

BACHELORARBEIT

Design und Implementierung eines Regelungsverfahrens zur Stabilisierung eines humanoiden Roboters bei verschiedenen Bewegungsabläufen auf Basis des NAO-Robotiksystems

Autor:

Finn Ole Poppinga

Betreuer: Patrick Götsch, M. Sc.

Prüfer: 1. Prof. Dr. Herbert Werner
2. Patrick Götsch, M. Sc.

20. März 2015

Titel:

Design und Implementierung eines Regelungsverfahrens zur Stabilisierung eines humanoiden Roboters bei verschiedenen Bewegungsabläufen auf Basis des NAO-Robotiksystems

Beschreibung:

Das humanoide Robotiksystem NAO der Firma Aldebaran ist derzeit die Standardplattform des Roboter-Fußball-Wettbewerbs RoboCup. Im Fußballspiel ist es notwendig, dass der Roboter schnelle Bewegungen ausführen kann und dabei nicht stürzt. Im Gedränge auf dem Platz treten häufig Kollisionen mit anderen Robotern auf – auch hier muss die Standstabilität gewährleistet sein. An diesen Anforderungen orientieren sich die Ziele dieser Arbeit.

Aufgaben:

1. Aufstellen einer Literaturübersicht von möglichen Regelungsmodellen für dynamische Bewegungen von humanoiden Robotern
2. Auswählen eines Regelungsmodells nach zu definierenden Kriterien
3. Simulation des Regelungsmodells zur Stabilisierung des Roboters nach Einwirken einer Störung im Stand
4. Erweiterung des Modells auf beliebige stabile Körperpositionen des Roboters
5. Implementierung und Verifikation des Modells auf Basis des humanoiden Roboters NAO im Software Framework von RobotING@TUHH e.V.

Betreuer: Patrick Götsch, M. Sc.

Prüfer: 1. Prof. Dr. Herbert Werner
2. Patrick Götsch, M. Sc.

Ausgabe der Arbeit: 5. Januar 2015

Abgabe der Arbeit: 22. März 2015

20. März 2015, Prof. Dr. H. Werner

Hiermit erkläre ich, die vorliegende Arbeit selbstständig durchgeführt und keine weiteren Hilfsmittel und Quellen als die angegebenen genutzt zu haben.

Hamburg, 20. März 2015

Finn Ole Poppinga

Inhaltsverzeichnis

1 Einleitung	1
1.1 Grundlagen	2
Typen von humanoiden Robotern	2
Das NAO-Robotiksystem	2
Dynamisches Gleichgewicht	3
Balancierungsstrategien	4
2 Regelmodelle für humanoide Roboter	6
2.1 Linear Inverted Pendulum Model	6
Herleitung der Bewegungsgleichungen des LIPM	6
Erweiterung in mehrere Dimensionen	8
2.2 Angular Momentum inducing Pendulum Model	8
Herleitung der Bewegungsgleichungen des AMPM	9
2.3 Variable Impedance Inverted Pendulum Model	10
Herleitung der Bewegungsgleichungen für das VIIP	10
2.4 Reaction Mass Pendulum Model	12
2.5 Vollständiges dynamisches Modell	13
2.6 Anforderungen an ein Regelungsmodell	13
2.7 Auswahl von Modellen zur Stabilisierung des Gleichgewichts	14
3 Sensorik und Messtechnik	17
3.1 Messtechnik auf dem NAO	17
Schwerpunktposition	17
Messverzögerung	18
Messung des Center of Pressure	20
Messrauschen	20
3.2 Filterung des Messsignals	21
Tiefpassfilter	21
Kalman-Filter	21
Vergleich der beiden Filtermethoden	23
3.3 Messdatenübertragung und -visualisierung	23
Protokoll zur Übertragung einer Auswahl von beliebigen Werten in Echtzeit an verschiedene Endpunkte	23

Visualisierung der Messdaten	24
4 Kriterien für dynamisches Gleichgewicht	25
4.1 Erweiterung des AMPM um das Konzept des ZMP	25
4.2 Stabilisierbarkeit mit der Ankle-Strategie	26
Orbitale Energie des LIPM und Capture Point	27
Anforderungen an die Position der Capture Points	28
4.3 Stabilisierbarkeit unter Verwendung der Flywheel-Strategie	29
Limitierungen des Flywheels	30
Der Bang-Bang Controller	30
Stabile Region unter Verwendung des Flywheels	31
4.4 Anwendung der Entscheidungsregionen auf verschiedene Posen	33
5 Modellverifizierung	35
5.1 Betrachtung der Übertragungsfunktion des LIPM	35
5.2 Vergleich der Eingangssprungantwort von LIPM und NAO	36
Auswertung der Messergebnisse	37
6 Dynamische Stabilisierung des Roboters	39
6.1 Steuerung des Motormoments im Knöchelgelenk	39
Stiffness	39
Steuerung des Drehmoments	40
Steuerung des CoP über das Moment im Fußgelenk	40
6.2 Regelung des Fußmomentes	41
Open-Loop Eigenschaften des Linear Inverted Pendulum Model (LIPM)	41
6.3 Verschiedene Regleransätze	43
Reglerparameter	46
6.4 Regelung des Oberkörpers	47
Abschaltung der CoM-Regelung	47
Anwendung der Flywheel-Strategie auf dem realen Roboter	48
6.5 Digitale Reglerimplementierung	49
Tustin-Approximation zur Bestimmung der digitalen Reglerparameter	49
6.6 Dynamische Adaption der Reglerparameter an verschiedene Roboterposen	50
7 Zusammenfassung und Ausblick	53

A Zero State Response der Flywheel-Strategie **55**

B Inhalte der CD **55**

Abbildungsverzeichnis

1.1	Foto des NAO-Robotiksystems	3
1.2	Darstellung des Supportpolygons	4
1.3	Darstellung der Bodenreaktionskraft	4
1.4	Schema verschiedener Balancierungsstrategien	5
2.1	Skizze des LIPM	7
2.2	Skizze des AMPM	10
2.3	Skizze des VIIP	11
2.4	Skizze des RMP	12
3.1	Zeichnung: Lage der IMU	18
3.2	Plot: Messverzögerung	19
3.3	Zeichnung: Lage der FSRs	20
3.4	Plot: Tiefpassfilterung eines Signals	22
3.5	Plot: Vergleich zwischen Tiefpass- und Kalman-Filter	22
3.6	Screenshot: Debug-Oberfläche	25
4.1	Skizze: Bodenreaktionskraft im AMPM	26
4.2	Plot: Phasendiagramm LIPM-Simulation	28
4.3	Skizze: Stabile Region mit Einzelfußkontakt	29
4.4	Plot: Zeitlicher Verlauf einer Zweipunktregelung	30
4.5	Plot: Stabilisierbare Region mit Flywheel-Strategie	33
4.6	Darstellung des Supportpolygons für beide Füße	34
5.1	Foto: Verschiedene Posen des Roboters	36
5.2	Plot: Messung der Sprungantwort des NAOs	37
5.3	Plot: Auswertung der Sprungantwort des NAOs	38
6.1	Geschlossener PD-Regelkreis.	41
6.2	Plot: Phasendiagramm des realen Roboters unter Anwendung der <i>Ankle-Strategie</i>	42
6.3	Positionen der Pole für das ungeregelte lineare invertierte Pendelmodell.	42
6.4	Plot: Wurzelortskurve des PD-geregelten Systems	44
6.5	Plot: Bodediagramm der verschiedenen Regleransätze	45
6.6	Plot: Wurzelortskurve des PID-geregelten Systems	45
6.7	Plot: Sprungantwort des geregelten Systems	46

6.8	Plot: Phasenverlauf des realen Roboters unter Verwendung der Flywheel-Strategie	48
6.9	Plot: Wurzelortskurven für verschiedene Posen	51
6.10	Plot: Automatische Berechnung der Reglerparameter	52

Tabellenverzeichnis

1	Reglerparameter für das simulierte LIPM.	47
2	Approximierte Reglerparameter für den realen Roboter.	50

Abkürzungsverzeichnis

LIPM	Linear Inverted Pendulum Model
RMP	Reaction Mass Pendulum
VIIP	Variable Impedance Inverted Pendulum
CTM	Cart Table Model
FDM	Full Dynamic Model
ZMP	Zero Moment Point
AMPM	Angular Momentum Pendulum Model
MBC	Momentum-Based Control
CoM	Center of Mass
CoP	Center of Pressure
DCM	Device Communication Manager
TCP	Transmission Control Protocol
JSON	JavaScript Object Notation
MISO	Multiple Input Single Output
MIMO	Multiple Input Multiple Output
SISO	Single Input Single Output
FSR	Force-Sensing Resistor
IMU	Inertia Measurement Unit
MRE	Magnetic Rotary Encoder

1 Einleitung

Die Forschung an humanoiden Robotern dient nicht nur dazu, neue Entwicklungen im Bereich der Robotik zu erlangen. Wissenschaftliche Arbeiten in diesem Bereich können auch dazu führen, die biomechanischen Abläufe im menschlichen Körper besser zu verstehen. Bewegungsabläufe und Reaktionen, die dazu dienen, das Gleichgewicht zu halten können zunächst beim Menschen analysiert und dann auf humanoiden Robotern implementiert werden. Dabei können Theorien über diese Bewegungsabläufe verifiziert werden, da ein physikalisches Verständnis der Reaktionen zur Umsetzung in einem Robotiksystem erforderlich ist.

Während junge und gesunde Menschen noch einen sehr guten Gleichgewichtssinn haben und sich auch bei starken Stößen noch abfangen können, kann es gerade bei älteren Menschen oft vorkommen, dass diese in Alltagssituationen stürzen. Gerade bei diesen Menschen sind die aus einem Sturz resultierenden Verletzungen schwerer und der Heilungsprozess benötigt eine längere Zeit. Die Implementierung einer Stabilisierung des Gleichgewichts bei einem humanoiden Roboter kann in Zukunft helfen, den Prozess des dynamischen Balancierens besser zu verstehen und damit auch Gründe für das Stürzen von Menschen zu finden.

Um mehr Menschen, Wissenschaftler wie Interessierte für die humanoide Robotik zu begeistern, wurde 1997 das Projekt *RoboCup* ins Leben gerufen. Im *RoboCup Soccer* treten Roboter in verschiedenen Ligen gegeneinander im Fußball an. Bei der Gründung des *RoboCups* war es das Ziel der Initiatoren bis 2050 eine Mannschaft aus humanoiden Robotern zu bilden, welche den dann amtierenden *FIFA*-Weltmeister der Menschen schlagen soll. An der Technischen Universität Hamburg-Harburg gibt es ein Team von Studierenden, welches an den Wettbewerben des *RoboCups* teilnimmt und hierzu ein eigenes Software-framework entwickelt [1]. Im Rahmen der Fußballspiele kommt es oft vor, dass Roboter zusammenstoßen oder sich gegenseitig behindern. Oft stürzen dabei mehrere Roboter. Kann einer der Spieler das Gleichgewicht jedoch halten, erlangt er dadurch einen großen Spielvorteil.

