



Universität Bremen
Fachbereich 4 Produktionstechnik

Masterthesis

im Studiengang Systems Engineering - Vertiefungsrichtung Eingebettete
Systeme und Systemsoftware

zur Erlangung des akademischen Grades
Master of Science

Thema: Synthese eines geeigneten Reglers für einen Ballspielroboter
mit redundanter Achsansteuerung

Autor: Timo Wörner <tiwo@uni-bremen.de>
MatNr. 3190282

Version vom: 21. Mai 2021

1. Prüfer: Prof. Dr.-Ing. Udo Frese
2. Prüfer: Prof. Dr.-Ing. Kai Michels

Zusammenfassung

Diese Arbeit befasst sich im Kern mit der Auswahl eines geeigneten Reglersystems für einen ballspielenden Roboter, sowie dessen Parametrierung. Die Aufgabe des Reglersystems besteht darin, den Aktuator eines dreiachsigen Ballspielroboters mit minimalem Zeitaufwand an eine beliebige Position in dessen Arbeitsbereich zu bewegen, um so mit einer präzisen Schlagbewegung einen auftreffenden Ball in eine gewünschte Richtung zu schlagen.

Die Aufgabe der Ballerkennung zählt nicht zum Inhalt dieser Arbeit und ist zum Zeitpunkt der Fertigstellung auch nicht nutzbar. Um dennoch eine Möglichkeit bereit zu stellen die Funktionalität der implementierten Regelung zu zeigen, wurde eine umfangreiche Steuersoftware entwickelt, die unter anderem die Bedienung mit einem Spielecontroller ermöglicht. Für die Ermittlung einer optimalen Reglerstruktur wurde neben einem heuristischen Verfahren zur Parameterbestimmung von PID Reglern, auch ein simulationsbasierter Ansatz untersucht. In den Kapiteln 4 und 5 sind die dazu nötigen Schritte beschrieben.

In Kapitel 6 wird die finale Lösung präsentiert und evaluiert. Außerdem sind hier frühe Testergebnisse abgebildet, die den Weg zur resultierenden Lösung erkenntlich machen sollen.

Abstract

The core of this thesis is the selection of a suitable controller system for a three axis ball batting robot and its parameterization. The task of the controller system is to move the actuator of the robot with minimal expenditure of time to any position in its work area in order to hit an impacting ball with a precise stroke movement and continue playing.

The tasks of ball recognition and track planning are not part of this work and cannot be used at the time of completion. In order to still provide the possibility to show the functionality of the implemented control algorithm extensive control software was developed, which among other things enables operation with a game controller. To determine an optimal controller structure, a heuristic method for determining the parameters of PID controllers and a simulation-based approach were investigated. In the chapters 4 and 5 the necessary steps are described.

In chapter 6 the final solution is presented and evaluated. In addition, early test results are shown here, which should describe the way to the resulting solution.

Inhaltsverzeichnis

Abbildungsverzeichnis	iii
Tabellenverzeichnis	iv
Abkürzungsverzeichnis	v
1 Einleitung	1
1.1 Motivation	1
1.2 Aufgabenbeschreibung	2
1.3 Übersicht	2
2 Wissenschaftliche und technische Grundlagen	4
2.1 Antrieb	4
2.1.1 Gleichstrommotoren	4
2.1.2 Thermisches Verhalten	6
2.2 Robotik	7
2.2.1 Bildung einer kinematischen Kette unter Anwendung der Denavit-Hartenberg-Konventionen	7
2.2.2 Berechnung der dynamischen Einflüsse auf die Gelenke	8
2.3 Regelungstechnische Grundlagen	9
2.3.1 Identifikation der Regelstrecke	10
2.3.2 Welche unterschiedlichen Arten von Reglern gibt es?	11
2.3.3 Bahnplanung	13
3 Der Ballspielroboter Doggy2	15
3.1 Die mechanischen Komponenten	15
3.2 Übersicht der Elektronik	16
3.3 Prüfstand und Grundlegende Funktionen des Roboters	16
4 Reglersynthese	19
4.1 Systemidentifikation der Regelstrecke	19
4.2 Auswahl der Reglerstruktur	20
4.3 Systementwurf auf Basis eines MATLAB-Simulink Models	24
4.4 Ein kaskadierter Momentenregler mit Vorsteuerung	26
4.4.1 Die innere Drehmomentreglerschleife	26
4.4.2 Die äußere Geschwindigkeitsreglerschleife	29
4.4.3 Die Vorsteuerung	29
4.5 Planung einer Bewegungsbahn mit dreieckigem Geschwindigkeitsverlauf	33
5 Systemintegration und Benutzerinterface	36
5.1 Zusammenbau und Inbetriebnahme	36
5.2 Die Doggy2-Elektronik	38
5.2.1 Strombegrenzung	38
5.2.2 Temperaturregelung	39
5.2.3 Das Kommunikationsinterface	40
5.2.4 Anti Windup Verfahren durch Begrenzung des I-Reglers	41
5.2.5 Initialisierungssequenz	41
5.2.6 Problembeschreibung kaputte Elektronik	42
5.3 Das Benutzerinterface	43

6	Evaluation	46
6.1	Entwicklungsbegleitende Systemanalyse	46
6.1.1	PI-Regler	47
6.1.2	Geschwindigkeitsvorsteuerung	49
6.2	Entwicklungsstand zum Projektende	50
6.2.1	Wiederholgenauigkeit	51
6.2.2	Schlagbewegung	52
7	Fazit und Ausblick	55
	Literaturverzeichnis	58
	Anhang	61
	Eidesstattliche Erklärung	63

Abbildungsverzeichnis

1	Gesamtansicht der CAD-Konstruktion	2
2	Ersatzschaltbild Gleichstrommotor	4
3	Thermisches Zweikörpermodell	6
4	Standardregelkreis	10
5	Reglervergleich Quelle: [Was14]	13
6	Doggy2 Prüfstand	17
7	Histogram Getriebeispiel	18
8	Kaskadierter Regler mit innerer und äußerer Reglerschleife	21
9	Sprungantwort der unregelmäßigten Regelstrecke mit eingezeichneter Wendepunkt tangente und Zeitparameter T_1 und T_t	22
10	Vergleich der Sprungantwort des unregelmäßigten Systems (rot) mit dem geschlossenen Regelkreis mit PID-Regler (blau)	23
11	Grafische Darstellung der Matlab Simulink Simulation des Doggy2 Roboters	25
12	PI-Regler Sprungantwort	27
13	PI-Regler Sprungantwort	28
14	Sprungantwort der äußerden P-Reglerschleife	29
15	Dynamischer Momentenverlauf der dritten Bewegungsachse bei einer Rotation aller Gelenke	30
16	Visualisierung der vereinfachten Massenmatrixelemente unter Variation von q_2 und q_3	32
17	Sättigung der Drehmoments Stellgröße	33
18	Positions-, Geschwindigkeits- und Beschleunigungsverlauf bei optimaler Trajektorie	34
19	Bildstrecke Montagearbeiten	37
20	PWM/Drehzahl-Kennlinie	39
21	Temperaturverlauf bei Motorlast	40
22	Das Benutzerinterface	43
23	Sprungantwort des realen Systems mit implementiertem PID-Regler	47
24	Sprungantwort des realen Systems mit implementiertem PI-Regler	48
25	Periodische Gelänkwinkeländerung	49
26	Bewegung der Achsen 2 und 3 mit starr bleibender Achse 1	50
27	Ergebnisse der Versuchsreihe zur Schlaggenauigkeit	51
28	Schlagversuch mit Ball an der Schnur	53
29	Schlagbewegung mit variierender Endgeschwindigkeit im Schlagpunkt	54
30	Simulink Simulation des Doggy2-Gesamtsystems mit Regler und Dynamik	62

Tabellenverzeichnis

1	Bedeutung der Denavit-Hartenberg-Parameter Quelle: [Cor17, S. 198]	8
2	Denevit-Hartenberg Parameter des Doggy2 Robotermodells	24
3	Reglereinstellungen nach Ziegler-Nichols (bekannte Regelstreckenparameter) Quelle: [Trö11]	61

Abkürzungsverzeichnis

CAD	computer aided design
DH	Denavit-Hartenberg
ESD	electro static discharge
GUI	graphical user interface
IMU	inertial measurement unit
JTAG	joint test action group
MOSFET	metal oxide semiconductor field effect Transistor
PID	proportional integral derivative
PLA	Polylactide
PWM	Pulsweitenmodulation
ROS	robotic operating system

1 Einleitung

Roboter unterschiedlichster Art finden in der heutigen Zeit immer mehr Verwendung im Alltag. Sie unterstützen uns Menschen in nahezu allen Bereichen, die dies zulassen. Längst etabliert haben sie sich in industriellen Anlagen, um beispielsweise einfache Transportaufgaben [Ind35] oder auch präzise Montage- und Schweißarbeiten [Kuk20] zu übernehmen. Aber auch bei hochkomplexen Herausforderungen, wie dem Durchführen von chirurgischen Eingriffen [Int21], finden Roboter ihre Anwendung.

Neben zahlreichen weiteren produktiven Aufgabenbereichen sind Roboter gleichermaßen in der Unterhaltungsbranche zu finden. Ein Beispiel hierfür sind Veranstaltungen, bei denen Roboter gegeneinander antreten, wie etwa als Fußballmannschaften beim RoboCup [Rob16]. Aber auch die Roboter-Mensch-Interaktion lässt sich spielerisch anhand von Unterhaltungsrobotern wie dem Torhüter-Roboter Robokeeper [4at21] demonstrieren. An dieser Stelle soll auch der in dieser Arbeit behandelte Roboter Doggy2 anknüpfen. Nicht etwa durch Herausfordern der menschlichen Mitspieler, sondern vielmehr durch ein spielerisches Miteinander soll hier eine Verbindung zwischen Mensch und Maschine hergestellt werden.

Ganz unabhängig vom Einsatzzweck liegt die Gemeinsamkeit, die sich hinter all diesen sehr unterschiedlichen Robotern verbirgt, in den Bewegungsabläufen, die die Roboter für die Bewerkstelligung ihrer Aufgaben benötigen. Für jede der genannten Aufgaben ist es nötig, hoch genaue und zeitlich präzise gesteuerte Bewegungen in jedem einzelnen Gelenk zu gewährleisten.

1.1 Motivation

Die Herausforderung, die sich stellt, wenn der Endeffektor eines Roboters auf eine gewünschte Position bewegt werden soll, ist neben der präzisen Ansteuerung der Aktorik, auch die Vorausberechnung der Bewegung. Dabei muss vor allem auf die gegenseitige Beeinflussung der mechanischen und physikalischen Einflüsse große Rücksicht genommen werden. Die Aufgabe der Regelung, die die Koordination der Bewegung einzelner Gelenke übernimmt, ist es diese Einflüsse auszugleichen und somit einen vorgegebenen Sollzustand zu erreichen. Unabhängig von der jeweiligen Aufgabe des Roboters werden so elektrische Servomotoren dazu benutzt, die Gelenke des Roboters zu bewegen.

Diese Arbeit bietet Einblicke in die systemtechnische Seite der Entwicklung von ganzheitlichen Regelungssystemen. Dabei soll in besonderer Weise der Fokus auf die Berücksichtigung der Anforderungen an spezielle Systemgegebenheiten gesetzt werden.

1.2 Aufgabenbeschreibung

Ziel dieser Arbeit ist es, für die zweite Generation des ballspielenden Roboters namens Doggy2 die Ansteuerung der Motoren zu entwerfen. Dabei wird zunächst eine geeignete Reglerstrategie unter Berücksichtigung der technischen und funktionalen Anforderungen synthetisiert. Auf Basis dieser Regelung wird dann ein Bahnplanungsverfahren implementiert, welches die Durchführung von Schlagbewegungen, sowie fließende Bewegungen des Roboteraktuators zulässt.



Abbildung 1: Gesamtansicht der CAD-Konstruktion

Abbildung 1 zeigt ein CAD-Modell des dreiaxigen Roboters, wie er im Verlauf dieser Arbeit entstehen wird. Parallel dazu wird noch der Stromverbrauch der Aktoren beobachtet, um so bei Bedarf überschüssige Energie über einen Bremswiderstand abzuführen und die Motoren vor Überhitzung zu schützen.

1.3 Übersicht

Für die Beantwortung der Frage, welcher Voraussetzungen es bedarf, um eine Regelung für einen ballspielenden Roboter zu implementieren, wird in Kapitel 2 zunächst auf die technischen Grundlagen eingegangen. Dabei wird, wie auch im weiteren Verlauf der Arbeit, zunächst auf die Details im Inneren Einblick gewährt, um dann sukzessive einen Überblick über das gesamte System zu schaffen.

So befasst sich der erste Abschnitt genauer mit den Gleichstrommotoren sowie deren technischen und dynamischen Eigenschaften.

Im Kapitel 2.3 werden kurz einige wichtige Aspekte der Regelungstechnik eingeführt, um den Lesenden einen besseren Einblick in die Entscheidungen zu geben, welche bei der tatsächlichen Reglersynthese zum beschriebenen Ergebnis geführt haben.

Der Systemeinblick wird mit der Betrachtung der Robotik abgeschlossen. Hier erfahren die Leser wofür die in den vorausgegangenen Kapiteln beschriebene Aktorik eingesetzt wird.

Kapitel 3 beinhaltet die Beschreibung des Roboters und erklärt genauer, wie die an ihn gestellten Aufgaben aussehen.

Die Synthese der Regelung ist in Kapitel 4 beschrieben. Außerdem wird hier ein Einblick in das Erstellen des Simulationsmodells sowie die Bahnplanung gegeben.

Auf welche Art und Weise die aus den vorausgehenden Kapiteln gewonnenen Erkenntnisse in die Praxis umgesetzt wurden, wird in Kapitel 5 erläutert. Hier werden auch die teilweise unvorhergesehenen Herausforderungen bei der Umsetzung der Aufgaben aufgezeigt. In Folge dessen werden lösungsorientierte Ansätze beschrieben, die es ermöglicht haben, trotz der Schwierigkeiten zu einem Ergebnis zu kommen.

Neben den Tests, die während des Entwicklungsprozesses für Einblicke in die Funktionsweise des Roboters boten, werden in Kapitel 6 die abschließenden Tests und deren Ergebnisse besprochen.

Das Fazit und einen weiteren Ausblick auf die nächsten Schritte im Projekt Doggy2 schließen die Arbeit in Kapitel 7 ab.

2 Wissenschaftliche und technische Grundlagen

Um zum Verständnis der vorliegenden Arbeit beizutragen, wird im Folgenden in Kürze auf die wesentlichen Bestandteile der Roboteraktuatorik und deren Funktionsweise eingegangen. Dabei werden neben den wichtigsten Komponenten auch relevante, physikalische Zusammenhänge deutlich gemacht. Anschließend werden noch einige gängige Verfahren betrachtet, die für die korrekte Berechnung von Bewegungsabläufen nötig sind.

2.1 Antrieb

Für die Ausführung von maschinellen Bewegungen ist der Aktuator das wichtigste Element. In der Robotik wird für die Umsetzung von Rotationsbewegungen in der Regel auf Servomotoren zurückgegriffen.

Ein Servomotor zeichnet sich maßgeblich dadurch aus, dass die Ausgangsgrößen des Motors wie Drehzahl und Drehmoment mittels der eingebauten Mess- und Regeleinrichtung hoch genau geregelt werden können [Sch08, S. 11]. Der Begriff Servomotor ist unabhängig von der eigentlichen Motortechnologie. Im Roboter Doggy2 sind drei Gleichstrommotoren der Firma Maxon Motors verbaut, die bereits über einen eingebauten Encoder sowie eine Getriebestufe verfügen.

2.1.1 Gleichstrommotoren

Gleichstrommotoren, Gleichstrommaschinen oder auch Kommutatormotoren eignen sich sehr gut als Aktuatoren in Robotergelenken. Mit einer großen Variation an Technologien sowie diversen unterschiedlichen Bauformen lässt sich für viele Anforderungen ein geeigneter Motor finden. Für die Ansteuerung dieser Motoren sind keine aufwändigen Frequenzumwandler nötig. Mit einer einfachen Gleichspannungsquelle kann der Motor angetrieben werden. Ist diese über eine geregelte Pulsweitenmodulation (PWM) am Motor angelegt, so können Drehzahl und Moment beliebig gesteuert werden [Cor17, S. 251].

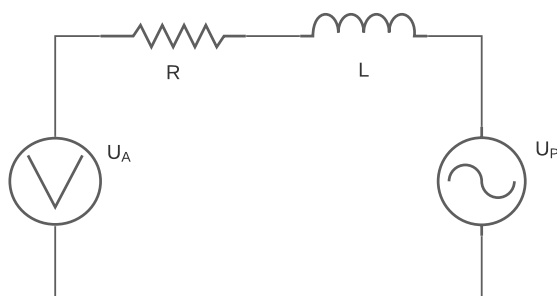


Abbildung 2: Ersatzschaltbild Gleichstrommotor

Die wichtigsten Komponenten dieser Motoren sind neben einem permanent magnetisierten Stator und einem mit Kupferwicklungen versehenen Rotor die beiden Anschlussklemmen, die mit den Bürsten verbunden sind. Außerdem besitzen sie einen Kommutator. Dieser sorgt für die Umpolung der Spannung in den Ankerwicklungen und somit für die rotierende Bewegung.

Um für Rotationen in beide Richtungen einen stetigen Ablauf zu erreichen, ist es notwendig, den Arbeitsbereich des Servomotors in allen vier Quadranten des Drehmoment/Drehzahl-Diagramms zu definieren. [Sch08, S. 15] Das heißt, dass für Rotationen in beide Richtungen jeweils sowohl beschleunigte als auch gebremste Bewegungen möglich sein sollen.

Daraus ergibt sich, dass der Motor nicht nur als Antrieb, sondern auch als Generator betrieben wird. Für den Fall, dass der Motor im Generatorbetrieb Energie an den Stromkreis abgibt, steigt die Spannung im System über die angelegte Spannung hinaus an. Das Ersatzschaltbild des Ankerstromkreis verdeutlicht diesen Effekt. Die Maschengleichung Gleichung 1 zeigt, wie die in den Ankerwicklungssträngen induzierte Spannung, hier als Polradspannung bezeichnet, die Masche beeinflusst [Bol18, S. 462]. Neben der Polradspannung sind zudem noch induktive und resistive Verluste im Gleichstrommotor zu finden. Der induktive Anteil sorgt durch seine differentielle Abhängigkeit vom Strom für eine unmittelbare Verkettung mit der Drehzahländerungsrate.

$$u_A = R \cdot i + \frac{di}{dt} \cdot L + u_p \quad (1)$$

Für den Fall, dass der Motor sich frei drehen kann, also keinerlei mechanischer Belastung - ausgenommen von Reibung in den Lagern und dem Getriebe-, ausgesetzt ist, stellt sich beim Anlegen einer Eingangsspannung die sogenannte Leerlaufdrehzahl ein. Da für diesen Fall der Ankerstrom sehr klein wird, und entsprechend keine großen zeitlichen Veränderungen erfährt, stellt sich also die Polradspannung nahezu gleich der Ankerspannung ein. Einzig die ohmschen Verluste sind noch zu berücksichtigen.

Anhand der angelegten Spannung und der daraus resultierenden Drehzahl kann so die Drehzahlkonstante k_n bestimmt werden [Max10]:

$$n = k_n \cdot u - \frac{\Delta n}{\Delta M} \cdot M \quad (2)$$

Über die daraus resultierende Kennlinie kann direkt über den Faktor k_n aus Gleichung 2 der momentane Motorstrom berechnet werden. Da zwischen Strom und Moment eine lineare Abhängigkeit besteht, kann hieraus wiederum der Moment und darüber die Beschleunigung der jeweiligen Achse berechnet werden. So kann alleine über den Encoder des Servomotors direkt auf die zu regelnden Größen Motorstrom, Moment und natürlich Position und Geschwindigkeit zurück geschlossen werden.

2.1.2 Thermisches Verhalten

Bei elektrischen Aktuatoren, die wechselnden Belastungen ausgesetzt sind, ist es wichtig, diese vor Beschädigung oder gar Zerstörung zu schützen. Da bei Servomotoren in Robotern unterschiedliche Bewegungen und Lastfälle möglich sind und bei der Dimensionierung der Aktorik meist neben den Anforderungen an die Leistung, oftmals auch noch Aspekte wie Kosten sowie Baugröße und Form zu berücksichtigen sind, werden Servomotoren in der Regel zumindest kurzzeitig oberhalb des Nennarbeitsbereichs betrieben. Bei einem thermischen Ersatzschaltbild ist die Spannung äquivalent zur Temperatur und der Strom zur Wärmemenge.

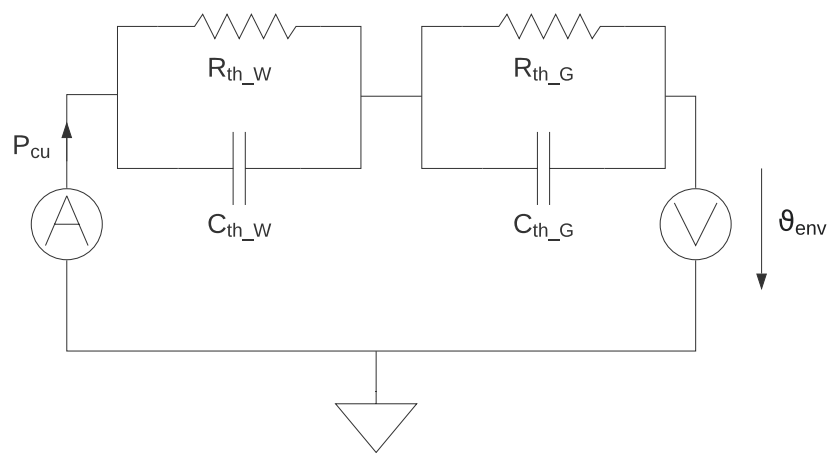


Abbildung 3: Thermisches Zweikörpermodell

Über den ermittelten Strom aus dem vorangegangenen Kapitel lässt sich hierfür die umgesetzte Verlustleistung berechnen. Diese wird zum überwiegenden Teil in Wärme umgesetzt. Über ein entsprechendes thermisches Modell kann somit die momentane Wicklungstemperatur rechnerisch bestimmt werden. Ein dafür benötigtes thermisches Ersatzschaltbild wird aus den vom Hersteller gelieferten Daten generiert. Da das Zweikörpermodell einige Vorteile bringt [NC18, S. 73] und für diese Arbeit benutzt wird, wird im Anschließendenden kurz auf dieses eingegangen.

Das in Grafik 3 dargestellte Ersatzschaltbild zeigt das thermische Verhalten eines Servomotors. Dabei sind die relevanten Servomotorkomponenten Wicklung und Gehäuse jeweils mit einem thermischen Widerstand sowie einer Kapazität dargestellt. Die zugeführte Wärmemenge P_{cu} berechnet sich nach Formel 3 aus der ohmschen Verlustleistung in den Wicklungen:

$$P_{cu} = R_{cu} \cdot i^2 \quad (3)$$

Unter Anwendung der Maschenregel lässt sich daraus die Übertragungsfunktion für das thermische System berechnen. Anhand dieser Übertragungsfunktion lässt sich somit über den zugeführten Strom die Temperatur im Inneren des Motors berechnen.

2.2 Robotik

Um Roboter bestmöglich für ihre jeweilige Aufgabe auszustatten, verfügen diese über mehrere Freiheitsgrade. Die Art und Anzahl dieser Freiheitsgrade hängt dabei stark von der jeweiligen Aufgabe ab. So können beispielsweise industrielle Schweißroboter mit ihren 6-Freiheitsgraden [Kuk20] jede Koordinate in ihrem Arbeitsbereich mit entsprechend ausgerichtetem Werkzeug anfahren.

