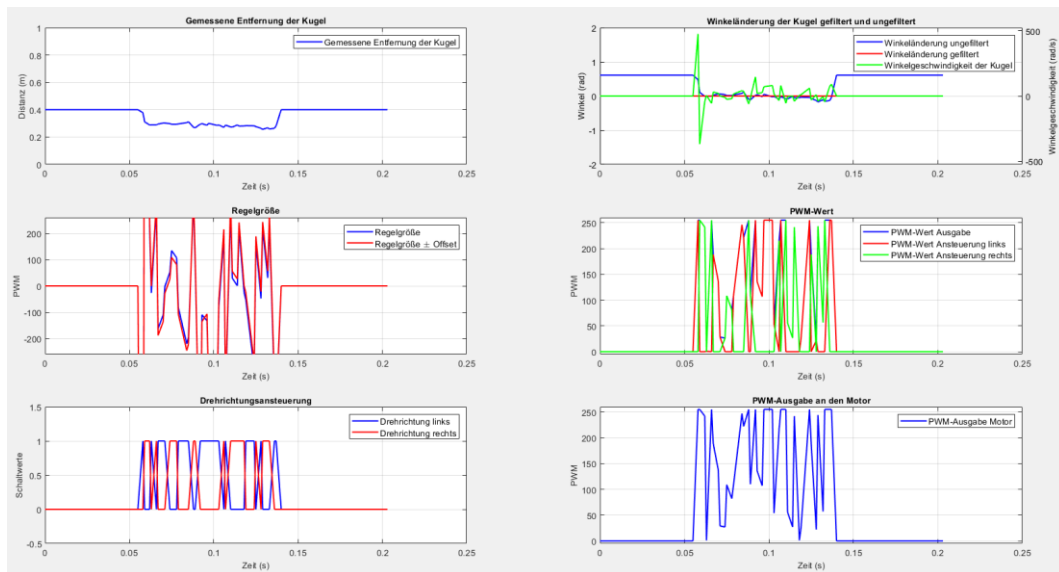


Abbildung 41: Systemtest 4 ohne Filterung mit  $D = 1,5$  und  $\omega_0 = 20$

Das System ist nicht in der Lage, die Kugel dauerhaft stabil auf dem Rad zu balancieren. Auch durch Variation der Regelparameter konnte kein zufriedenstellendes Verhalten erzielt werden. Infolgedessen waren auch weiterführende Tests, wie das Regeln der Kugel über einen Zeitraum von mehr als zwei Minuten, nicht erfolgreich durchführbar.



Das getestete Systemverhalten mit einer Holzkugel ist ähnlich zu dem der Gummikugel. Die Holzkugel konnte nicht balanciert und auf Stellgrößen konnte nicht reagiert werden, da die Kugel auf dem Kantenschutz eine zu geringe Haftung aufweist und somit von der Felge rutscht. Die erfassten Messergebnisse der Holzkugel sind im gleichen Format in der folgenden Abbildung 42 dargestellt.

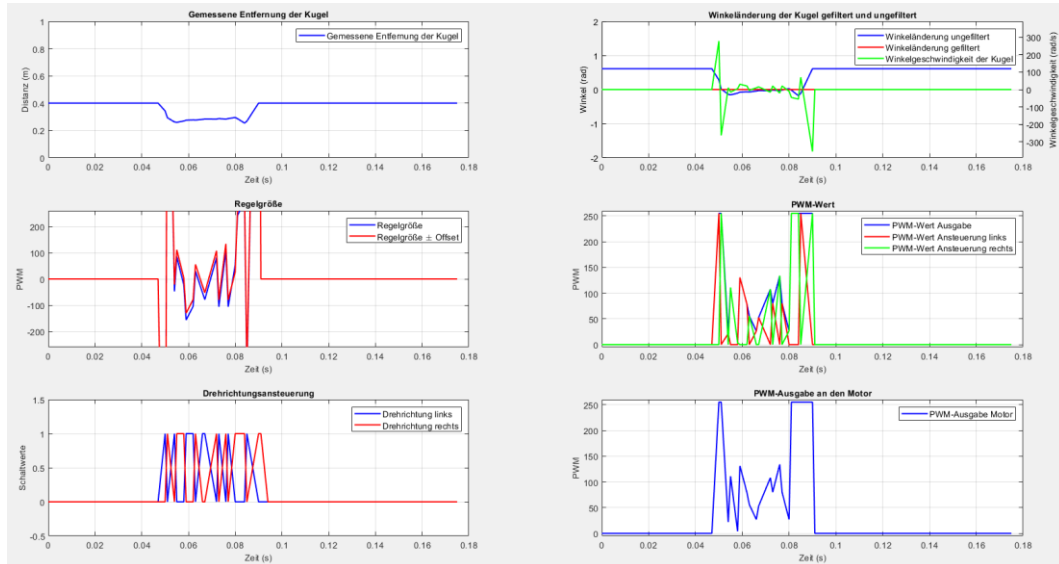


Abbildung 42: Systemtest 5 Holzkugel ohne Filterung mit  $D = 1,5$  und  $\omega_0 = 20$

Die Rückführung der Kugel aus den Auffangarmen auf das Rad konnte softwareseitig erfolgreich getestet werden. Die untere Abbildung 43 zeigt die erfassten Signale, sobald die Kugel im linken Auffangarm liegt. In dem oberen Plot sind die Signale für die Richtungsansteuerung dargestellt. Der untere Plot visualisiert die Signale *TST\_AnsteuerungSwitchBlock* und *TST\_Gegenimpuls*. Sobald der Taster gedrückt wurde und die Pause des Systems abgelaufen ist, springt das Signal *TST\_AnsteuerungSwitchBlock* auf 1, wodurch die Richtungsansteuerung ermöglicht wird. Wird der Gegenimpuls erzeugt, springen die Signale *TST\_Gegenimpuls* und *TST\_SchaltungGegenimpulsR* auf 1, wodurch der Gegenimpuls ausgelöst wird. Verlässt die Kugel den Auffangarm, fällt das Signal *TST\_SchaltungGegenimpulsR* wieder auf 0, da der Mikrotaster nicht mehr betätigt wird. Dies ist deutlich zu sehen, da dieses Signal signifikant kürzer ist als das des Gegenimpulses.



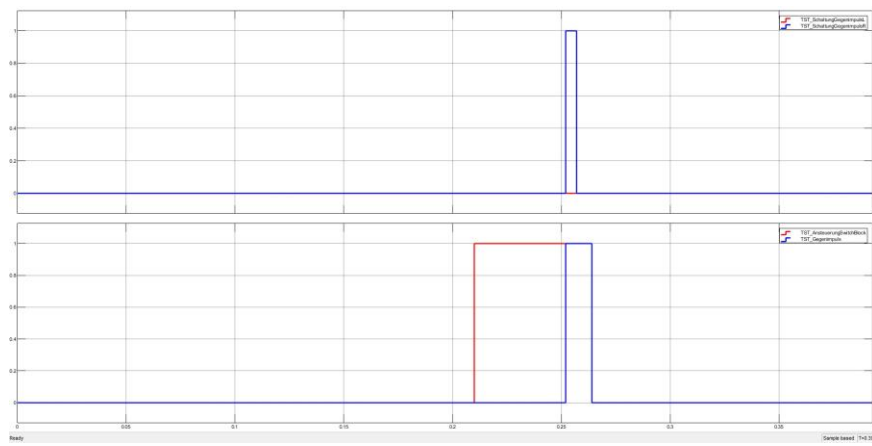


Abbildung 43: Systemtest 6 Betätigung des Mikrotasters im linken Auffangarm

Auch für den rechten Auffangarm konnte die Funktion der Software erfolgreich bestätigt werden, dargestellt in Abbildung 44.

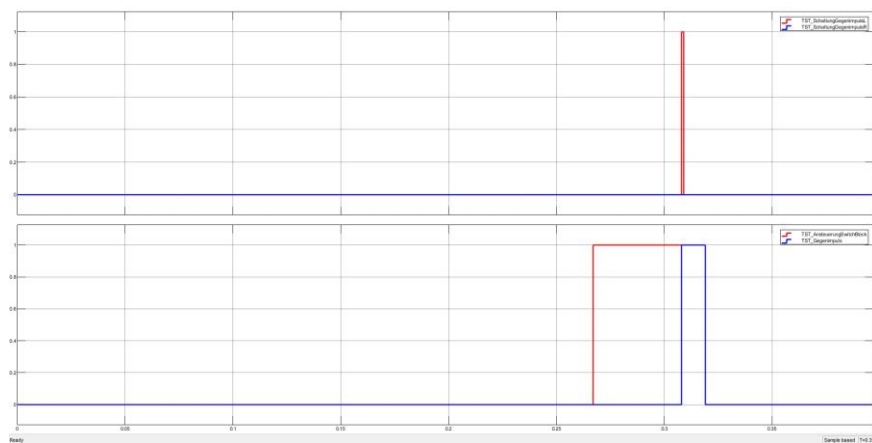


Abbildung 44: Systemtest 7 Betätigung des Mikrotasters im rechten Auffangarm

Probleme zeigten sich hinsichtlich der tatsächlichen Umsetzung: Zwar wird das Rad in die entsprechende Richtung beschleunigt und durch den Gegenimpuls gestoppt, jedoch ist das System nicht in der Lage, die Kugel zuverlässig auf die Sollposition zurückzubefördern.

Funktionstests, welche die Systemstabilität bei Unterbrechung und Wiederherstellung der Spannungsversorgung sowie das Verhalten bei Betätigung des Not-Aus betreffen, konnten hingegen erfolgreich absolviert werden. Die gewählten Systemparameter wurden nach dem Neustart korrekt übernommen und die Sicherheitsabschaltung funktionierte zuverlässig.

Insgesamt konnte durch die Tests festgestellt werden, dass diesem System keine Regelung der Kugel auf dem Rad möglich ist.

Sobald ein Gegenstand zu nah vor dem Sensor erfasst wird, ist es der Software nicht mehr möglich, einen Winkel zu berechnen, wodurch lediglich „NaN“ (Not a Number, z. Dt. Kein Wert) ausgegeben wird. Als Resultat muss das System neugestartet werden. Die folgende Abbildung 45 zeigt die Auswirkung auf die Komponente Signalverarbeitung. In den einzelnen Displays (rote Umrandungen) wird NaN angezeigt, wodurch es keine verarbeitbaren Daten gibt, die zu Regel- und Stellgrößen führen können. Zudem kam es zu einzelnen Aussetzern, bei denen der Sensor trotz unveränderter Hardware keine Messwerte lieferte. Zur Fehlerbehebung wurden sämtliche Kabel des Sensors auf Leitfähigkeit geprüft, andere Input-Ports am Arduino sowie ein anderer Arduino getestet. Die Ursache konnte dennoch nicht ausfindig gemacht werden.

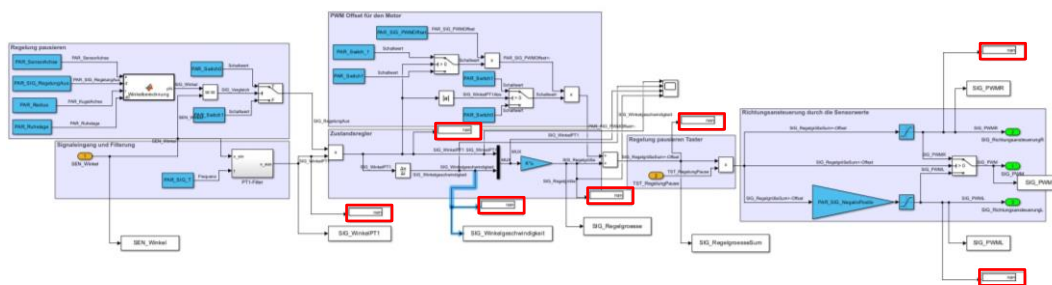


Abbildung 45: Sensorausgabe NaN

Die Grundfunktionen des Systems hingegen konnten erfolgreich getestet werden. Auch die weiteren Implementierungen sind vorhanden und funktionsfähig.