In dieser Arbeit soll ein Modell entwickelt und implementiert werden, dass die dynamische Balance eines humanoiden Roboters bewerten und durch eine Regelung verbessern kann. In Kapitel 2 werden zunächst einige Modelle vorgestellt und auf ihre Verwendbarkeit hin untersucht. Alle Modelle haben gemeinsam, dass sie als Grundlage ihrer Modellierung den Schwerpunkt des Roboters betrachten – genau wie Menschen bei der Bewertung ihres Gleichgewichts auch den Zustand ihres Schwerpunkts beobachten [2]. Am Ende dieses Kapitels steht eine Auswahl von zwei Modellen, die verschiedene Aspekte der Balancierung abdecken. In Kapitel 3 wird evaluiert, wie die Sensorik und Messtechnik des in dieser Arbeit verwendeten NAO-Robotiksystems verwendet werden kann, um eine Regelung des Schwerpunkts durchzuführen. Zunächst werden die Sensoren des NAOs jedoch in Kapitel 5 dazu verwendet, die vorher ausgewählten Modelle mit dem realen Robotiksystem zu vergleichen damit abgeschätzt werden kann, ob es diese zur Entwicklung einer Regelung verwendet werden können. Auf der Basis der ausgewählten Modelle und unter Berücksichtigung der Limitierungen des NAO-Robotiksystems wird in Kapitel 4 ausführlich hergeleitet, unter welchen Bedingungen der humanoide Roboter das Gleichgewicht

halten kann. Dafür werden zwei unterschiedliche Strategien angewendet, die Balancierung durch den Einsatz der Fußgelenke und Stabilisieren mithilfe einer ruckartigen Oberkörperneigung. Beide dieser Strategien werden auch von Menschen verwendet, wenn sie das Gleichgewicht halten sollen, ohne einen Ausfallschritt machen zu dürfen [3]. In Kapitel 6 werden schließlich Regler dimensioniert, die mit dem NAO-Robotiksystem zum Einsatz kommen und die theoretischen Grundlagen aus den vorhergehenden Kapiteln der Arbeit in der Praxis anwenden. Dabei wird berücksichtigt, wie sich die Modell- und Reglerparameter verändern, wenn der Roboter verschiedene Körperhaltungen annimmt.

1.1 Grundlagen

In diesem Abschnitt sollen einige Grundlagen und Begriffe eingeführt werden, auf die in den nachfolgenden Kapiteln häufig verwiesen wird. Diese Grundlagen sind Basis für den Entscheidungs- und Entwicklungsprozess dieser Arbeit.

Typen von humanoiden Robotern

Humanoide Roboter lassen sich über zahlreiche Kriterien kategorisieren. Im Kontext dieser Arbeit ist jedoch die Klassifizierung des Aktuatorsystems der Roboter von besonderer Bedeutung.

Werden die Körperteile eines Roboters durch hydraulische oder pneumatische Aktuatoren bewegt, kann die durch jeden dieser Aktuatoren aufgebrachte Kraft präzise gesteuert werden. Es wird dann von kräftebasierten Robotern gesprochen.

Häufiger sind die Körperteile humanoider Roboter mit Gelenken verbunden, die jeweils einen oder mehrere Elektromotoren enthalten. Bei solchen Motoren lässt sich üblicherweise nur das von ihnen aufgebrachte Drehmoment steuern. Diese Form der Ansteuerung ist platzsparender und preisgünstiger als eine kräftebasierte Steuerung. Diese Art der Ansteuerung ist dafür allerdings nicht so flexibel und so gut mit dem menschlichen Muskelapparat vergleichbar.

Der humanoide Roboter NAO ist ein solcher momentenbasierter Roboter.

Das NAO-Robotiksystem

Der praktische Teil dieser Arbeit wird auf dem NAO-Robotiksystem des französischen Herstellers Aldebaran durchgeführt. Das NAO-Robotiksystem ist ein kommerziell erhältlicher humanoider Roboter. Seit 2007 wird der NAO in der *Standard Platform League* des *RoboCup* eingesetzt. Der NAO ist ca. 57 cm groß und wiegt 5,4 kg (siehe Abb. 1.1).

Der humanoide Roboter NAO hat sieben Gelenke mit insgesamt 25 Motoren, die alle unabhängig voneinander angesteuert werden können. Mithilfe dieser Gelenke ist es dem NAO möglich, menschenähnliche Bewegungen auszuführen und auf zwei Beinen zu laufen.

Die Umwelt nimmt der NAO über seine Sensoren wahr. Zu den Sensoren zählen Kameras, Mikrofone und Ultraschallsensoren, mit denen die Umgebung des Roboters direkt unter-

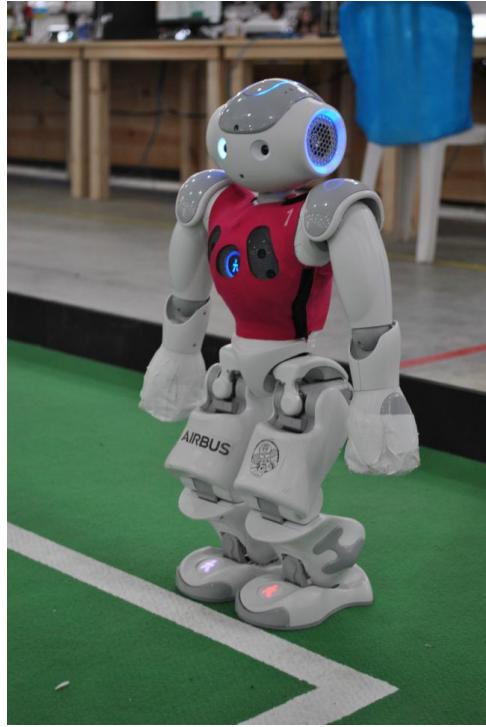


Abbildung 1.1: Der humanoide Roboter NAO auf einem Fußballfeld des *RoboCups*.

sucht werden kann. Durch die integrierten Druck- und Berührungssensoren kann der NAO mit Menschen interagieren. Weiterhin ist das NAO-Robotiksystem mit Sensoren ausgestattet, die die Lage und Position des Roboters im Raum messen können. Diese Sensoren werden in Kapitel 3 genauer beschrieben.

Dynamisches Gleichgewicht

Bei der Betrachtung eines sich in Ruhe befindlichen Humanoiden, lässt sich leicht beschreiben, wann sich dieser Roboter im Gleichgewicht befindet. Dazu wird zunächst eine Fläche beschrieben, welche die konvexe Hülle aller Punkte bildet, die mit dem Boden in Kontakt sind. Diese Fläche heißt *Supportpolygon* (siehe Abb. 1.2).

Um nicht umzufallen muss sich der Schwerpunkt (CoM) des Körpers direkt oberhalb des Supportpolygons befinden.

Diese Definition gilt nicht mehr, wenn bewegte Körper betrachtet werden. So liegt die Projektion des Center of Mass (CoM) zum Beispiel beim humanoiden Gang nicht zu jedem Zeitpunkt innerhalb des Supportpolygons, während der Gehende trotzdem nicht umfällt. Um ein dynamisches Gleichgewicht zu halten, muss demzufolge ein anderes Kriterium erfüllt werden.

Auf den Fuß eines humanoiden Roboters wirkt eine Bodenreaktionskraft. Diese Kraft ist eine auf das gesamte Supportpolygon verteilte Flächenlast, kann aber durch eine einzelne Kraft dargestellt werden, die im Flächenschwerpunkt der Flächenkraft angreift (siehe

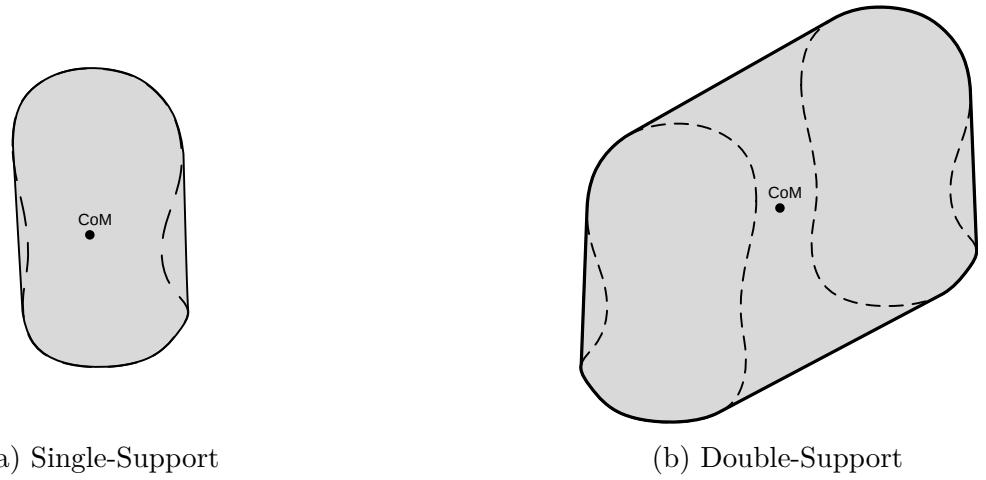


Abbildung 1.2: Das Supportpolygon ist die konvexe Hülle aller Flächen, die mit dem Boden in Kontakt sind [4].

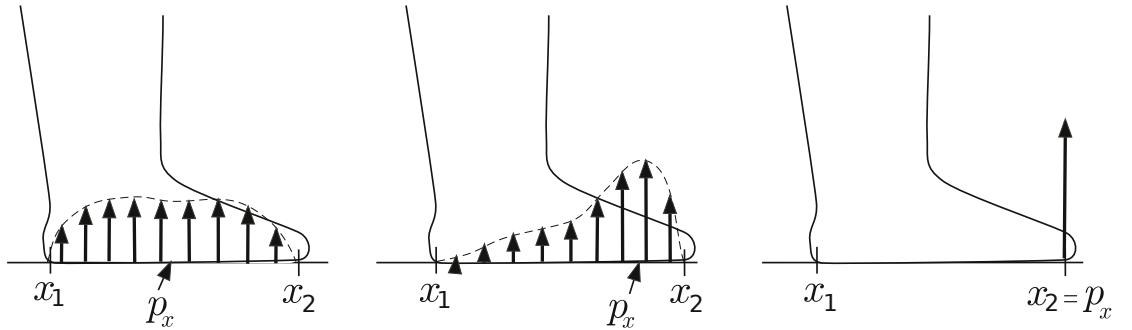


Abbildung 1.3: Die Bodenreaktionskraft als verteilte Flächenlast in verschiedenen Belastungssituationen [5]. Die Position des CoP ist mit p_x angegeben.

Abb. 1.3). Der Betrag der Bodenreaktionskraft ist, sofern keine Bewegung in z -Richtung stattfindet, gleich der Gewichtskraft des Roboters.