Neben rotatorischen Gelenken sind außerdem translatorische Gelenke möglich. Da der in dieser Arbeit beschriebene Roboter ausschließlich über rotatorische Gelenke verfügt, wird sich im Weiteren ausschließlich auf diese bezogen. Der Zusammenschluss dieser Gelenke, welcher den eigentlichen Roboter ausmacht, wird auch als Manipulator bezeichnet.

Die Robotik befasst sich mit den kinematischen und dynamischen Eigenschaften dieser Manipulatoren. Außerdem ist es ihre Aufgabe, mit den daraus gewonnenen Erkenntnissen die Bewegungsabläufe zu planen.

2.2.1 Bildung einer kinematischen Kette unter Anwendung der Denavit-Hartenberg-Konventionen

Um von den einzelnen Achsstellungen auf die dazugehörige Endeffektorstellung schließen zu können, ist es notwendig, die genaue Verkettung der einzelnen Gelenke untereinander zu kennen. Hierfür wird zunächst eine Nullposition definiert, in der die Achsen keine Auslenkung erfahren. Das heißt, für die einzelnen Gelenkwinkel beträgt die Auslenkung in dieser Position 0° . Hier wird in der Regel die Ruhelage oder ein aufrechter Stand gewählt, in der der Roboter eine genau definierte Position einnimmt.

Für den gesamten Manipulator werden an jedem einzelnen Gelenk lokale Koordinatensysteme definiert, die in Matrixform die Lage und Orientierung der Gelenkkoordinatensysteme untereinander beschreiben. Durch Matrixtransformation lässt sich so die Endeffektorlage zu den jeweiligen Gelenkstellungen berechnen [Mar20a, S. 56].

Bei der Matrixtransformation nach Denavit-Hartenberg (DH) wird die Lage zweier Koordinatensysteme zueinander mit nur 4 Parametern beschrieben. Dabei ist vor allem die Lage des Ursprungs sowie der Gelenkkoordinatensysteme zueinander von großer Bedeutung. Hierbei muss auf die Einhaltung der Konventionen geachtet werden. Im Detail kann dies unter anderem in *Robotics, Vision and Control* von Peter Corke [Cor17] nachgelesen werden. Die dabei ermittelten Parameter, zu sehen in Tabelle 1, reichen somit aus, um die Transformation zweier Koordinatensysteme vollständig zu beschreiben [Mar20a, S. 90].

Setzt man die DH Transformation vom Ursprungskoordinatensystem bis zum Endeffektor fort, so ergibt sich die kinematische Kette des Manipulators. Damit lässt sich in Abhängigkeit der einzelnen Gelenkwinkel q die Lage des Endeffektors berechnen.

θ_i	Drehwinkel um die z -Achse des vorangegangenen Gliedes $i - 1$ zwischen den x -Achsen i und $i - 1$
d_i	Strecke entlang der z -Achse des vorangegangenen Gliedes $i - 1$ bis zum Schnittpunkt mit der x_i -Achse
a_i	Strecke entlang der Normalen zu z_{i-1} in x_i -Richtung bis zum Ursprung von KS_i
α_i	Winkel zwischen z_i und z_{i-1} um x_i

Tabelle 1: Bedeutung der Denavit-Hartenberg-Parameter
Quelle: [Cor17, S. 198]

Dies wird im Allgemeinen auch als vorwärts oder direkte Kinematik bezeichnet. Im Umkehrschluss gibt es die inverse Kinematik, bei der abhängig von der Endeffektorlage die einzelnen Gelenkwinkel berechnet werden können. Somit kann die Position eines Roboters vollständig beschrieben werden. Um jedoch Rückschlüsse auf das Verhalten des Manipulators im realen Betrieb ziehen zu können, muss jedoch auch das dynamische Verhalten betrachtet werden.

2.2.2 Berechnung der dynamischen Einflüsse auf die Gelenke

Für die Durchführung präziser Bewegungen ist es von großem Vorteil, die Belastung des Roboters im Betrieb zu kennen. Da es sich bei einem Roboter um ein bekanntes System handelt, können durch physikalische Zusammenhänge die inneren und äußeren Kräfte ermittelt werden. Hierzu muss für die Betrachtung der Geschwindigkeiten zunächst die Jakobimatrix berechnet werden. Bei der Jakobimatrix handelt es sich um ein lineares Gleichungssystem, bei dem die Anzahl der Zeilen der Anzahl der Bewegungsfreiheitsgrade und die Anzahl der Spalten der Anzahl der Gelenke entsprechen [Mar20a, S. 156]. Analog zur Kinematik kann auch hier von der Endeffektorgeschwindigkeit auf die einzelnen Gelenkvariablen \dot{q} und umgekehrt geschlossen werden. Um letztendlich die inneren Kräfte zu berechnen, werden bei der Betrachtung der Dynamik auch noch die Beschleunigungswerte der Gelenkvariablen \ddot{q} betrachtet. Dies geschieht unter der Zuhilfenahme der Lagrange-Gleichung [Cor17]. Die Lagrange-Gleichung, zu sehen unter Gleichung 4, definiert sich aus der Differenz von kinetischer Energie T_i und potentieller Energie V_i [Mar20a, S. 222].

$$L_i = T_i - V_i \quad (4)$$

Das aus Gleichung 4 resultierende Differentialgleichungssystem (Gleichung 5) führt letztendlich durch Ableitung zur Lagrangeschen Bewegungsgleichung (Gleichung 6) [Cor17, S. 263].

$$\frac{d}{dt} \left(\frac{\partial L_i}{\partial \dot{q}} \right) - \frac{\partial L_i}{\partial q} = \frac{d}{dt} \left(\frac{\partial T_i}{\partial \dot{q}} \right) - \frac{\partial T_i - V_i}{\partial q} \quad (5)$$

Diese bietet die Grundlage für die Simulationen, welche letztendlich für die Auslegung von Reglersystemen für Roboteraktuatoren von elementarer Bedeutung sind.

$$\tau = M(q) \cdot \ddot{q} + C(q, \dot{q}) \cdot \dot{q} + F(\dot{q}) + G(q) + J(q)^T \cdot W \quad (6)$$

Durch diese Gleichung lassen sich alle Kräfte auf sämtliche Gelenke ermitteln. τ ist ein Vektor der Breite X , wobei X die Anzahl der Gelenke darstellt. Mit q als Positionsvektor aller Gelenkwinkel, sowie \dot{q} und \ddot{q} als dessen erste und zweite Ableitung, lässt sich der aktuelle Zustand des Roboters voll beschreiben. Die Terme beschreiben, in abgebildeter Reihenfolge, die Trägheitsmomente, die Korioliskräfte, Reibung, Gravitation sowie die Krafteinwirkung auf den Endeffektor [Mar20a, S. 210].

Die Trägheitsmomente lassen sich mittels der sogenannten Massenmatrix M berechnen. Wie der Bewegungsgleichung entnommen werden kann, ist diese von der Gelenkposition q abhängig. Durch diese Abhängigkeit gestaltet sich deren Berechnung entsprechend komplex. Die Trägheitsmomente der einzelnen Manipulatorsegmente bezüglich deren Gelenkkoordinatensystem lassen sich unter der Zuhilfenahme von CAD-Programmen bestimmen. Da beispielsweise auf das Gelenk am Rumpf eines Roboters die Trägheitsmomente aller daran befestigten Segmente wirken, berechnet sich der entsprechende Matrixeintrag aus einer Funktion dieser.

Der Term für die Korioliskräfte enthält Zentrifugalkräfte sowie Korioliskräfte. Er ist von Position und Geschwindigkeit abhängig. Da der Term für $\dot{q} = 0$ verschwindet, kann er für kleine \dot{q} vernachlässigt werden [Mar20a, S. 253].

Der Reibungsterm fasst die Haftreibung, Gleitreibung sowie die viskose Reibung zusammen. Diese Kräfte bzw. Momente wirken entgegen der Bewegungsrichtung und sind von der Geschwindigkeit im Gelenk abhängig [Mar20a, S. 245]. Die Gewichtskräfte, welche auf die einzelnen Manipulatorsegmente wirken, sind im Gravitationsterm zusammengefasst. Abhängig von der Position verändert sich so der Moment, welcher auf jedes einzelne Gelenk wirkt.

Zuletzt fällt noch die Krafteinwirkung auf den Endeffektor ins Gewicht. Da dieser üblicherweise in irgendeiner Form mit seiner Umwelt agiert, kommt es auch hier zu einem Kräfteeinfluss. Dieser berechnet sich abhängig davon, welche Arbeiten der Endeffektor verrichtet.

2.3 Regelungstechnische Grundlagen

Die Regelungstechnik ist den Ingenieurwissenschaften der Automatisierungstechnik zuzuordnen. Die Aufgabe eines Reglers besteht darin, den Ist-Wert eines Systems ei-

nem vorgegebenen Soll-Wert anzupassen. Beim Unterschied von Ist- und Soll-Wert wird auch von der sogenannten Regeldifferenz oder dem Regelfehler gesprochen. Das Minimieren dieses Fehlers wird durch das Berechnen einer Stellgröße y erreicht. Diese dient wiederum als Eingangsgröße des zu regelnden Systems. Abbildung 4 zeigt den standardmäßigen Aufbau eines Regelkreises.

Geregelte Systeme können in unterschiedlicher Form auftreten. Um einige Beispiele zu nennen, seien etwa die Temperaturregelung einer Kühlanlage oder die Füllstandsregelung eines Wasserspeichers genannt. Aber auch hinter den Bewegungen eines Baukrans oder vielfach innerhalb eines Kfz., beispielsweise beim ESP oder der Abstandsregelautomatik, sind Regelungen zu finden [Trö11, S. 21].

Entsprechend kann auch bei der Lage-, beziehungsweise Positionsregelung eines Roboters auf einen intelligenten Regler nicht verzichtet werden. Für die Auswahl des richtigen Reglers erfordert es neben exakten mathematischen Berechnungen auch eine gewisse ingenieurmäßige Intuition [Trö11, S. 6]. So sind für das Einstellen der Reglerparameter unterschiedliche, teilweise gegensätzliche, Qualitätskriterien zu erfüllen. Dies führt dazu, dass es den perfekten Allzweckregler nicht gibt. Vielmehr muss für jede einzelne Regelungsaufgabe ein entsprechender optimaler Regler gefunden werden, der genau diese spezielle Aufgabe zufriedenstellend erfüllen kann. Als Anforderungen an einen Regler lassen sich Genauigkeit, Schnelligkeit, Stabilität und Robustheit definieren. Da diese beim Einstellen der Parameter des Reglers teilweise im Gegensatz zueinander stehen, muss häufig ein Kompromiss beim Erfüllen dieser Kriterien gefunden werden [Trö11, S. 208–210].

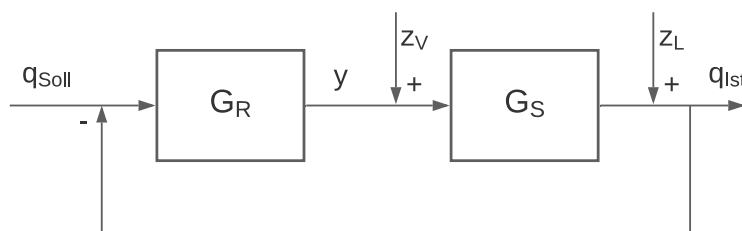


Abbildung 4: Standardregelkreis

Um diese Auswahl treffen zu können, muss die Regelstrecke genau bekannt sein. Daher ist im ersten Schritt eine Systemidentifikation der Regelstrecke durchzuführen. Abbildung 4 zeigt den einschleifigen Standardregelkreis mit der Regelstrecke G_S , dem Regler G_R sowie einer Rückführung der Ausgangsgröße q_{Ist} zum Eingang des Reglers.

2.3.1 Identifikation der Regelstrecke

Für die Auswahl des richtigen Reglers ist es, wie bereits erwähnt, von großer Bedeutung die Regelstrecke so gut wie möglich zu kennen. Um deren Parameter zu ermitteln, werden in der Praxis verschiedene Ansätze verfolgt. Ein gängiges Verfahren, ein Modell einer unbekanntenen Regelstrecke empirisch zu entwickeln, ist es, deren Sprungantwort

auszuwerten [Trö11, S. 205].

Dieser Ansatz setzt voraus, dass das zu regelnde System als solches bereits vorhanden ist. Abhängig vom Aufbau des Systems kann es zudem nötig sein, das System in einzelne Teilsysteme aufzutrennen und diese gesondert zu betrachten. Um nun für ein zu untersuchendes System die zugehörige Sprungantwort zu erhalten, muss am Eingang eine sprunghafte Veränderung der Stellgröße vorgenommen werden. Zeitgleich ist durch geeignete Messtechnik der Ausgangszustand des Systems zu erfassen. Dieses messtechnische Verfahren eignet sich sehr gut, wenn keine beziehungsweise unvollständige Informationen über das zu regelnde System vorliegen. Anhand des zeitlichen Verlaufs können dann die Parameter der Systemgleichung ermittelt werden. Systemgleichungen setzen sich aus den Gleichungen elementarer Systeme zusammen [Trö11, S. 137]. Dieses Verfahren setzt also voraus, dass der Regelungstechniker anhand der Sprungantwort das System einem, oder mehreren elementaren Systemen zuordnen kann. Als elementare Systeme sind P-, I-, D- und T-Glieder definiert. Das T-Glied ist ein Verzögerungssystem und wird daher auch Totzeit-System genannt. T-Glieder sorgen für eine zeitliche Verzögerung des eingegangenen Signals und damit für ein nicht lineares Verhalten des Systems [Trö11, S. 140]. Die Systemgleichungen der P-, I-, und D-Glieder gleichen derer der Reglerarten, welche in Kapitel 2.3.2 näher beschrieben sind.

Eine Systemgleichung aus den elementaren Systemgleichungen kann nun dazu benutzt werden, weitere mathematische Berechnungen und Simulationen des Systems anzustellen.

Ein weiterer Ansatz ein bekanntes System mathematisch zu beschreiben, ist es aus einem Modell die Gleichung herzuleiten. Dabei wird darauf zurückgegriffen, dass sich physikalische Zusammenhänge mittels Gleichungen darstellen lassen. Wird der Energiefluss vom Eingang zum Ausgang eines Systems verfolgt, so ergibt sich ein Differentialgleichungssystem. Durch Lösen dieses Gleichungssystems erhält man die Systemgleichung. Für die Erstellung des Systemmodells müssen die einzelnen Komponenten bekannt sein.

2.3.2 Welche unterschiedlichen Arten von Reglern gibt es?

Es gibt genau drei verschiedene Arten von Reglern. Um jedoch für ein jeweiliges Regelproblem einen geeigneten Regler zu finden, können diese Reglerarten nahezu beliebig kombiniert werden.

P-Regler oder auch Proportionalregler zeichnen sich dadurch aus, dass sie ein zur Reglerdifferenz $e(t)$ proportionales Ausgangssignal $y(t)$ erzeugen. Der P-Regler verstärkt so die am Reglereingang anliegende Differenz zwischen der Führungsgröße q_{soll} und der zurückgeführten Regelgröße q_{ist} um den Faktor K_P [Trö11, S. 249].

$$y(t) = K_P \cdot e(t) \quad (7)$$

Der P-Regler zeichnet sich durch moderate Regelgeschwindigkeit und seine einfache Handhabung aus. Allerdings ist er ungenau, was sich durch eine bleibende Regelabweichung zeigt [Was14].

I-Regler integrieren die Reglerdifferenz über der Zeit. Eine andauernde Abweichung der Reglergröße q_{ist} von der Führungsgröße q_{soll} sorgt für eine immer größer werdende Stellgröße $y(t)$ am Ausgang des Reglers. Mit dem Integrierbeiwert K_I kann eingestellt werden, wie stark sich der integrale Anteil auf die Stellgröße auswirkt [Trö11, S. 253].

$$y(t) = K_I \int_0^t e(t) dt \quad (8)$$

Der I-Regler kann eine bleibende Regelabweichung vollständig eliminieren. Er ist jedoch langsam im Vergleich zu den anderen Regelgliedern und kann zu schwingendem Verhalten führen [Was14]. Ein weiteres Problem, das durch den Gebrauch eines I-Reglers in einem realen System aufkommt, ist das sogenannte Windup [Ort12, S. 13]. Dieses Verhalten tritt auf, wenn die Stellgröße in irgendeiner Form beschränkt wird. Hat diese ihr Maximum erreicht und ist jedoch die Reglerdifferenz noch nicht eliminiert, so vergrößert sich das Integral des Fehlers immer weiter. „Heftige Oszillationen, ein extrem verlangsamtes Ausregelverhalten oder sogar Instabilität“ [Ort12, S. -2] sind mögliche Folgen, wenn diese Effekt außer Acht gelassen wird.

D-Regler sind differentielle Regler, die als solche in alleinstehender Form nicht vorkommen. Der D-Regler reagiert auf Änderungen in der Reglerdifferenz $\dot{e}(t)$. Verändert sich die Reglerdifferenz über der Zeit, so wird diese Veränderung mit dem Faktor K_D verstärkt als Ausgangssignal dieses Reglers erzeugt. Zeitlich konstante Abweichungen können so nicht ausgeglichen werden, weshalb ein reiner D-Regler in der Praxis keinen Sinn ergibt [Trö11, S. 255].

$$y(t) = K_D \cdot \dot{e}(t) \quad (9)$$

Der D-Anteil in einem Regler sorgt für sehr schnelles Ausregeln von Regeldifferenzänderungen. Dies kann jedoch bei besonders verrauschten Signalen gefährlich werden, da das Rauschen ebenfalls verstärkt wird. Zudem können auch große Änderungen der Eingangsgröße dazu führen, dass der Regler die Stellgröße über die Maxima hinaus berechnet. Das kann dazu führen, dass der D-Anteil gar keinen oder zumindest nicht den berechneten Effekt auf den Regelvorgang hat [Was14].

Der PID-Regler vereint alle Vorteile der oben genannten elementaren Regler. Durch das Einstellen der Reglerparameter im Arbeitspunkt können beim PID-Regler die besten Ergebnisse erzielt werden. In Abbildung 5 ist ein Vergleich gängiger Reglertypen zu sehen. Der PID-Regler sticht hier durch seine sehr schnelle Einregelzeit, die Genauigkeit sowie das nicht vorhandene Überschwingen hervor. Aus diesem Grund fällt bei der Wahl des geeigneten Reglertyps bei vielen Anwendungen die Entscheidung auf einen PID-Regler.

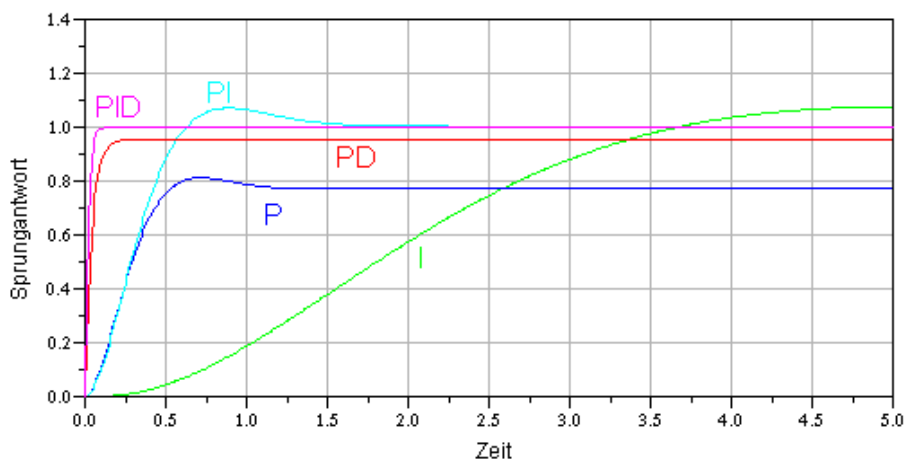


Abbildung 5: Reglervergleich

Quelle: [Was14]

Um die Reglerparameter einzustellen, wird sich üblicherweise einem der empirischen Verfahren bedient. Diese bieten ausgehend von der Sprungantwort einer P - T_2 Regelstrecke, Faustformeln für die jeweiligen Parameter. Für diese Arbeit wurde exemplarisch das Verfahren nach Ziegler-Nichols herangezogen, welches in Kapitel 4.2 näher erläutert wird [Trö11].

Sobald die Antriebe beherrscht werden, geht es darum sinnvolle Bewegungsabläufe zu ermöglichen. Hierfür liefert die Bahnplanung die notwendigen Berechnungen.

2.3.3 Bahnplanung

Für die Planung der Bewegungsbahnen eines Roboters sind in der Regel mehrere Schritte notwendig. Zunächst muss sicher gestellt werden, dass der Roboter eine gewünschte Endeffektorstellung auch auf einem realisierbaren Weg erreichen kann. Das heißt, es dürfen keine Hindernisse im Weg vorkommen, der Roboter darf nicht mit seinen eigenen Gelenken kollidieren und auf der Bewegungsbahn dürfen keine Singularitäten auftreten. Singularitäten sind Endeffektorstellungen, für die das Lösen der inversen Kinematik zu einer leeren Lösungsmenge führt.

Die Beschaffenheit des in dieser Arbeit behandelten Roboters erlaubt es diese Kriterien außer Acht zu lassen, da sie durch die Achsenbeschränkung automatisch eingehalten werden.

Neben dem kartesischen Verlauf der Bewegungsbahn, welche auch als Pfadplanung bezeichnet wird, spielt der zeitliche Verlauf in der vorliegenden Arbeit eine deutlich größere Rolle. So ist es üblich für eine definierte Bewegung entlang des geplanten Pfades mehrere Punkte anzugeben, in denen für definierte Zeitpunkte definierte Geschwindigkeiten herrschen sollen.

Die Bahnplanung soll nun diese Punkte durch eine Trajektorie miteinander verbinden. Eine Möglichkeit dies zu bewerkstelligen ist es, die Punkte mit einem Polynom zu interpolieren. Die kubische Spline-Interpolation ist ein gängiges Verfahren dies zu realisieren. Durch das Polynom 3. Grades können zwei Punkte miteinander verbunden werden und gleichzeitig die Steigungen in beiden Kurven angegeben werden. Für die Bewegung bedeutet das, dass sowohl der Zeitpunkt, an dem die Koordinate erreicht wird, als auch die Geschwindigkeit, mit der der Endeffektor diese erreicht, vorgegeben werden kann. Durch Differenzieren des Positionsverlaufs erhält man dann einen parabelförmigen Geschwindigkeitsverlauf. Durch erneutes Differenzieren ergibt sich eine Gerade, die den Verlauf der Beschleunigung beschreibt. Um die Bahn auch für den Roboter realisierbar zu machen, ist es wichtig, dass die maximal zulässige Beschleunigung nicht überschritten wird [Mar20b, S. 20].

Ein weiterer Ansatz ist der eines dreieckigen beziehungsweise trapezförmigen Geschwindigkeitsverlaufs. Hierbei kann eine zulässige Beschleunigung für die Roboterachse definiert werden. Diese gibt die Steigung des Geschwindigkeitsverlaufs an. Für den Regelfall, dass der Endeffektor zu einem definierten Zeitpunkt mit definierter Geschwindigkeit eine exakte Koordinate erreicht, ergibt dies einen trapezförmigen Geschwindigkeitsverlauf. Das Gelenk erfährt zunächst eine Beschleunigung, wird dann auf der selben Geschwindigkeit gehalten und anschließend bis zum Erreichen des Endpunktes wieder abgebremst. Für den Grenzfall, dass die Endkoordinaten schnellstmöglich erreicht werden sollen, entfällt der Bereich mit konstanter Geschwindigkeit, sodass das Gelenk zunächst maximal beschleunigt wird und im Anschluss direkt maximal abgebremst wird. Dadurch wird der zeitlich optimale Verlauf der Bewegungsbahn erreicht [Mar20b, S. 29].