## 4.2 Vergleich von Simulation und realem System

In der Simulation zeigte das System ein stabiles Verhalten. Die Kugel verblieb in ihrer Ruhelage, bis gezielt eine äußere Störung eingebracht wurde. Die idealisierten Modellannahmen, perfekte Sensorik und verzögerungsfreie Reaktionen, führen zu dieser Stabilität. Durch die kontrollierte Umgebung war es möglich, unter gleichbleibenden Bedingungen Variationen von Dämpfungs- und Eigenkreisfrequenzwerten zu untersuchen und das damit verbundene Systemverhalten zu analysieren. Diese Rahmenbedingungen ermöglichen das System zu idealisieren und stabilisieren, was im realen Aufbau jedoch nicht möglich ist.

Im realen Aufbau verlässt die Kugel die Ruhelage bereits unmittelbar nach der Positionierung auf dem Rad. Dieses Verhalten zeigt, dass das physikalische

System eine sehr hohe Empfindlichkeit gegenüber kleinsten Störungen besitzt und entsprechend hohe Anforderungen an die Präzision und Reaktionsfähigkeit der Regelung stellt. Eine Anforderung, die unter den gegebenen Bedingungen nicht ausreichend erfüllt werden konnte. Da die Kugel bereits nach wenigen Zehntelsekunden ihre Sollposition verlässt, konnten lediglich kleine Datenfenster für die Abstimmung genutzt werden. Eine präzise Optimierung des Reglers war somit nicht möglich. Die Kugel wurde bei der Verwendung verschiedener Dämpfungs- und Eigenkreisfrequenzwerte in allen Testfällen weniger als 0,15 s auf dem Rad gehalten.

### **4.3 Ergebnisanalyse**

Der vorangegangene Systemtest hat gezeigt, dass das entwickelte Gesamtsystem, bestehend aus Sensorik, Mechanik, Software und Regelung, derzeit nicht vollfunktionsfähig ist. Das Gesamtergebnis ist ein instabiles und unruhiges System, das nicht in der Lage ist, die Kugel zuverlässig zu balancieren. Im Folgenden wird analysiert, welche konkreten Ursachen zu diesem Ergebnis geführt haben.

Ein wesentlicher Schwachpunkt liegt in der verwendeten Sensorik. Aufgrund des 25 ° breiten Sichtfelds kann der Sensor nicht präzise und punktuell auf die Kugel ausgerichtet werden. Für das umgesetzte Projekt ist dieses breite Sichtfeld ein Nachteil. Zwar lässt sich die Sichtweite softwareseitig begrenzen, der große Erfassungswinkel bleibt jedoch unverändert und beeinträchtigt die Messgenauigkeit. Zusätzlich werden durch die schwankenden Sensorwerte kontinuierlich Regelgrößen ermittelt, welche die Kugelposition korrigieren sollen, auch wenn die Kugel auf der Sollposition liegt. Die Folge dabei ist, dass die Kugel „heruntergeregelt“ wird, selbst, wenn diese stabil ist. Besonders kritisch ist hierbei, dass bereits kleine Positionsabweichungen durch die Ableitung zur Bestimmung der Winkelgeschwindigkeit erheblich verstärkt werden. Ein Filtern der Sensorsignale ist trotz dieser Probleme nicht möglich, da es zu einer zu starken Verzögerung und Trägheit des Systems führen würde, sodass das System nicht mehr reaktionsfähig wäre.

Der eingesetzte Zustandsregler zeigt in der Simulation ein vielversprechendes Regelverhalten mit stabiler Kugelposition und gezielter Reaktion auf äußere Störungen. Im realen Aufbau ist die Regelgüte des Reglers durch die

unzureichende Sensorqualität, die starke Störanfälligkeit und die hohe Empfindlichkeit des physikalischen Systems nicht ausgeschöpft. Insbesondere die starken Schwankungen der Messsignale und deren direkte Ableitung zur Ermittlung der Winkelgeschwindigkeit führen zu übersteuerten Stellgrößen, was die Regelung zusätzlich destabilisiert. Auch wenn der Zustandsregler theoretisch eine gute Grundlage zur Kompensation von Positionsabweichungen bietet, führt die fehleranfällige Eingangsbasis zu instabilen Regelgrößen und letztlich zu einer unzureichenden Regelleistung. Eine Verbesserung der Sensorik sowie eine robuste Signalverarbeitung wären notwendige Voraussetzungen, um die Wirksamkeit des Zustandsreglers im realen System sinnvoll nutzen zu können.

Ebenfalls weisen die weiteren Hardware-Komponenten Qualitätsmängel auf. Das Rad verfügt über Unwuchten. Trotz des manuellen Auswuchtens konnten diese nicht komplett beseitigt werden. Durch die Verklebung des Kantenschutzes gibt es vereinzelte Erhebungen, welche die Kugel bei hohen Drehzahlen springen lässt. Diese Unwuchten führen zur zusätzlichen Destabilisierung der Kugel. Ein weiteres Problem ergibt sich bei den eingesetzten Mikrotastern mit Rollenhebel. Liegt die Kugel auf dem Hebel auf und wird beschleunigt, erfolgt eine seitliche Auslenkung des Hebels. Dadurch driftet die Kugel von ihrer zentrierten Position ab.

Trotz dieser Probleme kann festgehalten werden, dass die entwickelte Software grundsätzlich funktionsfähig ist. Alle vorgesehenen Komponenten und Funktionen wurden erfolgreich umgesetzt.

## **5 Fazit und Ausblick**

Als Abschluss dieser Arbeit erfolgt die Zusammenfassung der Ergebnisse mit einer kritischen Betrachtung und Analyse potenzieller Verbesserungen für eine Fortführung dieses Projekts.

### **5.1 Zusammenfassung der Ergebnisse**

Ziel dieser Masterarbeit war es, das Projekt *Kugel auf Rad* erfolgreich umzusetzen. Die Entwicklung des Systems brachte eine Reihe technischer Herausforderungen mit sich, insbesondere im Bereich der Hardware, welche mehrfach überarbeitet und angepasst werden musste.

Ein wichtiger Bestandteil war die mathematische Beschreibung, um das Verhalten des Systems analysieren und verstehen zu können. Ergänzend dazu wurde durch eine Simulation das Systemverhalten visualisiert. Dabei wurde die Systemdynamik vereinfacht, indem ausschließlich die Zustände der Kugel betrachtet wurden, während die Dynamik des Rads unberücksichtigt blieb. Dennoch konnten grundlegende Eigenschaften, wie die Instabilität des Systems bestätigt werden. Zusätzlich war es möglich, erste Dämpfungswerte sowie die Eigenkreisfrequenz auszulegen, um ein schnelles und schwingungsarmes Regelverhalten zu erzielen. Mit diesen Werten wurden anschließend Tests am realen System durchgeführt.

Für die Entwicklung des realen Systems wurde anhand des V-Modells vorgegangen: Planung, Umsetzung und Tests. Somit konnte während der Entwicklung der Software- als auch Hardware-Komponenten Fehler direkt ausfindig gemacht und behoben werden. Für die Realisierung war neben der Modellierung von Bauteilen mittels SolidWorks auch die praktische Umsetzung gefragt, um den Aluminiumrahmen sowie die Scheibe zu fertigen.

Das Gesamtsystem ist durch den Aufbau kompakt und leicht transportierbar. Alle Komponenten sind mechanisch stabil befestigt und ausgerichtet.

Dennoch konnte das System bei der Zusammenführung von Software und Hardware nicht das gewünschte Ergebnis erzielen. Die grundlegende Funktion ist trotzdem gegeben und bildet eine solide Basis für zukünftige Weiterentwicklungen.

Wie in Abbildung 37 bis Abbildung 42 dargestellt, ist für eine präzise Umsetzung die Messeinrichtung ein ausschlaggebender Punkt. Der verwendete Sensor konnte zwar bei den durchgeführten Tests in Kapitel 3.2.3 mit seiner Genauigkeit und Zuverlässigkeit überzeugen, im realen System ist diese jedoch aufgrund der hochgradigen Instabilität nicht mehr ausreichend genug.

Bei der Zusammenführung von Software und Hardware traten aufgrund der Qualitätseigenschaften der elektronischen Komponenten Implementierungsprobleme auf. Diese Komplikationen verdeutlichen die Bedeutung instabiler Regelstrecken, sensorischer Einschränkungen und mechanischer Einflussgrößen.

Ein früherer Entwicklungsstand des Projekts war in der Lage, die (Holz-)Kugel erfolgreich zu stabilisieren. Durch notwendige Änderungen an Hard- und Software ging diese Funktionalität in der aktuellen Version jedoch verloren. Die entwickelte Software selbst konnte in allen durchgeführten Tests überzeugen und arbeitet zuverlässig, die Umsetzung ist in diesem Teilbereich daher als erfolgreich zu bewerten.

## **5.2 Potenzielle Verbesserungen**

Im Verlauf des Projekts wurde deutlich, dass insbesondere die reale Umsetzung des Systems vor erheblichen Herausforderungen steht. Trotz eines funktionierenden Simulationsmodells und der Implementierung eines Zustandsreglers.

Um eine stabile Regelung des hochgradig instabilen Systems auch in der Praxis zu ermöglichen, ergeben sich folgende Optimierungspotenziale:

- Sensorik: Einsatz eines hochwertigen, gezielt einstellbaren Sensors, der die Kugel punktuell erfasst.
- Filterung: Einsatz eines echtzeitfähigen Filters, der Störungen zuverlässig unterdrückt, ohne das dynamische Verhalten des Systems merklich zu beeinflussen.
- Mechanik: Verwendung eines präzise ausgewuchteten und zentrierten Rads, um Unwuchten und störende Vibrationen zu minimieren.
- Modell: Erweiterung des mathematischen Modells durch die vollständige Berücksichtigung aller vier Zustände des Systems (Kugel- und Radzustände), um die reale Dynamik genauer abzubilden.
- Datenerfassung: Ergänzung der Sensorik durch einen Drehzahlsensor für das Rad, um zusätzlich zur Kugelbewegung auch die Drehdynamik in die Regelung einbeziehen zu können.
- Lageerfassung: Verwendung von Lichtschranken oder stabileren Mikrotastern mit Rollenhebel.

Mit diesen Optimierungen erscheint die Hardwarebasis deutlich vielversprechender, sodass eine erfolgreiche Umsetzung in zukünftigen Projekten möglich wird.