Den Angriffspunkt dieser Kraft nennt man den CoP. Den Punkt, an dem die Bodenreaktionskraft angreifen müsste, um das Momentengleichgewicht für den Roboter herzustellen nennt man den Zero Moment Point (ZMP). Die Bedingungen für das dynamische Gleichgewicht sind erfüllt, wenn die Position des CoP mit der des ZMP überein stimmt.

Das Prinzip des Zero Moment Point wurde erstmals 1968 von Vukobratovic et al. [6] eingeführt und ist eines der wichtigsten Stabilitätskriterien in der humanoiden Robotik.

Balancierungsstrategien

Werden Menschen gestoßen, können sie sich durch verschiedene Maßnahmen sehr gut und schnell selbst vor einem Sturz bewahren. Dabei kommen verschiedene Balancierungsstrategien zum Einsatz, von denen einige hier kurz vorgestellt werden sollen. Da der primäre

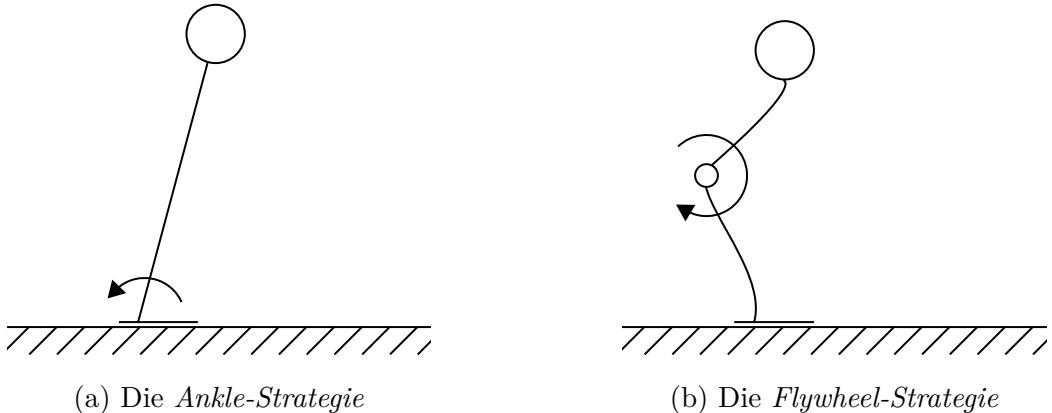


Abbildung 1.4: Je nachdem, wie stark der Stoß war, nutzten die Probanden entweder die *Ankle-Strategie*, oder für starke Störungen, die *Flywheel-Strategie*.

Fokus dieser Arbeit auf Balancierungsmaßnahmen liegt, welche erfordern, dass die Füße des Roboters stationär bleiben, werden hier auch nur solche Strategien vorgestellt, die dieser Anforderung genügen.

Maki et al. [2] haben eine Studie durchgeführt, in der erforscht wird, wie menschliche Probanden ihr Gleichgewicht halten, wenn sie keine Ausfallschritte machen dürfen. Dazu stellten sich die Probanden auf eine kurze Stufe, die sie nicht verlassen durften. Danach wurden die Probanden angestoßen und ihre Reaktionen aufgezeichnet. Nach mehreren Wiederholungen des Experiments erlernten viele der Probanden zwei ähnliche Strategien zum Balancieren nach der Einwirkung des Stoßes.

Bei der Anwendung der *Ankle-Strategie* (siehe Abb. 1.4a) bringen Probanden durch ihre Unterschenkelmuskulatur ein Moment um das Sprunggelenk ein, welches die Störung durch den Stoß ausgleichen kann. Bei der Anwendung der *Flywheel- oder Hip-Strategie* (siehe Abb. 1.4b) wird eine ruckartige Oberkörperbewegung ausgeführt, mit der selbst das Abfangen von starken Stoßen möglich ist. Die *Flywheel-Strategie* kommt vor allem dann zum Einsatz, wenn die Anwendung der *Ankle-Strategie* alleine nicht ausreicht, um das dynamische Gleichgewicht zu halten.

Beide dieser Strategien, die Menschen sehr erfolgreich einsetzen, sollen im Rahmen dieser Arbeit auf einem humanoiden Roboter implementiert und getestet werden. Auch die Entscheidung über den Einsatz einer Strategie soll, ähnlich wie bei Menschen, auf einer Bewertung der Stabilisierbarkeit beruhen und dynamisch getroffen werden können.

2 Regelmodelle für humanoide Roboter

Humanoide Roboter sind Systeme mit einer hohen Zahl von Freiheitsgraden. So hat zum Beispiel der Roboter *NAO* insgesamt 25 Freiheitsgrade. Soll nun die Kinematik eines solchen Systems beschrieben werden, um daraus Informationen über die dynamische Stabilität des Roboters zu erhalten, ist es fast immer ausreichend, dieses System durch ein geeignetes mathematisches Modell anzunähern, welches angemessene Vereinfachungen vornimmt [5].

In diesem Kapitel sollen einige dieser Modelle vorgestellt und erläutert werden um einen Überblick über verschiedene Möglichkeiten der dynamischen Vereinfachung eines humanoiden Robotiksystems zu gewinnen. Am Ende des Kapitels steht die Auswahl eines Modells, anhand dessen im weiteren Verlauf dieser Arbeit eine Regelung zur dynamischen Stabilisierung des Roboters entwickelt wird.

2.1 Linear Inverted Pendulum Model

Ein einfacher und sehr verbreiteter Ansatz zur Modellierung von bipedalen Robotern ist das LIPM nach Kajita et al. [7]. Die Grundidee hinter diesem Modell ist, anzunehmen, die gesamte Masse des Roboters sei in seinem Schwerpunkt konzentriert und verhalte sich wie ein invertiertes Pendel. Zur Linearisierung des Modells wird die Höhe des Schwerpunkts über dem Boden als konstant festgesetzt, das Pendel kann sich demzufolge nur in einer Ebene parallel zum Boden bewegen. Das Pendel muss deshalb eine variable Länge r haben.

Das LIPM dient als Grundlage für viele andere Modelle, die als Erweiterungen des Pendelmodells verstanden werden können und von denen einige in den nächsten Abschnitten vorgestellt werden sollen.

Herleitung der Bewegungsgleichungen des LIPM

Im Ersten Schritt sollen die Bewegungsgleichungen des LIPM für den zweidimensionalen Fall eines Roboters, der sich nur in der xz -Ebene bewegen kann, hergeleitet werden. Durch das Aufstellen eines Kräfte- und eines Momentengleichgewichts mithilfe eines Freikörperbilders (siehe Abb. 2.1) können so Differentialgleichungen für die Bewegung des Roboterschwerpunkts in der x -Richtung aufgestellt werden [5]. Die Erweiterung des Modells auf dreidimensionale Schwerpunktsbewegungen kann durch Superposition der hier hergeleiteten Lösung durchgeführt werden.

Das resultierende Moment um den Koordinatenursprung, also die Festlagerung des Pendels, wird mit $\tau^{(y)}$, die Masse des Roboters bzw. des Pendels mit M benannt.

Aufgrund des Momentengleichgewichts in y -Richtung gilt dann:

$$\tau^{(y)} = \underbrace{r^2 \ddot{\theta} M}_{\text{d'Alambertsches Moment}} + \underbrace{2r\dot{r}\dot{\theta} M}_{\text{Drehimpuls}} - \underbrace{r g \sin(\theta) M}_{\text{Gewichtskraft}} \quad (2.1)$$

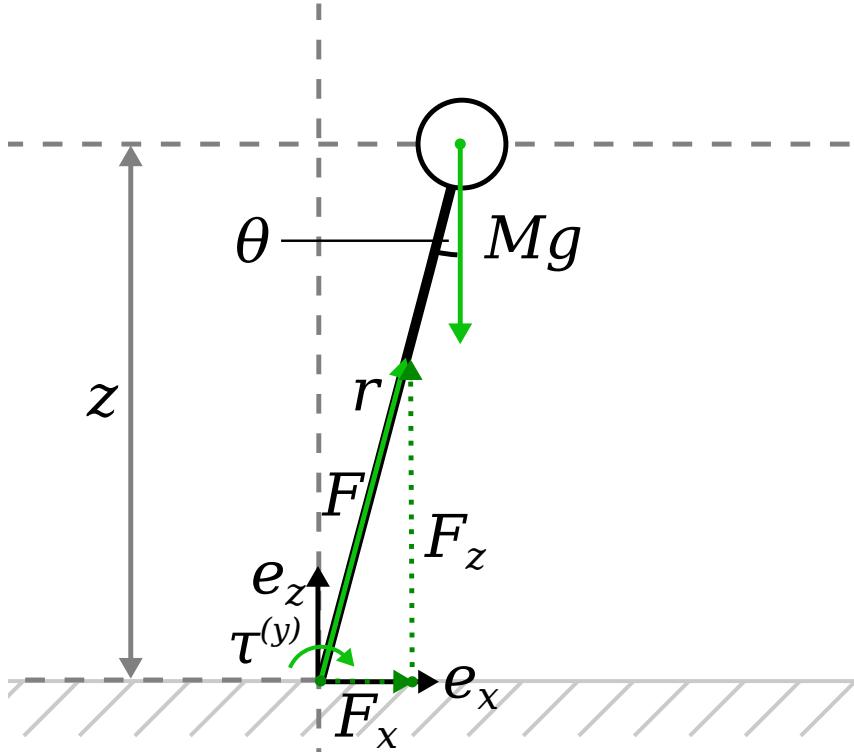


Abbildung 2.1: Skizze des LIPM. Die Position des Pendels ist in z -Richtung limitiert.

Das Kräftegleichgewicht im Lager ergibt sich zu:

$$F = \underbrace{\ddot{r}M}_{\text{Trägheit}} + \underbrace{g \cos(\theta) M}_{\text{Gewichtskraft}} - \underbrace{r \dot{\theta}^2 M}_{\text{Zentrifugalkraft}} \quad (2.2)$$

Aus (2.1) und (2.2) lassen sich nun zwei Gleichungen ableiten, die das LIPM vollständig beschreiben.

$$\frac{\tau^{(y)}}{M} = r^2 \ddot{\theta} + 2r \dot{r} \dot{\theta} - rg \sin(\theta) \quad (2.3)$$

$$\frac{F}{M} = \ddot{r} + g \cos(\theta) - r \dot{\theta}^2 \quad (2.4)$$

Bei der schwerpunktsbasierten Regelung eines humanoiden Roboters benötigt man ein Gleichungssystem, welches die Position und die Geschwindigkeit des CoM des Roboters in karthesischen Koordinaten beschreibt [8]. Die Gleichungen (2.3) und (2.4) liegen allerdings in Polarkoordinaten vor.