3 Der Ballspielroboter Doggy2

Der für diese Arbeit aufgebaute Roboter ist die 3. Generation des ballspielenden Roboters der Arbeitsgruppe Multisensorische Systeme der Universität Bremen. Die Aufgabe dieses ballspielenden Roboters ist es, eigenständig einen zugeworfenen Ball erkennen zu können und die Flugbahn des herannahenden Balls zu berechnen. Anhand der daraus ermittelten Daten wird eine Schlagbewegung an dem berechneten Auftrittspunkt koordiniert. Ähnlich zu einem Fußballspieler soll der Roboter den Ball per Kopfball spielen können. Von seinem Vorgänger Doggy unterscheidet sich Doggy2 im Wesentlichen in den mechanischen Komponenten. Doggy entstand aus dem Projekt Piggy, welches im Gegensatz zu den drei Achsen bei Doggy lediglich über zwei steuerbare Achsen verfügte. Den Namen Doggy erhält dieser Roboter von seinem hundeähnlichen Kostüm, welches die Technik unter sich verbirgt.

Wurde Doggy noch mit Gleichstrommotoren angetrieben, die über ein Zahnriemengetriebe die Gelenke bewegten, so wurde für Doggy2 dagegen eigens ein Getriebe aus 3D-Druck Zahnrädern entwickelt, welches für eine direktere Verbindung zwischen Aktuator und Effektor sorgt. Aus diesem Grund soll die Ansteuerung deutlich einfacher zu beherrschen sein.

Zudem wird bei Doggy2 auf Gleichstrommotoren der Firma Maxon Motors zurück gegriffen. Diese zeichnen sich durch ihre patentierten eisenlosen Rotoren aus, was bei geringen Trägheitsmomenten für eine hohe Leistungsdichte im Motor sorgt [Max21].

3.1 Die mechanischen Komponenten

Die Konstruktion der Baugruppen wurde von Herrn Prof. Dr. Frese durchgeführt. Er hat bei dieser überarbeiteten Version des Doggy Roboters, die Zahnriemengetriebe der einzelnen Gelenke durch zweistufige Zahnradgetriebe ersetzt.

Die erste Getriebestufe ist bereits ab Werk an den Gleichstrommotoren montiert und liefert eine Untersetzung von 13:3. Für die zweite Stufe sorgt ein auf der Getriebewelle aufgebrachtes Zahnrad, welches die Rotation auf ein selbst konstruiertes Zahnrad mit einem Verhältnis von 84:15 überträgt. Das Zahnrad wurde mittels additivem Fertigungsverfahren aus Iglidur hergestellt. Iglidur ist ein von der Firma Igus eigens für verschleissfreie Zahnräder entwickeltes Polymer, das sich zudem sehr gut mit 3D-Druckverfahren verarbeiten lässt[igu21]. Durch die Schrägverzahnung werden zusätzliche Laufruhe sowie ein kontinuierlicher Momentenfluss ermöglicht.

Aus den beiden Stufen resultiert ein gesamtes Untersetzungsverhältnis von $24,2\bar{6}:1$, welches für die drei Gelenke äquivalent ist.

Alle drei Bewegungsachsen schneiden sich zudem in einem Punkt im Zentrum des Körpers von Doggy2. Gemäß der Denavit Hartenberg Konvention, wie sie in Kapitel 2.2.1 beschrieben wird, findet demnach lediglich eine Drehung der Koordinatenachsen statt.

Einzig die Entfernung zwischen Kopf und diesem Achsenschnittpunkt stellt hier eine Ausnahme dar.

3.2 Übersicht der Elektronik

An den elektronischen Komponenten hat sich, bis auf einen zuschaltbaren Bremswiderstand im Wesentlichen nichts verändert. Dieser dient dazu die überschüssige Energie für den Fall, dass die Motoren im Generatorbetrieb laufen, abzuführen. Somit kann der Bremsvorgang nicht durch die gegeninduzierte Spannung verzögert werden. Als Treiberkomponente wird der DRV8842 von Texas Instruments benutzt. Dieser ist ein Gleichstromtreiber mit n-Kanal MOSFETS, welche als H-Brücke angeordnet sind. Durch diese Anordnung können die 4 Transistoren so geschaltet werden, dass der Strom in beide Richtungen wirken kann.

Gesteuert werden die Motoren zentral von einem ARM Cortex-M4 Mikrokontroller mit 32Bit und Fließkommaarithmetik. Dieser enthält die Regelung sowie die Kommunikation zum Bedienrechner.

Gesteuert wird der Roboter über einen Bedienrechner, auf dem die Betriebsmodi sowie Bewegungsabläufe des Doggy2 implementiert sind. Hierfür wird das Open Source System von Robot Operating System (ROS) genutzt. ROS ermöglicht es, diverse Funktionalitäten in einzelne Pakete aufzutrennen. Zudem bietet es zahlreiche Softwarebibliotheken und Werkzeuge, die bei deren Entwicklung helfen.

3.3 Prüfstand und Grundlegende Funktionen des Roboters

Zum Zeitpunkt der Entwicklung dieser Arbeit stand der fertige Roboter noch nicht zur Verfügung. Aus diesem Grund wurde der wesentliche Teil des Projektes mittels eines einachsigen Prüfstandes entwickelt und erprobt.

Da die elektronischen Komponenten vom Vorgänger übernommen wurden, konnte an dieser Stelle bereits auf den vollen Funktionsumfang zurückgegriffen werden. Die einzelne Achse, die am Prüfstand vorhanden ist, spiegelt in etwa das 3. Gelenk des Doggy2 Roboters wider. Bestehend aus einem rotatorischen Gelenk sowie einem Verlängerungsstab zum Endeffektor lässt sich damit eine Schlagbewegung im Bereich zwischen 70° und -70° nachstellen. An Stelle des Kopfes wurde am Ende des Verlängerungsstabes eine entsprechende Masse befestigt. Zu sehen ist dies in Abbildung 6 in leicht angewinkelter Haltung.

Mit der Möglichkeit einen zweiten Motor an der zweiten Getriebestufe zu befestigen konnte zudem ein beliebiger Lastfall simuliert werden. Hierfür wird zu Beginn des Projektes der Antriebsmotor mit konstant hohem PWM-Wert betrieben. Dabei



Abbildung 6: Doggy2 Prüfstand

ist die Karbonstange, die später zum Hals von Doggy2 wird, zunächst nicht montiert, um die beweglichen Teile frei drehen zu können. Der zweite Lastmotor wurde dann in entgegengesetzter Richtung mit schrittweise erhöhten PWM-Werten bis zum maximal zulässigen Dauerstrom betrieben. Ziel dieses Verfahrens war es, eine qualitative Aussage über die Güte der mechanischen Komponenten, insbesondere des großen Zahnrades, zu treffen. Mit dem Ergebnis, dass, bis auf geringe Abriebspuren, die Stabilität der Teile zufriedenstellend ist, wurde der Test beendet. Die Abriebspuren sind auf eine leichte, fertigungsbedingte Unwucht zurückzuführen. Diese stellte aber auch bei späteren Tests kein Problem dar, womit eine erste Bestätigung der Verbesserung der Mechanik gewonnen ist.

Um die Qualität des Getriebes zu testen, wurden mit den Magnetschaltern für die Nullpositionsbestimmung eine Messreihe der Encoderwerte durchgeführt.

Bei der Messung des in Abbildung 7 dargestellten Histogramms, wurde der Schläger im Prüfstand über den Zeitraum von über einer Stunde immer vom rechten zum linken Anschlag und wieder zurück bewegt. Dabei wurden die Encoderwerte bei steigender sowie fallender Flanke des Schalters erfasst und an den Bedienrechner gesendet. Die beiden zusammengehörenden Ausschläge sind jeweils beim Anfahren beziehungsweise Verlassen des Magnetschalters auf einer Seite entstanden und verdeutlichen so sehr genau das Getriebespiel, was sich hier auf etwa 200 Ticks beläuft. Dies entspricht

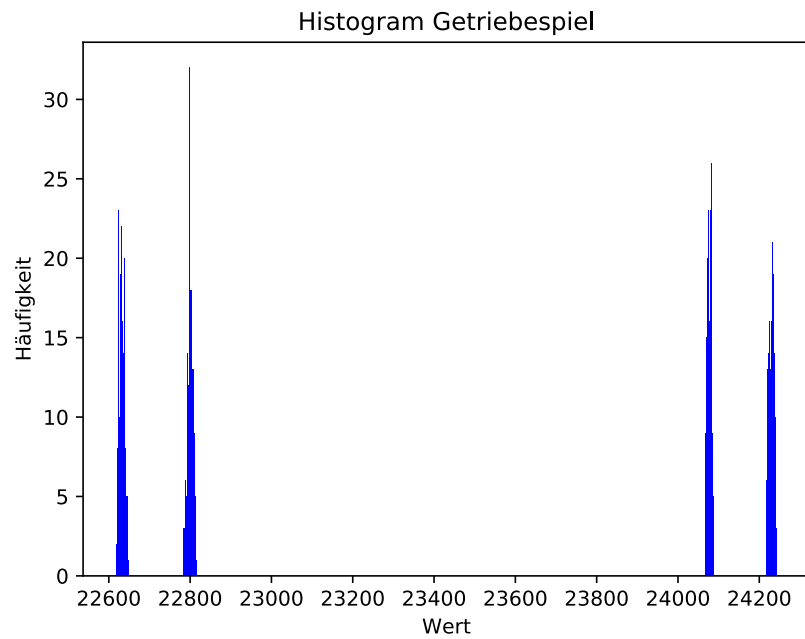


Abbildung 7: Histogram Getriebespiel

was bei einer Auflösung von $2000 \cdot 24,2666 = 48\,533$ Ticks pro Umdrehung nur etwa $1,48^\circ$. Zu erkennen ist auch, dass die Werte der Schaltpunkte des Magnetschalters über die Zeit der Messung nicht abdriften. Demnach ist im Vergleich zur ersten Doggy-Generation kein Schlupf im Getriebe zu erkennen. Das Problem des Vorgängers, welches dem Riemengetriebe geschuldet war, ist demnach auch behoben worden.

4 Reglersynthese

Der Kern dieser Arbeit befasst sich mit der Auswahl sowie der Einstellung des geeigneten Reglers. Um den Lesenden den Prozess dieser Auswahl zu verdeutlichen, wird zunächst der letztendlich nicht zielführende Entwurf eines PID Reglers beschrieben. Mit den daraus gewonnenen Erkenntnissen wurde dann in einem zweiten Ansatz der D-Anteil aus dem Regler entfernt, da dieser für sehr ungleichmäßige Bewegungen gesorgt hat. Auch die Einstellverfahren der Parameter haben sich für diese praktische Anwendung als nicht zielführend erwiesen. In einem neuen Ansatz wurde daher ein Simulationsmodell unter Berücksichtigung der technischen Möglichkeiten erstellt. Anhand dieses Modells wurden dann empirisch die geeigneten Parameter ermittelt. Nachdem die Ergebnisse der Simulation auf dem Prüfstand aus Kapitel 3.3 verifiziert werden konnten, wurde der Regelkreis noch durch eine Vorsteuerung ergänzt.

In Vorarbeit für die Auswahl des Reglers muss jedoch zunächst die Regelstrecke genauer betrachtet und identifiziert werden. Das folgende Kapitel soll die dafür erforderlichen Arbeitsschritte veranschaulichen.

4.1 Systemidentifikation der Regelstrecke

Ziel der Systemidentifikation ist es, eine Transferfunktion zu erhalten, welche die Systemeigenschaften der Regelstrecke mathematisch beschreibt. Hierfür muss der Energiefluss vom Eingang zum Ausgang der Regelstrecke betrachtet werden. Die Stellgröße des Reglers wird letztendlich die Spannung der Motortreiber sein. Diese kann mittels PWM des Mikrocontrollers variabel eingestellt werden und sorgt somit für die resultierende Positionsveränderung des Robotergelenks. Zwischen der am Motor anliegenden Spannung und der letztendlichen Gelenkposition sitzt die gesuchte Regelstrecke.

Für die elektrische Komponente dieser Strecke kann das Ersatzschaltbild des Gleichstrommotors aus Kapitel 2.1.1 betrachtet werden. Die Maschengleichung Gleichung 1 liefert den ersten Term für die Systemgleichung. Im nächsten Schritt wird der Zusammenhang des Stroms mit dem Wirkmoment des Gleichstrommotors genutzt. Gleichung 10 zeigt diesen Zusammenhang [Gui99]. Da der Strom sowie die Winkelbeschleunigung in zeitlicher Abhängigkeit stehen, gilt dieser Zusammenhang analog für deren Differentiale:

$$J_i \cdot \dot{\omega} = M = k_M \cdot i \quad (10)$$

Die Reibung wurde hier vernachlässigt. Der Hersteller des Motors und der ersten Getriebestufe macht hierzu keine Angaben. Da noch eine zweite Getriebestufe in der Konstruktion hinzugefügt wurde, ist offensichtlich, dass das Reibmoment nochmals vergrößert wird. Es wird später die Aufgabe des Reglers sein, diese Ungenauigkeit des Modells auszugleichen.

Mit den Gleichungen 1 und 10 lässt sich durch Einsetzen und Differenzieren die Systemgleichung Gleichung 11 berechnen:

$$u = L \cdot \frac{J_i}{k'_M} \cdot \ddot{w}' + R \cdot \frac{J_i}{k'_M} \cdot \dot{w}' + \frac{1}{k'_n} \cdot w' \quad (11)$$

Durch Laplacetransformation dieser Differentialgleichung ergibt sich letztendlich die Übertragungsfunktion Gleichung 12 im Bildbereich. Die Sprungantwort dieser Funktion ist in Abbildung 9 zu sehen.

$$G(s) = \frac{\Omega'(s)}{U(s)} = \frac{k'_n}{1 + \frac{R \cdot J_i \cdot k'_n}{k'_M} s + \frac{L \cdot J_i \cdot k'_n}{k'_M} s^2} \quad (12)$$

Durch Substitution der Parameter lässt sich die Transferfunktion in die Form eines P-T₂-Gliedes bringen.

Für die korrekte Interpretation der Funktion ist es wichtig zu definieren, an welchen Stellen die Parameter ihre Gültigkeit besitzen. Da für die Bewegung des Roboters weniger die Motordrehzahl von Interesse ist, sondern vielmehr die des rotatorischen Gelenks, findet diese in der Transferfunktion ihre Verwendung. Demnach berechnet sich die Gelenkwinkelgeschwindigkeit w' aus der Motordrehzahl w und dem Untersetzungsverhältnis n mit $w' = w/n$. Entsprechend müssen $k'_M = k_M \cdot n$ sowie $k'_n = k_n/n$ für die Berechnung der Größen nach dem Getriebe ebenfalls mit dem Untersetzungsverhältnis umgerechnet werden. Beide Parameter werden vom Hersteller Maxon Motors im Datenblatt angegeben [Max21].

$$J_3 = J_{Gelenk} + J_{Zahnrad} \cdot n_2^2 + J_{Motor} \cdot (n_1 \cdot n_2)^2 \quad (13)$$

Das Massenträgheitsmoment wird entsprechend der Gleichung 13 berechnet [Urs20]. Für das Motorträgheitsmoment J_{Motor} liefert Maxon Motors ebenfalls den Wert im Datenblatt [Max21]. Die beiden Trägheitsmomente des Zahnrades aus der zweiten Übersetzungsstufe n_2 $J_{Zahnrad}$ sowie des jeweiligen Gelenks J_{Gelenk} wurden aus der CAD-Konstruktion ermittelt. Das genutzte CAD-Werkzeug Autodesk Inventor bietet die Funktion diese Komponenteneigenschaft berechnen zu lassen.

4.2 Auswahl der Reglerstruktur

Nachdem die Regelstrecke R_S identifiziert wurde, wird eine geeignete Strategie für die Reglerstruktur entwickelt. Da die Regelstrecke als Ausgangsparameter die Drehzahl des Gelenks vorweist, wird noch ein Integrator benötigt, womit nun die aktuelle Gelenkposition q ausgegeben wird. Des Weiteren wird hier eine kaskadierte Regelung angewendet. Die in Abbildung 8 dargestellte Skizze veranschaulicht dies. Hierbei re-

gelt sich die äußere Reglerschleife R_A auf die Position ein und die innere R_I auf die Geschwindigkeit. Entsprechend ist es die Aufgabe der äußeren Reglerschleife eine Sollgeschwindigkeit für die innere Reglerschleife vorzugeben. Daher wird im Folgenden der Regler der äußeren Reglerschleife auch als Geschwindigkeitsregler bezeichnet.

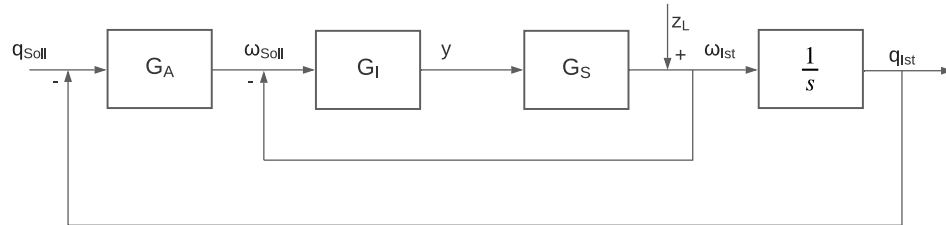


Abbildung 8: Kaskadierter Regler mit innerer und äußerer Reglerschleife

Zunächst liegt jedoch der Schwerpunkt auf der inneren Reglerschleife. Hier stellt sich grundsätzlich erst einmal die Frage, welche Größe geregelt werden soll. Technisch unumgänglich ist, dass letztendlich die Stellgröße y die Spannung am Motor ist. Analog zur äußeren Reglerschleife wäre es jedoch sinnvoll, hier direkt die Beschleunigung einzustellen. Dieses Vorhaben stößt allerdings mit dem Wissen aus Kapitel 2.2.2 auf das Problem, dass sich die Kräfteverhältnisse des Roboters abhängig von seiner Lage verändern. Demnach muss, um die Beschleunigung zu regeln, auf diese Veränderungen eingegangen werden. Ein weiteres Problem ist, dass für denselben Beschleunigungsverlauf in der einen Position viel mehr Energie benötigt wird als in einer anderen. Damit würde abhängig davon, wie der Arbeitsbereich des Reglers beschränkt wird, nicht das volle Potential der Motoren ausgenutzt, oder aber die Motoren über den spezifizierten Werten hinaus betrieben werden. Für den gewählten Ansatz wird daher das Drehmoment geregelt. Durch die lineare Abhängigkeit über den Faktor k_M zwischen Drehmoment und Strom lässt sich so der Arbeitsbereich auf den maximal zulässigen Motorstrom beschränken. Gleichung 1 liefert den benötigten Strom, wobei die Annahme besteht, dass der Strom im aktuellen Mikrocontroller-Systemtick konstant bleibt. Entsprechend fällt der induktive Term aus der Gleichung heraus. Der dabei gemachte Fehler fällt zu Gunsten der Sicherheit aus. Das heißt, dass der berechnete Strom in der Regel größer ist als der tatsächlich vorkommende. Der Regler wird diesen Fehler später kompensieren.

PID-Drehmomentsregler Im ersten Ansatz wird versucht die Regelungsaufgabe durch einen PID Regler zu lösen. Wie bereits in Kapitel 2.3.2 erwähnt wurde, dienen empirische Verfahren zur Berechnung der Reglerparameter als erster Anhaltspunkt, um für bestimmte Regelstrecken geeignete Reglerkreise zu entwerfen [Trö11, S. 338]. Hierfür wird die Sprungantwort des Systems, wie sie in Abbildung 9 zu sehen ist, genauer betrachtet. Bei dem gewählten Verfahren nach Ziegler und Nichols spielt die Ordnung

des Systems keine große Rolle. So können recht einfach komplexe Systeme mit einem geeigneten Regler versehen werden. Dabei wird die Sprungantwort des Systems analysiert und mit einem P-T₁-T_t-System approximiert. In Abbildung 9 ist die Tangente im Wendepunkt der Sprungantwort abgebildet. Am Schnittpunkt mit der X-Achse lässt sich die Zeitkonstante T_t für das Totzeitverhalten ablesen. Die Zeitkonstante T_1 ergibt sich beim Erreichen des Endwertes K_{PS} . Bei der abgebildeten normierten Sprungantwort entspricht dies dem Wert 1.

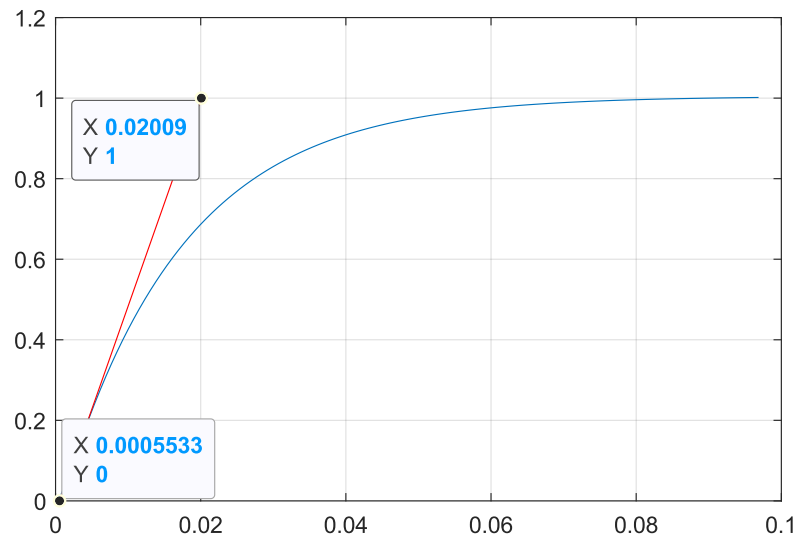


Abbildung 9: Sprungantwort der unregelten Regelstrecke mit eingezeichneter Wendepunkt tangente und Zeitparameter T_1 und T_t

Die Berechnung der Einstellparameter für den PID-Regler erfolgt gemäß der im Anhang angefügten Tabelle 3. Für die aus Abbildung 9 ermittelten Parameter ergibt das einen PID-Regler mit der in Abbildung 10 gezeigten Sprungantwort. Bei der abgebildeten Matlab-Simulation wurde der Verstärkungsparameter K_{PR} bereits angepasst, da der berechnete Wert zu großem Überschwingen führte.