## Literaturverzeichnis

- [1] AMAZON, DOPPEL H-BRÜCKE TECHNISCHE DATEN. VERFÜGBAR UNTER: [HTTPS://AMZN.EU/D/HAHTOSA](https://amzn.eu/d/HAHTOSA). [ZUGEGRIFFEN: 9. JANUAR 2025]
- [2] ARDUINO, ARDUINO MEGA 2560 REV3 DATENBLATT, 2025. VERFÜGBAR UNTER: [HTTPS://DOCS.ARDUINO.CC/RESOURCES/DATASHEETS/A000067-DATASHEET.PDF?\\_GL=1\\*1NQINL6\\*\\_UP\\*MQ..\\*\\_GA\\*MTA4MJE5NDAuMTc0NzQ3ODQwMw..\\*\\_GA\\_NEXN8H46L5\\*cze3NDc0NzG0MDEkbzEkZzAKdDE3NDc0NzG0MDEKAJAKBDakADIWNjMxOTQ4MjQ](https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf?_gl=1*1nqinl6*_up*MQ..*_ga*MTA4MJE5NDAuMTc0NzQ3ODQwMw..*_ga_NEXN8H46L5*cze3NDc0NzG0MDEkbzEkZzAKdDE3NDc0NzG0MDEKAJAKBDakADIWNjMxOTQ4MjQ). [ZUGEGRIFFEN: 6. JANUAR 2025]
- [3] CONRAD ELECTRONIC, M5STACK U010 ENTFERNUNGSMESSER 1 ST. PASSEND FÜR (ENTWICKLUNGSKITS): ARDUINO. VERFÜGBAR UNTER: [HTTPS://WWW.CONRAD.DE/DE/P/M5STACK-U010-ENTFERNUNGSMESSER-1-ST-PASSEND-FUER-ENTWICKLUNGSKITS-ARDUINO-2306335.HTML#PRODUCTDOWNLOADS](https://www.conrad.de/de/p/m5stack-u010-entfernungsmesser-1-st-passend-fuer-entwicklungskits-arduino-2306335.html#productdownloads). [ZUGEGRIFFEN: 10. MÄRZ 2025]
- [4] CONRAD ELECTRONIC/M5STACK, TOF UNIT (VL53L0X) – ENTFERNUNGSMESSER, DATENBLATT. M5STACK, 2020, VERFÜGBAR UNTER: [HTTPS://ASSET.CONRAD.COM/MEDIA10/ADD/160267/C1/-/EN/002306335DS00/DATENBLATT-2306335-M5STACK-U010-ENTFERNUNGSMESSER-1-ST-PASSEND-FUER-ENTWICKLUNGSKITS-ARDUINO.PDF](https://asset.conrad.com/media10/add/160267/c1/-/en/002306335DS00/datenblatt-2306335-m5stack-u010-entfernungsmesser-1-st-passend-fuer-entwicklungskits-arduino.pdf). [ZUGEGRIFFEN: 2. FEBRUAR 2025] ZUSÄTZLICH AUFGEFÜHRT IM ANHANG K.
- [5] E. SCHRÖGER UND G. HARTWIGSEN, *BIOLOGISCHE PSYCHOLOGIE: EIN ÜBERBLICK FÜR PSYCHOLOGIESTUDIERENDE UND -INTERESSIERTE*. SPRINGER, 2024.
- [6] F. ENGELMANN, *MASCHINENELEMENTE KOMPAKT: AUSWAHL, GESTALTUNG UND DIMENSIONIERUNG IN THEORIE UND PRAXIS*. SPRINGER-VERLAG, 2019.
- [7] F. HÜNING, *EMBEDDED SYSTEMS FÜR IOT*. SPRINGER-VERLAG, 2018.
- [8] F. MAHR, S. HENNINGER, M. BILLER UND J. JÄGER, *ELEKTRISCHE ENERGIESYSTEME: WISSENSVERNETZUNG VON STROMRICHTER, NETZBETRIEB UND NETZSCHUTZ*. SPRINGER VIEWEG, 2021.
- [9] F. VOLKEN, F. ZURBRIGGEN, D. AMBÜHL UND S. WIESER, „PRAKTIKUM MESS- UND REGELTECHNIK ANLEITUNG ZUM VERSUCH BALL ON WHEEL“, OKT. 2015. VERFÜGBAR UNTER: [HTTPS://ETHZ.CH/CONTENT/DAM/ETHZ/SPECIAL-INTEREST/MAVT/DYNAMIC-SYSTEMS-N-CONTROL/IDSC-DAM/LECTURES/CONTROL-LAB/BALLONWHEELSTUDENT.PDF](https://ethz.ch/content/dam/ethz/special-interest/mavt/dynamic-systems-n-control/idsc-dam/lectures/control-lab/ballonwheelstudent.pdf). [ZUGEGRIFFEN: 3. JANUAR 2025]
- [10] H. WALTER, *ZUSTANDSREGELUNG: ANALYSE UND SYNTHESE VON ZUSTANDSREGELUNGEN EINSCHLIEßLICH REGLEROPTIMIERUNG*. SPRINGER VIEWEG, 2019.

- [11] H. WINNER, S. HAKULI UND G. WOLF, *HANDBUCH FAHRERASSISTENZSYSTEME: GRUNDLAGEN, KOMPONENTEN UND SYSTEME FÜR AKTIVE SICHERHEIT UND KOMFORT*. SPRINGER-VERLAG, 2009.
- [12] HANDSON TECHNOLOGY, 775 BALL BEARING DC MOTOR, VERFÜGBAR UNTER: [HTTP://HANDSONTEC.COM/INDEX.PHP/PRODUCT/775-BALL-BEARING-DC-MOTOR/?CURRENCY=MYR](http://HANDSONTEC.COM/INDEX.PHP/PRODUCT/775-BALL-BEARING-DC-MOTOR/?CURRENCY=MYR). [ZUGEGRIFFEN: 12. JANUAR 2025]. ZUSÄTZLICH AUFGEFÜHRT IM ANHANG L.
- [13] HESCHEN, RED SIGN EMERGENCY STOP, VERFÜGBAR UNTER: [HTTPS://HESCHEN.COM/DE/PRODUCTS/HESCHEN-EMERGENCY-STOP-SWITCH-PUSH-BUTTON-SWITCH](https://HESCHEN.COM/DE/PRODUCTS/HESCHEN-EMERGENCY-STOP-SWITCH-PUSH-BUTTON-SWITCH). [ZUGEGRIFFEN: 2. FEBRUAR 2025]. ZUSÄTZLICH AUFGEFÜHRT IM ANHANG M.
- [14] ITEM INDUSTRIETECHNIK GMBH, ITEM ENGINEERINGTOOL, VERFÜGBAR UNTER: [HTTPS://TOOLS.ITEM24.COM/DEDE/TOOLS/ENGINEERINGTOOL/](https://TOOLS.ITEM24.COM/DEDE/TOOLS/ENGINEERINGTOOL/). [ZUGEGRIFFEN: 16. APRIL 2025]
- [15] J. A. DE CARVALHO GONÇALVES, J. L. S. DE MAGALHÃES LIMA, J. P. COELHO, F. J. GARCÍA-PEÑALVO UND A. GARCÍA-HOLGADO, *PROCEEDINGS OF TEEM 2023: THE ELEVENTH INTERNATIONAL CONFERENCE ON TECHNOLOGICAL ECOSYSTEMS FOR ENHANCING MULTICULTURALITY*, 1. AUFL. SPRINGER NATURE, 2024. DOI: 10.1007/978-981-97-1814-6
- [16] J. HÄRTERICH UND A. ROOCH, *DAS MATHE-PRAXIS-BUCH: WIE INGENIEURE MATHEMATIK ANWENDEN - PROJEKTE FÜR DIE BACHELOR-PHASE*. SPRINGER-VERLAG, 2013.
- [17] L. PAVLAKOS UND L. PAVLAKOS, „ARTICLE → WHAT IS A TIME OF FLIGHT SENSOR (ToF)?“, *OMNIROBOTIC - ADVANCED ROBOTIC MACHINES THAT ANYONE CAN USE*, 20. SEPTEMBER 2024. VERFÜGBAR UNTER: [HTTPS://OMNIROBOTIC.COM/NEWS\\_AND\\_EVENTS/ARTICLES/WHAT-IS-A-TIME-OF-FLIGHT-SENSOR/](https://OMNIROBOTIC.COM/NEWS_AND_EVENTS/ARTICLES/WHAT-IS-A-TIME-OF-FLIGHT-SENSOR/). [ZUGEGRIFFEN: 11. DEZEMBER 2024]
- [18] PROF. DR.-ING. MIREK GÖBEL. MESS- UND REGELUNGSTECHNIK. VORLESUNG. VORLESUNGSSKRIPT. LIPPSTADT, HOCHSCHULE HAMM-LIPPSTADT, 2019.
- [19] PROF. DR.-ING. MIREK GÖBEL. FORMELSAMMLUNG MESS- UND REGELUNGSTECHNIK. HOCHSCHULE HAMM-LIPPSTADT, JUNI 2023. VERFÜGBAR UNTER: [HTTPS://MYMOODLE.HSHL.DE/PLUGINFILE.PHP/1940325/MOD\\_RESOURCE/CONTENT/0/MESS\\_U\\_REGELUNGSTECHNIK\\_FORMELSAMMLUNG.PDF](https://MYMOODLE.HSHL.DE/PLUGINFILE.PHP/1940325/MOD_RESOURCE/CONTENT/0/MESS_U_REGELUNGSTECHNIK_FORMELSAMMLUNG.PDF). [ZUGEGRIFFEN: 05. MAI 2025]
- [20] R. MARCHTHALER UND S. DINGLER, *KALMAN-FILTER: EINFÜHRUNG IN DIE ZUSTANDSSCHÄTZUNG UND IHRE ANWENDUNG FÜR EINGEBETTETE SYSTEME*. SPRINGER-VERLAG, 2024.
- [21] S.-O. LINDERT, C. HÖFLER UND K. SCHLACHER, „NICHTLINEARE REGELUNG MIT EINEM RASPBERRY PI“, *E+I ELEKTROTECHNIK UND INFORMATIONSTECHNIK*, BD. 135, NR. 3, S.



286–293, APR. 2018, DOI: 10.1007/s00502-018-0613-8. VERFÜGBAR UNTER: [HTTPS://DOI.ORG/10.1007/s00502-018-0613-8](https://doi.org/10.1007/s00502-018-0613-8)

[22] STMICROELECTRONICS, VL53L0X – TIME-OF-FLIGHT RANGING SENSOR DATASHEET. STMICROELECTRONICS, MAI 2016. VERFÜGBAR UNTER: [HTTPS://M5STACK.OSS-CN-SHENZHEN.ALIYUNCS.COM/RESOURCE/DOCS/DATASHEET/HAT/VL53L0X\\_EN.PDF](https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/docs/datasheet/hat/vl53l0x_en.pdf).

[ZUGEGRIFFEN: 2. FEBRUAR 2025]. ZUSÄTZLICH AUFGEFÜHRT SIND DIE RELEVANTEN SEITEN 28 UND 30 IM ANHANG N.

[23] UNIVERSITÄT STUTTGART, INSTITUT FÜR TECHNISCHE UND NUMERISCHE MECHANIK PROFS. EBERHARD/FEHR/HANSS UND INSTITUT FÜR TECHNISCHE UND NUMERISCHE MECHANIK, „PRAKTIKUMSVERSUCH BALANCIERTER BALL AUF FELGE“. VERFÜGBAR UNTER: [HTTPS://WWW.ITM.UNI-STUTTGART.DE/LEHRE/PRAKTIKUM-TECHNISCHE-DYNAMIK/PDFFILES/POSTER/PRAKTIKUMSVERSUCH\\_BALLAUFSELGE.PDF](https://www.itm.uni-stuttgart.de/lehre/praktikum-technische-dynamik/pdf/files/poster/praktikumsversuch_ballauffelge.pdf). [ZUGEGRIFFEN: 27. NOVEMBER 2024]

[24] W. ZEUGE, *NÜTZLICHE UND SCHÖNE GEOMETRIE: EINE ETWAS ANDERE EINFÜHRUNG IN DIE EUKLIDISCHE GEOMETRIE*. SPRINGER SPEKTRUM, 2021.

[25] WIKIPEDIA-AUTOREN, „INVERSES PENDEL“, 1. MÄRZ 2010. VERFÜGBAR UNTER: [HTTPS://DE.WIKIPEDIA.ORG/WIKI/INVERSES\\_PENDEL#/MEDIA/DATEI:CART-PENDULUM.SVG](https://de.wikipedia.org/wiki/Inverses_Pendel#/Media/Datei:Cart-pendulum.svg).

[ZUGEGRIFFEN: 27. NOVEMBER 2024]

[26] Z. LI, C. ZHANG, G. HUANG UND H. MA, „DESIGN OF CABLE PIPELINE FAULT LOCATION DEVICE BASED ON DYNAMIC LASER SCANNING“, *JOURNAL OF PHYSICS CONFERENCE SERIES*, BD. 2366, NR. 1, S. 012008, NOV. 2022, DOI: 10.1088/1742-6596/2366/1/012008. VERFÜGBAR UNTER: [HTTPS://DOI.ORG/10.1088/1742-6596/2366/1/012008](https://doi.org/10.1088/1742-6596/2366/1/012008).