Die Bodenreaktionskraft F lässt sich in eine x - und eine y -Komponente zerlegen:

$$F_z = F \cos (\theta) \quad (2.5)$$

$$F_x = F \sin (\theta) \quad (2.6)$$

Die Kernannahme, auf der das Konzept des LIPM basiert ist, dass der Schwerpunkt des Modells einen Konstanten Abstand zur Bodenebene hat. Dieser Abstand wird mit z bezeichnet. Aus dieser Annahme folgt, dass die resultierende Kraft in z -Richtung gleich der Gewichtskraft der Pendelmasse sein muss (2.7).

$$F_z \stackrel{!}{=} Mg \quad (2.7)$$

Nun setzt man (2.5) ein und löst nach F auf und erhält:

$$F = \frac{Mg}{\cos (\theta)} \quad (2.8)$$

In x -Richtung gilt der Impulssatz:

$$F_x = F \sin (\theta) = M\ddot{x} \quad (2.9)$$

Setzt man nun (2.8) ein, erhält man mit

$$M\ddot{x} = \frac{Mg}{\cos (\theta)} \sin (\theta) = Mg \tan (\theta) = Mg \frac{x}{z} \quad (2.10)$$

eine gewöhnliche Differentialgleichung, die die Position des CoM beschreibt [5].

$$\ddot{x} = \frac{g}{z}x \quad (2.11)$$

Erweiterung in mehrere Dimensionen

Das LIPM beschreibt so, wie es bisher betrachtet wurde, lediglich die Bewegung des Schwerpunktes in einer Dimension. Da jedoch in (2.11) keine Abhängigkeit zum Zustand des Schwerpunkts im Hinblick auf dazu orthogonale Richtungen besteht, kann das LIPM für beide horizontale Koordinatenrichtungen unabhängig betrachtet werden. Soll dieses Modell also für die Approximation von Bewegungen in mehrere Richtungen herangezogen werden, können alle Prozesse, wie zum Beispiel der Entwurf von Reglern, für jede Dimension einzeln durchgeführt werden und am Ende superponiert werden [5].

2.2 Angular Momentum inducing Pendulum Model

Das im vorhergehenden Kapitel beschriebene LIPM bildet die Dynamik eines humanoiden Roboters allein durch die Bewegung des Massenschwerpunktes ab. Zur Stabilisierung des

Roboters nach einer Einwirkung von außen sollten aber neben einer Regelung des Schwerpunkts auch andere Aktuatoren des Systems verwendet werden – zum Beispiel eine Bewegung der Arme oder ein Neigen des Oberkörpers des Roboters. Diese Bewegungen können als das Einbringen eines Drehimpulses in das System zusammengefasst werden. Um die Auswirkungen dieser Bewegungen erfolgreich vorhersagen zu können, muss das verwendete Modell in der Lage sein, diese Aktionen zu berücksichtigen. Eine Erweiterung des LIPM, die exakt das Einbringen dieses Drehimpulses vorsieht, ist das Angular Momentum Pendulum Model (AMPM).

Herleitung der Bewegungsgleichungen des AMPM

Pratt et al. wählen den Ansatz, die durch den eingebrachten Drehimpuls τ_h resultierende Kraft in die Modellgleichungen aufzunehmen [9]. Ein Freikörperbild des AMPM ist in Abb. 2.2 dargestellt. Die Gleichung für das Kräftegleichgewicht des LIPM (2.2) lässt sich, analog zu (2.5) in die Koordinatenrichtungen x und z zerlegen und wird durch das Hilfsmoment dann entsprechend erweitert:

$$\sum F^{(x)} = F \sin(\theta) - \underbrace{\frac{\tau_h}{r} \cos(\theta)}_{F_{h,x}} = M\ddot{x} \quad (2.12)$$

$$\sum F^{(z)} = -Mg + F \cos(\theta) + \underbrace{\frac{\tau_h}{r} \sin(\theta)}_{F_{h,z}} = M\ddot{z} \quad (2.13)$$

Die Kräftegleichgewichte zeigen, dass das Hilfsmoment Auswirkungen auf beide Komponenten der Bodenreaktionskraft hat. Es ist jedoch davon auszugehen, dass die Wirkung des Hilfsmoments im Verhältnis zur Gewichtskraft des Pendels klein ist. Die Höhe z des Pendelschwerpunkts über dem Boden ist wie beim LIPM konstant. Für kleine Auslenkungen des Pendels gilt dann die Vereinfachung:

$$r \approx z \quad (2.14)$$

Es folgt also aus (2.12) für die Bewegungsgleichungen des AMPM

$$\ddot{x} = \frac{g}{z}x - \frac{1}{Mz}\tau_h \quad (2.15)$$

Setzt man hier das Hilfsmoment $\tau_h = 0$, so folgt aus (2.15) direkt die Differentialgleichung für das LIPM, (2.11).

Zur Herleitung der Bewegungsgleichungen des AMPM müssen gewisse Vereinfachungen angenommen werden um eine Linearität der Gleichungen sicherzustellen. Es wird sich im Verlauf dieser Arbeit jedoch zeigen, dass die Abschätzung der Auswirkung der Oberkörperbewegungen auf diese Art möglich und sinnvoll ist. Das AMPM wurde zur Bewertung der Stabilisierbarkeit mehrmals erfolgreich von anderen Gruppen eingesetzt [8, 9, 10].

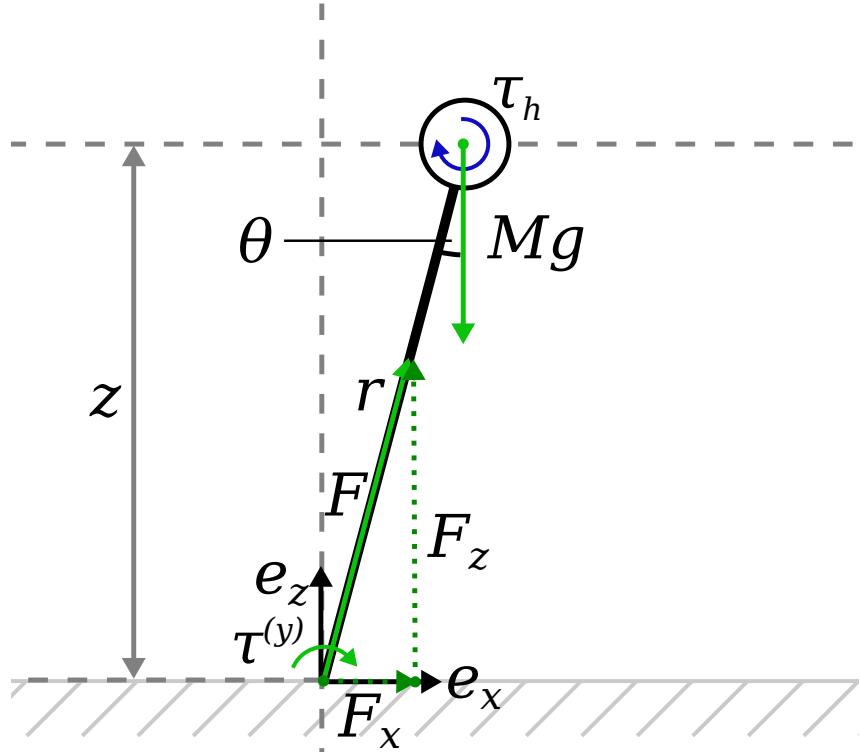


Abbildung 2.2: Skizze des AMPM. Die Bewegung des Oberkörpers wird durch ein Moment τ_h modelliert.

2.3 Variable Impedance Inverted Pendulum Model

Auch das Variable Impedance Inverted Pendulum (VIIP) ist eine Erweiterung des LIPM, welches von Sugihara et al. [11] erstmals eingeführt wurde. Dieses Modell betrachtet neben dem Schwerpunkt des Roboters auch die Beine und deren Aktuatoren. Entwickelt wurde das VIIP vor Allem, um die Auswirkung von Unebenheiten und Veränderung der Bodenbeschaffenheit für das zweibeinige Laufen von Humanoiden abilden zu können. Durch eine Modellierung der im Roboterbein auftretenden Kräfte kann auf ein Nachgeben des Bodens reagiert werden. Ein Funktionsschema des VIIP befindet sich in Abb. 2.3.

Herleitung der Bewegungsgleichungen für das VIIP

In der Herleitung der Bewegungsgleichungen für das VIIP wird analog zur Herleitung des LIPM vorgegangen, jedoch wird die Beschränkung der wirkenden Kräfte in z -Richtung (2.7) nicht angenommen und die Dynamik des Pendels in z -Richtung gesondert betrachtet. Für die Bewegungen des Schwerpunkts in der xy -Ebene können weiterhin die Gleichungen des LIPM verwendet werden.

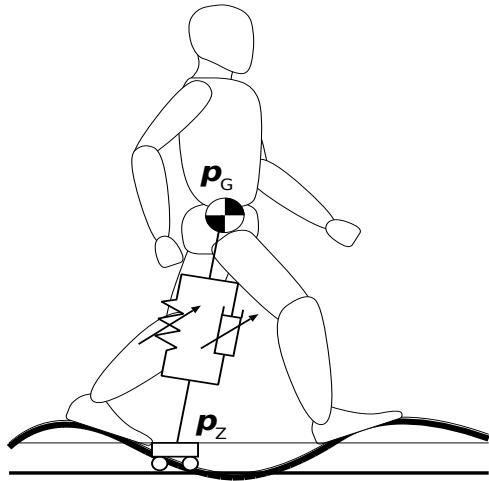


Abbildung 2.3: Skizze des VIIP. Die variablen Dämpfungs- und Federkomponenten sind entsprechend dargestellt und die Position des Schwerpunkts p_G und des ZMP p_Z markiert [12].

Für die Beschleunigung in z -Richtung gilt allgemein der Impulssatz im Schwerpunkt:

$$\ddot{z} = \frac{F_z}{M} - g \quad (2.16)$$

wobei z die Höhe des Schwerpunkts über dem Boden darstellt. Dadurch, dass der Roboter nicht am Boden fixiert ist, kann die entsprechende Reaktionskraft lediglich in positive z -Richtung zeigen.

$$F_z \geq 0 \quad (2.17)$$

Das Ziel des VIIP ist es in diesem Zusammenhang, durch einen kraftgeregelten Roboter eine Referenz für den zeitlichen Verlauf der Kraft in z -Richtung vorzugeben, um komplexe Bewegungen wie das Abfedern eines Sprunges oder das Dämpfen einer Stoßbewegung durch eine Beinbewegung abbilden zu können. Durch eine geeignete dynamische Vorgabe des Kräfteverlaufs können die Beine des Roboters als System modelliert werden, was genutzt werden kann um ein besseres dynamisches Verhalten in verschiedenen Situationen zu erlangen.