Die Abbildung zeigt deutlich, wie sich das Verhalten des gesamten Systems verbessert. Der Soll-Wert wird nach wie vor asymptotisch ohne Schwingen erreicht. Jedoch ist die Einregelzeit um Faktoren besser als beim unregelten System. In einer frühen Implementierung konnte diese Art der Regelung auch auf dem Prüfstand getestet werden. Durch das in Kapitel 5.2.3 beschriebene Problem war es zu diesem Zeitpunkt leider nicht möglich, visualisierbare Daten zu generieren. Zu einem späteren Zeitpunkt wurden die Tests jedoch mit einer deutlich weniger aggressiven Version des PID-Reglers durchgeführt (Siehe auch Kapitel 6.1). Diese ließen schnell darauf schließen, dass der berechnete Regler nicht für den geforderten Zweck geeignet ist. Das Verhalten der Bewegung lässt sich als sehr ruckelig und stotternd beschreiben. Beim Halten einer Position wurde leichtes Schwingen beobachtet, sodass der Roboterarm zu keiner Zeit in der Ruheposition verharren konnte. Insgesamt war die Akrobatik zudem sehr laut, was

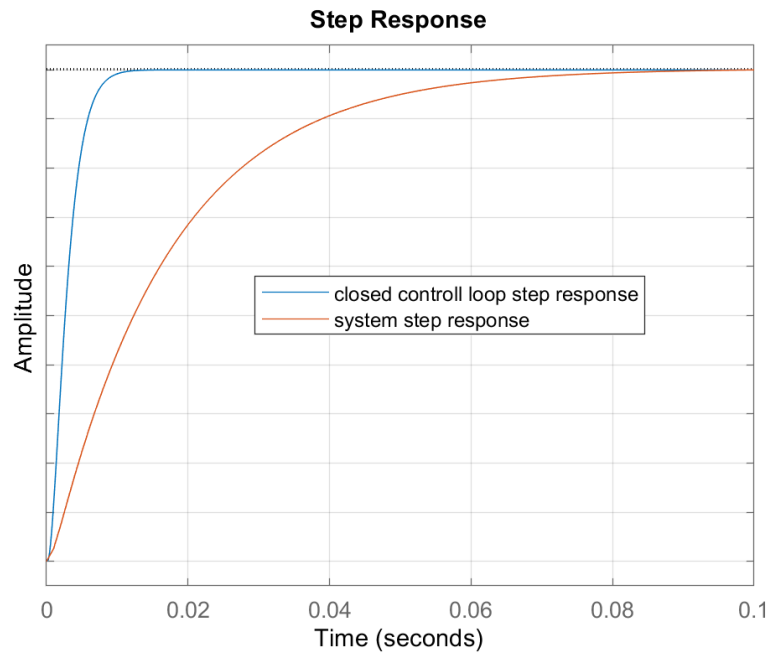


Abbildung 10: Vergleich der Sprungantwort des unregulierten Systems (rot) mit dem geschlossenen Regelkreis mit PID-Regler (blau)

auf das aggressive Verhalten des Reglers zurückzuführen ist.

Als Ursache für das schlechte Abschneiden des Reglers lässt sich schnell der Unterschied zwischen realem System und idealem System bestimmen. Bisher wurde nur eine Systemidentifikation des Motors sowie der Mechanik gemacht. Da hier neben einigen Vereinfachungen, alle Berechnungen auf physikalischen Grundlagen basieren, ist davon auszugehen, dass hier wenig Optimierungspotential liegt.

Der Regler hingegen wurde bisher als optimal angenommen. Wird allerdings genauer betrachtet, wie sich die hard- und softwareseitige Implementierung gestaltet, so wird schnell festgestellt, dass diese keinesfalls der Realität entspricht. Bei der Regelung mit einem Mikrocontroller kommt es zu einem Quantisierungsfehler [BU02, S. 17]. Neben der Wertequantisierung spielt bei der genutzten Implementierung auch die Zeitquantisierung eine große Rolle. Durch die Taktrate des Zustandsautomaten sind so lediglich 250 Reglereingriffe pro Sekunde möglich. Die Wertequantisierung spielt bei den Fließkommaoperationen eher keine Rolle. Allerdings ist durch die Auflösung des Encoders für die Bestimmung der Position sowie die daraus berechnete Geschwindigkeit eine gewisse Ungenauigkeit zu berücksichtigen. Die Auflösung von knapp 138 Ticks pro Grad sind für die Positionierung des Roboterarms zwar mehr als ausreichend, bei kleinen Geschwindigkeiten kann es an dieser Stelle aber zu signifikanten Ungenauigkeiten kommen.

Der Erkenntnisgewinn, der bei diesem ersten Lösungsansatz gemacht wurde, führt zu dem Ergebnis, dass der heuristische Ansatz für die gestellte Aufgabe keine zufrie-

denstellende Lösung liefert. Daher wurde im nächsten Schritt, bevor mit der Auslegung des Reglers fortgefahren wird, ein genaueres Modell des Gesamtsystems benötigt. Eine Möglichkeit, ganzheitliche Modelle von Robotersystemen zu erzeugen, bietet ROS. Da dieses Software Framework bereits für die Schnittstelle zum Kontrollrechner benutzt wurde, fiel die Wahl zunächst auf ein Robotermodell auf Basis von ROS. Der große Einarbeitungsaufwand, sowie einige Unklarheiten bei der Planung, führten jedoch letztendlich dazu, von dieser Lösung wieder Abstand zu nehmen.

Mit der Robotics Toolbox von Peter Corke wird eine Open-Source Toolbox für MATLAB angeboten, welche über eine sehr umfangreiche Dokumentation in Form eines Buches (Robotics, Vision and Control [Cor17]) sowie über viele Beispiele verfügt. Durch die Nutzung von MATLAB-Simulink fällt zudem der Einarbeitungsaufwand geringer aus.

4.3 Systementwurf auf Basis eines MATLAB-Simulink Models

Die Grundlage für alle Berechnungen der Robotics Toolbox bietet ein detailliertes Modell des Roboters. Hierzu sind neben den Kenndaten der Aktorik und den DH-Parametern noch die dynamischen Parameter des Roboters anzugeben.

Die Kenndaten der Aktorik sind aus dem vorangegangenen Kapitel zur Systemidentifikation bereits bekannt. Um das Robotermodell zu beschreiben wurden diese einfach übernommen. Für das Aufstellen der kinematischen Kette, unter Anwendung der DH-Konventionen, wurde wie in Kapitel 2.2.1 beschrieben vorgegangen. Da sich bei Doggy2 alle Rotationsachsen in einem Punkt schneiden, wird genau dieser Punkt auch als Ursprung des Weltkoordinatensystems genutzt. Zum Zeitpunkt dieser Arbeit stand noch nicht fest, worauf der fertige Roboter montiert werden soll. Daher erfolgte der provisorische Aufbau auf einer Holzplatte, sodass er quasi direkt auf dem Boden stehend montiert wurde. Die Höhendifferenz muss für die spätere Nutzung beim Kalibrieren der Kameras jedoch berücksichtigt werden. In Tabelle 2 sind die DH-Parameter für den Doggy2 Roboter gelistet. Aufgrund der Gelenkanordnung ergibt sich für die einzelnen Gelenkverbinder lediglich eine Rotation der Koordinatensysteme. Einzig die Verbindung zwischen Gelenk 3 und Endeffektor, also das Stück zwischen Rumpf und Kopf, weist einen Längenversatz auf.

doggy2:: 3 axis, RRR, stdDH, slowRNE

joint	θ	d	a	α	offset
1	q1	0	0	$-\frac{\pi}{2}$	0
2	q2	0	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$
3	q3	0	-0.8	0	$\frac{\pi}{2}$

tool: t = (0, 0.800, 0), RPY/xyz = (-90, 0, -90) deg

Tabelle 2: Denevit-Hartenberg Parameter des Doggy2 Robotermodells

Mit dem Aufstellen der kinematischen Kette ist die geometrische Anordnung des Roboters vollständig beschrieben. Für das Erzeugen einer real getreuen Simulation fehlen jedoch noch die Masseverhältnisse der einzelnen Manipulatorsegmente.

Durch die Nutzung des CAD-Werkzeuges Autodesk Inventor [Aut20] ist es möglich, die Massenträgheitstensenoren direkt aus der Software zu exportieren. Inventor stellt diese Parameter in den Baugruppeneigenschaften zur Verfügung. Hierbei ist jedoch zu beachten, dass die Orientierung mit den in der kinematischen Kette gewählten Gelenkkordinatensystemen übereinstimmt. Die Abweichungen wurden entsprechend angepasst.

Aus den bis zu dieser Stelle ermittelten Parametern lässt sich nun das Modell für die Simulation mit der Robotics Toolbox beschreiben. Die Toolbox liefert für das beschriebene Modell bereits einen fertigen Simulink Baustein sowie eine animierte Grafik (Abbildung 11) des Manipulators. Diese können nun in eine Gesamtsystemsimulation übernommen werden.

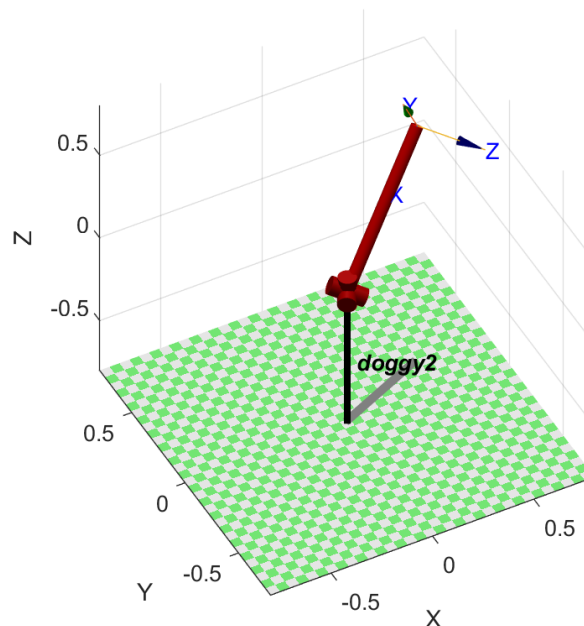


Abbildung 11: Grafische Darstellung der Matlab Simulink Simulation des Doggy2 Roboters

Eine Übersicht aller Komponenten der Simulation ist im Anhang 30 zu sehen. Hier wurde neben dem Robotermodell auch bereits die innere und äußere Reglerschleife integriert. Das Robotermodell wird eingangsseitig mit dem Gelenkmoment angesteuert. Entsprechend fehlt zwischen Regler und Robotergelenk der elektrische Teil der Motoren. Dieser wurde in der Simulation daher durch ein PT1-Übertragungsglied ergänzt. Für die Feder an den Gelenken 2 und 3, die den Roboter besser in die Ausgangsposition zurück bringen soll, wird in der Simulation zusätzlich noch ein zur Auslenkung q proportionales Moment am Eingang des Robotermodells subtrahiert. Zudem sind zahlreiche

Blöcke für die Visualisierung der Bewegungsparameter q , \dot{q} und \ddot{q} sowie der auftretenden Momente τ und den dynamischen Parametern für Gravitation und Massenträgheit vorhanden. Eingangsseitig ist neben einem Sinusgenerator noch der Robotics Toolbox Block für die Bahnplanung vorhanden. Diese Toolbox-Funktion bietet die Möglichkeit, über eine kubische Spline-Interpolation eine Bahnkurve zu berechnen. Zudem wird noch der Mikrocontroller-Takt simuliert, um dem Problem der zeitlichen Diskretisierung, welches aus Kapitel 4.2 extrahiert wurde, nachzugehen. Für die Beschränkung der Stellgröße des Reglers wurde zudem der Wert für den maximal zulässigen Strom definiert. Dieser wird später softwareseitig limitiert. Als Schwächen des Modells seien genannt, dass die Reibung sowie das Getriebespiel nicht berücksichtigt sind. Außerdem ist die Wertequantisierung der Motor-PWM nicht in der Simulation vorhanden. Anhand dieses Modells wurde nun im Folgenden die Reglerstruktur spezifiziert.

4.4 Ein kaskadierter Momentenregler mit Vorsteuerung

Für die grundlegende Reglerstruktur wurde bereits in Kapitel 4.2, wie auf Abbildung 8 skizziert, eine kaskadierte Struktur gewählt. Aus dem Kapitel 2.2 zur Robotik lässt sich die Erkenntnis ziehen, dass sich die Zustände des Roboters anhand seiner Modellparameter präzise berechnen lassen. Daher bietet sich die Zuhilfenahme einer Vorsteuerung an. Der Begriff Steuerung impliziert, dass die Vorsteuerung entgegen der Regelung keinerlei Rückführung des IST-Zustandes benötigt. Eine Steuerung zeichnet sich dadurch aus, einem System einen Soll-Wert vorzugeben allein unter der Berücksichtigung der vom Anwender spezifizierten Systemeigenschaften. Der große Vorteil der Vorsteuerung liegt darin, den Regler zu entlasten, sodass dieser lediglich kleinere Abweichungen vom Soll-Wert ausgleichen muss. Die Bahnplanung berechnet auf der Grundlage eines gegebenen Positionsverlaufs q die dafür benötigten Winkelgeschwindigkeiten \dot{q} und Beschleunigungen \ddot{q} . Zusätzlich lassen sich mittels der dynamischen Berechnungen die jeweils dafür notwendigen Gelenkmomente τ berechnen. Werden diese Informationen einer dem Regler vorgelagerten Steuerung zugeführt, so bleibt dem Regler lediglich die Abweichung zu dieser berechneten Trajektorie auszugleichen. Die Reglerdifferenz wird entsprechend dadurch minimiert und der Regler entlastet [Mar20b, S. 129]. Eine detaillierte Beschreibung der für die Vorsteuerung notwendigen Schritte findet sich in Kapitel 4.4.3.

4.4.1 Die innere Drehmomentreglerschleife

Bei der ersten Implementierung unter Verwendung des PID-Regler-Ansatzes ist vor allem das aggressive Verhalten der Aktorik negativ aufgefallen.

Eine Ursache hierfür kann der D-Anteil des Reglers sein. Der D-Anteil sorgt beispielsweise dafür, dass das Rauschen des Sensorsignals noch weiter verstärkt wird. Die Be-

schaffenheit des Encoders lässt vermuten, dass dieser Effekt in der vorliegenden Anwendung eine eher untergeordnete Rolle hatte. Größere Sprünge des Eingangssignals sind jedoch eine weitere Ursache für ein unruhiges Verhalten des D-Anteils. Diese lassen sich aufgrund der Taktung der Kommunikationsschnittstelle allerdings nicht vermeiden. Die Sprünge in der Reglerdifferenz sorgen dafür, dass der D-Anteil des Reglers die Stellgröße in die Begrenzung treibt, wodurch das Einschwingverhalten nicht wie berechnet ausfällt [Was14].

Aus dieser Erkenntnis heraus wurde im inneren Reglerkreis auf den D-Anteil verzichtet, sodass aus dem ehemaligen PID-Regler nun ein PI-Regler wurde. Der zusätzliche Benefit zur Beherrschung der Aggressivität ist die einfachere Handhabung. Durch das Wegrationalisieren des D-Anteils fällt eine unbekannte Größe aus der Reglerschleife weg, die in Abstimmung mit den anderen Parametern eingestellt werden muss.

Die Einstellung der Parameter erfolgte nun auf Basis der Ergebnisse aus der Simulation. Als erster Anhaltspunkt dienen die aus der Literatur entnommenen Werte für den Verstärkungsfaktor des Proportionalanteiles $K_v = 1$ sowie den des Integrators $K_i = 10$ [Cor17, S. 260]. Um die Simulation für die Einstellung des inneren Reglers nutzen zu können, wurde der Verstärkungsfaktor K_p der äußeren Reglerschleife auf '0' gesetzt. Somit greift dieser nicht in den Regelvorgang ein. Der Eingangssprung wurde dementsprechend nicht für den Positionssollwert sondern für die Geschwindigkeitsvorsteuerung eingestellt.

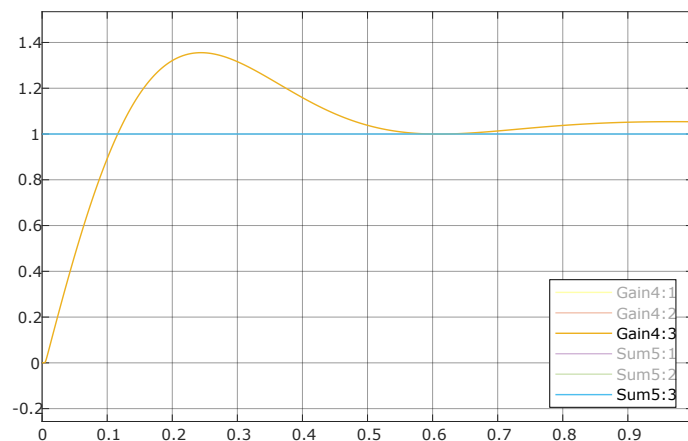
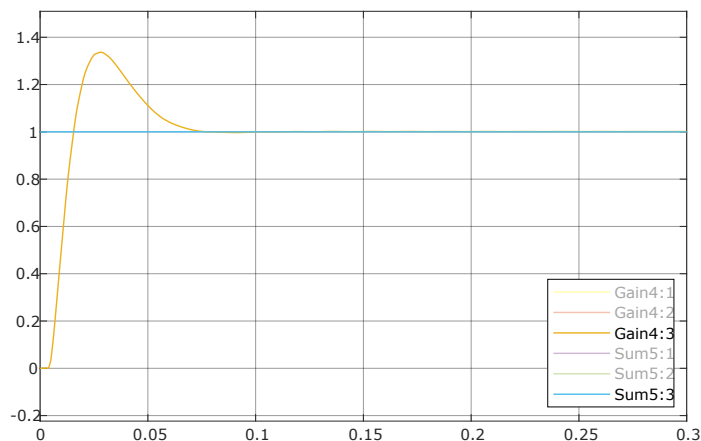


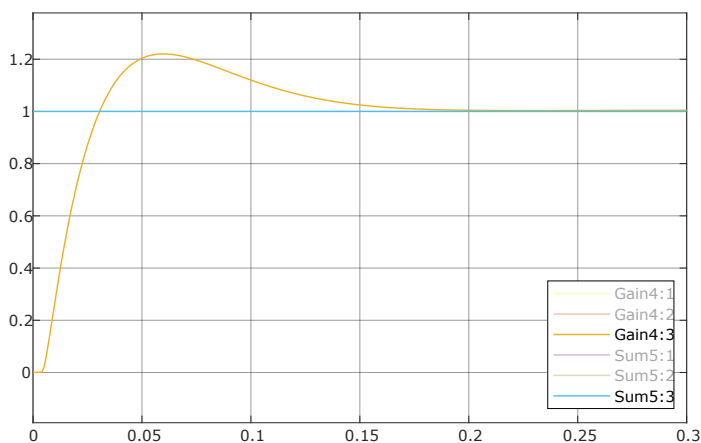
Abbildung 12: PI-Regler Sprungantwort

Zunächst soll also lediglich die Anpassung des PI Reglers anhand der Sprungantwort geschehen. Abbildung 12 zeigt diese für die Parameter aus dem Literaturbeispiel. Das Überschwingen ist typisch für einen PI-Regler und daher auch nicht gänzlich zu beseitigen. Auffällig ist jedoch, dass die Einregelzeit sehr hoch ist. Der dargestellte Bereich von einer Sekunde lässt immer noch keine Aussage darüber treffen, wann der Soll-Wert er-

reicht wird. Mit diesem Ergebnis kann der Roboter also nicht präzise gesteuert werden.



(a) $Kv = 9$ und $Ki = 400$



(b) $Kv = 5$ und $Ki = 100$

Abbildung 13: PI-Regler Sprungantwort

Um die Performanz zu verbessern, wurde zunächst der Proportionalanteil des Reglers erhöht, bis ein Schwingen bemerkbar wurde. Die Einregelzeit wurde dadurch deutlich verkürzt, jedoch schlägt jetzt die bleibende Reglerabweichung, bekannt aus dem Grundlagenkapitel 2.3.2, zu buche. Daher muss nun der Integralanteil erhöht werden. Abbildung 13 a zeigt die Sprungantwort der dadurch angenäherten Parameter. Dort ist zu sehen, dass die Geschwindigkeit nach etwa 70ms den Soll-Wert stabil erreicht hat. Der dort abgebildete Regler ist somit auch der durch Simulation ermittelte beste Regler hinsichtlich Performanz und Stabilität. Die spätere Implementierung am Prüfstand konnte diese Aussage bestätigen. Dadurch lässt sich auch die Aussage treffen, dass die Einstellung der Reglerparameter anhand der Simulation zwar deutlich mehr Aufwand erfordert als die Berechnung mittels heuristischer Verfahren, jedoch durch die realitätsnahe Umgebung deutlich zielführender ist.

Auch wenn das gemessene Ergebnis des Reglers auf den ersten Blick überzeugen

konnte, so ist das Ergebnis hinsichtlich der Laufruhe noch nicht gänzlich zufriedenstellend. Daher wurde dem Regler wieder etwas an Aggressivität entzogen und er dadurch bewusst langsamer, aber somit auch ruhiger gemacht. Der tatsächlich implementierte Regler besitzt demnach die Sprungantwort aus Abbildung 13 b.

Diese Maßnahme wird dadurch begründet, dass eine hohe Geräuschemission stets auf eine mechanische Beanspruchung zurückzuführen ist. Um die Komponenten des Roboters entsprechend zu schützen und auch das Bedienerlebnis zu erhöhen, wurde hier dieser Kompromiss eingegangen.

4.4.2 Die äußere Geschwindigkeitsreglerschleife

Für die äußere Reglerschleife wurde ein P-Regler veranschlagt. Durch die Vorsteuerung ist es die Aufgabe der äußeren Reglerschleife für den Ausgleich von minimalen Abweichungen von der Solltrajektorie zu sorgen. Der P-Regler wird somit den Positionsfehler verstärken und damit die Stellgröße des inneren Reglers liefern [Sch08, S. 104]. Wegen der unterschiedlichen Taktung des Mikrocontrollers und der Datenübertragung von der Bahnplanung wird auch hier auf einen D-Anteil verzichtet. Die bleibende Reglerabweichung wird von dem I-Anteil des inneren Reglers bewerkstelligt.

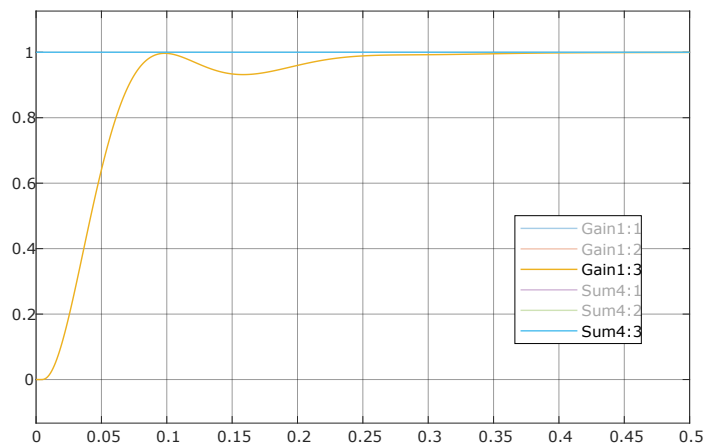


Abbildung 14: Sprungantwort der äußerden P-Reglerschleife

Die in Abbildung 14 abgebildete Sprungantwort zeigt ein leichtes Schwingen beim Erreichen des Soll-Wertes, welches jedoch bereits nach einer Periode abklingt. Die bleibende Reglerabweichung ist nach 300ms bereits nur noch minimal. Damit ist der äußere Regler etwas langsamer als der innere, was für das gewünschte Verhalten zweckmäßig ist [Trö11, S. 354].

4.4.3 Die Vorsteuerung

Wie bereits im Kapitel 4.4 angemerkt wurde, soll der Regler über eine Vorsteuerung verfügen. Bei der für die Vorsteuerung nötigen Berechnungen sind die dynamischen Einflüsse aus Kapitel 2.2.2 von elementarer Bedeutung. Die in Gleichung 6 aufgeführten

Terme liefern die zustandsabhängigen Momente für das jeweilige Gelenk. Die Robotics Toolbox liefert hierfür auf Basis des Robotermodells fertige Blöcke, mit denen diese Momente berechnet werden können.

Das Schaubild aus Abbildung 15 zeigt exemplarisch einen in der Simulation aufgezeichneten Verlauf der Massenträgheitsmomente (gelb), Korioliskräfte (blau) und der Gravitationskräfte (rot) der dritten Bewegungsachse. Bei der gemessenen Bewegung wurde eine Rotation aller drei Gelenke simuliert, um dadurch die gegenseitige Beeinflussung zu verdeutlichen. Insbesondere der Koriolisterm würde bei einer Bewegung nur einer Achse keinen Effekt haben, da dieser durch Rotation des Bezugssystems entsteht.