## Anhang A SVN-Ordner Link

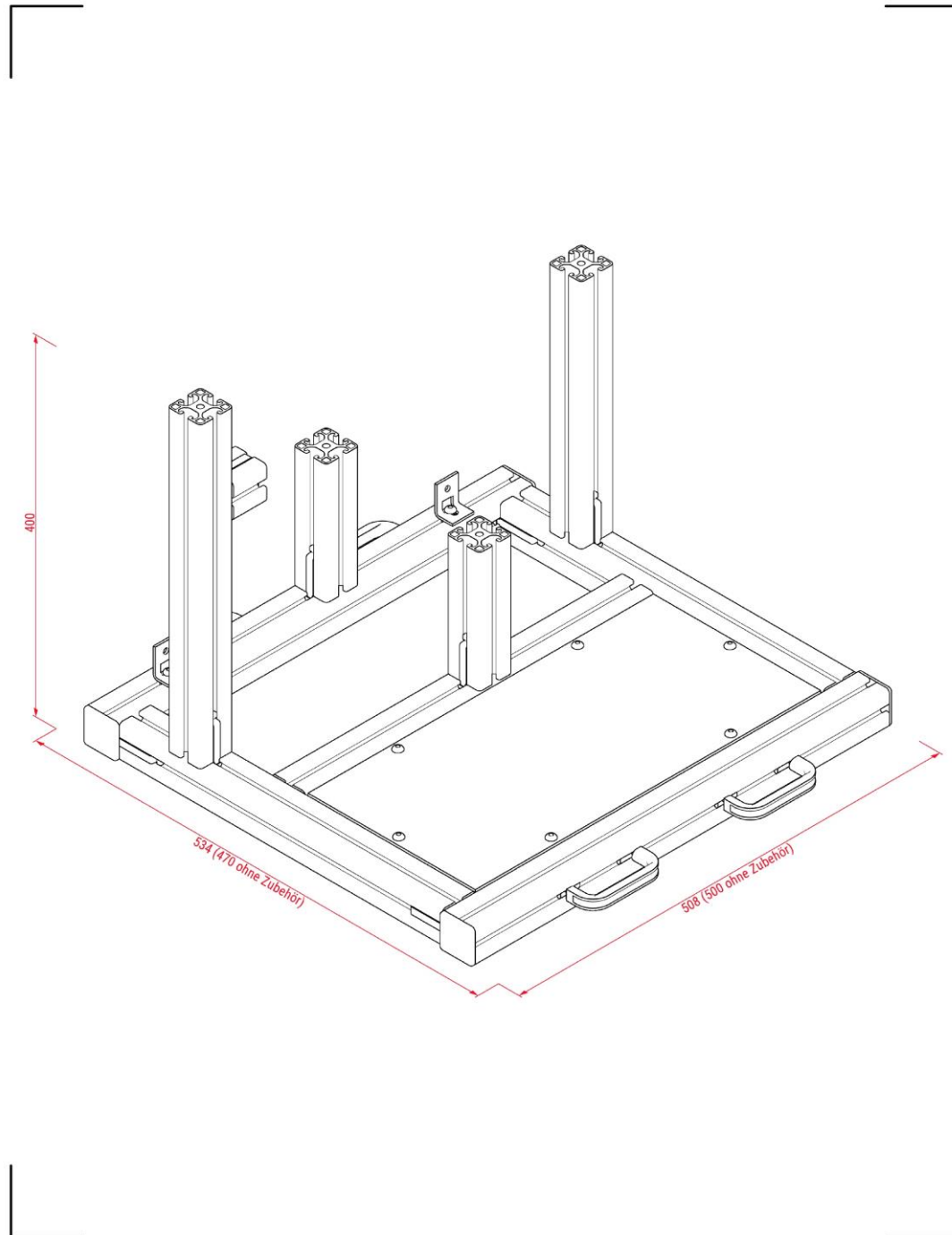
[https://svn.hshl.de/svn/HSHL\\_Projekte/trunk/Regelung\\_KugelAufEinemRad](https://svn.hshl.de/svn/HSHL_Projekte/trunk/Regelung_KugelAufEinemRad)

## Anhang B Aluminiumrahmen

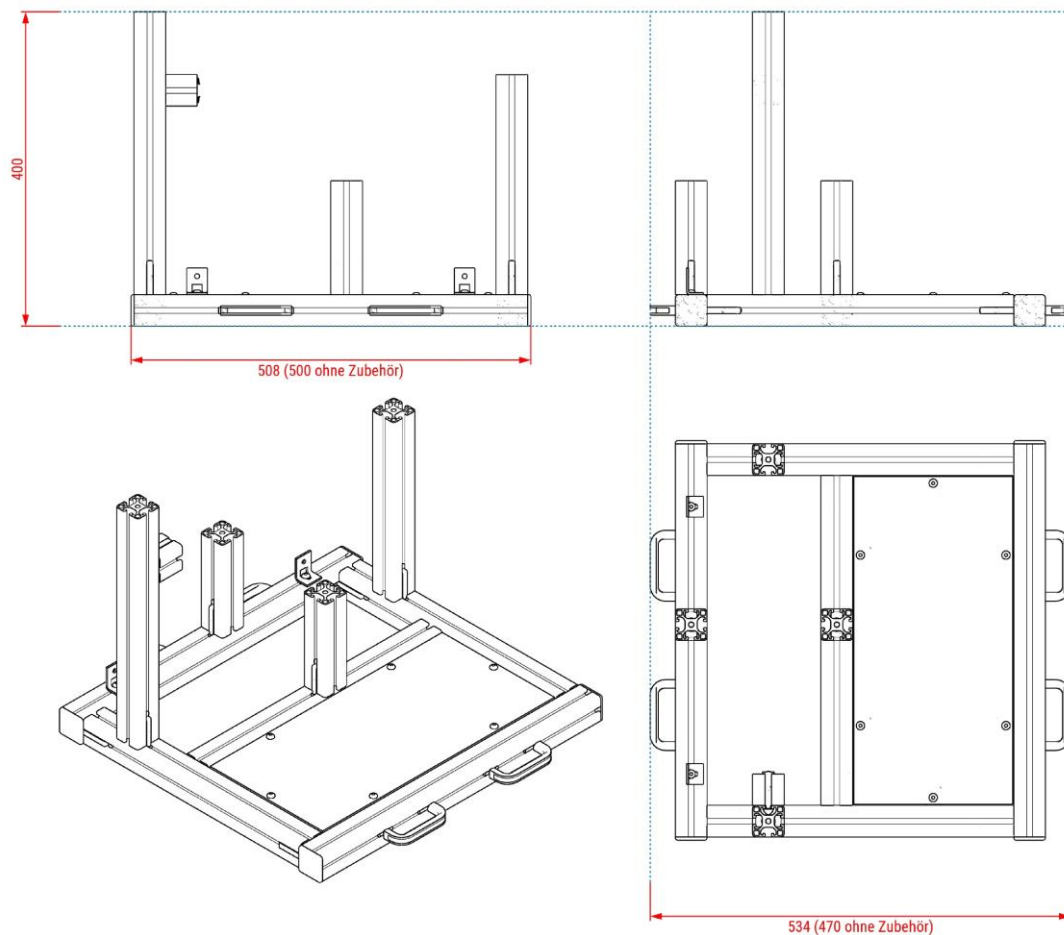
**item**

item. Ihre Ideen sind es wert.®

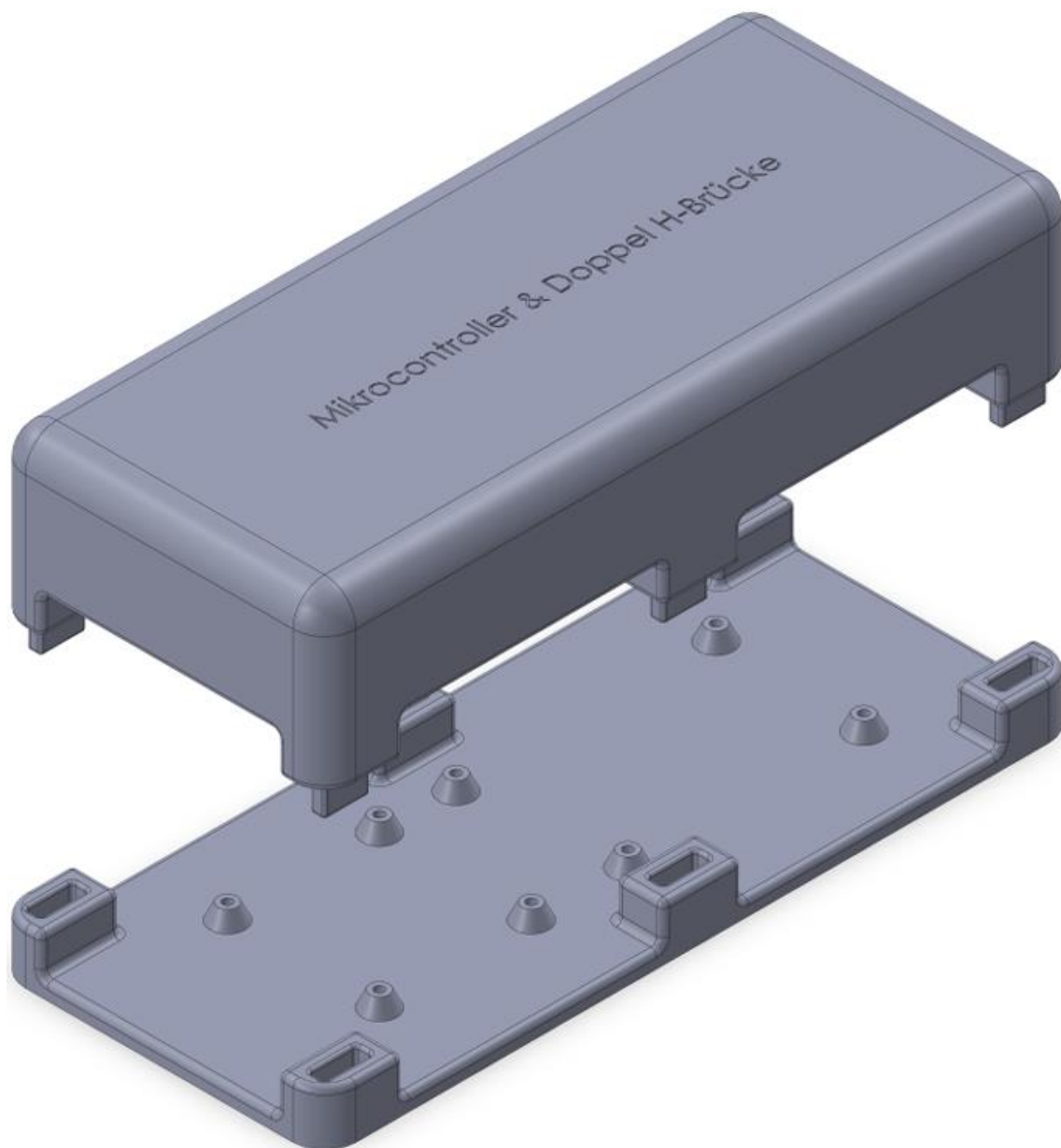
Isometrische Ansicht Baugruppe 1 / 1



# Dreitafelprojektion Baugruppe 1 / 1



## Anhang C Komponentenhaltung



## Anhang D Parameterdatei

```
%*****
%                               Hochschule Hamm-Lippstadt
%*****
% Modul      : PAR_KugelaufRad.m
%
% Datum      : 17.05.2025
%
% Funktion    : Parameterdatei für das Modul Kugel auf Rad
%
% Implementation : MATLAB 2023b
%
% Author      : Ken Hilz
%
% Letzte Änderung : 17-05-2025
%*****

%% Parameter
% Sample Time
ST = 0.001;                % Abtastrate min. 10x schneller als das System

% Switch
PAR_Switch0 = 0;           % Schaltwert für die Switch-Blöcke
PAR_Switch1 = 1;           % Schaltwert für die Switch-Blöcke
PAR_Switch_1 = -1;         % Schaltwert für die Switch-Blöcke

%Kosinussatz
PAR_SensorAchse = 0.32;    % in Metern
PAR_Radius = 0.2;          % in Metern
PAR_Ruhelage = 0.29;      % in Metern

%% Sensorik
% Umrechnung
PAR_SEN_Umrechnung = 1000; % Umrechnung Millimeter in Meter

% Messbereich Begrenzung
PAR_SEN_STUP = 400;        % in Millimetern
PAR_SEN_STDOWN = 0;        % in Millimetern

%% Signalverarbeitung
% Regelung deaktivieren
PAR_SIG_RegelungAus = 0.4; % in Metern

% PWM Offset
PAR_SIG_PWMOffset = 26;    % Offset, welches der Motor benötigt um zu drehen

% PWM Begrenzung
PAR_SIG_PWMUP = 255;       % Stellgrößenbegrenzung
PAR_SIG_PWMDOWN = 0;       % Stellgrößenbegrenzung
PAR_SIG_NegativPositiv = -1; % Multiplikation der negativen Stellgröße

% Filter Frequenz
PAR_SIG_T = 15;           % in Hz
```

```

%% Taster
% PWM
PAR_TST_PWM = 170;           % Definierter Wert zum Beschleunigung des Motors bei Tast
PAR_TST_PWMGI = 255;        % Definierter Wert zum Abbremsen des Motors durch den Ge

%% Berechnung Rückführvektor
% Parameter
g = 9.81;                    % m/s^2

% Rad
R = 0.16;                    % Radius Rad in Metern
RS = 0.2;                    % Abstand Radachse bis Kugelmittelpunkt
m_r = 0.4;                   % Masse Rad in Kilogramm
r_1 = 0.3;                   % Innerer Radius in Metern
r_2 = 0.32;                  % Äußerer Radius in Metern
J_r = 0.5 * m_r * (r_1^2 + r_2^2); % Massenträgheitsmoment Rad

% Kugel
r = 0.025;                   % Radius Kugel in Metern
rS = 0.022;                  % Abstand Mittelpunkt Kugel bis Auflagefläche
m_k = 0.08;                  % Masse Kugel in Kilogramm
J_k = m_k * (r^2);           % Massenträgheitsmoment Kugel

% Gesamtmassenträgheitsmoment
L = J_k * J_r + m_k * (R^2 * J_k + rS^2 * J_r); % Gesamtmassenträgheitsmoment

% Dämpfung und Eigenkreisfrequenz
D = 1;                       % Dämpfungswert
w0 = 12;                     % Eigenkreisfrequenz
K1 = ((-w0^2 * RS * L) - (m_k * g * (R^2 * J_k + rS^2 * J_r))) / (R * J_k); % Berechnung K1
K2 = -((2 * D * w0 * RS * L) / (R * J_k)); % Berechnung K2

K = [K1 K2];                 % Rückführvektor K