Soll das Bein zum Beispiel als Stoßdämpfer modelliert werden, kann für die Bodenreaktionskraft eine Referenzkraft F_z^{ref} folgender geschwindigkeitsabhängiger Ausdruck verwendet werden:

$$F_z^{\text{ref}} = M \cdot [K_D (\dot{z}^{\text{ref}} - \dot{z})] \quad (2.18)$$

Setzt man nun (2.18) in (2.16) ein, erhält man die Differentialgleichung:

$$\ddot{z} = K_D (\dot{z}^{\text{ref}} - \dot{z}) - g \quad (2.19)$$

Das VIIP ist analog zu (2.18) auch um einen proportionalen Term erweiterbar[12], der sich wie eine Federung verhält. Die Bewegungsgleichung ergibt sich dann zu:

$$\ddot{z} = K_P (z^{\text{ref}} - z) + K_D (\dot{z}^{\text{ref}} - \dot{z}) - g \quad (2.20)$$

2.4 Reaction Mass Pendulum Model

Ein weiterer Ansatz um die Dynamik eines humanoiden Roboters zu modellieren, ist das RMP. Dieses Modell ist, im Gegensatz zu den vorherigen besprochenen Modellen, keine Erweiterung des LIPM im engeren Sinne, da hier nicht mehr davon ausgegangen wird, dass die Masse des Roboters in seinem Schwerpunkt konzentriert ist [13]. Stattdessen wird das Flächenträgheitsmoment des Roboters erfasst und dieser durch ein Ellipsoid mit äquivalentem Flächenträgheitsmoment modelliert. Dieses Ellipsoid hat seinen Ursprung im Roboterschwerpunkt in einer bestimmten Entfernung über dem Boden. Wie beim LIPM wird die Höhe des Schwerpunkts über dem Boden auch beim RMP als konstant angenommen.

Das Ellipsoid selbst kann wiederum durch einen hantelähnlichen Aufbau von sechs Punktmassen entlang seiner Hauptachsen beschrieben werden. Diese Massenpunkte können sich radial zum Schwerpunkt bewegen, wodurch die Variation des Flächenträgheitsmomentes erreicht wird. Um den Schwerpunkt an einer konstanten Position zu halten haben zwei gegenüberliegende Punktmassen stets den gleichen Abstand zum CoM[14].

Auch für das RMP lassen sich Bewegungsgleichungen für den Schwerpunkt herleiten [14]. Jedoch ist der eigentliche Nutzen des RMP ein anderer: Mit einer entsprechenden in-

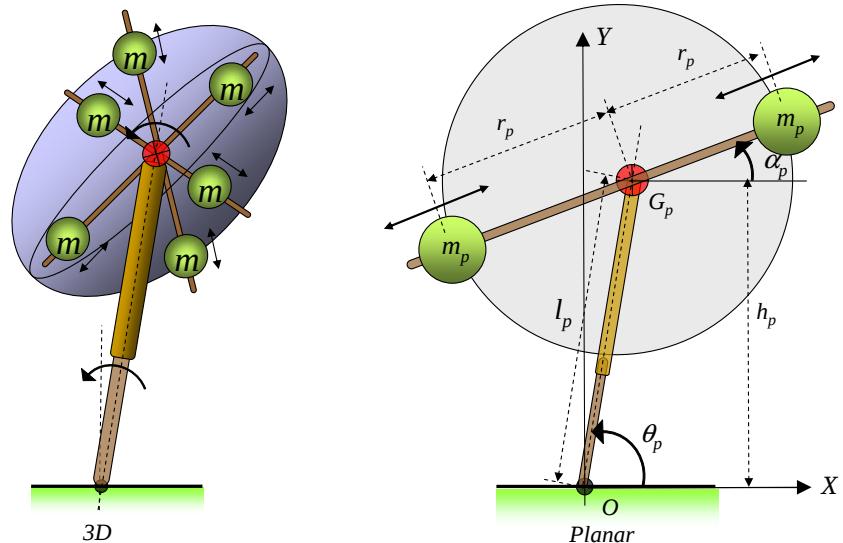


Abbildung 2.4: Skizze des RMP. Die radial zum Schwerpunkt des Roboters orientierten Massen modellieren den Roboter durch ein äquivalentes Ellipsoid [13].

versen Kinematik ist es möglich, über die Steuerung des Flächenträgheitsmoments automatisch Posen für den Roboter zu generieren, die dieses Flächenträgheitsmoment haben. Im RMP-Modell sind Aktuation und Modellierung also viel enger verknüpft als in den anderen bisher besprochenen Modellen.

Prinzipiell kann das RMP als eine Erweiterung des AMPM angesehen werden, in diesem Modell wird zusätzlich zum resultierenden Drehimpuls um den Schwerpunkt auch noch die Variation des Flächenträgheitsmomentes berücksichtigt.

2.5 Vollständiges dynamisches Modell

Die Entwicklung eines vollständigen dynamischen Modells für den NAO wurde im Rahmen einer Bachelorarbeit untersucht [15]. Das Ziel der Entwicklung eines vollständigen dynamischen roboterspezifischen Modells ist es, den Fehler, der durch Vereinfachungen und Dimensionsreduzierungen in anderen Modellen entsteht zu reduzieren. Da in einem vollständigen Modell die Wirkung aller Momente und Kräfte im System berücksichtigt wird, können mit dieser Art des Modells theoretisch alle möglichen Stabilisierungsstrategien erfasst werden. Durch eine geeignete Erweiterung gehört dazu zum Beispiel auch eine Interaktion mit dem Umfeld, wie das Abstützen an Wänden oder die Benutzung von Griffen.

Das vollständige dynamische Modell berechnet die Position des ZMP (siehe Kapitel 1.1) über das Moment, welches im Fußgelenk wirkt. Dieses ist Abhängig von Position, Beschleunigung und Geschwindigkeit aller Motoren. Durch diese sehr detaillierte Betrachtung der gesamten Roboterdynamik lassen sich die Auswirkungen aller Bewegungen des Roboters theoretisch genau vorhersagen. Diese Genauigkeit wird maßgeblich von der Sensorik des vorliegenden Systems und einer genauen Erfassung der Parameter aller Roboterelemente beeinflusst.

Die Anzahl der Parameter, die das vollständige dynamische Modell spezifizieren ist in der Größenordnung von 100, was den Aufwand einer mit diesem Modell entworfenen Regelung auf einem anderen Robotiksystem einzusetzen deutlich erhöht. Unter der Voraussetzung, dass der prinzipielle Aufbau des anderen Robotiksystems gleich wäre, müssten dennoch alle Parameter des Modells neu dimensioniert werden.

2.6 Anforderungen an ein Regelungsmodell

Im Kontext der Aufgabenstellung dieser Bachelorarbeit und unter Berücksichtigung der Limitierungen des NAO-Robotiksystems können Anforderungen an die Regelungsmodelle gestellt werden. Bei der Auswahl eines der in den vorhergehenden Abschnitten vorgestellten Regelungsmodells für den weiteren Verlauf dieser Arbeit sind diese Kriterien zu berücksichtigen.

Wie in Kapitel 1.1 erläutert, gibt es zwei wesentliche Strategien, die Menschen zu ihrer Stabilisierung anwenden, wenn sie keinen Ausfallschritt machen dürfen. Die Entscheidung

über die Strategie, welche angewendet wird um einen Stoß abzufangen treffen Menschen anhand der Zustandsgrößen ihres Körperschwerpunkts [3].

Anforderung 1: Das Modell muss geeignet sein um Position und Geschwindigkeit des Schwerpunkts des Roboters zu bestimmen.

Da zur Stabilisierung des Roboters neben der *Ankle-Strategie* auch die *Flywheel-Strategie* zum Einsatz kommen soll, muss das Modell eine Bewegung des Oberkörpers eines humanoiden Roboters hinreichend genau abbilden.

Anforderung 2: Das Modell muss ein im Oberkörper des Roboters eingebrachtes Moment in seinen Bewegungsgleichungen berücksichtigen.

Um Entscheidungshorizonte zur Auswahl einer geeigneten Regelungsstrategie definieren zu können, muss es möglich sein, die Auswirkungen von Regelungsstrategien auf das Modell theoretisch zu zeigen. Da die Auswertung der Entscheidungshorizonte in Echtzeit auf dem NAO-Robotiksystem geschehen muss, sollten hierfür möglichst keine numerischen Berechnungen erforderlich sein.

Anforderung 3: Es ist möglich die Modellgleichungen analytisch für den Zeit- und den Frequenzbereich zu lösen.

Falls eine weitere Balancierungsstrategie implementiert werden soll, sollte es möglich sein, diese entweder in das bestehende Modell zu integrieren, oder das Modell entsprechend zu erweitern.

Anforderung 4: Das Modell muss einfach um weitere Komponenten erweiterbar sein.

Um die von Regelungsverfahren, die auf dem ausgewählten Modell basieren, auf dem NAO-Robotiksystem implementieren zu können, müssen sie durch momentenbasierte Regelung realisierbar sein.

Anforderung 5: Die Verwendung des Modells erfordert keine präzisen Kraftmessungen oder genaue Steuerung von Kräften.

2.7 Auswahl von Modellen zur Stabilisierung des Gleichgewichts

Das LIPM ist eine weit verbreitete und sehr gut erweiterbare Lösung, mit der das dynamische Verhalten eines humanoiden Roboters beschrieben werden kann. Die Modellgleichungen zeichnen sich durch ihre Einfachheit und die Möglichkeit zur symbolischen Lösung im Zeitbereich aus, was Möglichkeiten zur Beobachtung des dynamischen Gleichgewichts eröffnet.

Wird nur das LIPM zur Beschreibung der Roboterbewegungen verwendet, können Einflüsse, die durch die Aktuation des Roboteroberkörpers oder der Arme entstehen nur bedingt, nämlich durch die von diesen Bewegungen hervorgerufene Schwerpunktsbewegung, berücksichtigt werden.

Das AMPM bildet durch die Erweiterung um das Hilfsmoment genau die Drehimpulse ab, die vom Oberkörper des NAOs eingebracht werden können. So lässt sich zum Beispiel das Balancieren mit einer ruckartigen Oberkörperneigung, die *Flywheel-Strategie*, damit analysieren und ihre Auswirkungen auf das gesamte dynamische Gleichgewicht des Roboters besser abschätzen.

Weder das AMPM noch das LIPM berücksichtigen eine Veränderung der Bodenreaktionskraft, um die Stabilität des NAOs zu erhöhen. So könnte zum Beispiel durch gezieltes Nachlassen der Kraft in einem Standbein ein Stoß abgefangen werden. Das VIIP bietet durch das Vorgeben einer Referenzkraft in der z -Richtung eine Möglichkeit, das Bein zum Beispiel als Dämpfungsglied zu modellieren. Hierzu ist jedoch eine präzise Steuerung der Kräfte im Bein notwendig, die der NAO aufgrund seiner momentenbasierten Konstruktion nicht ohne weiteres ermöglicht. Auch die Sensorik des NAOs ist, was die Messung von Kräften betrifft, nicht ausreichend genau (siehe Kapitel 3) um eine Regelung der Bodenreaktionskraft zu ermöglichen.

Das RMP bietet durch den alternativen Ansatz in der inversen Kinematik die Möglichkeit, auf ein von außen wirkendes Moment mit dem ganzen Körper gleichzeitig zu reagieren. Aus dem Modell resultiert jedoch eine große Menge neuer Anforderungen. Einerseits müsste auf dem NAO eine vollständig neue inverse Kinematik implementiert werden, damit die dem Modell entsprechenden Posen eingenommen werden können. Andererseits gibt es noch keine Referenzimplementierung auf einem realen System, die Bedingungen wie den Kontakt zwischen dem Boden und den Füßen des Roboters berücksichtigen kann. Die Implementierung einer RMP-basierten Regelungsstrategie wäre ein hoher Aufwand, wobei Aufgrund der hohen Ähnlichkeit des Grundprinzips gleichzeitig nur eine geringe Verbesserung der Modellgenauigkeit im Verhältnis zum AMPM zu erwarten wäre.