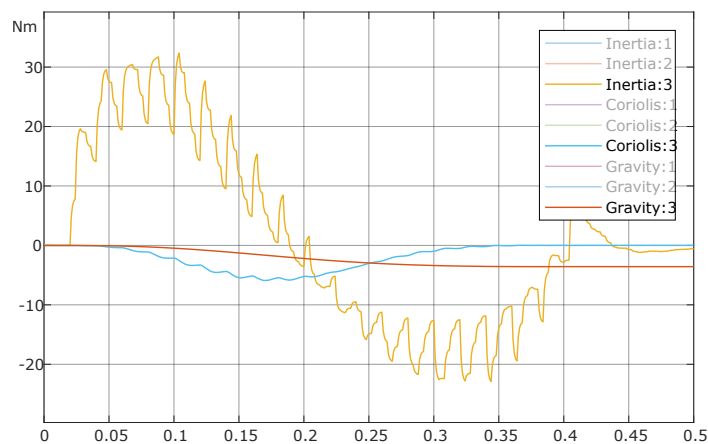


Abbildung 15: Dynamischer Momentenverlauf der dritten Bewegungsachse bei einer Rotation aller Gelenke

Die gezeigte Grafik soll dabei helfen, die Relevanz der einzelnen Terme für die Vorsteuerung zu bewerten. Auffallend ist im ersten Moment der Einfluss der Massenträgheit. Der gezackte Verlauf ist auf die unterschiedliche Taktung der Sollwertvorgabe des Reglers zurückzuführen. Da die Massenträgheit entgegen der beschleunigenden Kraft wirkt, lässt sich der Verlauf dieser Kraft, also der des Motordrehmoments, darin erkennen.

Das Gravitationsmoment verhält sich in Abhängigkeit zum Auslenkungswinkel des Gelenks. In der aufrecht stehenden Position, was einem q_3 von 0° entspricht, ist der Einfluss der Gravitation auf die dritte Achse aufgrund der symmetrischen Bauweise verschwindend gering. Bei der gezeigten Simulation beträgt der maximale Ausschlag 40° und führt zu einer Gravitationskraft von etwas über 4Nm . Maximal wäre diese Größe bei einem Ausschlag von 90° , was jedoch von dem Gelenk nicht erreicht werden kann.

Die Korioliskraft sowie die Zentrifugalkraft treten hier initiiert durch die Bewegungen der Achsen 1 und 2 auf. Die Kraft ist maximal an der Stelle, an der die Winkelgeschwindigkeit dieser beiden Achsen ihren Maximalwert besitzt. Der Vollständigkeit halber sei erwähnt, dass die Reibung sowie die Kräfteeinwirkung durch das Schlagen eines Balls hier nicht weiter berücksichtigt sind.

Für die Vorsteuerung wurde die Entscheidung getroffen, lediglich die Terme der Massenträgheit sowie der Gravitation zu berücksichtigen. Die Implementierung lässt zu, dass die Krafterwirkung des Balls zu einem späteren Zeitpunkt ohne großen Aufwand ergänzt werden kann.

Da für die Reibung keine Parameter spezifiziert sind, wird die Aufgabe, diese Einflüsse zu kompensieren, dem Regler überlassen. Sollte die Praxis zeigen, dass an dieser Stelle optimiert werden muss, so können diese Parameter experimentell bestimmt werden.

Auch die Kompensation des Koriolisterms sei zunächst dem Regler überlassen. Kapitel 2.2.2 zeigt, dass für die Berechnung ein nicht unerheblicher Aufwand betrieben werden muss. Angesichts des verhältnismäßig geringen Einflusses wird dieser ebenfalls vernachlässigt. Der Gravitationsterm schlägt betragsmäßig nicht viel mehr ins Gewicht als der Koriolisterm. Allerdings ist die Berechnung durch einfache trigonometrische Mathematik trivial. Zudem ist der Einfluss der Gravitation in Ruhelage besonders zu berücksichtigen, um den Roboter stabil in bestimmten Positionen ruhen zu lassen. Aus diesem Grund wird die Berechnung dieses Terms direkt von der Steuerelektronik bewältigt.

Das Massenträgheitsmoment soll für die Vorsteuerung vom Bedienrechnersystem berechnet werden. Dieser Berechnung liegt die im folgenden Kapitel behandelte Bahnplanung zugrunde. Die daraus resultierende Trajektorie q sowie die Beschleunigung \ddot{q} dienen als Eingangsparameter für den Massenträgheitsterm aus Gleichung 6.

Für die Massenmatrix $M(q)$ liefert die Robotics Toolbox eine eigene Funktion, die anhand der mechanischen Parameter des Robotermodells eine positionsabhängige Massenmatrix liefert. Die berechnete Massenmatrix wiederum ist eine zur Hauptdiagonalen symmetrische 3x3 Matrix. Die Dimension der Matrix ergibt sich aufgrund der drei Bewegungsachsen des Roboters. Die einzelnen Einträge der Matrix sind komplexe numerische Ausdrücke der Verkuppelungen zwischen den einzelnen Gelenksegmenten. Die Implementierung der Vorsteuerung wie sie in Kapitel 5.3 beschrieben wird, erfordert eine schnelle Berechnung der Matrixelemente. Zudem ist auch die hohe Genauigkeit der numerischen Variante nicht erforderlich, da, wie oben bereits angedeutet, der Regler die Abweichungen zwischen den von der Vorsteuerung berechneten und den tatsächlichen Verläufen der Zustandsparameter ausgleichen wird. Daher wurden die von der Robotics Toolbox berechneten komplexen Ausdrücke mit einfachen Sinus beziehungsweise Exponentialfunktionen approximiert. Den Ansatz für diese Vereinfachung liefert die Kurvenanpassung einer Sinusförmigen Funktion [Sta14]. Um ein Gespür dafür zu bekommen, wie sehr diese Vereinfachung die Berechnungsformel verkürzt, sei ein Blick auf den Term M_{11} der Massenmatrix geworfen. Die numerische Funktion liefert hierfür einen Term mit über 5650 Zeichen, der aus einer Summe von Multiplikationen und Divisionen verschiedener trigonometrischer Funktionen besteht. Die vereinfachte Form ist ein Polynom 4. Grades.

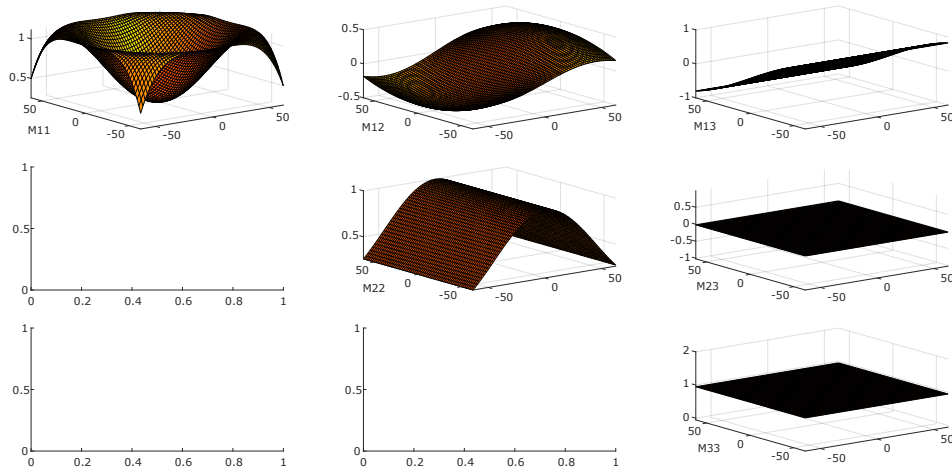


Abbildung 16: Visualisierung der vereinfachten Massenmatrixelemente unter Variation von q_2 und q_3

Aufgrund der Rotation um die senkrecht stehende z-Achse des ersten Gelenks hat eine Veränderung von q_1 keinen Einfluss auf die Massenmatrix. Daher sind die Matrixelemente lediglich von q_2 und q_3 abhängig und lassen sich somit in einem dreidimensionalen Koordinatensystem visualisieren. Abbildung 16 zeigt die bereits vereinfachten Matrixelemente. Auf einen visuellen Vergleich mit den numerischen Parametern wird an dieser Stelle verzichtet. Der einzige bemerkbare Fehler findet sich im Eintrag M_{11} , der jedoch im Arbeitsbereich der Achsen zwei und drei in den Extrempunkten unter 10% bleibt. Aufgrund der deutlich schnelleren Berechnung wird dieser Fehler jedoch hingenommen. Für die weiteren Einträge ist der Fehler vernachlässigbar klein.

Aus der somit gewonnenen vereinfachten Massenmatrix wird damit in Abhängigkeit der von der Bahnplanung bestimmten Position und der vorgegebenen Beschleunigung die Trägheitsmomentenvorsteuerung berechnet.

Die Gewichtskraft des Schlagarmes wirkt je nach Auslenkung auf die beiden Gelenke 2 und 3. Somit kann für das erste Gelenk angenommen werden, dass hierauf kein Einfluss genommen wird. Die Berechnungsformeln für die anderen beiden Gelenke können dem Funktionsblock der Robotic Toolbox entnommen werden:

$$\begin{aligned}\tau_2 &= -1,26 \cdot \cos(q_2) + 0,49 \cdot \sin(q_2) - 6,39 \cdot \cos(q_3) \cdot \sin(q_2) + 0,023 \cdot \sin(q_2) \cdot \sin(q_3) \\ \tau_3 &= -\cos(q_2) * \frac{3,49 \cdot \cos(q_3) + 976 \cdot \sin(q_3)}{6547}\end{aligned}\tag{14}$$

Die Gleichungen 14 zeigen die Formeln für den Gewichtskraftanteil der jeweiligen Ge-

lenke. Aufgrund der nicht zu den Gelenkachsen zentrierten Massenverteilung sowie der gegenseitigen Einflussnahme beider Gelenke lässt sich die mathematische Beschreibung nicht mit einfacher technischer Mechanik lösen. Die angegebenen Formeln berechnen das Gelenkmoment in Abhängigkeit des aktuellen Gelenkwinkels im Bogenmaß. Der berechnete Wert wird von dem vom Regler berechneten Soll-Moment subtrahiert, um somit den Anteil der Gewichtskraft zu kompensieren.

4.5 Planung einer Bewegungsbahn mit dreieckigem Geschwindigkeitsverlauf

Der in Kapitel 2.3.3 beschriebene Ansatz einer Spline-interpolierten Bewegungsbahn liefert bei erster Betrachtung eine optimale Lösung für das Problem der Schlagbewegung. Für relativ zeitkritische Schlagbewegungen jedoch kann es vorkommen, dass die gesamte dem System zur Verfügung stehende Leistung benötigt wird.

Abbildung 17 zeigt den Momentenverlauf eines in der Simulation durchgeführten Experimentes, bei dem der Schlagarm über Gelenk 2 aus der Nullposition auf 60° Auslenkung bewegt wird. Zu sehen ist, neben dem in grün dargestellten Verlauf der Vorsteuerung des Massenträgheitsmoments, der Verlauf des Motordrehmoments in rot. Beim Abbremsen kommt es dazu, dass das Motormoment seine Sättigung erreicht. Dies hat ein Überschwingen zur Folge und führt dazu, dass der Regler noch einmal in die andere Richtung nachjustieren muss. Somit ist die Soll-Position erst nach 650ms erreicht und nicht wie gewünscht nach 500ms. Es sei angemerkt, dass zum Zeitpunkt der Simulationsdurchführung die Feder im System noch nicht vorhanden war. Durch die der Gewichtskraft entgegenwirkende Federwirkung wird das Verhalten deutlich verbessert. Trotzdem zeigt das Experiment, wie durch einen stetigen Verlauf des Drehmoments das Potential der Aktorik nicht voll ausgeschöpft wird.

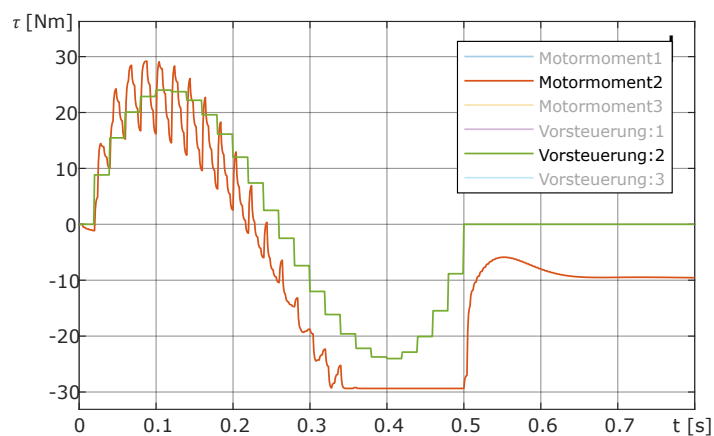


Abbildung 17: Sättigung der Drehmoments Stellgröße

Die ebenfalls in Kapitel 2.3.3 beschriebene Methode einer konstanten Beschleunigung mit einem dreiecksförmigen Geschwindigkeitsverlauf bietet hingegen eine gute

Möglichkeit, die maximale Leistung besser auszunutzen. Aus diesem Grund wurde für die Implementierung diese Art des Bahnplanungsverfahrens verwendet. Da zunächst noch nicht das Ziel besteht, einen Ball zu einem exakten Zeitpunkt zu treffen, wurde versucht, den Schläger mit der schnellstmöglichen Bewegung auf seine Endposition zu bewegen. Hierfür wird im ersten Schritt die maximal zulässige Beschleunigung definiert.

In der Simulation wurde für die minimale Verzögerung in der Sättigung des Motormoments, ein Wert von etwa $1600^\circ/\text{s}^2$ gemessen. Um dem Regler etwas Reserve zu lassen, wurde für die Tests mit einem Wert von $1200^\circ/\text{s}^2$ für \ddot{q}_{max} gerechnet. Die Abbildung 18 zeigt einen beispielhaften Verlauf einer solchen Trajektorie. Im gezeigten Beispiel startet das Robotergelenk aus der 0° -Position in der Ruhelage. Danach wird es bis 200ms konstant beschleunigt und anschließend wieder abgebremst. In der Endposition q_{end} bei 45° beträgt die Endgeschwindigkeit \dot{q}_{end} noch etwa $60^\circ/\text{s}$.

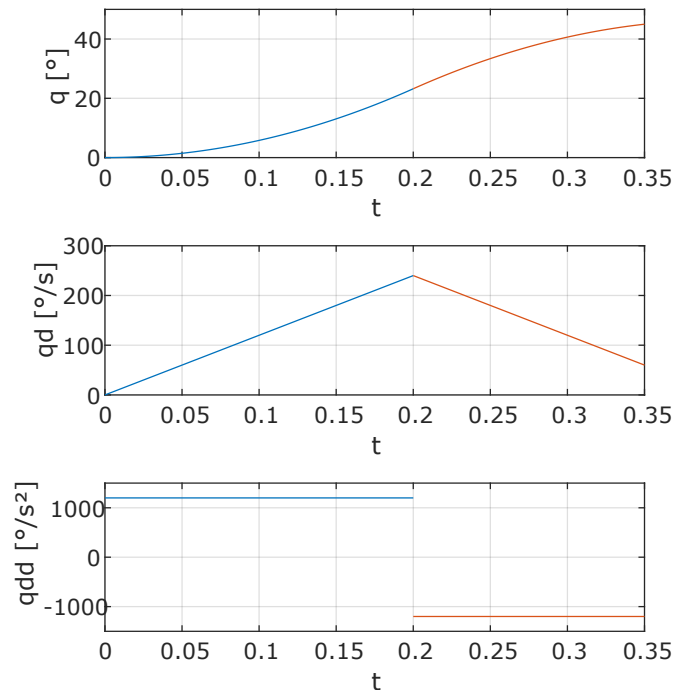


Abbildung 18: Positions-, Geschwindigkeits- und Beschleunigungsverlauf bei optimaler Trajektorie

Der Trajektorienverlauf von q im oberen Schaubild ist parabelförmig und besteht aus zwei Parabeln, die sich zum Zeitpunkt t_m berühren. Das heißt, in diesem Punkt ist die Steigung beider Parabeln gleich groß. Für die Berechnung der Trajektorie ist dieser Punkt sehr wichtig. Durch Gleichsetzen der beiden Parabelgleichungen wird der Zeitpunkt t_m ermittelt. Bei variablen Anfangs- und Endgeschwindigkeiten ist dieser Punkt dementsprechend nicht zwingend bei der Hälfte der Bewegungsbahn zu finden. Nachdem über t_m auch der Endzeitpunkt t_{end} ermittelt wurde, lässt sich für jeden Punkt

auf der Trajektorie die Position, Geschwindigkeit und Beschleunigung angeben.

Diese Berechnungen finden gelenkseitig statt. Um nun die Bewegung aller Gelenke miteinander zu synchronisieren, wird im nächsten Schritt für eine bestimmte Bewegung der errechnete Endzeitpunkt aller drei Gelenke miteinander verglichen. Der dabei maximale Wert gibt letztlich den Endzeitpunkt der gesamten Bewegung vor.

Um zum Endzeitpunkt nicht nur alle Gelenkwinkel entsprechend der Vorgabe einstellen zu können, sondern auch die Gelenkgeschwindigkeiten, ist es nötig, die Bewegungen der beiden schnelleren Gelenke zu verzögern. Hierfür wird die rechte der beiden Parabeln bis zum berechneten Endzeitpunkt verschoben. Die dadurch entstehende Lücke wird mit einer Geraden geschlossen, die tangential an beiden Parabeln anliegt. Dadurch entsteht ein Geschwindigkeitsverlauf, der einem Trapez gleicht und keine Beschleunigung im mittleren Teil erfährt.

Somit ist sichergestellt, dass neben dem Positionsvektor q auch der Geschwindigkeitsvektor \dot{q} im Endpunkt wie gewünscht dem vorgegebenen Wert entspricht.

5 Systemintegration und Benutzerinterface

Dieses Kapitel verdeutlicht, welche praktischen Arbeitsschritte benötigt wurden, um den Entwicklungsstand zum Ende dieser Arbeit zu erreichen. Dabei ist besonders hervorzuheben, dass es sich um ein ganzheitliches mechatronisches System handelt. Entsprechend der Definition von mechatronischen Systemen sind alle drei ingenieurstechnischen Disziplinen Mechanik, Elektronik sowie Informatik vertreten [HROA16, S. 13]. Dieses Kapitel stellt in eben dieser Reihenfolge dar, welche Rolle jede einzelne dieser Disziplinen für die vorliegende Arbeit gespielt hat.

Im Folgenden wird zunächst der Zusammenbau der mechanischen Komponenten erörtert. Dabei soll lediglich ein kleiner Einblick gewährt werden, wie aus dem von Herr Frese konstruierten Modell der aktuelle Stand des Roboters entstanden ist.

Der elektronische Part dient dazu, einen Einblick in die Entwicklungsphase des Regelungsteils zu gewähren. Dabei wird gezeigt, mit welchen teilweise unvorhergesehenen Herausforderungen bei der Umsetzung der Ergebnisse in die Praxis umgegangen werden musste. Da Elektronik und Software mikrocontrollerseitig nur schwer zu trennen sind, werden im Folgenden beiden Themenfeldern keine separaten Kapitel zugeordnet, sondern bei der Beschreibung der jeweiligen Komponenten die Relevanz für die Arbeit erläutert. In Kapitel 5.3 wird letztendlich die Software des Bedienrechners genauer betrachtet. Dabei wird ein Überblick über die ROS-Komponenten und deren Interaktion sowie das Benutzerinterface gegeben, welches unter Benutzung des Qt-Frameworks implementiert wurde.

5.1 Zusammenbau und Inbetriebnahme

Der Zusammenbau des eigentlichen Roboters erfolgte im Rahmen dieser Arbeit. Die Bilderstrecke unter Abbildung 19 soll einen Einblick in die Entstehung des Roboters geben. Zunächst wurden die Biegebleche mit einschraubbaren Gewindemuffen versehen. Der Einbau der Motoren sowie das Zusammenschrauben der Gelenke auf den Bildern (a) und (b) wurden zunächst ohne die Zahnräder durchgeführt, da diese sich noch im Druck befanden. Der Kopf des Roboters besteht aus einer Styroporkugel. Die Kugel ist mit einer Karbonstange am Rumpf befestigt (c). Das Verbindungsstück zwischen Stange und Kopf wurde aus Polylactide (PLA) gedruckt. Die IMU ist ebenfalls innerhalb des Kopfes an diesem Verbindungsstück montiert. Der Kopf wurde zunächst provisorisch zusammen geklebt, da später noch mechanische Komponenten für die Augen verbaut werden sollen.

Nachdem auch die Zahnräder verfügbar (d) waren, konnten alle mechanischen Komponenten zusammengefügt werden. Die meisten Komponenten wurden dabei verklebt oder verschraubt. Sämtliche beweglichen Teile besitzen wartungsarme, schmierfreie Gleitlager.

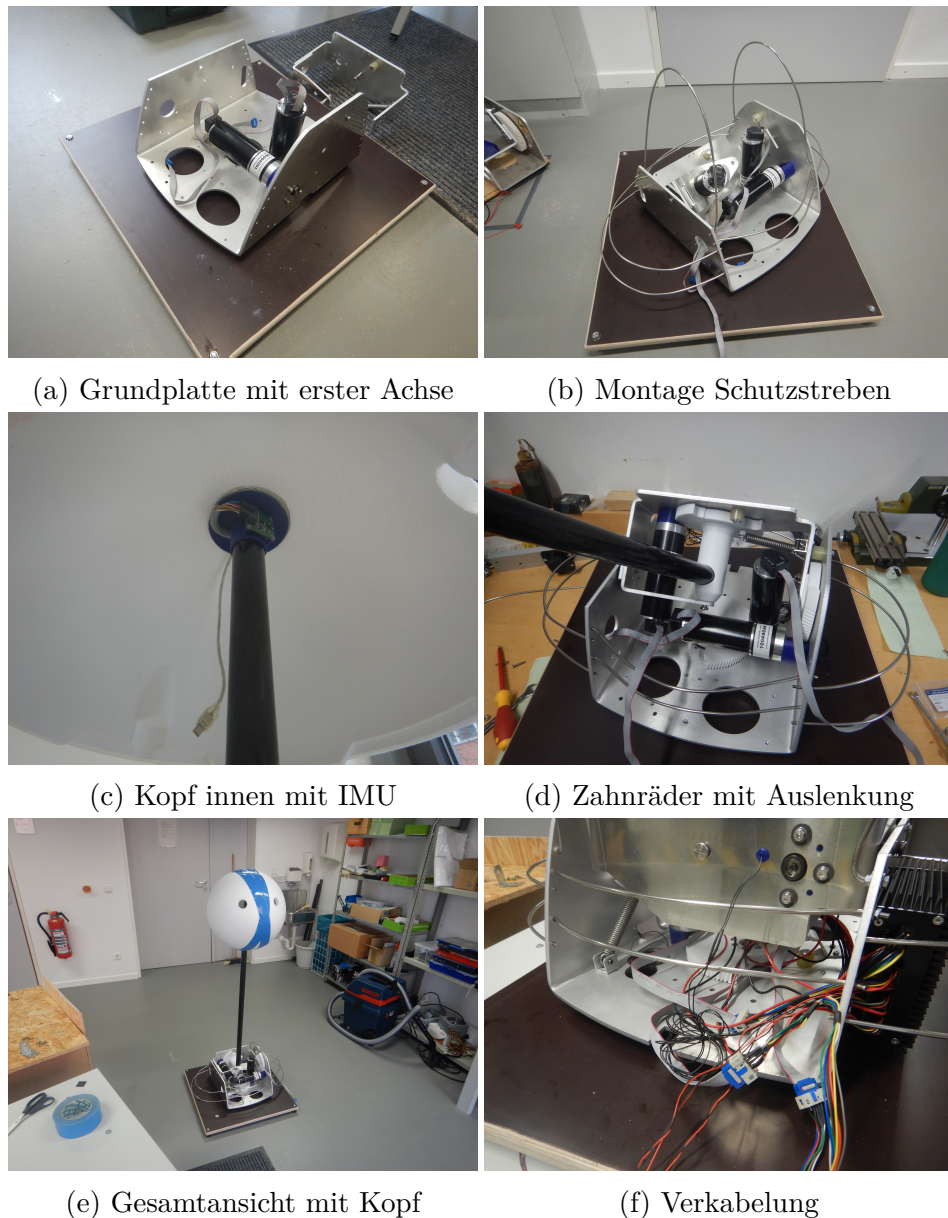


Abbildung 19: Bildstrecke Montagearbeiten

An den Gelenken 2 und 3 sind zudem Federn verbaut worden, die dabei helfen sollen, den Roboter in die Nullposition zurückzubringen. Im letzten Schritt (f) konnte dann die Elektronik verbaut sowie die Verkabelung der Motoren, Sensoren und der Versorgung mit der Steuerelektronik durchgeführt werden. Nachdem alle mechanischen und elektrischen Komponenten an ihrem Platz befestigt wurden, konnten die ersten Tests mit dem Roboter und der Regelung mit manueller Steuerung gemacht werden. Bei den ersten Bewegungen des Roboters ist bedauerlicherweise die Befestigung der Carbonstange an Gelenk 3 gebrochen. Durch die Rotationsbewegung und die träge Masse im Kopf des Roboters entstehen in Gelenk 3 an der Befestigungsstelle sehr hohe Torsionskräfte. Da die Stange zunächst nicht verklebt wurde, konnte das Verbindungsstück diesen Kräften nicht stand halten. Die Bruchstelle verläuft entlang der durch

den Druck entstandenen Schichten. Um später auch Schlagbewegungen durchführen zu können, sollte die Konstruktion an dieser Stelle massiv verstärkt oder auf andere Materialien zurückgegriffen werden. Für die Fertigstellung dieser Arbeit wurde das Verbindungsstück provisorisch repariert und die Tests daher ohne den voluminösen Kopf durchgeführt.