```

## Anhang E Winkelberechnung MATLAB Function

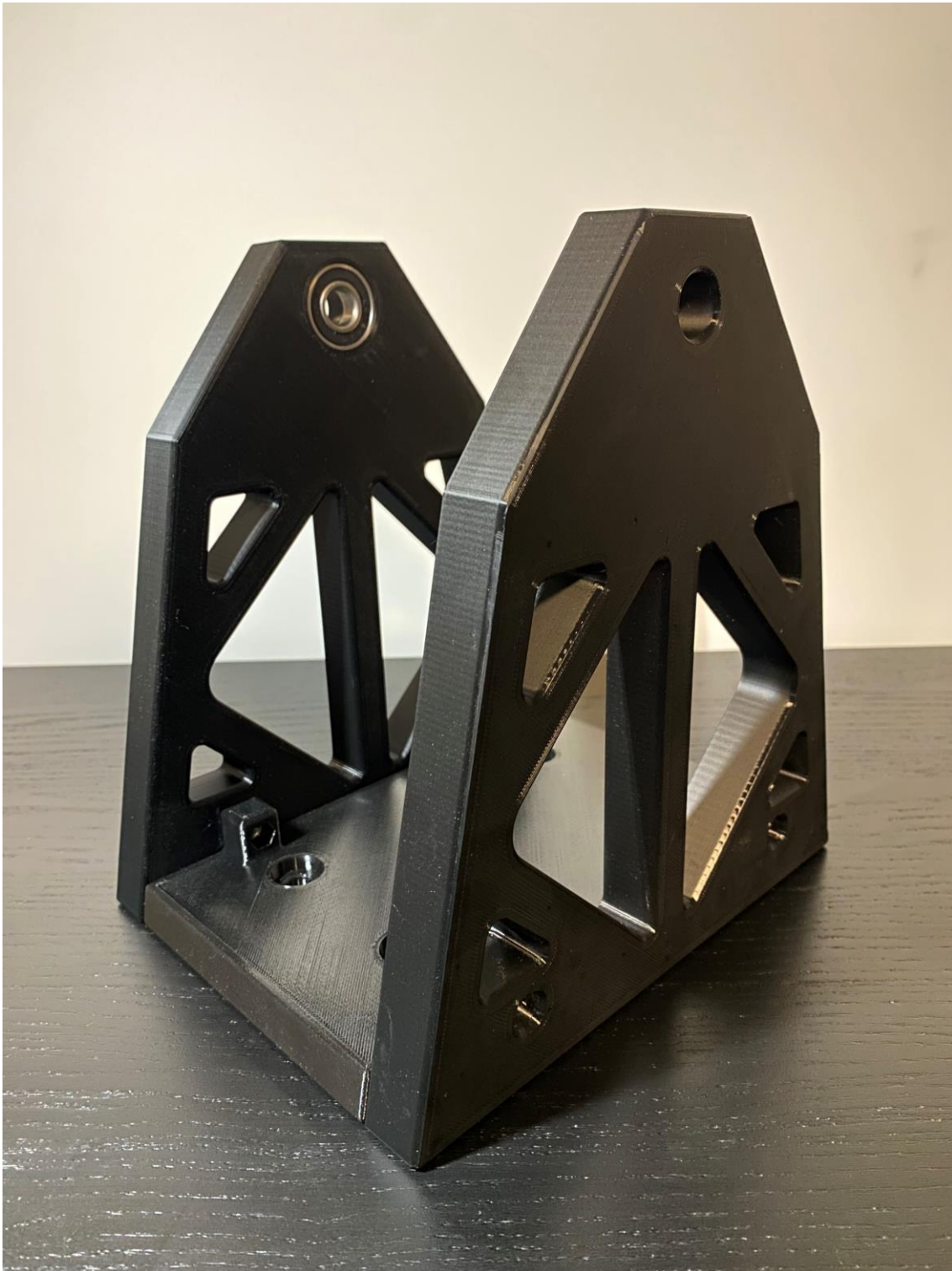
```
MATLAB Function1
KugelaufRadZustandsregler ▶ Sensorik ▶ MATLAB Function1
1 function phi = Winkelberechnung(x, d, r, d0)
2 % x = Abstand Sensor zur Radachse (konstant)
3 % d = Ist-Entfernung der Kugel zum Sensor
4 % r = Abstand Kugelmittelpunkt zur Radachse (konstant)
5 % d0 = Entfernung Kugel zum Sensor in der Ruhelage (konstant)
6
7 phi_ist = acos((x^2 + r^2 - d^2) / (2 * x * r)); % Berechnung des aktuellen Winkels mithilfe des Kosinussatzes.
8 phi_soll = acos((x^2 + r^2 - d0^2) / (2 * x * r)); % Berechnung des Winkels in der Ruhelage mithilfe des Kosinussatzes.
9 phi = phi_ist - phi_soll; % Winkelabweichung von der Ruhelage (Differenz der beiden Winkel)
10 phi = round(phi * 1e4) / 1e4; % Rundet "phi" auf 4 Nachkommstellen auf.
11 end
```