Eine Implementierung einer Stabilisierung anhand eines vollständigen dynamischen Modells kann theoretisch die besten Resultate erzielen, in denen der gesamte Körper des Roboters verwendet wird, um einen Sturz zu verhindern [16]. Auch eine Erweiterung des Modells aus [15] um die Berücksichtigung von extern wirkenden Kräften um das Abstützen des Roboters zum Beispiel an einer Wand abzubilden, wäre denkbar. Ein großes Problem der Nutzung des roboterspezifischen Modells ist jedoch die hohe Anzahl an Parametern und die geringe Flexibilität, was die Anwendung der gleichen Lösung auf anderen Robotiksystemen anbetrifft.

Die Wahl eines Modells muss immer ein Kompromiss aus der Realitätsnähe des Modells und der Komplexität der Modellgleichungen sein. Es ist das Ziel eines Modells, durch eine Reduktion der Komplexität eines Systems die Möglichkeit zu schaffen, das allgemeine dynamische Verhalten des Roboters verstehen zu können und das Design und die Implementierung eines Regelungsverfahrens zu vereinfachen. Das LIPM ist zunächst ein sehr einfaches Modell, welches zahlreiche Einschränkungen annimmt, dafür jedoch nur von einem Parameter abhängt. Ansätze wie das AMPM und das VIIP zeigen jedoch, dass

es möglich ist, dieses Modell so anzupassen und zu erweitern, dass es den individuellen Problemen einer bestimmten Fragestellung gerecht werden kann.

Alle vorgestellten Modelle erfüllen die Anforderung, den Zustand des Schwerpunkts des Roboters hinreichend genau darstellen zu können. Wie hergeleitet wurde, ist die Erweiterung des LIPM, das AMPM, geeignet die *Flywheel-Strategie* durch das Berücksichtigen eines in den Oberkörper des Roboters eingebrachten Moments zu analysieren. Auch das RMP oder das roboterspezifische Modell können hierfür verwendet werden. Die Verwendung des RMP erhöht jedoch die Komplexität der Modellgleichungen im Verhältnis zu denen des AMPM um vier Dimensionen. Da sich der Vorteil des AMPM erst auswirkt, wenn eine entsprechende inverse Kinematik implementiert wurde, sinkt das Verhältnis von Nutzen zu Aufwand der Verwendung dieses Modells im Kontext dieser Bachelorarbeit weiter.

Das vollständige dynamische Modell ist direkt von allen Gelenkpositionen und -geschwindigkeiten des Roboters abhängig und macht eine symbolische Lösung der Modellgleichungen deswegen und wegen der hohen Anzahl an Parametern sehr schwierig. Dafür ist dieses Modell um jeden physikalischen Effekt erweiterbar.

Die Kombination aus LIPM und AMPM erfüllt alle im vorhergehenden Abschnitt eingeführten Anforderungen an ein Modell, welches zur dynamischen Stabilisierung eines humanoiden Roboters genutzt werden soll. Im weiteren Verlauf dieser Arbeit werden diese beiden Modelle dazu genutzt, verschiedene Balancierungsstrategien zu bewerten und Regler zu implementieren, die diese Balancierungsstrategien auf dem NAO-Robotiksystem anwenden.

3 Sensorik und Messtechnik

Das NAO-Robotiksystem verfügt über fünf verschiedene Sensorsysteme [17]. Dazu gehören Ultraschallsensoren, mit denen Hindernisse wahrgenommen werden können genau so, wie verschiedene Berührungssensoren und Taster, über die der Roboter mit Menschen interagieren kann. In jedem Motor ist ein Magnetic Rotary Encoder (MRE) verbaut, über den die Winkelstellung des Motors gemessen werden kann. Nahe dem Ursprung des Roboterkoordinatensystems, im Oberkörper des NAOs, ist eine Inertia Measurement Unit (IMU) vorhanden, mit der Beschleunigungen und Rotationsraten gemessen werden können. In den Füßen des Roboters sind Force-Sensing Resistor (FSR) Sensoren verbaut, mit denen die Bodenreaktionskraft und deren Angriffspunkt gemessen werden kann.

Einige dieser Sensorsysteme können verwendet werden, um Informationen über das Gleichgewicht des NAOs zu erlangen. Diese Sensoren werden in diesem Kapitel vorgestellt und bewertet.

3.1 Messtechnik auf dem NAO

Schwerpunktposition

Zur Regelung des Gleichgewichts des NAOs wird die Position des CoM als Referenzgröße benötigt. Diese muss messtechnisch erfasst werden. Der humanoide Roboter NAO verfügt herstellerseitig über ein Modul, welches den Schwerpunkt des Roboters mit einer Frequenz von 100 Hz aus den Motorwinkeln und Massen aller Körperglieder ausrechnen kann. Im Softwareframework des *RobotING@TUHH e. V.* wird dieser Schwerpunkt relativ zum Schwerpunkt des Torsos des Roboters angegeben. Das gewählte Modell benötigt jedoch den Schwerpunkt des NAOs relativ zum Fußgelenk. Die Umrechnung der Messgröße erhält man durch eine Koordinatentransformation:

$$\mathbf{c}' = T^{-1} \mathbf{c} \quad (3.1)$$

Gestrichene Koordinaten sind im Fußkoordinatensystem, ungestrichene im Torsokoordinatensystem angegeben. Die Transformationsmatrix T bildet vom Torsokoordinatensystem in das Fußkoordinatensystem ab und kann auf zwei Weisen ermittelt werden:

1. Berechnung der Transformation von einem Fußkoordinatensystem in das Torsokoordinatensystem durch Vorwärtsskinematik, also aus den Winkeln aller Gelenkmotoren bis zum Körpermittelpunkt. Diese Methode hat den Nachteil, dass bei der Berechnung davon ausgegangen wird, dass der zur Transformation herangezogene Fuß auf dem Fußboden aufliegt. Dies ist nicht in jeder Situation gewährleistet, da der Roboter durch starke von außen wirkende Störungen in eine Kipplage gebracht werden kann, in denen nur ein teilweiser Bodenkontakt besteht. Die Messung des Schwerpunktzustands weicht dann erheblich von der tatsächlichen Position und Geschwindigkeit des CoM ab, was eine Regelung zur Stabilisierung des Roboters unmöglich macht.

- Verwendung des im NAO-Roboter vorhandenen Sensorsystems aus Beschleunigungs- und Rotationsratensensor, der so genannten IMU (siehe Abb. 3.1), zur Erfassung der Oberkörperbewegung. Diese Methode erlaubt eine Projektion des Schwerpunkts auf die Bodenebene auch dann, wenn kein Bodenkontakt besteht. Das Sensorsystem ist dafür stark rauschbehaftet. Ein weiterer Fehler wird durch Beschleunigungen des Roboters in der xy -Ebene eingeführt, da das Accelerometer nur die resultierende Beschleunigung des Roboters und nicht die Erdbeschleunigung direkt messen kann. Im *tuhhSDK* wird versucht, diese Nachteile durch die Anwendung eines Kalmanfilters zu kompensieren [4].

Die erste Messmethode erlaubt eine gute Messung nur unter den Umständen, dass der Roboter mit dem Referenzfuß vollständig auf dem Boden aufliegt. Dieser Bodenkontakt ist allerdings gleichzeitig auch eine Bedingung dafür, dass überhaupt eine Stabilisierung mit der *Ankle-Strategie* möglich ist.

Die Verwendung der IMU hat eine kürzere Reaktionszeit als die Verwendung der Gelenkdaten aus den Motoren (siehe auch Kapitel Messverzögerung), es gibt hier jedoch Probleme mit einem variablen Offset zum Ursprung, der eine häufige Kalibrierung des Systems erfordert [15]. Die Verwendung der IMU löst auch die Abhängigkeit von Motorgelenkdaten nicht vollständig auf, da die Position des CoM noch immer aus den Motorstellungen berechnet wird.

Messverzögerung

Durch die in Kapitel 5.2 durchgeführten Experimente und während der Implementierung eines Regelungsverfahrens für den realen Roboter stellte sich heraus, dass die Messwerte der Winkelpositionen der Motoren erst mit einer gewissen Verzögerung von ca. 0,08 s bis 0,1 s (siehe Abb. 3.2) zur Verfügung stehen.

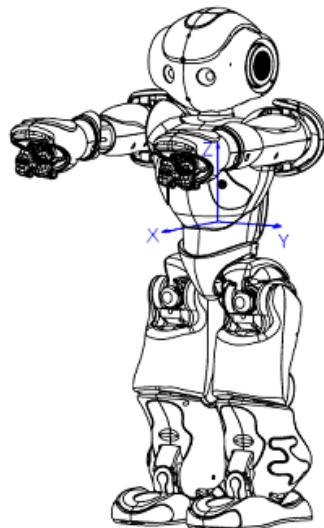


Abbildung 3.1: Position der IMU im Torso des Roboters [17].

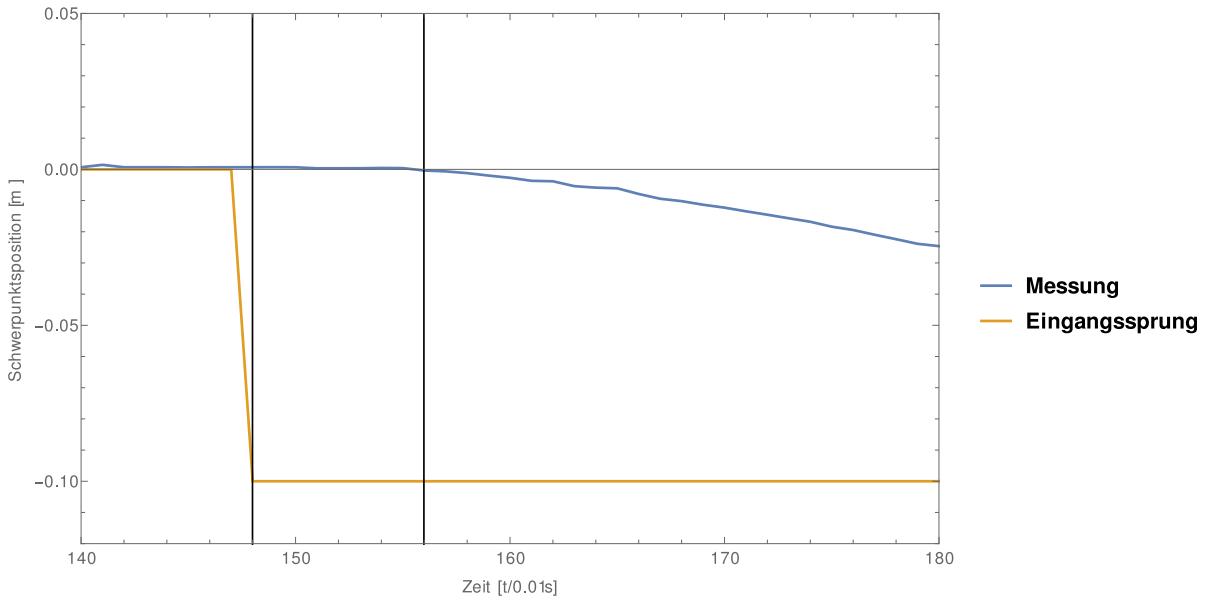


Abbildung 3.2: Die Messung reagiert verzögert auf eine Änderung des Systemzustands. Hier beträgt die Messverzögerung $T_{\text{latenz}} = 0,08 \text{ s}$

Wenn Stöße abgewehrt werden sollen, um ein Fallen des Roboters zu verhindern, ist es wichtig, dass die Reaktionszeit des Gesamtsystems möglichst kurz ist. Das hier verwendete NAO-Robotiksystem reagiert zwar physikalisch innerhalb von ca. 0,01 s auf die Änderung der Eingangsgröße, dem Regler liegen jedoch bei der Messung der Schwerpunktsposition nur verzögerte Messwerte vor. In Abb. 3.2 sind die Auswirkungen dieses Effekts sichtbar. Dieses Verhalten wurde zum Beispiel auch von Alcaraz et al. [18] beobachtet.