5.2 Die Doggy2-Elektronik

Dieses Kapitel vermittelt einen Eindruck über die Arbeitsschritte, die in Software und Hardware am Roboter benötigt wurden und nicht direkt mit der Regelung in Verbindung gebracht werden können.

5.2.1 Strombegrenzung

Um sowohl die Leistungselektronik als auch den Aktuator vor Beschädigung zu schützen, wurde der maximal zulässige Strom begrenzt. Wie bereits aus Kapitel 2.1.1 bekannt, besteht eine Abhängigkeit zwischen Spannung und Drehzahl. Um dieses lineare Verhalten bestätigen zu können, wurde die Kennlinie aus Schaubild 20 experimentell ermittelt. Für die Messung wurde schrittweise die Motorspannung U aus Gleichung 1 erhöht, um dann die daraus resultierende Drehzahl im Leerlauf zu messen. Mittels der aus dem Schaubild ermittelten Geradensteigung lässt sich so die Leerlaufspannung von der momentanen Drehzahl ableiten.

Alternativ bzw. zur Verifikation kann dieser Faktor auch aus dem Datenblatt des Gleichstrommotors [Max20] entnommen werden. Hier ist die Drehzahlkonstante mit $39,5 \text{ min}^{-1}/\text{V}$ gegeben.

Unter Berücksichtigung des Übersetzungsverhältnisses des Doggy2-Getriebes ergibt sich so eine Drehzahlkonstante von $1,63 \text{ min}^{-1}/\text{V}$. Mit der maximalen Spannung von 30V resultiert daraus eine Drehzahl von etwa $48,8 \text{ min}^{-1}$. Dieser Wert entspricht sehr genau dem experimentell ermittelten.

Gleichung 15 zeigt wie sich der Motorstrom, mittels des Gesamtwiderstands der Leistungselektronik berechnen lässt.

$$I_{Motor} = (U_{Motor} - U_{Leerlauf}) \cdot R_G \quad (15)$$

Die Differenz der angelegten Spannung und der Leerlaufspannung liefert die über den ohmschen Anteilen abfallende Spannung. Der Gesamtwiderstand berechnet sich aus der Summe des Durchlasswiderstands der H-Brücke sowie der ohmschen Verluste des Motors.

Im Umkehrschluss lässt sich so auch die maximale Spannungsdifferenz für einen gegebenen maximalen Strom berechnen. Anhand dieser Spannungsdifferenz kann nun ein gültiger Bereich für den PWM-Wert des folgenden Taktes angegeben werden. Liegt

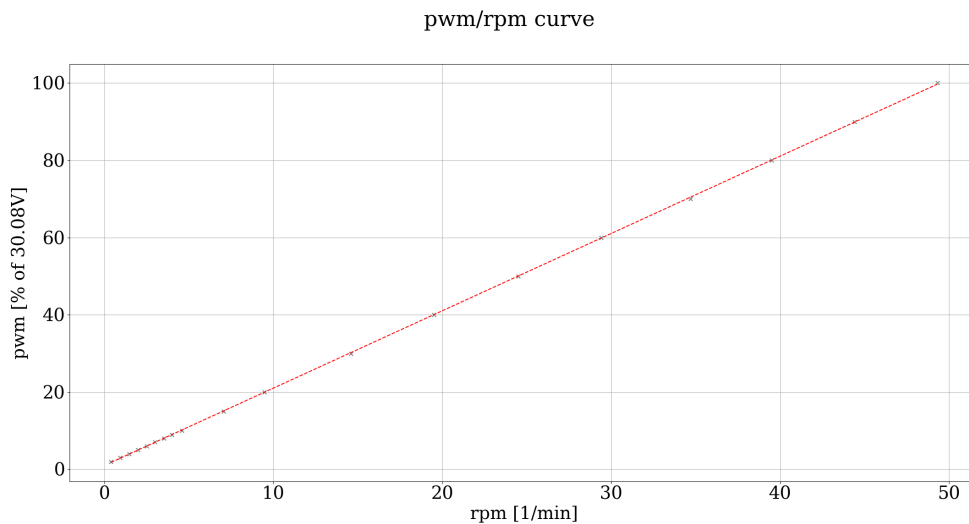


Abbildung 20: PWM/Drehzahl-Kennlinie

der vom Regler berechnete Wert außerhalb dieses Bereichs, so greift die Stellgrößenbeschränkung ein.

5.2.2 Temperaturregelung

Für die Überwachung der Windungstemperatur wurde anhand eines thermischen Modells des Gleichstrommotors eine Vereinfachung der integralen Wärmeverlustleistung gemacht. Die hierfür benötigten Informationen wurden vom Hersteller Maxon Motors im Datenblatt [Max20] bereit gestellt. Zunächst wurde anhand des in Kapitel 2.1.2 beschriebenen Verfahrens ein thermisches Zweikörpermodell entworfen. Die Analogie der thermischen Ersatzschaltbilder zu den elektrischen ermöglicht es, Simulationen mit gängigen Werkzeugen der Elektrotechnik zu erstellen.

Für die Simulation des thermischen Verhaltens des Doggy2-Antriebs wurde ein Spice-Modell erzeugt, welches mit den Parametern aus dem Motordatenblatt [Max20] parametrisiert wurde. Mittels der Software LTspice von Analog Devices (ehem. Linear Technologies) [Lin21] wurde dieses Modell simuliert und ausgewertet. LTspice bietet die Möglichkeit, elektrische Schaltungen zu entwerfen und deren zeitliches Verhalten zu analysieren. Ein- und Ausgänge können beliebig parametrisiert und an diversen Knoten innerhalb der Schaltung Messpunkte abgegriffen werden. Im Scope-Fenster lassen sich so die entsprechenden Strom- und Spannungsverläufe visualisieren.

Für die Simulation wurden zwei unterschiedliche Lastfälle untersucht: Zum einen der resultierende Temperaturverlauf, wenn der Motor dauerhaft mit dem Nennstrom von 1,89A betrieben wird, und zum Anderen der Fall, dass der Motortreiber den maximal zulässigen Überstrom von 5A liefert [Tex16].

Wie der in Grafik 21 dargestellte Verlauf zeigt, wird die maximal zulässige Wicklung-

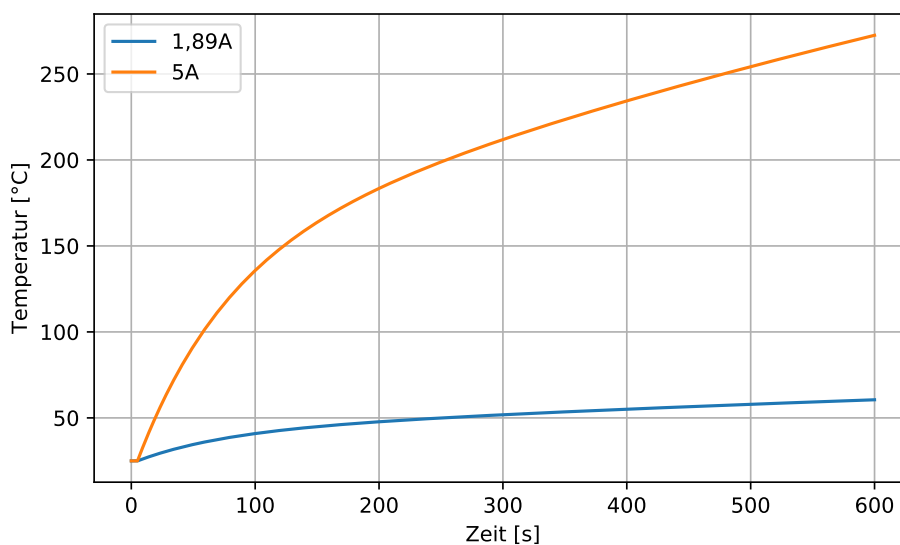


Abbildung 21: Temperaturverlauf bei Motorlast

temperatur von 125°C bei einem Betrieb mit 1,89A nicht erreicht. Für einen Betrieb mit 5A dauert es hingegen nur etwa 80 Sekunden, bis die Wicklungen zu heiß werden. Auch wenn solch eine Belastung nur im Störfall denkbar ist, so erfordert dies die Überwachung der Wicklungstemperatur. Insbesondere, da bei wiederholtem, kurzfristigem Überstromen auf längere Zeit eine Erhöhung der Temperatur gefährlich werden kann.

5.2.3 Das Kommunikationsinterface

Um dem System im Vergleich zum Vorgänger etwas Komplexität zu nehmen, wurde bei Doggy2 auf den Gebrauch der Ethernet Schnittstelle verzichtet. Die ohnehin für die Konfiguration des Doggy benötigte USB-Schnittstelle wird nun auch als Datenbus für die Steuerbefehle genutzt.

Um auch hier eine Abwärtskompatibilität gewährleisten zu können, wurde das vorhandene Interface ROS-Paket lediglich um die benötigten USB Botschaften erweitert. Da es jedoch bei der Übertragung zunächst zu einer Verzögerung auf dem Bus gekommen war, mussten Änderungen am Data Link Layer vorgenommen werden.

Die Ursache für die Verzögerung war, dass die Daten auf dem Bus erst dann gesendet wurden, wenn der Sendebuffer der rechnerseitigen USB-Schnittstelle mit 64 Bytes befüllt wurde. Zudem wurde die vom Ros-Paket eingestellte Senderate durch einen Parameter im USB-Treiber überschrieben. Dieses Problem wurde bei Untersuchungen unter Zuhilfenahme eines Oszilloskops festgestellt. Nachdem diese Probleme behoben waren, konnte die Schnittstelle wie gewünscht genutzt werden. Die Übertragung der Steuerlsignale funktioniert nach dem Request/Response-Prinzip. Ein Zustandsautomat im Steuerrechner sorgt dafür, dass bei regulärem Betrieb zyklisch Anfragen gesendet werden. Mit einer Senderate von 50 Botschaften pro Sekunde werden so die Positi-

onsanfragen alle 20ms an den Roboter gesendet. Der Roboter quittiert den Empfang dieser Botschaft mit dem Senden seiner aktuellen Koordinaten. Da der Zustandsautomat im Roboter mit 250 Zyklen pro Sekunde deutlich schneller läuft, werden die Daten maximal um 4ms verzögert an den Datenbus übergeben. An dieser Stelle sei jedoch vermerkt, dass es durch das Sendeverhalten des Busses sowie Lese- und Schreibzugriffe der jeweiligen Register zu weiteren Verzögerungen kommen kann. Diese wurden hier jedoch nicht weiter untersucht. Um später den Zeitpunkt des Schlages exakt abstimmen zu können, kann es daher nötig sein, dass der Steuerrechner diese Verzögerung in seine Berechnungen mit einfließen lassen muss.

5.2.4 Anti Windup Verfahren durch Begrenzung des I-Reglers

Beim Betrieb von Reglern in realen Systemen stößt man auf Probleme, die es im Modell zunächst nicht gibt. So besteht zum Beispiel das Problem, dass die Stellgröße in der Realität durch die technischen Eigenschaften des Systems beschränkt ist. Beim Doggy2-Roboter ist der Strom durch die Motortreiber beziehungsweise die Temperaturregelung beschränkt. Wie in Kapitel 2.3.2 beschrieben, kann es bei I-Reglern zum sogenannten Windup kommen, sobald die Regeldifferenz zu einer Überschreitung der Stellgrößenbeschränkung führt. Bei ersten Versuchen wurde der Effekt dadurch beobachtet, dass das Gelenk des Prüfstandes bei schnellen Bewegungen mit voller Wucht gegen den Endanschlag gefahren wurde. Der I-Anteil des Reglers integriert den Fehler auch dann immer weiter auf, wenn die berechnete Stellgröße schon gar nicht mehr erreicht werden kann. Dies führt zu einem Übertrag, der wiederum fehlerhaftes Verhalten des Reglers zur Folge hat.

Aus diesem Grund wurde der Wirkungsbereich des Integrators beschränkt. Sobald entweder die Strombegrenzung aktiv ist oder die maximale Spannung erreicht wurde, wird das Integral nicht weiter aufsummiert und verharrt auf dem aktuellen Wert. Dadurch wird verhindert, dass sich der Übertrag, der für den Windup-Effekt zuständig ist, bildet. Befindet sich die Stellgröße wieder im Arbeitsbereich der Aktorik, so wird der Integrator wieder aktiv. Der vorher gespeicherte Wert ist jetzt immer noch gültig, sodass die Regeldifferenz auf diesen Wert weiter aufsummiert werden kann.

5.2.5 Initialisierungssequenz

Für die Initialisierungssequenz muss sichergestellt werden, dass der Roboter für jedes Gelenk aus jeder Lage die Nullposition findet. Dies muss gewährleistet sein, obwohl die Gelenkstellung zu Beginn nicht bekannt ist. Die Gelenkposition vor der Initialisierung kann also im gesamten Bewegungsfeld des Roboters liegen.

Noch bevor es zu einer Bewegung kommt, muss der Zähler für den Encoder mit einem Startwert initialisiert werden. Da es sich um 16-Bit Zähler für Gelenk 1 und 2 sowie einen 32-Bit Zähler für Gelenk 3 handelt, wird der Initialisierungswert auf 32767 ge-

setzt. Dieser entspricht genau der Mitte einer vorzeichenlosen 16-Bit Ganzzahl. Mit dem verbauten Encoder und dem gegebenen Übersetzungsverhältnis kann jedes Gelenk aus dieser Position heraus in beide Richtungen um mehr als 243° rotieren, ohne ein Überlaufen des Zählers zu verursachen. Diese Grenze wird in keinem der drei Gelenke von Anschlag zu Anschlag überschritten.

Für die genaue Erfassung der Nullposition dient ein Zustandsautomat, der sicherstellen soll, dass die Initialisierungssequenz ordnungsgemäß durchgeführt wird. Um nun die Nullposition zu finden, muss zunächst der Magnetsensor erkannt werden. Hierfür wird das Gelenk mit konstanter Geschwindigkeit in eine Richtung bewegt. Der Positionsregler ist dabei nicht relevant, da noch keine absoluten Positionen bekannt sind. Für die Geschwindigkeit ist jedoch nur die relative Position, also die Differenz der Encoderwerte von zwei aufeinander folgenden Systemtackten, von Bedeutung. Diese Bewegung wird nun solange fortgeführt, bis entweder ein Anschlag erreicht wird - in diesem Fall beträgt die Geschwindigkeit 0 - oder der Magnetsensor erkannt wird. Für den Fall, dass die Geschwindigkeit 0 ist, wird die Richtung umgekehrt und der Zustandsautomat verharrt in diesem Zustand. Ist jedoch der Magnetsensor erkannt worden, so werden die Encoderwerte jeweils beim Abschalten in beiden Richtungen erfasst und daraus die Mittelposition berechnet. Der Grund, warum nicht einmal der Wert beim Auftreffen auf den Schalter, also beim ersten Einschalten und dann wieder beim Abschalten, erfasst wird, liegt an dem Getriebeispiel. Durch diese Maßnahme soll garantiert werden, dass egal von welcher Richtung der Magnetsensor erkannt wird, immer der selbe Nullpunkt ermittelt wird. Zudem wird die Geschwindigkeit beim Ausmessen der Schaltpunkte nochmals verringert, um zusätzliche Genauigkeit zu garantieren.

5.2.6 Problembeschreibung kaputte Elektronik

Als im Zuge der finalen Inbetriebnahme versucht wurde, den Mikrocontroller erneut mit der Software zu bespielen, wurde festgestellt, dass die Elektronik einen Defekt hat. Die Symptome waren, dass der Mikrocontroller über die JTAG-Schnittstelle nicht mehr angesteuert werden konnte. Die Vermutung liegt nahe, dass es zu einem ESD-Schaden bei der Montage der elektronischen Komponenten gekommen sein muss. Da für die Steuerelektronik kein Ersatz verfügbar ist, musste der Mikrocontroller auf der vorhandenen Platine ausgetauscht werden.

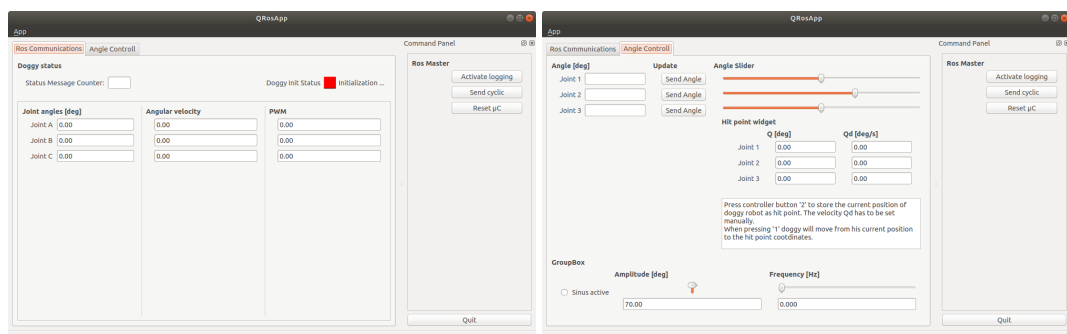
Erschwerend kam hinzu, dass der exakt gleiche Mikrocontroller leider zu dieser Zeit nicht verfügbar war und auf einen baugleichen Mikrocontroller mit dem selben Prozessor zurückgegriffen wurde. Auf die Funktionalität der Elektronik hatte diese Änderung keine Auswirkungen. Allerdings spielt die geänderte Hardware für die Software eine große Rolle. So konnte der für den alten Mikrocontroller übersetzte Binärcode nicht auf dem neuen Mikrocontroller gestartet werden.

Um die Steuerelektronik wieder nutzbar zu machen, musste unter Verwendung der

STM32CubeIDE-Entwicklungsumgebung von STMicroelectronics neuer Code für die Hardwaretreiberebene konfiguriert und generiert werden. Da sich im Vergleich zum ursprünglichen Projektcode auch die untere Treiberschicht, die vom Chiphersteller bereit gestellt wird, verändert hat, musste auch die Anbindung der bereits vorhandenen Funktionalitäten wie der Ansteuerung der Motoren, das Ablesen der Encoder sowie der USB-Schnittstelle, angepasst werden. Die beschriebenen Anpassungen wurden ebenfalls im Rahmen dieser Arbeit vorgenommen, weil sie für die Verifikation der Ergebnisse benötigt wurden. Da sie jedoch für die Arbeit an sich keine Relevanz besitzen, wird auf eine detaillierte Beschreibung verzichtet.

5.3 Das Benutzerinterface

Für die Durchführung der Reglersynthese sowie den Tests der implementierten Funktionalitäten ist es während des gesamten Entwicklungszyklus wichtig, Daten mit dem Robotersystem austauschen zu können. Für die Verbindung zwischen dem Roboter und dem Bedienrechner sorgt das Kommunikationsinterface. Um die Daten auch für den Benutzer greifbar zu machen, wurde für die Steuerung und Visualisierung dieser Daten ein Benutzerinterface in Form einer GUI entworfen. Abbildung 22 zeigt die beiden Tabs dieses Benutzerinterfaces. Im Folgenden soll ein Einblick in die Entwicklung der Bedienelemente und deren Funktionsweise gewährt werden.



(a) Bewegungsdaten

(b) Positionssteuerung

Abbildung 22: Das Benutzerinterface

Für die Entwicklung der Benutzeroberfläche wurde das Qt-Framework verwendet [Qt 21]. Qt bietet eine einfache Lösung, eine Benutzeroberfläche mit ROS zu verbinden. Das Benutzerinterface wird als ROS-Knoten vom ROS-Core verwaltet und kann ebenfalls in C++ beschrieben werden. Dadurch ist auch der Austausch der Daten mit dem Kommunikationsinterface, welches ebenfalls einen eigenen ROS-Knoten darstellt, über die ROS-internen Messages und Services gegeben.

Die Benutzeroberfläche verfügt über zwei eigenständige Tabs. Diese dienen der besseren Übersicht. Zudem ist am rechten Rand ein Menü fixiert, welches grundlegende

Funktionen wie den Reset, das zyklische Senden sowie die Steuerung der Aufzeichnungsfunktion jederzeit erreichbar macht. Auf dem Tab "ROS Communication", zu sehen in Abbildung 22 (a), sind die Bewegungsdaten abgebildet, die vom Roboter gesendet werden. Von links nach rechts werden dort die aktuelle Winkelposition, Winkelgeschwindigkeit sowie der momentane Prozentwert der PWM abgebildet. Zusätzlich kann der Status des Roboters anhand des Botschaftszählers und einer Status-LED erkannt werden. Der Botschaftszähler soll auf einen Blick erkennbar machen, wenn die Kommunikation aktiv ist. Man sieht dort einen fortlaufenden Zähler, der sich im Sendezyklus iteriert. Die Status-LED schwenkt von rot auf grün, sobald die Initialisierungssequenz des Roboters erfolgreich abgeschlossen ist, und soll den Anwendern signalisieren, dass der Roboter bereit für das Empfangen von Daten ist.

Hinter dem Tab "Angle Controll" verbirgt sich eine Auswahl an Bedienelementen, die den Anwendern verschiedene Möglichkeiten bieten, den Roboterarm zu manipulieren. Die Auswahl der Bedienelemente ist mit zunehmender Funktionalität des Roboters gewachsen. Zunächst wurde die Einstellung des Winkels über ein einfaches Eingabefeld vorgenommen. Mit dem Klick auf den "SSend AngleButton" wird der eingegebene Winkel einmalig an den Roboter gesendet. Diese Funktion steht nicht zur Verfügung, wenn das zyklische Senden aktiv ist. Um die Eingabe des Soll-Winkels zu vereinfachen, wurde zusätzlich ein Fader-Element integriert, welches zudem im zyklischen Sendemodus flüssige Bewegungen durch die Steuerung mit der Maus zulässt. Ein weiteres Feature stellt der Sinusgenerator dar. Mit dem Sinusgenerator kann eine periodische, sinusförmige Trajektorie vorgegeben werden. Die Amplitude und Frequenz kann über Fader-elemente eingestellt werden. Der Sinusgenerator wird über eine Checkbox aktiviert. Ist dieser aktiv, so dominiert er über die anderen Eingabemetoden und macht somit eine manuelle Steuerung nicht mehr möglich.