## Anhang F Verlötung der Kondensatoren am Motor





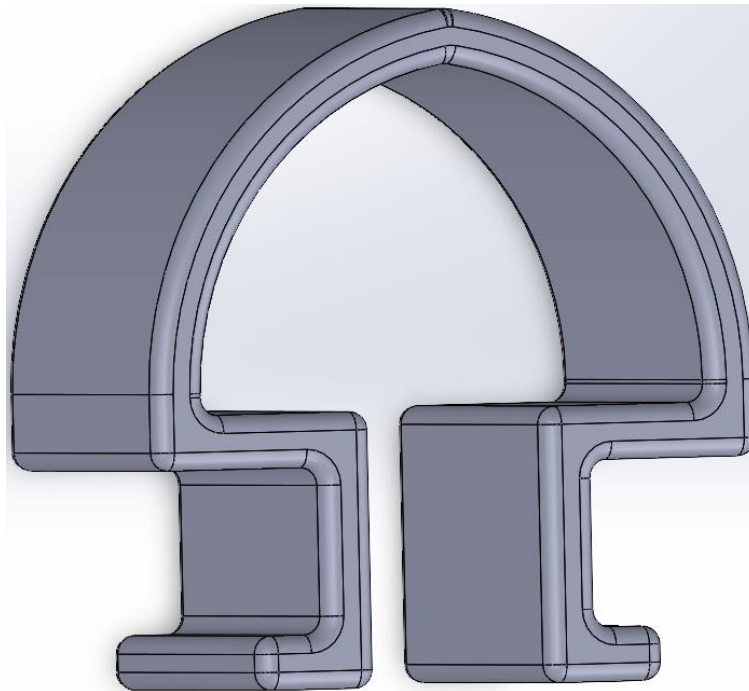
## Anhang G Verworfenne Radhalterung





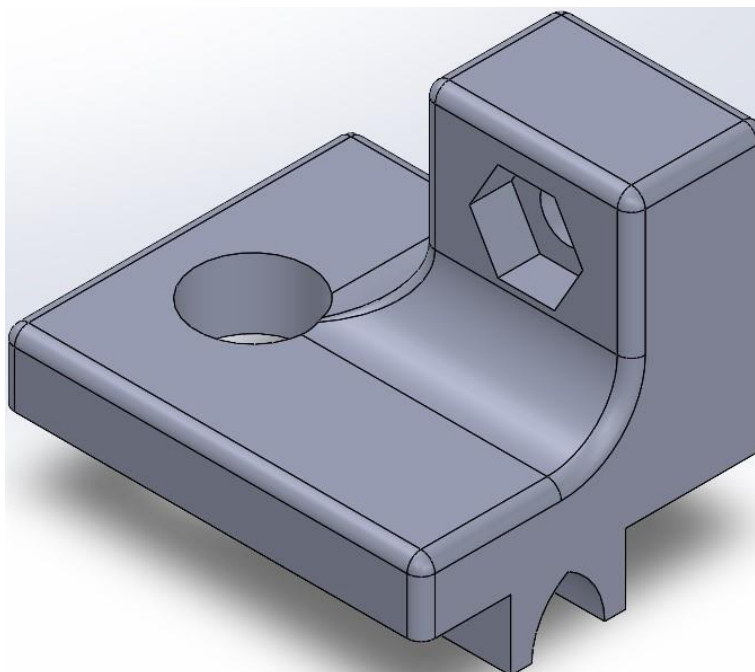
## Anhang H Halterungen

### Kabelhalterung

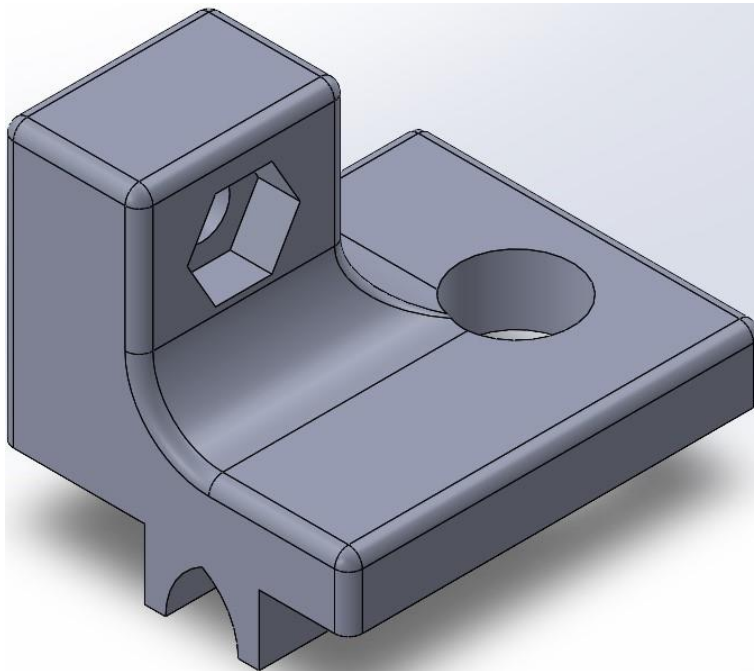


### Plexiglashalterungen

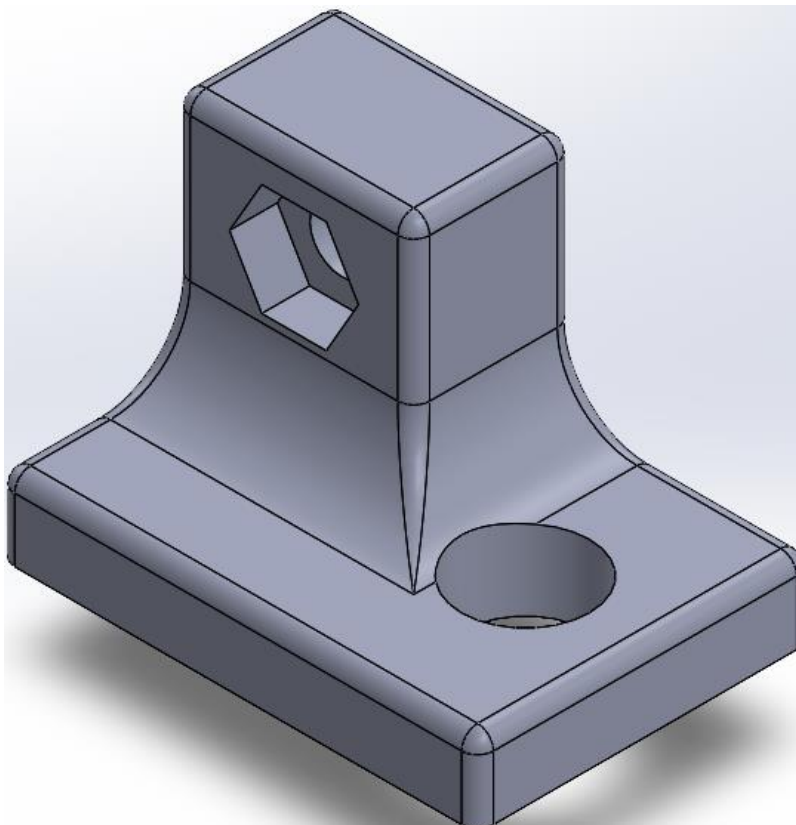
#### Links



**Rechts**

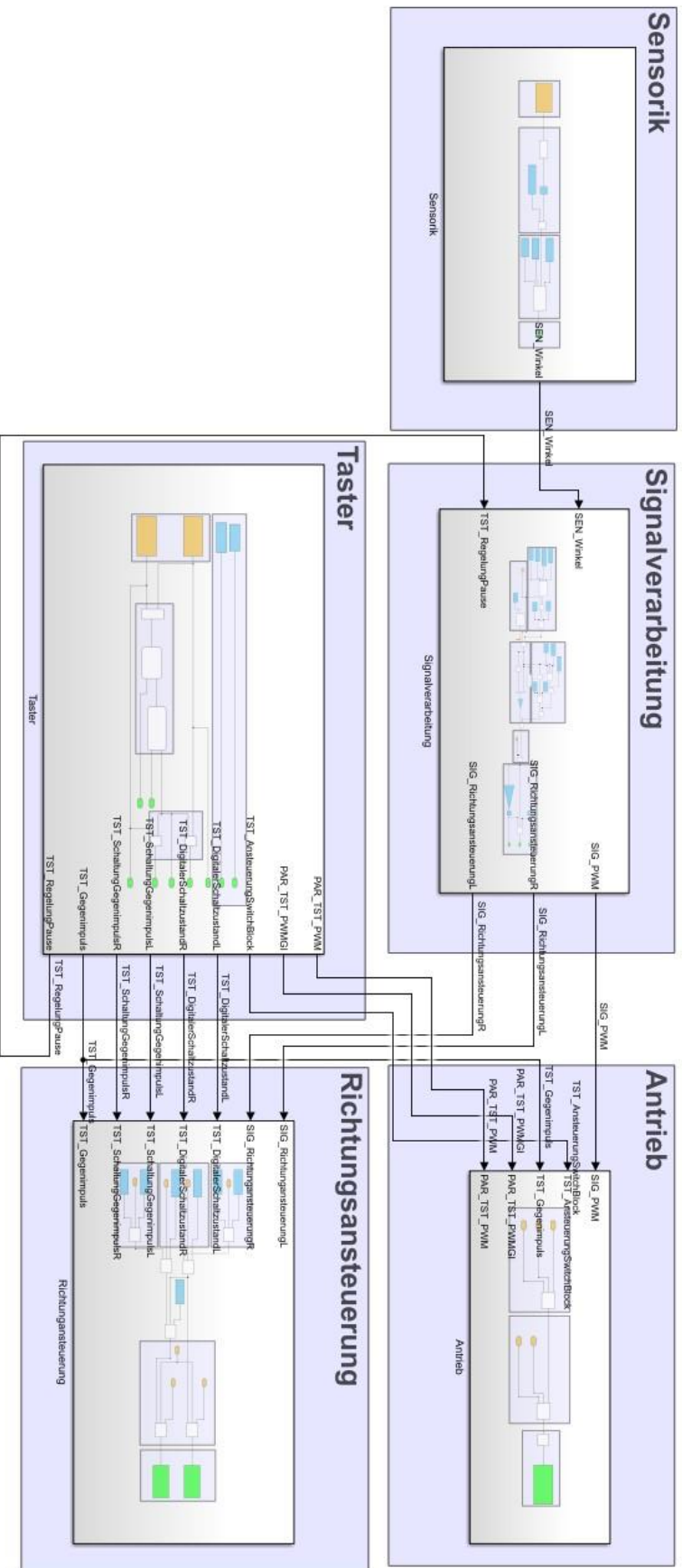


**Halterung am Sensor**



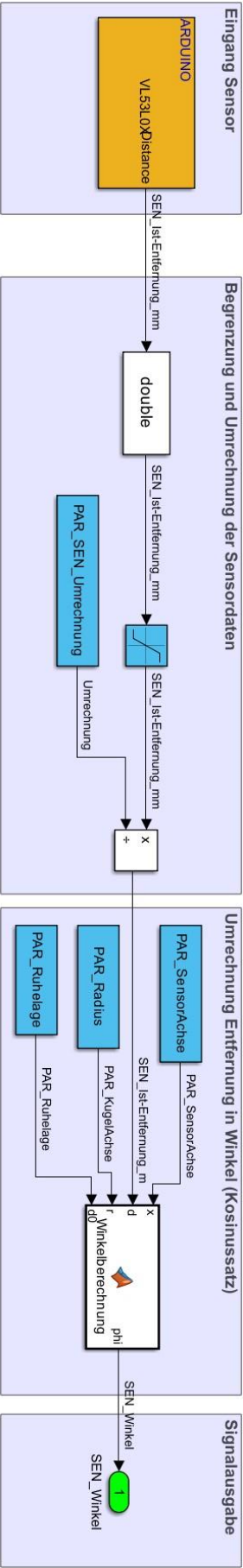
# Modul: KugelaufRad.slx

Datum: 09.05.2025  
Funktion: Balancierung einer Kugel auf einem Rad  
Implementierung: Simulink R2023b  
Toolbox: Simulink Support Package für Arduino Hardware  
Autor: Ken Hiltz  
Letzte Änderung: 09.05.2025