Verstreicht eine gewisse Zeit, bevor ein System auf eine Änderung der Eingangsgröße reagiert, spricht man von einem totzeitbehafteten System. Dies ist zum Beispiel der Fall, wenn in einer Anlage ein Förderband zum Einsatz kommt, welches ein Material transportiert. Wird als Reaktion auf eine Messgröße eine große Menge des Materials auf das Förderband geladen, wirkt sich diese Änderung trotzdem erst aus, nachdem das Material am Ende des Förderbands angekommen ist [19].

Das hier vorliegende System ist demzufolge nicht in erster Linie ein totzeitbehaftetes System, da die Änderung der Eingangsgröße physikalisch ohne Verzögerungen umgesetzt wird. Lediglich das Messen der Ausgangsgröße unterliegt einer deutlichen Latenz.

Durch Maßnahmen die der Rauschunterdrückung dienen, wie zum Beispiel eine Tiefpassfilterung oder die Verwendung von Kalman-Filtern, wird die Reaktionszeit des Messsignals unweigerlich weiter verschlechtert werden. Vor der Verwendung eines Filters ist daher zu klären, welcher Kompromiss aus Messrauschen und -verzögerung gemacht werden kann, um eine individuelle Problemstellung zu lösen.

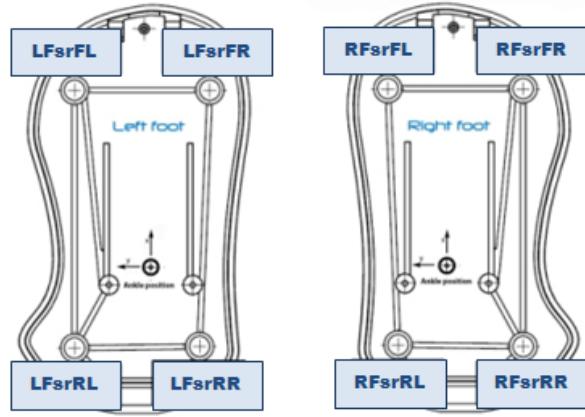


Abbildung 3.3: Lage der FSR-Sensoren in den Füßen des NAO-Robotiksystems [17].

Messung des Center of Pressure

Das in Kapitel 1.1 eingeführte Prinzip des ZMP kann theoretisch dazu verwendet werden, das dynamische Gleichgewicht des NAOs über die Regelung des Zero Moment Points zu erreichen. Die Position des CoP kann über *Force-Sensing Resistor*-Sensoren in den Füßen des NAOs messtechnisch erfasst werden (siehe Abb. 3.3). Die Messgenauigkeit der Sensoren wurde an vielen Stellen bereits bemängelt [4, 15, 17] und ist bereits systembedingt nicht besonders gut. Die Verwendung der FSRs wird in der Arbeit mit dem NAO-Robotiksystem daher selten angewendet um dynamische Balance zu halten. Üblicher ist bei der Regelung des ZMPs die Verwendung eines Modells, welches die Position des ZMPs anhand anderer Messungen vorhersagt.

Die Verwendung der FSR Sensoren als Grundlage einer feedbackbasierten Kraftregelung des Motors, wie sie für die Anwendung des VIIP erforderlich wäre, wäre prinzipiell denkbar. Mit einem Messfehler in einem Bereich von 20% und mehr [17, 15] wird eine kraftgeregelte Steuerung des Roboters jedoch erheblich erschwert.

Messrauschen

Die Messung der Gelenkwinkel in jedem Motor ist rauschbehaftet. Prinzipiell ist der Messwert für die Position der Gelenkwinkel gut. Die Genauigkeit der MREs wird vom Hersteller mit $0,01^\circ$ angegeben [17]. Das Rauschen der Messwerte wird durch die Ableitung des Gelenkwinkels zur Bestimmung einer Motorgeschwindigkeit erhöht. Da bei der Berechnung der Schwerpunktposition die Position aller Gelenkwinkel und die Massen aller Roboterteile berücksichtigt werden müssen, pflanzt sich bei der Messung der Schwerpunktsposition der Messfehler durch alle Gelenkketten bis zum Ursprung des Referenzkoordinatensystems fort.

Das Messrauschen zeigt sich besonders dann, wenn das ungefilterte Messsignal in einer geschlossenen Regelschleife an den Reglereingang zurückgeführt wird. Durch das große Geschwindigkeitsrauschen des Messwertes wird ein hochfrequentes Signal an den Fußmo-

tor gesendet. Dieses Eingangssignal führt zu einer starken Vibration des Roboters, die in machen Fällen einen Sturz des NAOs zur Folge hat.

3.2 Filterung des Messsignals

Um den im vorhergehenden Abschnitt beschriebenen Vibrationseffekt zu vermeiden, muss das Messsignal gefiltert und geglättet werden. Hierzu können verschiedene Verfahren angewandt werden, die sich in Komplexität, Erfolgswahrscheinlichkeit und Implementierungsaufwand unterscheiden.

Tiefpassfilter

Da das Messrauschen im Allgemeinen von höherer Frequenz ist als die Frequenz des Regelsignals [19], kann versucht werden, dieses durch eine Tiefpassfilterung zu unterdrücken. Eine mögliche Implementierung eines Tiefpassfilters für ein digitales System ist zum Beispiel die Berechnung eines gleitenden Mittelwerts. Der gleitende Mittelwert ist softwareseitig einfach zu implementieren und glättet den Verlauf von Messdaten durch die gleichzeitige Berücksichtigung der letzten n Messwerte für den Filterausgang. Ein beispielhafter Verlauf eines verrauschten Messsignals und dem entsprechend tiefpassgefilterten Signals ist in Abb. 3.4 abgebildet.

Für ein aktuelles Messsignal x_0 und die letzten $n - 1$ Messsignale x_i gilt dann für den Ausgang des Filters, y :

$$y = \frac{1}{n} \sum_{i=0}^n x_i \quad (3.2)$$

Diese Form des Tiefpassfilters kann durch die Verwendung einer *double-ended queue* leicht und performant implementiert werden. Durch die Veränderung des Parameters n lässt sich die Stärke der Tiefpassfilterung einstellen.

Die Zeitverzögerung, die durch das Filtern des Messsignals eingebracht wird, beträgt im Mittel $T_{\text{delay}} = n \cdot T_s$. Diese Verzögerung erhöht die Latenz zwischen Messung und realem Signalverlauf zusätzlich. Bei der hier verwendeten Filtergröße von $n = 15$ folgt mit der Samplingzeit der Messwerte $T_s = 0,01$ s eine Verzögerung von $T_{\text{delay}} = 0,15$ s

Kalman-Filter

Zur Unterdrückung des Messrauschen der Schwerpunktposition kann ein Kalman-Filter eingesetzt werden. Die Dimensionierung eines solchen Filters, das auf dem LIPM basiert erfolgte im Nachgang seiner Diplomarbeit bereits durch Kaufmann [4] und konnte durch die Verwendung des Softwareframeworks des *RobotING@TUHH e.V.* leicht übernommen werden. Ein Kalman-Filter setzt die Genauigkeit von Messung und Modell in Relation und berücksichtigt beide bei der Bestimmung des Nutzsignals.

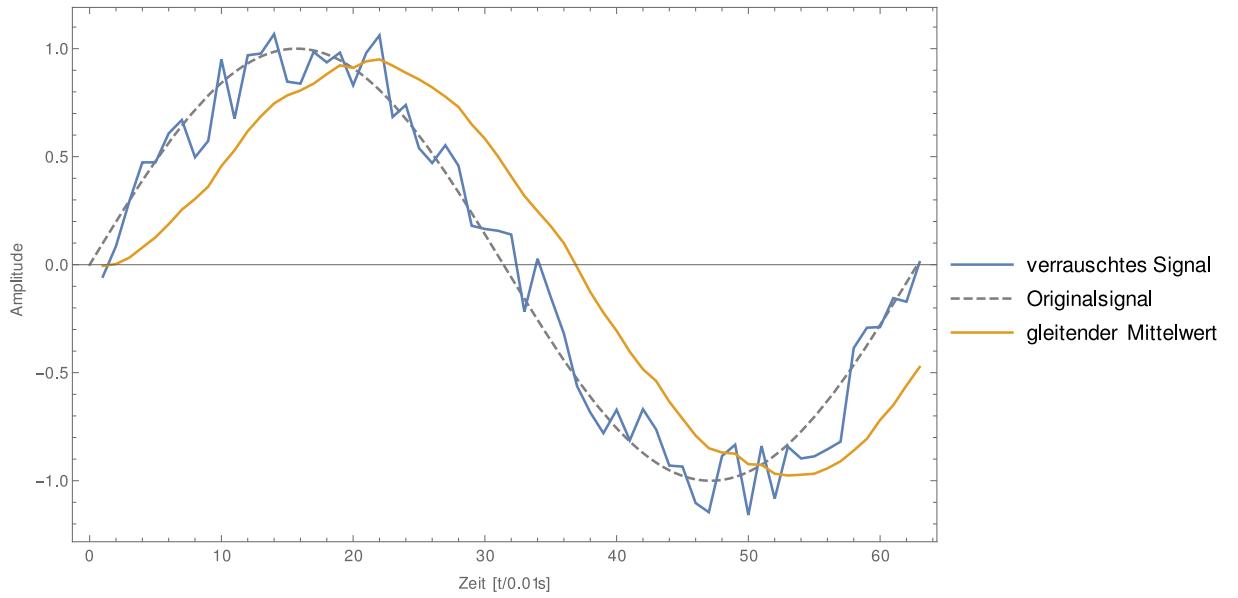


Abbildung 3.4: Glättung eines Messsignals mit einem gleitenden Mittelwert der Größe $n = 10$ als Tiefpassfilter. Das gefilterte Signal hat eine deutliche Zeitverzögerung.