Oberhalb des Sinusgenerators befinden sich noch die Anzeige- und Eingabefelder des Trajektorienplaners. Diese können in Verbindung mit dem Gamepad-Kontroller genutzt werden. Über das Betätigen einer Taste kann die aktuelle Position des Roboters eingespeichert werden. Soll etwa an dieser Position eine Schlagbewegung durchgeführt werden, so besteht zusätzlich zu den Koordinaten die Möglichkeit zur Eingabe eines Geschwindigkeitsvektors. Wird nun der Roboterarm an eine andere Position gefahren, so kann mit dem Betätigen der Schlag-Taste der Service zur Berechnung der Trajektorie angefordert werden.

Die Berechnung der Trajektorie findet in einem separaten ROS-Knoten statt, der als Serviceprovider implementiert ist. Von der Benutzeroberfläche werden die beiden Koordinaten der aktuellen sowie der Endposition an den Bahnplanungsservice gesendet. Die Berechnung der Trajektorie zwischen diesen beiden Koordinaten erfolgt dann gemäß des in Kapitel 4.5 beschriebenen Verfahrens. Anschließend wird eine Liste mit Punk-

ten vom Bahnplanungsservice an den Klienten übermittelt. Zu den einzelnen Punkten liefert der Service zudem die dazugehörige Geschwindigkeit und Beschleunigung, die für die Vorsteuerung der kaskadierten Reglerschleife genutzt werden sollen. Auch die Berechnung der Trägheitsmomente wird vom Bahnplanungsservice übernommen und für jeden einzelnen Trajektorienpunkt an das System übermittelt.

6 Evaluation

Im Laufe der verschiedenen Entwicklungsstufen des Projektes wurden zahlreiche Messungen durchgeführt. Dadurch konnten Änderungen direkt bewertet und somit etwa die Einstellung der Reglerparameter zunächst am Prüfstands Aufbau durchgeführt werden.

Mit der Implementierung der USB-Kommunikation wurde die Möglichkeit geschaffen, ohne den Einsatz eines Debuggers einen Einblick in den Zustandsautomaten des Roboters zu gewähren. Die gesendeten Zustandsdaten wurden dann vom Bedienrechner empfangen und aufgezeichnet. Später wurden die aufgezeichneten Daten visualisiert und boten somit eine gute Möglichkeit, den vorliegenden Entwicklungsstand zu bewerten.

Dieses Kapitel soll an einigen Stellen Einblicke in den Stand der Entwicklung geben. Dadurch soll gezeigt werden, auf welcher Basis die Entscheidungen gefällt wurden und somit den Lesenden verdeutlicht werden, wie die Regelung entstanden ist.

6.1 Entwicklungsbegleitende Systemanalyse

Zunächst wurde die Implementierung des Reglers am einachsigen Prüfstand des Roboters durchgeführt. Dabei wurde der Fokus auf die korrekte Funktionsweise des Reglers an sich sowie der Strombegrenzung und der Positionsbegrenzung gelegt.

Bei einem sehr frühen Versuch wurden die Daten aus Abbildung 23 aufgezeichnet. Für die Parametrierung wurde zunächst der in Kapitel 4.2 beschriebene Ansatz implementiert und getestet. Der Regler wurde bei diesem Versuch mit einem Sprung beaufschlagt, der als Soll-Winkel in blau dargestellt ist. Ungefähr bei Sekunde 2,1 ist der Sprung von -20° auf 20° zu sehen. Der in orange dargestellte Ist-Winkel zeigt das schwingende Verhalten, welches bei dieser Implementierung der Reglerschleife als PID-Regler, wie anfangs beschrieben für Probleme, gesorgt hat. Für den Regler konnten zwar Parameter eingestellt werden, die den Sprung ohne Schwingen ausregeln, jedoch wurden die erzielten Ergebnisse wieder schlechter, sobald man die Gelenkposition änderte. Der große Arbeitsbereich, geschuldet durch die variierenden Einflüsse der Gewichtskraft, ist hier das Problem.

Weiter zeigt das Schaubild die Geschwindigkeit, den PWM-Prozentwert sowie den Reglerinput, -output und -error.

Für den Entwicklungsprozess liefert das Schaubild wichtige Erkenntnisse, die zur Verbesserung der Funktionsweise beigetragen haben. Prägnant ist an diesem Schaubild der Verlauf des Regler-Outputs. Aufgrund der hohen Differenz zwischen Soll- und Ist-

Winkel (Regler Input) stellt sich direkt ein hoher Wert für die Stellgröße ein. Am PWM-Verlauf ist leicht zu erkennen, dass die Elektronik dieser Stellgrößenvorgabe nicht folgen kann, da die Spannung bei einem PWM-Wert von 100% nicht weiter erhöht werden kann.

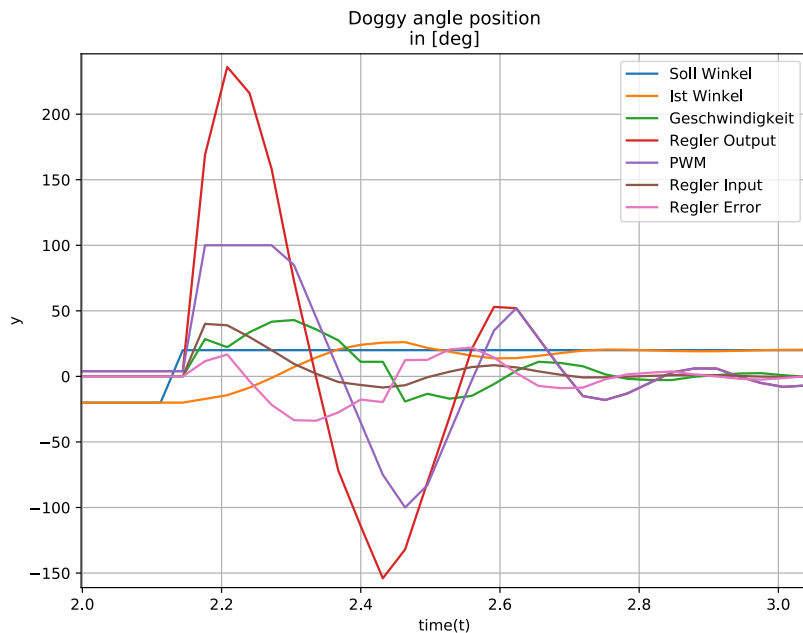


Abbildung 23: Sprungantwort des realen Systems mit implementiertem PID-Regler

Eine erste Maßnahme gegen dieses Verhalten wäre es, den Regler weniger performant auszulegen. Diese Maßnahme hätte allerdings zur Folge, dass der Regler auf sehr kleine Sprünge ebenfalls langsamer reagieren würde, was wiederum eine Verschlechterung der Ergebnisse nach sich zieht. Durch die Beschränkung der Stellgröße auf den maximal möglichen PWM-Wert wird dieses Problem gelöst, ohne eine Verschlechterung der Reglerperformanz zu erhalten. Für die Proportional- und Derivativateile spielt diese Beschränkung keine größere Rolle. Der Integralanteil jedoch integriert die Regeldifferenz immer weiter auf und sorgt dadurch für ein fehlerhaftes Verhalten, das sich durch deutliches Überschwingen oder sogar Instabilität bemerkbar macht.

An dieser Stelle kommt das anti-windup-Verfahren aus Kapitel 5.2.4 zum Tragen. Dabei beharrt der Integrator des PID-Reglers auf dem selben Wert, solange die Stellgröße außerhalb des Arbeitsbereichs der Elektronik ist.

6.1.1 PI-Regler

Aufgrund der in der Simulation gewonnenen Erkenntnisse wurde nach den ersten Versuchen mit einer PID-Reglerstruktur der D-Anteil auf 0 gesetzt. Dadurch wurde dem Regler Komplexität entzogen. Um die optimale Parameterkonfiguration für den Prüf-

stand zu finden, wurden zahlreiche Schleifen durchlaufen, bei denen einzelne Parameter verändert wurden, um anschließend den Effekt zu testen. Dadurch wurde iterativ die optimale Konfiguration für den Prüfstand gesucht. Durch die Reduzierung der Reglerparameter wurde diese Arbeit erleichtert. Da sich später am fertigen Roboter im Vergleich zum Prüfstand die Last auf den einzelnen Gelenken noch einmal ändern wird, muss dieser Schritt wiederholt werden. Die Verringerung der Komplexität ist somit auch an dieser Stelle ein Zugewinn.

Abbildung 24 zeigt erneut die Sprungantwort des Prüfstandes, hier jedoch mit implementiertem PI-Regler in der inneren Reglerschleife. Der Ist-Winkel nähert sich erst schnell, dann etwas langsamer dem Soll-Winkel an. Daraus lässt sich schließen, dass der Regler an dieser Stelle etwas aggressiver eingestellt werden kann. Da beim Sprung auf 20° die Gewichtskraft noch nicht ihren maximalen Effekt hat, wird es hier geduldet, dass sich der Regler etwas langsamer verhält als theoretisch möglich.

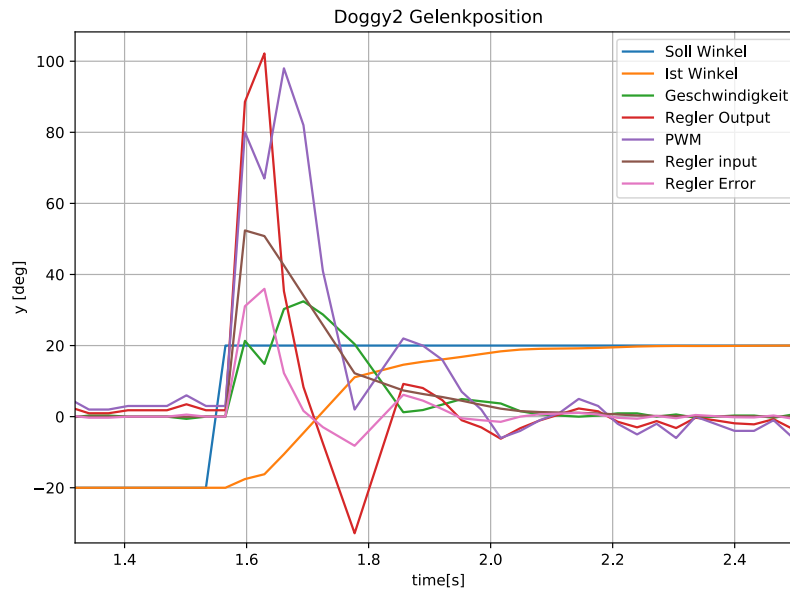


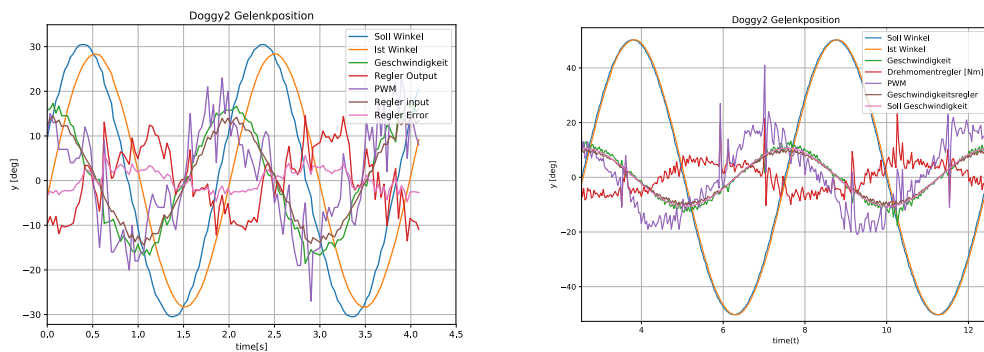
Abbildung 24: Sprungantwort des realen Systems mit implementiertem PI-Regler

Der Verlauf des Regler Inputs, in Abbildung 24 braun dargestellt, zeigt, wie der Positionsfehler ausgeglichen wird. Dieser springt am Anfang und nähert sich nach etwa 0,5 Sekunden dem Nullwert an. Der in rosa dargestellte Geschwindigkeitsfehler weist ein leichtes Schwingen auf, wird jedoch deutlich schneller ausgeglichen als der Positionsfehler. Die innere Reglerschleife ist also deutlich schneller als die äußere. Dies ist, wie in Kapitel 4.4.2 beschrieben, Voraussetzung für ein fehlerfreies Funktionieren des kaskadierten Reglers.

6.1.2 Geschwindigkeitsvorsteuerung

Die Auswertung der Sprungantworten einer Positionsänderung haben ergeben, dass die Soll-Position sich sehr langsam dem Soll-Wert anpasst. Beim Versuch, ein periodisches Oszillieren des Gelenkwinkels nachzustellen, zeigt sich, dass der gewünschte Positionsverlauf durchaus erreicht werden kann. In Abbildung 25 (a) ist gut zu sehen, dass der Verlauf des tatsächlichen Winkels dem des Soll-Winkels mit einem zeitlichen Versatz in etwa entspricht. Auch der abgebildete PWM-Verlauf bleibt deutlich unter dem 100%-Wert, was darauf schließen lässt, dass die Aktorik deutlich unter der Belastungsgrenze bleibt.

Aus der Messung wurde eine zeitliche Verzögerung des Winkelverlaufs von etwa 140ms ermittelt. Diese muss deutlich verringert werden, um die Durchführung eines präzise gesteuerten Schlages zu ermöglichen. Hierfür soll die in Kapitel 4.4.3 beschriebene Vorsteuerung sorgen.



(a) ohne Geschwindigkeitsvorsteuerung (b) mit Geschwindigkeitsvorsteuerung

Abbildung 25: Periodische Gelenkwinkeländerung

Durch den deutlich schneller agierenden inneren Reglerkreis soll erreicht werden, dass der Verlauf der Gelenkstellung durch die Vorgabe der geforderten Winkelgeschwindigkeit deutlich präziser erreicht wird. Für die sinusförmige Bewegung lässt sich diese Winkelgeschwindigkeit durch die Ableitung des Soll-Winkels einfach bestimmen. In Abbildung 25 (b) ist diese Soll-Geschwindigkeit in rosa dargestellt. Die in grün dargestellte tatsächliche Geschwindigkeit zeigt, wie gut der Vorsteuerung Folge geleistet wird. Die äußere Reglerschleife dient nach wie vor dazu, Abweichungen vom Soll-Winkel auszugleichen. In braun ist die Differenz aus Vorsteuerung, äußerer Reglerschleife sowie der gemessenen Geschwindigkeit zu sehen. Dieser Wert ist der Soll-Wert für die innere Reglerschleife.

Die Stellgröße am Reglerausgang (in rot dargestellt) entspricht bei diesem Stand der Implementierung dem Drehmoment im Gelenk. Dieser wird letztendlich in die tatsächliche Stellgröße PWM (lila) mit den aus Kapitel 5.2.1 bekannten Formalitäten umgerechnet.

Um die Verbesserung der Vorsteuerung bewerten zu können, wurde auch bei der Messung in Abbildung 25 (b) der Phasenversatz abgelesen. Mit 33,9ms kann der mit dem bloßen Auge erkennbare Fortschritt auch beziffert werden.

Bedenkt man den über 20ms großen Versatz aufgrund der Latenz beim Übermitteln der Daten zwischen Bedienrechner und Steuerelektronik, ist dieser Wert für die Durchführung einer präzisen Bewegung angemessen.

6.2 Entwicklungsstand zum Projektende

Für die abschließende Bewertung der Regelung wurde eine Reihe von Tests mit dem zusammengebauten Roboter durchgeführt. Für die Einordnung der Ergebnisse ist es von großer Bedeutung, dass bei der Durchführung der Tests der voluminöse Kopf, sowie das Kostüm des Roboters nicht angebracht waren. In Kapitel 5.1 wurde beschrieben, dass bei ersten Bewegungen die Halterung für die Karbonstange gebrochen ist. Um zu verhindern, dass eine weitere Beschädigung des Roboters den Projektabschluss verzögert, wurden die Versuche daher lediglich mit der Karbonstange am Gelenk befestigt durchgeführt. Für die Überprüfung der Bahnplanung wurden zunächst einige Schläge aus verschiedenen Positionen gemacht, bei denen die Trajektorien aller Gelenke aufgezeichnet wurden. Dabei wurde der Roboter mit manueller Steuerung mittels Gamepad an beliebige Positionen gebracht, um von dort aus über die Schlagen-Taste die Trajektorienplanung zu aktivieren und den Schlag auszuführen.

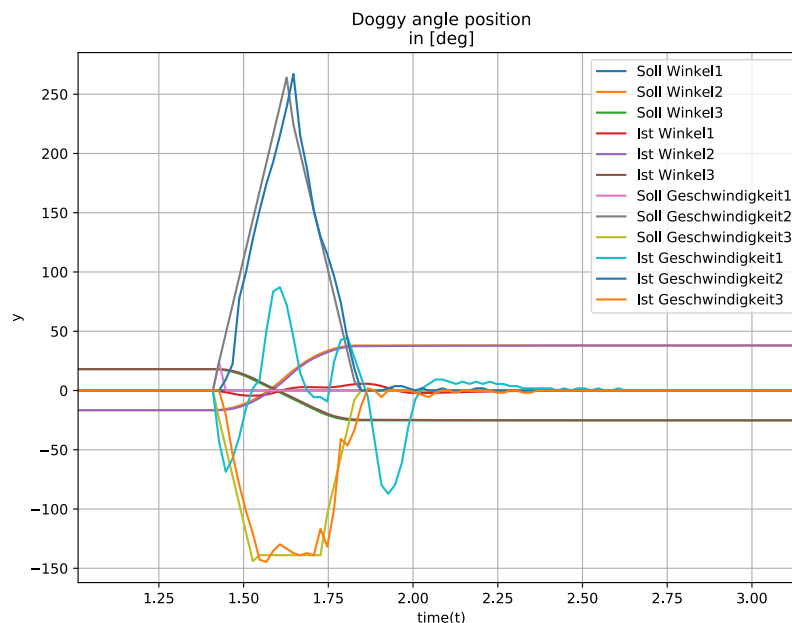


Abbildung 26: Bewegung der Achsen 2 und 3 mit starr bleibender Achse 1

Abbildung 26 zeigt die aufgezeichneten Daten bei einer Bewegung der Gelenke 2 und 3 mit berechneter Trajektorie. Dem Schaubild können zu allen drei Gelenken die Soll-

und Ist-Werte der Position und der Geschwindigkeiten entnommen werden. Anhand dieses Schaubildes kann gut die Funktionsweise der Bahnplanung entnommen werden. Für Gelenk 2 ist ein dreieckiger Geschwindigkeitsverlauf (grau) berechnet worden. Das dritte Gelenk dagegen weist einen trapezförmigen Geschwindigkeitsverlauf (gelb) auf. Gelenk 1 hingegen sollte laut Bahnplanung auf seiner Position verharren. Der dazugehörige Ist-Geschwindigkeitsverlauf (türkis) zeigt jedoch, dass hier sehr deutliche Bewegungen zu erkennen sind. Die Ursache dafür liegt in der Momentenvorsteuerung. Da die Versuche lediglich mit dem Hals des Roboters und ohne dessen massigen Kopf durchgeführt wurden, kommt es zu diesem Fehler. Die berechnete Massenträgheit übertrifft die tatsächliche deutlich. Somit ist der Moment, der eingestellt wird, um diese zu kompensieren, viel größer als benötigt. Dadurch führt dieser zu einer Beschleunigung des Gelenks. Diese Bewegung kann erst relativ spät vom Regler kompensiert werden, da die Momentenvorsteuerung direkt auf die Stellgröße am Reglerausgang beaufschlagt wird. Der Regler kann also erst dann eingreifen, wenn es tatsächlich zur Bewegung kommt.

6.2.1 Wiederholgenauigkeit

Bei einer weiteren Versuchsreihe wurde die Wiederholgenauigkeit der Schläge überprüft. Hierfür wurde an der Spitze der Karbonstange ein Filzstift befestigt. Der Endpunkt der Bewegungsbahn wurde dann auf einem Blatt Papier vor dem Roboter definiert.



(a) Winkelbeschleunigung $1200^\circ/s^2$

(b) Winkelbeschleunigung $200^\circ/s^2$

Abbildung 27: Ergebnisse der Versuchsreihe zur Schlaggenauigkeit

Erneut wurde der Roboter dann mittels Gamepad an beliebige Positionen gesteuert, um von dort aus mit dem Betätigen der Schlagtaste auf der berechneten Trajektorie den Punkt auf dem Papier zu erreichen.

Die beiden Fotos unter Abbildung 27 zeigen die Ergebnisse zweier Versuchsreihen. Bei der ersten Versuchsreihe unter Abbildung 27 (a) wurde der Schlagarm mit einer Winkelbeschleunigung von $1200^\circ/s^2$ bewegt. Damit kann innerhalb einer halben Sekunde

eine Winkeldifferenz von mindestens 75° erreicht werden, was für das Schlagen des Balls ausreichend ist. Wie die Aufnahme zeigt, kann hierbei eine breite Streuung des Schlagpunktes beobachtet werden. Entgegen erster Befürchtungen liegt die Ursache für diese Streuung nicht in der Ungenauigkeit, mit der die berechnete Bahn abgefahren wird. Durch das starke Abbremsen gerät die Karbonstange in Schwingung. Dieses Schwingen sorgt für die beobachtete Streuung.

Um eine Aussage über die Wiederholgenauigkeit der Schlagbewegung treffen zu können, wurde daher eine weitere Messreihe mit deutlich reduzierter Beschleunigung durchgeführt. Auf Abbildung 27 (b) ist das Ergebnis dieser Messreihe zu sehen. Unter den selben Bedingungen wie bei der ersten Versuchsreihe wurde hier der Schlag aus diversen Positionen wiederholt. Die Aufnahme zeigt, dass die Streuung deutlich geringer ausfällt als bei den Versuchen mit hoher Beschleunigung. Daraus lässt sich zusammenfassen, dass der Roboter durchaus in der Lage ist, eine definierte Bewegung reproduzierbar durchzuführen. Eine Zusammenfassung dieser Versuche ist unter [Doggy2_Wiederholgenauigkeit.mp4](#) zu sehen.

Das Problem des schwingenden Endeffektors wird im Kapitel Fazit nochmals aufgegriffen. Die auf Abbildung 27 (a) gemessene Verteilung der Punkte liegt, bis auf einige Extremwerte, innerhalb eines 5cm breiten Feldes. Bei einer Stangenlänge von 0,8m entspricht das einer Ungenauigkeit von etwa $3,5^\circ$. Dies kann zu Problemen bei der Treffsicherheit führen. Durch die etwas längere Stange und die höhere Masse des späteren Kopfes von Doggy2 wird dieser Effekt zusätzlich verstärkt.

6.2.2 Schlagbewegung

Im nächsten Schritt wurde überprüft, ob die Dynamik des Systems dazu ausreicht, dem Ball einen ausreichenden Impuls zu geben, um die Schlagbewegung durchzuführen. Zunächst wurde angedacht, den Ball von einer Halterung in eine Art Tor zu schlagen, um so zugleich die Zielgenauigkeit des Roboters zu überprüfen. Aufgrund des nur provisorischen Schlägers, stellt es sich jedoch schwer dar, den Ball mit einer geeigneten Halterung in eine beliebige Richtung zu schlagen. Der Schläger eignet sich lediglich dazu, den Ball nach unten zu schlagen, wenn sich die Halterung vor dem Roboter befindet. Hierfür müsste eine deutlich aufwändigere Halterung konstruiert werden.

Aus diesem Grund wurde für die ersten Schläge der Ball frei schwingend an einer Schnur befestigt.