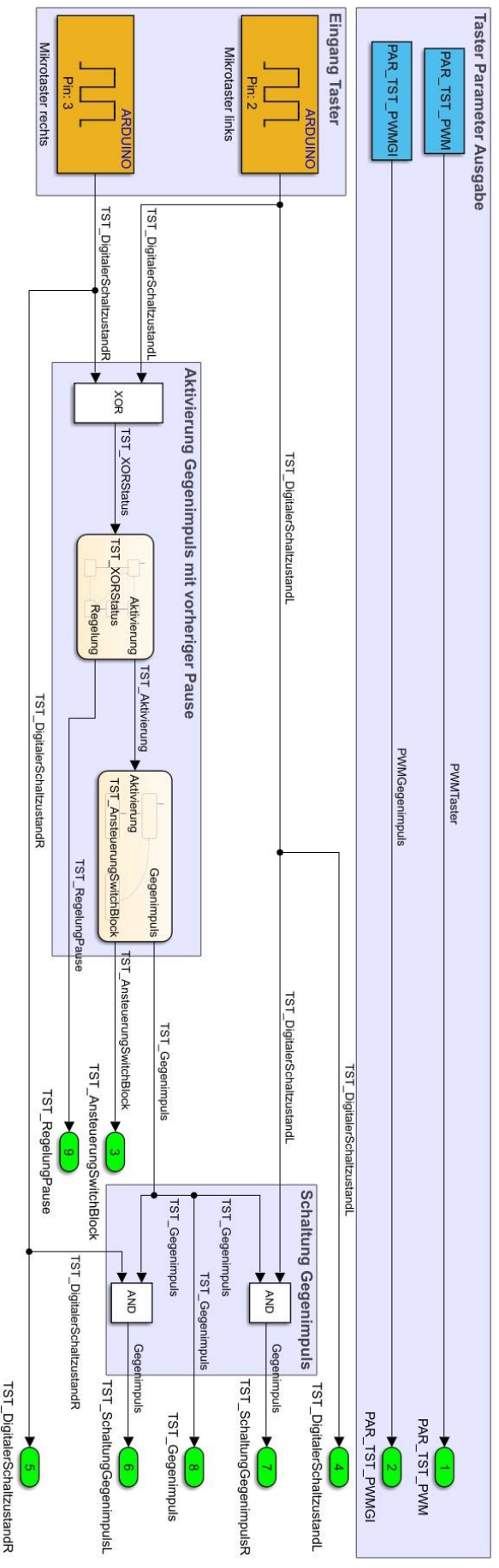


## Anhang I Software

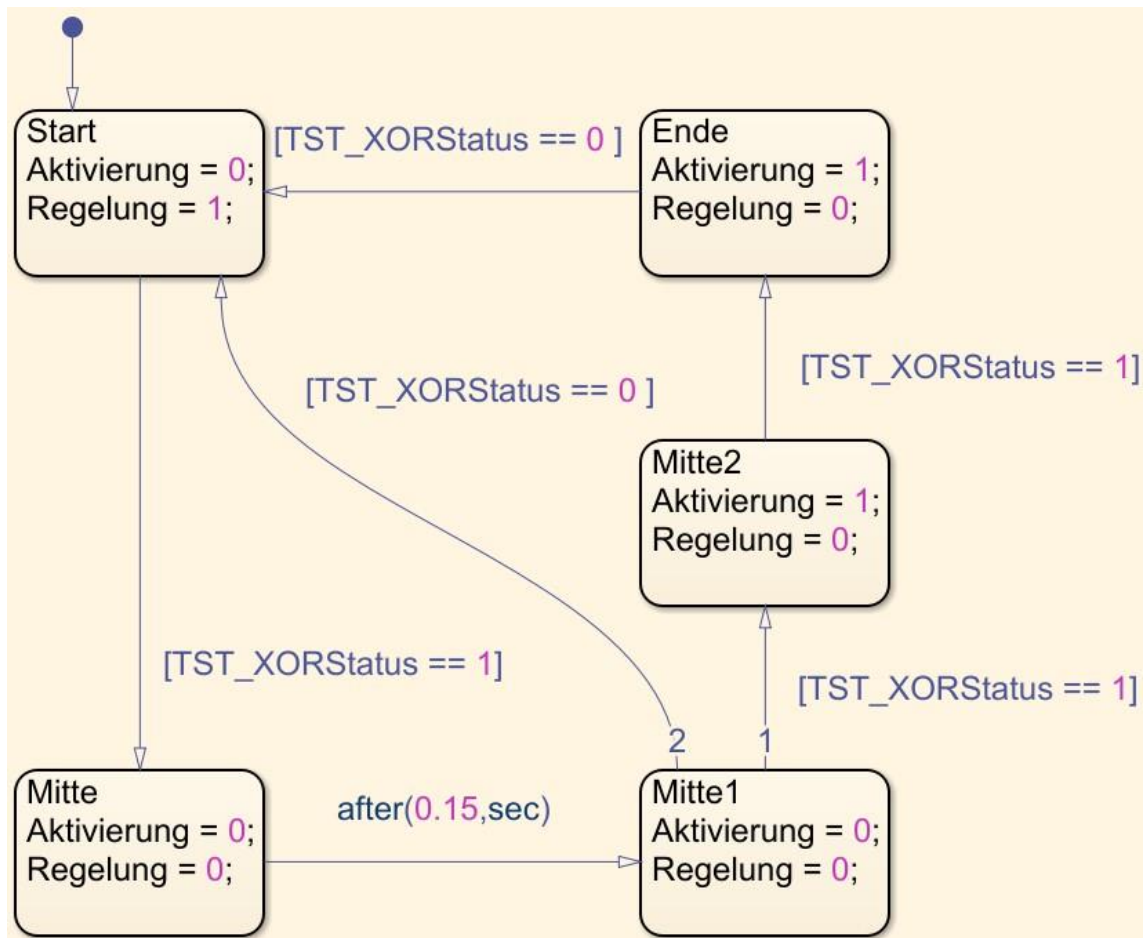
### Gesamt-Software



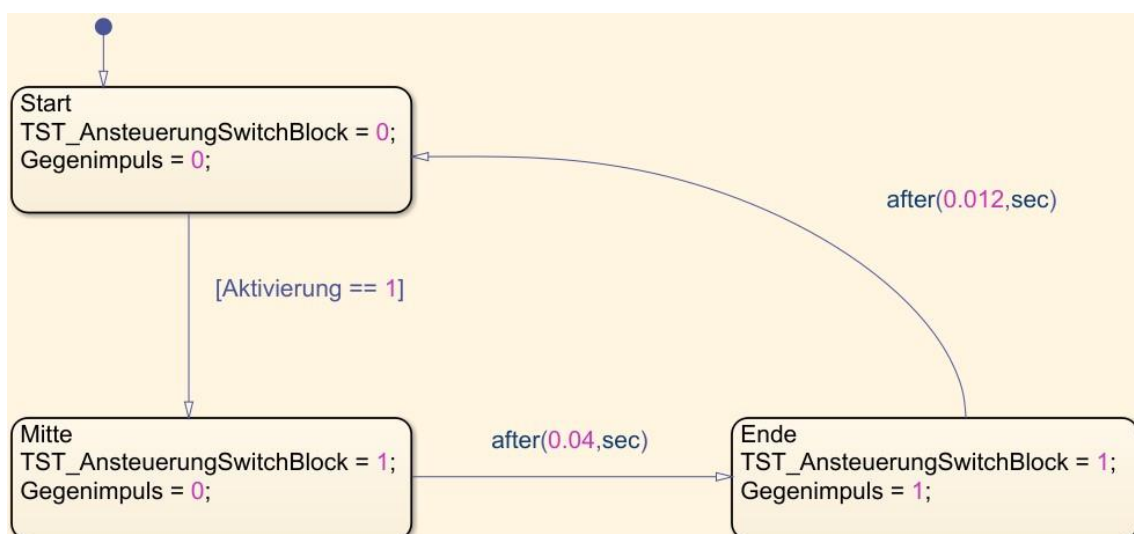
## Taster

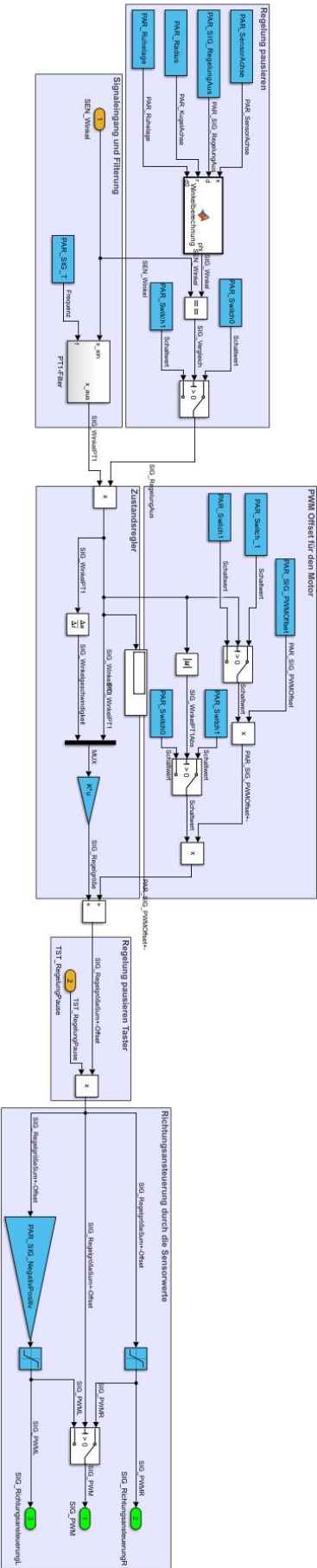


## Stateflow-Chart Pausierung des Systems



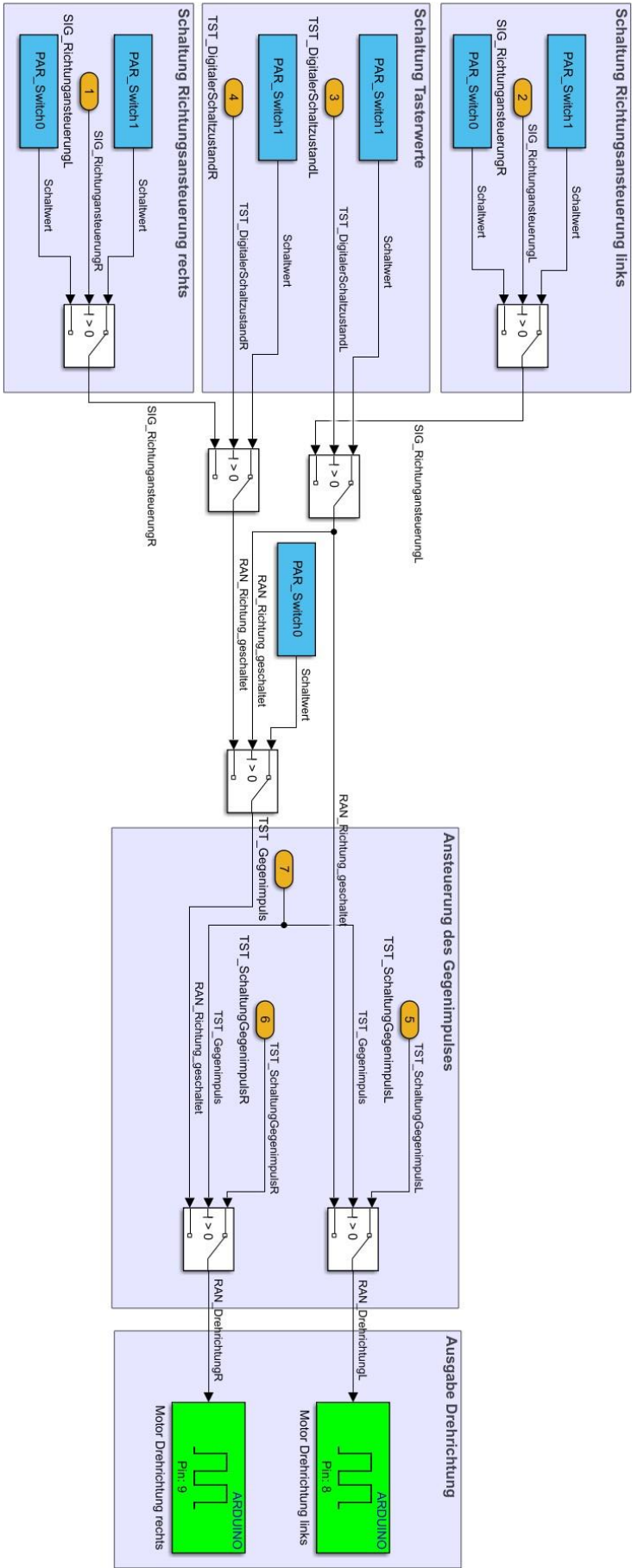
## Stateflow-Chart Gegenimpuls



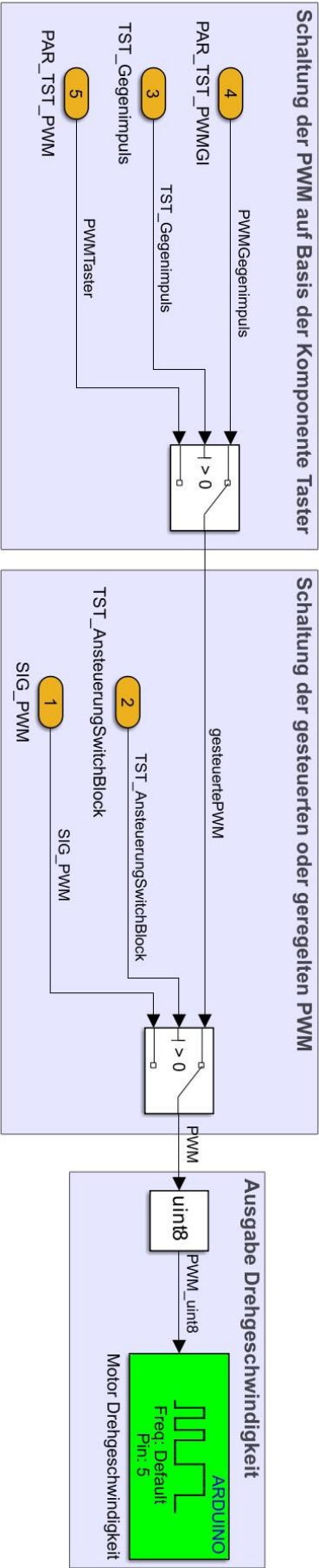




Richtungsansteuerung







## Anhang J Bill of Materials

<i>Nr.</i>	<i>Bauteil</i>	<i>Stückzahl</i>	<i>Bezugsquelle</i>
1	Aluminiumprofil 40x40 Nut 8	Nach Bedarf	HSHL
2	Aluminiumprofil Endkappen	5	Amazon
3	Antriebsriemen 500mm	1	Amazon
4	Arduino Mega	1	Amazon
5	DC-Motor	1	Amazon
6	Diverse Schrauben, Muttern und Unterlegscheiben	Nach Bedarf	Baumarkt/Internet/ HSHL
7	Doppel H-Brücke	1	Amazon
8	Dupont Stecker	Nach Bedarf	HSHL
9	Filament	4	Amazon
10	Gewinde Adapter M6 zu M4	2	Amazon
11	Gewindestange M4	1	Baumarkt
12	Griffe Alurahmen Set	4	Amazon
13	Gummikugel	1	Amazon
14	Hammermutter Nutenstein Nut 8, M6	2	HSHL
15	Kantenschutz	1	Amazon
16	Kondensator	4	HSHL
17	Kugellager	2	Amazon
18	Mikrotaster	2	Amazon
19	Motorhalterung	1	Amazon
20	Netzteil Arduino	1	Conrad
21	Netzteil Motor	1	Conrad

22	Not-Aus-Knopf	1	Amazon
23	Nutensteine Nut 8, M8	21	HSHL
24	Plexiglasscheibe	1	HSHL
25	Rad	1	Amazon
26	Radachse	1	Amazon
27	Schraubenrosetten	2	Baumarkt
28	VL53L0X ToF-Sensor	1	Conrad
29	XT60-Stecker Paar	1	HSHL
30	Zahnriemenscheibe Motor	1	Amazon
31	Zahnriemenscheibe Rad	1	Amazon

# ToF

SKU:U010



## Description

**ToF** that employs time-of-flight techniques to resolve distance between the emit point and the reach point of a subject, measuring the round trip time of an artificial light signal provided by a laser.

This unit integrated a distance measuring sensor VL53L0x providing accurate distance measurement whatever the target reflectance, unlike conventional technologies. It can measure absolute distances up to 2m in less than 30ms.

This unit communicates with M5Core via I2C(0x29).

- In this case, make sure you use the 3.3V on SDA and SCL, M5Core GROVE provide 3.3V to data pins, 5V to power pin. only 3.3v allowed on VL53L0x.

## Product Features

- High precision
- Measure absolute distances up to 2m
- Measurement accuracy  $\pm 3\%$

- The wavelength of laser: 940nm
- Program Platform: Arduino, UIFlow(Blockly, Python)
- Two Lego-compatible holes

## | Include

---

- 1x ToF Unit
- 1x Grove Cable

## | Applications

---

- 1D gesture recognition
- Laser Ranging
- 3D structured light imaging (3D sensing)
- Camera assist (ultra fast autofocus and depth of field)

## | Specification

---

Resources	Parameter
Measuring distance	0.3-2m
Measurement accuracy	±3%
Net weight	4g
Gross weight	17g
Product Size	32*24*8mm
Package Size	67*53*12mm

## | Related Link

## | EasyLoader

---

EasyLoader is a concise and fast program writer, which has a built-in case program related to the product. It can be burned to the main control by simple steps to perform a series of function verification.

[Download Windows Version Easyloader](#)

[Download MacOS Version](#)

[Easyloader](#)

### Description:

The screen displays the current ranging data.

## | Schematic

---

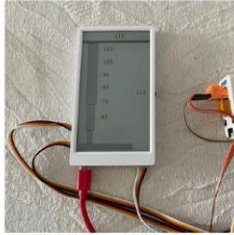
### ToF Schematic



### PinMap

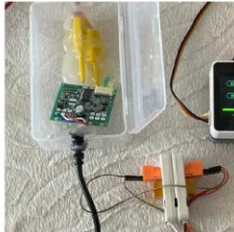
M5Core(GROVE A)	GPIO22	GPIO21	5V	GND
ToF Unit	SCL	SDA	5V	GND

## | Learn



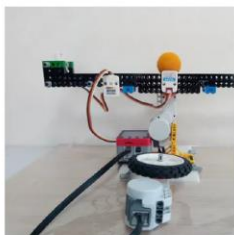
### Control sit/stand Desk height with a Paper E-Ink Display

Control IKEA Bekant Desk sit/stand height with M5Stack M5Paper using two Relay units and a ToF (time-of-flight sensor VL53L0x) unit.



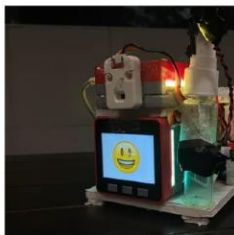
### Control sit/stand Desk height with a touch screen

Control IKEA Bekant Desk sit/stand height with M5Stack Core2 using two Relay units and a ToF (time-of-flight sensor VL53L0x) unit.



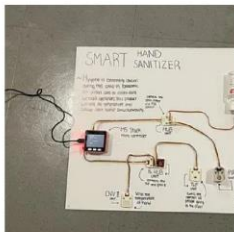
### M5Stack Based PID Control Learning Platform

The PID Control Learning Platform is based on an M5Stack Fire & BaseX with TOF and Accel Units, and some Lego NXT parts.



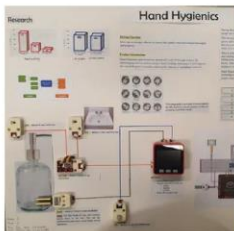
### Q-Bot

We aim to build a robot that is effective in sanitizing surfaces without putting cleaners in harm's way.



### 3 in 1 Sanitiser Dispenser

We have devised with an innovation that dispenses hand sanitiser, measures the temperature of user, shows the count of people in vicinity.



### Hand Hygienics

Hand Hygienics aims to reduce spread of Covid-19 through contact.



### SEAM [Safe-Entry Access Machine]

An automatic system to perform the tasks of Safe Entry. It consist an adjustable Thermometer, RFID / QR Code Check in and Counting.

## | Example

---

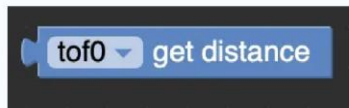
### Arduino

- [Click here to download the Arduino example](#)

### UIFlow

#### Feature Introduction

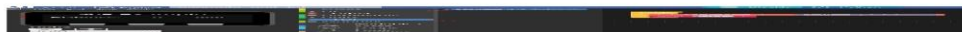
Measuring distance up to 2 meters



- **Get distance** Return distance mm

#### Usage

- [Click here to download the UIFlow example](#)





Last updated: 2020-12-14



# Handson Technology

Data Specs

## 775 Ball Bearing DC Motor

Ball bearing DC motor with built-in cooling fan. High torque with wide operating voltage 12~36 Vdc with speed of 3600~9000 RPM. Suitable for motor tools application and DIY projects.



SKU: [MCH-1166](#)

### Specifications:

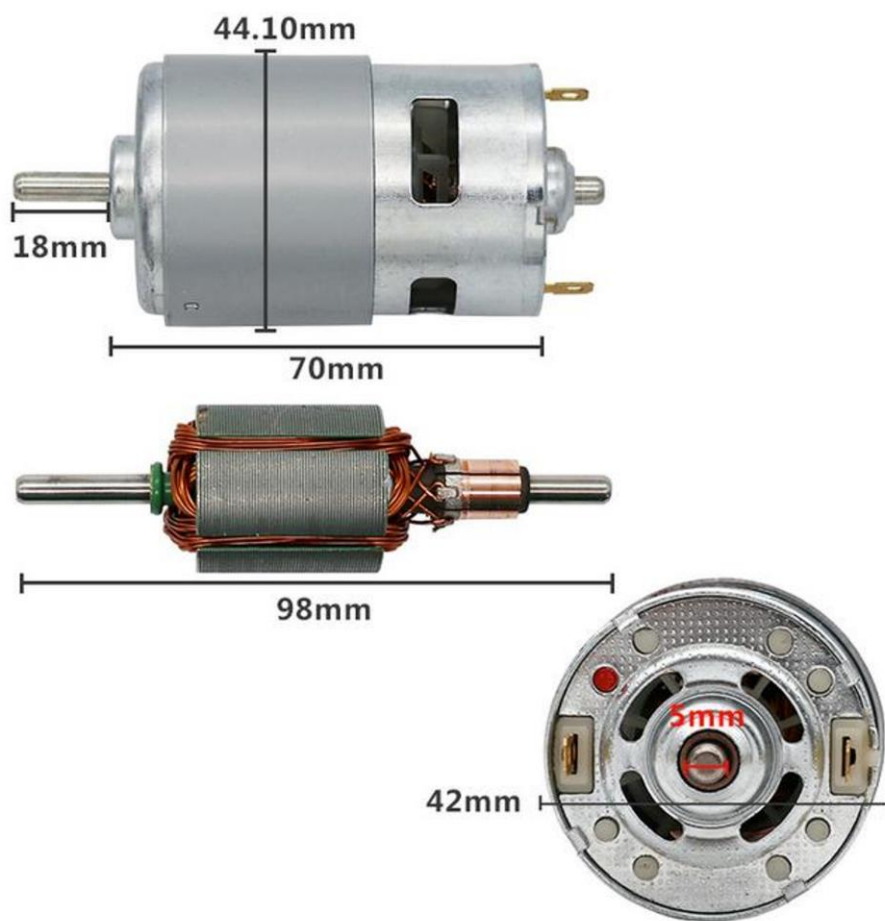
- Motor type: 775.
- Rated voltage: DC12V-24V (Max 36V).
- No Load Speed: 3500~9000 RPM.
- Rated current: 1.2A @ 12V.
- Cooling Fan: Internal
- Overall Size: 98x42mm.
- Shaft: Full Round Type Ø5mm.
- Mounting Hole Size: M4.

### Application:

- CNC Machine
- Robotics
- Linear Motion

### Mechanical Dimension:





# Anhang M Not-Aus-Knopf

## Technical Parameter

Conforming standard	EN/IEC60947-5-1, EN/IEC60947-5-4, GB14048.1-93, GB14048.5-2008		
Vibration resistance	Mushroom head button: 8g, other buttons: 15g (40Hz<frequency<500Hz) Conform to IEC6Q06826		
Shock resistance	Button, button with lamp: 70g, mushroom head button:15g, selective switch; 120g, conform to IEC60068-2-27		
Protection grade	IP 65 (front)		
Reliability performance	(programmable controller input 24VDC )		
Wiring	Screw-in terminal, Wiring capability: min 1x0.5mm <sup>2</sup> , max, with or without terminal strip:2*1.5mm <sup>2</sup> or 1*2.5mm <sup>2</sup>		
Rated insulation voltage ui (V)	660V	Mechanical life:	Normal button≥3 million times, knob≥0.3 million times
Conventional thermal current Ith(A)	10A	Electrical life:	>0.6 million times, others 0.1 million times
Rated capacity	AC-15	Recommended torque:	0.9Nm
	DC-13		
	220V/6A, 380V/4A, 660V/2A		
	440V/0.3A, 220V05A, 110V/1A		
Power frequency voltage	2500V/60S	Terminal:	Suitable for IEC No. 2 screwdriver (+/- type) and bolt with DIN washer
Contact resistance	<25mΩ	Temperature:	-25~+55℃
Short circuit protection	When the maximum short circuit current is 1KA, fuseI 6A (common) 10A (delay)		

5.3.2 Range profile examples

Table 13 details typical performance for the four example ranging profiles, as per measurement conditions in Section 5.3: Ranging accuracy.

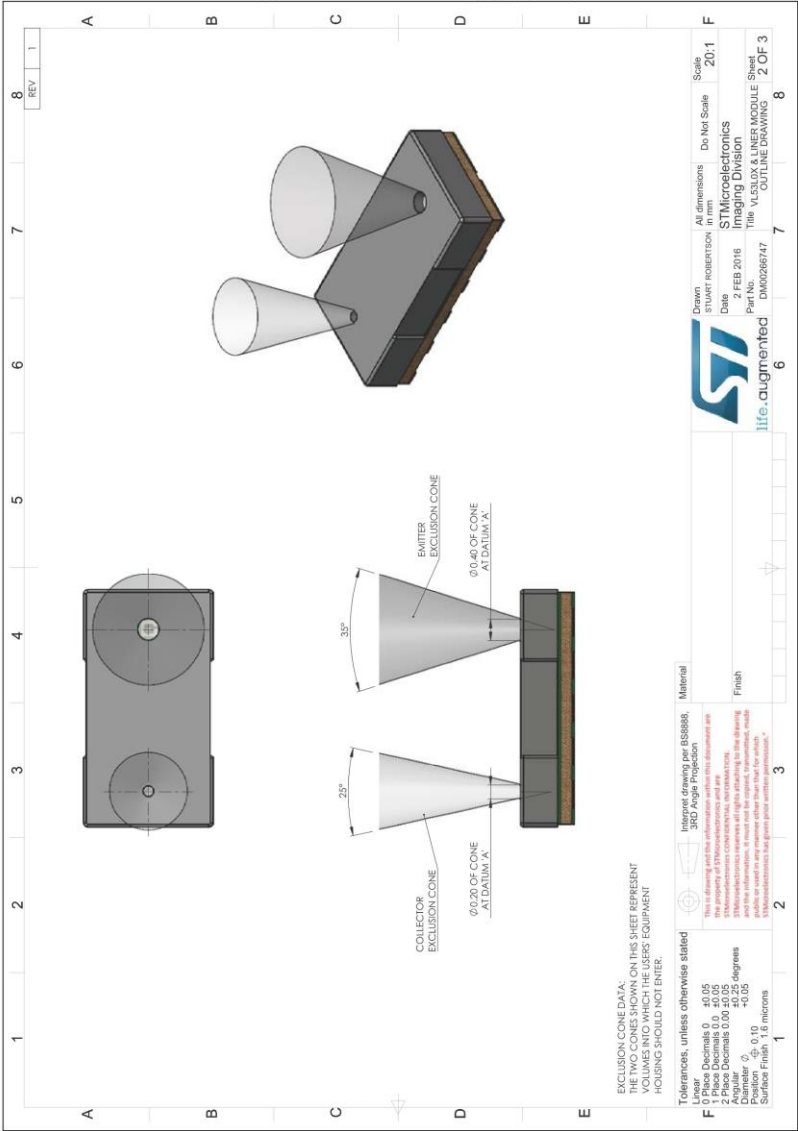
Table 13. Range profiles			
Range Profile	Range timing budget	Typical performance	Typical application
Default mode	30ms	1.2m, accuracy as per Table 12	standard
High accuracy	200ms	1.2m, accuracy < +/- 3%	precise measurement
Long range	33ms	2m, accuracy as per Table 12	long ranging, only for dark conditions (no IR)
High speed	20ms	1.2m, accuracy +/- 5%	high speed where accuracy is not priority

5.3.3 Ranging offset error

The table below shows how range offset may drift over distance, voltage and temperature. Assumes offset calibrated at 10cm. See VL53L0X API User Manual for details on offset calibration.

Table 14. Ranging offset				
	Nominal Conditions	Measure point	Typical offset from nominal	Maximum offset from nominal
Ranging distance	Offset calibration at 10cm ("zero")	White 120cm (indoor) Grey 70cm (indoor) White 60cm (outdoor) Grey 40cm (outdoor)		< 3%
Voltage drift	2.8V	2.6V to 3.5V	+/- 10mm	+/- 15mm
Temperature drift	23°C	-20°C to +70°C	+/- 10mm	+/- 30mm

Figure 23. Outline drawing (page 2/3)



## Eidesstattliche Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Literatur und Hilfsmittel angefertigt habe. Wörtlich übernommene Sätze und Satzteile sind als Zitate belegt, andere Anlehnungen hinsichtlich Aussage und Umfang unter Quellenangabe kenntlich gemacht.

Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen und ist auch noch nicht veröffentlicht.

Lippstadt, 26.05.2025

Ort, Datum

Ken Hitz

Unterschrift