Abbildung 28 zeigt den Versuchsaufbau, bei dem der Ball direkt vor dem Roboter an einer Schnur befestigt von der Decke hängt. Zudem ist eine Zusammenfassung dieser Versuche als Video unter [Doggy2_Schlag.mp4](#) zu sehen. Für diesen Versuch wurde der Schlagpunkt direkt an den Berührungspunkt des Balls gesetzt. Mit dem Betätigen der Schlagtaste kann der Ball von jeder Position aus an dieser Stelle geschlagen werden. Aus der Nullposition heraus findet die Schlagbewegung fast ausschließlich über Gelenk

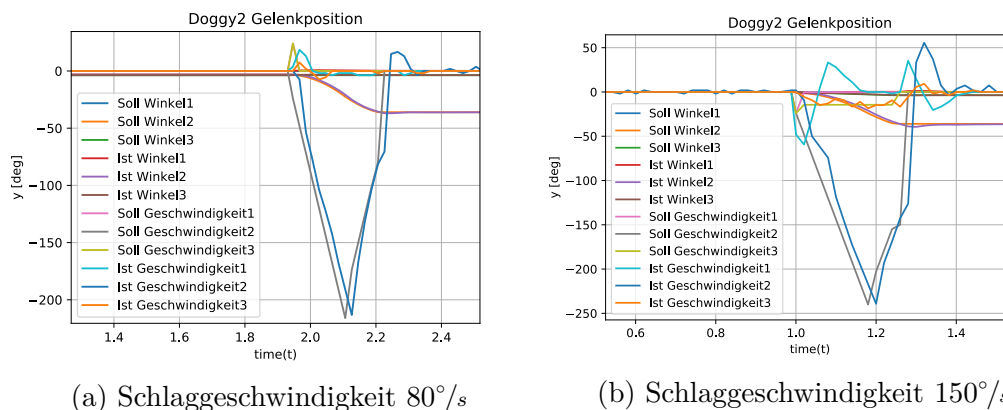


Abbildung 28: Schlagversuch mit Ball an der Schnur

2 statt. Zunächst wurde die Geschwindigkeit in diesem Punkt noch mit $0^\circ/s$ definiert. Der Schläger bewegt sich dann auf direktem Weg an die Schlagposition. Der Ball wird dabei jedoch nicht geschlagen, sondern nur leicht berührt. Bei einigen Bewegungen mit sehr spitzem Winkel, bei denen der Großteil der Bewegung von Gelenk 1 absolviert werden muss, wird der Ball seitlich geschlagen und beginnt zu rotieren.

Um dem Ball einen Schlagimpuls zu verschaffen, wurde im nächsten Schritt für Gelenk 2 eine Geschwindigkeit von $-80^\circ/s$ im Schlagpunkt definiert. Dies ist bereits ausreichend, um den Ball aus der Ruhelage heraus gegen die Wand zu schlagen. Der ursprüngliche Ansatz, den Impuls anhand der Auslenkung zu berechnen, wurde verworfen, da dieser mit der Masse des Kopfes deutlich größer wird und somit die Aussage dieser Größe nicht direkt auf den fertigen Roboter übertragbar ist. Die auf Abbildung 29 (a) abgebildeten Soll- und Ist-Werte geben einen Einblick in die genauen Abläufe der Bewegung. Die Bewegungen der Gelenkwinkel 1 und 3 sind dabei marginal. Der Schlag erfolgt aus der aufrechten Position in eine 35° Neigung des zweiten Gelenks. Dabei wird die geplante Geschwindigkeit im Schlagpunkt in etwa erreicht. Die Genauigkeit der Darstellung gerät hier an ihre Grenzen, da die Daten lediglich im 20ms-Takt aktualisiert werden. Dennoch lässt sich daraus erkennen, dass der tatsächliche Verlauf der Bewegung im Bereich der vorgegebenen Bahn liegt.

Auch ein leichtes Überschwingen des Ist-Winkels ist in Gelenk 2 zu sehen. Dies ist so gewollt, da die Geschwindigkeit beim Erreichen der Endposition nicht 0 beträgt. Die Bahnplanung ist in dem Moment beendet, in dem das Gelenk die Schlagposition erreicht hat. Das Abbremsen und Zurückregeln auf die Soll-Position erfolgt dann komplett über die Reglerschleife ohne Vorsteuerung von Geschwindigkeiten und Momenten. Abbildung 29 (b) zeigt eine Wiederholung des Versuches mit erhöhter Endgeschwindigkeit. Beim Auftreffen auf den Ball mit einer Winkelgeschwindigkeit von $-150^\circ/s$ wird



(a) Schlaggeschwindigkeit 80°/s

(b) Schlaggeschwindigkeit 150°/s

Abbildung 29: Schlagbewegung mit variierender Endgeschwindigkeit im Schlagpunkt

dieser bereits mit großer Wucht gegen die Wand geschlagen. Die aufgezeichneten Daten lassen erkennen, dass der Soll-Winkel hier mit deutlich höherer Geschwindigkeit erreicht wird. Auch die maximal erreichte Geschwindigkeit ist entsprechend höher im Vergleich zu der in Abbildung 29 (a) dargestellten. Insgesamt vergehen vom Berechnen der Bewegungsbahn bis zum Erreichen der Schlagposition knapp 260ms, was für die Durchführung des Schlages ein brauchbares Ergebnis ist.

7 Fazit und Ausblick

Die vorliegende Arbeit zeigt den gesamten Verlauf des Entwicklungsprozesses der Ansteuerung und Regelung der Akrorik des ballspielenden Roboters Doggy2. Hierfür wurde zunächst anhand eines Prototypenprüfstandes die Regelung für eine Achse implementiert. Dabei wurde ein heuristischer Ansatz zur Bestimmung der Reglerparameter gewählt. Nachdem erste Tests ergeben haben, dass dieser Ansatz für die gestellte Aufgabe nicht zielführend ist, fand der Entwurf der Regelung, parallel zur Implementierung im Prüfstand, in einer MATLAB-Simulink-Simulation statt. Dabei wurde unter Verwendung der Robotics-Toolbox von Peter Corke [Cor17] ein Modell des Roboters erstellt. Aufgabe des Modells war es, das Verhalten des Roboters so realitätsnah wie erforderlich nachzubilden, sodass der Regler in einer virtuellen Umgebung eingestellt werden kann.

Neben der eigentlichen Aufgabe wurde Doggy2 zudem mit einer umfangreichen Anwendersoftware ausgestattet, die es den Benutzern ermöglicht, den Roboter frei zu bewegen und somit die implementierte Regelung zu testen. Da der Roboter im Zuge dieser Arbeit auch gefertigt und montiert wurde, konnten gegen Projektende die ersten Tests mit Teilen der finalen Hardware durchgeführt werden.

Bei diesen Tests sowie der entwicklungsbegleitenden Analyse konnte verifiziert werden, dass das ausgearbeitete Konzept für die gestellte Aufgabe funktioniert. In der finalen Version kommt ein kaskadierter Regler mit Vorsteuerung zum Einsatz. Der Regler verfügt über eine innere sowie eine äußeren Reglerschleife. Die innere Reglerschleife ist als PI-Regler realisiert und bekommt neben der Geschwindigkeitsvorsteuerung die Stellgröße der äußeren Reglerschleife vorgegeben. Bei der äußeren Reglerschleife handelt es sich lediglich um einen Proportionalregler, der Abweichungen von der Soll-Position in eine Geschwindigkeitsvorgabe umrechnet.

Das von der Reglerkaskade berechnete Moment wird auf die Summe der aus der Vorsteuerung berechneten Trägheitsmomente sowie der durch die Gewichtskraft verursachten Momente aufaddiert. Über die pulsweitenmodulierte Motoransteuerung wird dieses berechnete Moment auf die einzelnen Gelenke übertragen.

Für die Bahnplanung der Vorsteuerung wurde eine Variante gewählt, bei der die maximale Beschleunigung ausgenutzt wird. Die Bahnplanung berechnet dazu für das Gelenk mit dem längsten Verfahrensweg einen dreieckigen Verlauf der Geschwindigkeit. Für die anderen Gelenke wird entsprechend ein trapezförmiger Geschwindigkeitsverlauf berechnet, sodass alle Gelenke zum selben Zeitpunkt die Endstellung erreichen.

Um zum vorliegenden Ergebnis zu gelangen, mussten neben der eigentlichen Implementierung noch weitere Herausforderungen bewerkstelligt werden. So wurde beispielsweise die Funktionsweise der Datenübertragung an die neuen Anforderungen angepasst.

Die besondere Herausforderung dabei war es, stets Kompatibilität zu der ersten Generation des Ballspielroboters Doggy zu gewährleisten.

Viel Zeit hat zudem die Implementierung der Treiber in Anspruch genommen. Nachdem die elektronischen Komponenten unerwartet beschädigt wurden, mussten für den neuen Mikrocontroller sämtliche Treiber der genutzten Peripherie neu implementiert werden.

Die anfangs gestellte Aufgabe, die Regelung für die Gelenkansteuerung des ballspielenden Roboters zu synthetisieren, wurde damit erfüllt. Auch wenn es dem Roboter zum jetzigen Zeitpunkt aufgrund des nicht montierten Kopfes und der fehlenden Erkennungssoftware nicht möglich ist an einem Ballspiel teilzunehmen, wurde gezeigt, dass bereits dynamische Bewegungen der neu entwickelten Mechanik möglich sind. Durch die Funktion der Bahnplanung wird der Endeffektor auf dem schnellstmöglichen Weg zwischen zwei frei definierbaren Punkten bewegt.

Die eigens für die Verifikation implementierte Bediensoftware ermöglicht es den Benutzern, den Roboter mittels Gamepad-Controller zu steuern. Dadurch kann bereits ein erster Einblick in die Interaktion mit dem Roboter geboten werden. Für zukünftige Entwicklungsschritte liefert diese Software zudem einen guten Einstiegspunkt in die Arbeit mit Doggy2.

Die nächsten Schritte in diesem Projekt werden es sein, die finalen Tests mit dem montierten Kopf nochmals zu wiederholen. Die Versuche zur Wiederholgenauigkeit haben gezeigt, dass der Endeffektor Vibrationen ausgesetzt ist und dadurch in Schwingung gerät. Das Erhöhen der Last am Ende der Karbonstange verstärkt diesen Effekt noch zusätzlich. Dies kann dazu führen, dass die Feineinstellung der Reglerparameter nochmals durchgeführt werden muss. Dazu empfiehlt es sich, die Sprungantworten der einzelnen Gelenke analog zu Kapitel 4.4.1 auszuwerten und anhand dieser die Parameter zu optimieren. Sollte sich das Verhalten dadurch nicht wie gewünscht verbessern, so gibt es zudem die Möglichkeit, die im Kopf verbaute inertielle Messeinheit zu nutzen, um das Schwingen durch gezieltes Ansteuern zu unterbinden.

Mit der im Zuge dieser Arbeit implementierten Bahnplanung wurde die Möglichkeit geschaffen, den Weg zwischen zwei unterschiedlichen Gelenkstellungen auf einer zeitlich optimierten Bewegungsbahn zu berechnen. Die Versuche haben gezeigt, dass diese Bewegung nicht ausreicht, um aus allen möglichen Positionen heraus eine beliebige Schlagposition zu erreichen. Auch wird die Redundanz der Gelenke an dieser Stelle nicht ausgenutzt. Eine zukünftige Implementierung könnte die Bewegungsbahn auf Grundlage der geforderten Schlagrichtung berechnen. Dafür können mehrere Punkte zwischen Ausgangspunkt und Schlagpunkt vorgegeben werden. So könnte der Roboter beispielsweise eine Ausholbewegung machen oder die erforderliche Endeffektorstellung auf einer energetisch optimierten Bewegungsbahn erreichen. Zudem kann hier bei einem Schlag unter Anwendung der inversen Kinematik die Redundanz der Gelenke intelligent

genutzt werden. Die implementierte Bahnplanung ermöglicht eine optimale Berechnung der Strecke zwischen zwei Gelenkstellungen und liefert somit eine gute Grundlage für all diese Erweiterungen.

Bei der Bewertung der Ergebnisse ist zu berücksichtigen, dass aufgrund der teilweise unvorhergesehenen Herausforderungen, zu denen es im Verlaufe des Projektes gekommen ist, die anfängliche Planung mehrfach angepasst wurde. Dies hatte unter anderem zur Folge, dass die Tests nicht mit dem vollen Funktionsumfang durchgeführt werden konnten. So war es zum Beispiel nicht möglich die Präzision der Bewegung in Kombination mit dem Schlagen eines Balles zu testen.

Abschließend kann gesagt werden, dass die gestellte Aufgabe anhand des simulationsbasierten Entwicklungsmodells dennoch erfolgreich bewerkstelligt wurde. Der ballspielende Roboter Doggy2 ist mit seinem neuen Antriebskonzept somit für zukünftige Aufgaben hinreichend ausgestattet.

Literaturverzeichnis

- [4at21] 4ATTENTION GMBH & CO. KG: *Robokeeper*. <https://www.robokeeper.com/de/>. Version: 2021
- [Aut20] AUTODESK ; COPYRIGHT 2020 AUTODESK, INC. ALLE RECHTE VORBEHALTEN (Hrsg.): *Inventor*. <https://www.autodesk.de/products/inventor/overview?term=1-YEAR&support=null>. Version: 2020
- [Bol18] BOLTE, Ekkehard: *Elektrische Maschinen*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2018. <http://dx.doi.org/10.1007/978-3-662-54688-8>. <http://dx.doi.org/10.1007/978-3-662-54688-8>
- [BU02] BRINKSCHULTE, Uwe ; UNGERER, Theo: *Mikrocontroller und Mikroprozessoren*. Berlin and Heidelberg : Springer, 2002 (Springer-Lehrbuch). <http://dx.doi.org/10.1007/978-3-662-08746-6>. <http://dx.doi.org/10.1007/978-3-662-08746-6>. – ISBN 9783662087466
- [Cor17] CORKE, Peter: *Robotics, Vision and Control*. Bd. 118. Cham : Springer International Publishing, 2017. <http://dx.doi.org/10.1007/978-3-319-54413-7>. <http://dx.doi.org/10.1007/978-3-319-54413-7>
- [Gui99] GUIDO KRAMANN: *Modelierung eines Gleichstrommotors*. http://www.kramann.info/62_Regelungssysteme/05_Regleroptimierung/06_Gleichstrommotor/index.php. Version: 1999
- [HROA16] HEIMANN, Bodo ; RISSING, Lutz ; ORTMAIER, Tobias ; ALBERT, Amos: *Mechatronik: Komponenten - Methoden - Beispiele : mit 292 Bildern, 44 Tabellen und 80 ausführlich durchgerechneten Beispielen*. 4., überarbeitete und ergänzte Auflage. München : fv Fachbuchverlag Leipzig im Carl-Hanser-Verlag, 2016. <http://dx.doi.org/10.3139/9783446445338>. <http://dx.doi.org/10.3139/9783446445338>. – ISBN 978-3-446-44451-5
- [igu21] IGUS GMBH: *iglidur® Zahnräder*. <https://www.igus.de/info/zahnraeder>. Version: 2021
- [Ind35] INDUSTRIE MAGAZIN: *Der Markt für Logistik-Roboter boomt*. <https://industriemagazin.at/a/der-markt-fuer-logistik-roboter-boomt>. Version: 28.10.2020 09:35
- [Int21] INTUITIVE SURGICAL: *DaVinci: Surgical Systems*. <https://www.intuitive.com/en-us/products-and-services/da-vinci/systems#>. Version: 2021
- [Kuk20] KUKA: *Industrieroboter von Kuka*. <https://www.kuka.com/de-de/produkte-leistungen/robotersysteme/industrieroboter>. Version: 2020
- [Lin21] *LTspice*. <https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html>. Version: 21.01.2021

- [Mar20a] MARECZEK, Jörg: *Grundlagen der Roboter-Manipulatoren – Band 1*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2020. <http://dx.doi.org/10.1007/978-3-662-52759-7>. <http://dx.doi.org/10.1007/978-3-662-52759-7>
- [Mar20b] MARECZEK, Jörg: *Grundlagen der Roboter-Manipulatoren – Band 2: Pfad- und Bahnplanung, Antriebsauslegung, Regelung*. 1st ed. 2020. 2020. <http://dx.doi.org/10.1007/978-3-662-59561-9>. <http://dx.doi.org/10.1007/978-3-662-59561-9>. – ISBN 9783662595619
- [Max10] MAXON MOTORS GROUP: *Motordaten und Betriebsbereiche von DC Motoren*. https://www.maxongroup.ch/medias/sys_master/8797816094750.pdf. Version: 2010
- [Max20] MAXON MOTORS GROUP (Hrsg.): *RE 50 maxon DC motor 50 mm, Graphitbürsten, 200 Watt*. 578299. maxongroup: Maxon Motors Group, 2020. (RE50) . https://www.maxongroup.de/medias/sys_master/root/8839846559774/DE-142.pdf
- [Max21] MAXON MOTORS GROUP ; MAXON MOTORS GROUP (Hrsg.): *RE 50, Graphitbürsten, 200 Watt*. <https://www.maxongroup.de/maxon/view/product/motor/dcmotor/re/re50/578296>. Version: 2021
- [NC18] NÉMETH-CSÓKA, Mihály: *Thermisches Management elektrischer Maschinen*. Wiesbaden : Springer Fachmedien Wiesbaden, 2018. <http://dx.doi.org/10.1007/978-3-658-20133-3>. <http://dx.doi.org/10.1007/978-3-658-20133-3>
- [Ort12] ORTSEIFEN, Andreas: *Modellbasierte Anti-Windup-Methoden*, TU Darmstadt, Dissertation, 2012. https://tuprints.ulb.tu-darmstadt.de/3748/1/Andreas_Ortseifen_Entwurf_von_Anti-Windup-Methoden.pdf
- [Qt 21] *Welcome to the ROS Qt Creator Plug-in wiki*. <https://ros-qtc-plugin.readthedocs.io/en/latest/>. Version: 15.01.2021
- [Rob16] ROBOCUP FEDERATION: *RoboCup*. <https://www.robocup.org/>. Version: 2016
- [Sch08] SCHULZE, Manfred: *Elektrische Servoantriebe: Baugruppen mechatronischer Systeme ; mit Beispielen mit Lösungen*. München : Fachbuchverl. Leipzig im Hanser-Verl., 2008 <http://www.hanser-elibrary.com/isbn/9783446414594>. – ISBN 9783446416772
- [Sta14] STAR STRIDER ; MATHWORKS (Hrsg.): *Curve fitting to a sinusoidal function*. <https://uk.mathworks.com/matlabcentral/answers/121579-curve-fitting-to-a-sinusoidal-function>. Version: 2014
- [Tex16] TEXAS INSTRUMENTS (Hrsg.): *DRV8842 DC Motor Driver IC*. DRV8842. TI: Texas Instruments, 2016. <https://www.ti.com/lit/gpn/drv8842>
- [Trö11] TRÖSTER, Fritz: *Steuerungs- und Regelungstechnik für Ingenieure*. 3., überarb. und erw. Aufl. München : Oldenbourg, 2011. <http://dx.doi.org/10.1524/9783486705256>. <http://dx.doi.org/10.1524/9783486705256>

- [Urs20] URS KAFADER: *Getriebe: Massenträgheitsmoment*. <https://support.maxongroup.com/hc/de/articles/360006129633-Getriebe-Massentr%C3%A4gheitsmoment>. Version: 2020
- [Was14] WASTE ; RN-WISSEN (Hrsg.): *Regelungen*. https://rn-wissen.de/wiki/index.php/Regelungstechnik#Der_Regler. Version: 2014

Anhang

Regler	K_{PR}	T_n	T_v
P	$\frac{T_I}{K_{PS} \cdot T_t}$	-	-
PI	$0,9 \cdot \frac{T_I}{K_{PS} \cdot T_t}$	$3,3 \cdot T_t$	-
PID	$1,2 \cdot \frac{T_I}{K_{PS} \cdot T_t}$	$2,0 \cdot T_t$	$0,5 \cdot T_t$

Tabelle 3: Reglereinstellungen nach Ziegler-Nichols (bekannte Regelstreckenparameter)

Quelle: [Trö11]

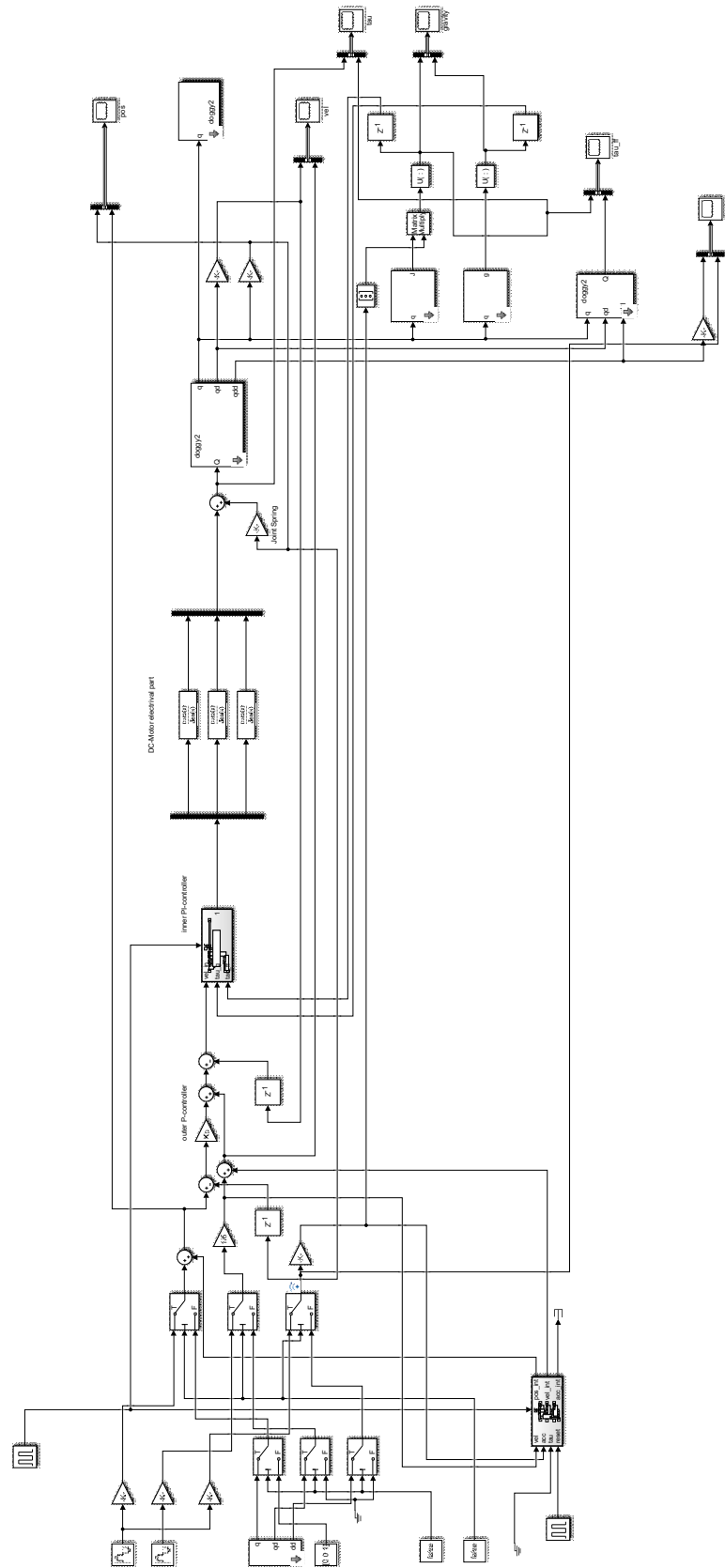


Abbildung 30: Simulink Simulation des Doggy2-Gesamtsystems mit Regler und Dynamik

Eidesstattliche Erklärung

Eidesstattliche Erklärung zur Master-Arbeit

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Unterschrift :

Ort, Datum :