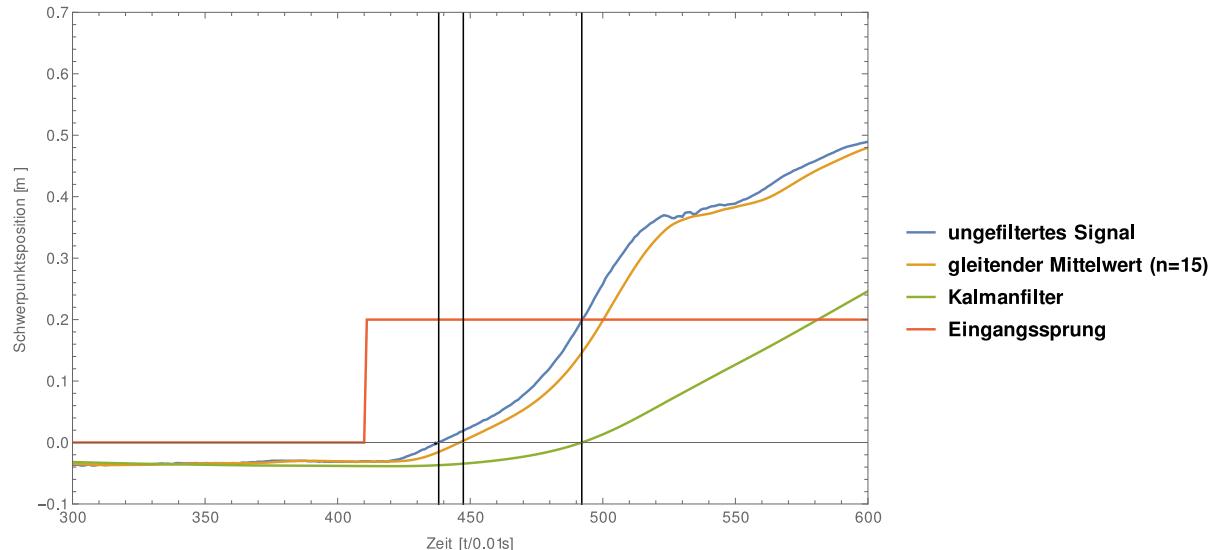


Abbildung 3.5: Die Sprungantwort des NAOs mit den Ausgangssignalen verschiedener Filter.

Das Kalman-Filter funktioniert rekursiv, das heißt, dass das Filtersystem für eine Aktualisierung seines Ausgangswert nur den letzten aktuellen Messwert benötigt und die vorhergehenden Messwerte im Systemzustand gespeichert werden.

Vergleich der beiden Filtermethoden

In Abb. 3.5 zeigt sich, dass das Tiefpassfilter eine größere Ähnlichkeit des Ausgangsverlauf zum Signalverlauf aufzeigt. Das Kalman-Filter orientiert sich in der hier getesteten Konfiguration eher am Modell und hat lediglich einen qualitativ übereinstimmenden Verlauf mit dem Messsignal auf. Durch die Einstellung der Kovarianzmatrizen des Kalmanfilters lässt sich das Vertrauen in die Messung oder in das Modell variieren. Gute Werte für die Kovarianzmatrizen lassen sich zum Beispiel experimentell bestimmen [4].

Die Signalverzögerung durch das Tiefpassfilter wurde bereits mit 0,15 s angegeben. Experimentell konnte eine große Latenz des Kalman-Filters in dieser Konfiguration von ca. 0,5 s bestimmt werden. Die Verzögerung, die durch das Kalman-Filter entsteht, ist nicht konstant.

Zur Filterung des Messsignals kommen bei der Implementierung einer Regelung im Rahmen dieser Bachelorarbeit beide vorgestellten Filter zur Anwendung. Zur Filterung des Messwerts der Schwerpunktsposition wird das Kalman-Filter verwendet. Die Geschwindigkeit des Schwerpunkts wird direkt aus dem Messsignal berechnet und mit einem gleitenden Mittelwert über $n = 15$ Samples geglättet.

3.3 Messdatenübertragung und -visualisierung

Die Aufzeichnung und Visualisierung von Messdaten ist eine essentielle Voraussetzung um empirische Schlüsse aus Entwicklungen und Experimenten ziehen zu können. Diese Anforderung wird von einer Software adressiert, die in Zusammenarbeit mit anderen Mitgliedern der Gruppe *RobotING@TUHH e.V.* entwickelt wurde¹.

Zwar wurde von Kaufmann in [4] bereits eine Messdatenaufzeichnung *NaoObserve* für den NAO entwickelt, jedoch unterstützt diese, wie in der Arbeit schon beschrieben, weder die Aufzeichnung von beliebigen benutzerkonfigurierbaren Messdaten, noch die selektive Übertragung von Messdaten an unterschiedliche Clients.

Protokoll zur Übertragung einer Auswahl von beliebigen Werten in Echtzeit an verschiedene Endpunkte

Die Speicherung und Übertragung von Messungsergebnissen erfolgt in mehreren Schritten. So werden alle Messungen zunächst temporär auf dem Roboter gespeichert und danach in einem einzigen Paket auf die Endpunkte übertragen.

Die Erfassung beliebiger Messdaten erfolgt softwareintern in einem *Dictionary*-Container, in dem menschenlesbaren Schlüsseln Werte verschiedenen Typs zugeordnet werden können. Das hier vorgestellte Kommunikationsprotokoll soll eine generische Lösung zur Übertragung beliebiger Daten in möglichst großer Menge darstellen und muss deshalb entsprechende Techniken zur Verfügung stellen um einerseits zahlreiche Messdaten vorhalten zu können, jedoch andererseits zur Datenübertragung in Echtzeit eine möglichst geringe

¹Maßgeblich hat Robert Oehlmann zur Entwicklung beigetragen.

Übertragungsrate benötigen. Aus diesem Grund ist das verwendete Debug-Protokoll so aufgebaut, dass nur die Messwerte übertragen werden, die von einem Endpunkt – also zum Beispiel von einer Visualisierungsanwendung – angefordert wurden. Ferner wurde die Möglichkeit vorgesehen, nur Daten zu Übertragen, die sich seit der letzten Übertragung geändert haben. Eine Implementierung dieses Features wurde jedoch noch nicht durchgeführt.

Zur Datenübertragung wird in diesem Protokoll Transmission Control Protocol (TCP) verwendet, da das Auseinanderhalten verschiedener Endpunkte durch das explizite Herstellen von TCP-Verbindungen bereits inhärent gegeben ist. Jeder Endpunkt kann beim Server, also beim NAO, eine Liste der Schlüssel anfordern, die zum Abruf zur Verfügung stehen. Durch das Senden eines **Subscribe**-Befehls kann der Client Updates zu einem Key-Value-Paar abonnieren, von nun an sendet der Server in Echtzeit Updates des Wertes an den Endpunkt. Diese Updates enthalten neben dem Key-Value-Tupel noch den Zeitpunkt, zu dem der Messwert auf dem Server erfasst wurde, sowie Informationen darüber, von welchem Datentyp der gesendete Wert ist.

Benötigt der Endpunkt keine Updates zu einem Messwert mehr, zum Beispiel weil die Messaufzeichnung gestoppt wurde oder das Experiment beendet worden ist, kann durch das Senden eines **Unsubscribe**-Befehls das Abonnement beendet werden.

Anders als das *NaoObserve* Modul wird das Debug Protokoll in einem anderen Prozess als die Bewegungssteuerung auf dem NAO ausgeführt. Das Sammeln und Übertragen einer großen Menge von Messwerten im Kontext der zu überwachenden Anwendung hat somit keinen nennenswerten Einfluss auf die Performance derselben und beeinflusst allenfalls das Echtzeitverhalten der Key-Value-Updates. Zum Zeitpunkt der Messungen, die im Rahmen dieser Bachelorarbeit durchgeführt wurden, konnten ohne Verlangsamung des Roboterzyklus' von 100 Hz mehr als 100 Werte gleichzeitig empfangen werden.

Die Updates selbst werden menschenlesbar im standardisierten JSON-Format [20] übertragen, was eine Visualisierung, Auswertung oder anderweitige Weiterverarbeitung der Daten in faktisch jeder Programmiersprache auf nahezu jeder Plattform ermöglicht.

Visualisierung der Messdaten

Die Messdaten werden während der Nutzung des Debugprotokolls in einer eigens entwickelten, auf *JavaScript* und *HTML5* basierenden, Weboberfläche in Echtzeit angezeigt. Bei Bedarf kann eine Messreihe aufgenommen und in einem Schritt in ein mit *Matlab* oder *Mathematica* kompatibles Textformat exportiert werden. In dieser Software können die Messdaten dann einfach weiterverwendet werden um diese tiefer zu analysieren oder Berechnungen mit ihnen durchzuführen.



Abbildung 3.6: Webbasierte Benutzeroberfläche des Debug-Tools. Die Nutzung einer Weboberfläche ermöglicht die Echtzeitauswertung von Messdaten auf beliebigen Endgeräten.

4 Kriterien für dynamisches Gleichgewicht

Die in Kapitel 2 erarbeiteten Modelle wurden bisher für den Fall betrachtet, dass der Roboter nur in einem Punkt Kontakt mit dem Boden hat. Ein echter Roboter hat einen Fuß, welcher in einer Fläche mit dem Fußboden in Kontakt ist. Die Größe dieser Fläche beeinflusst die Stabilität des Roboters. Je größer diese Fläche ist, desto mehr Moment kann die Bodenreaktionskraft auf den Roboter ausüben und desto größer kann also eine von außen wirkende Störung sein, ohne dass der Roboter umfällt.

In diesem Kapitel wird anhand theoretischer Betrachtung zum dynamischen Gleichgewicht humanoider Roboter hergeleitet, welche Auswirkungen verschiedene Balancierungsstrategien auf den Zustand des Roboters haben können und in welchen Fällen eine Balancierungsstrategie erfolgsversprechend sein kann.

4.1 Erweiterung des AMPM um das Konzept des ZMP

Möchte man das AMPM so erweitern, dass der ZMP berücksichtigt wird, muss ein Supportpolygon definiert werden. Für die in Kapitel 2.2 hergeleiteten Formeln stellt das Supportpolygon ein Intervall entlang der x -Achse dar (siehe Abb. 4.3).

Wenn der CoP nicht in der Projektion des CoM liegt, entsteht ein durch das Kräftepaar aus Gewichts- und Bodenreaktionskraft hervorgerufenes Moment. Durch ein Momentengleichgewicht im Fußgelenk erhält man (siehe Abb. 4.1):

$$F_x z = F_z (x_{\text{CoM}} - x_{\text{CoP}}) - \tau_h \quad (4.1)$$

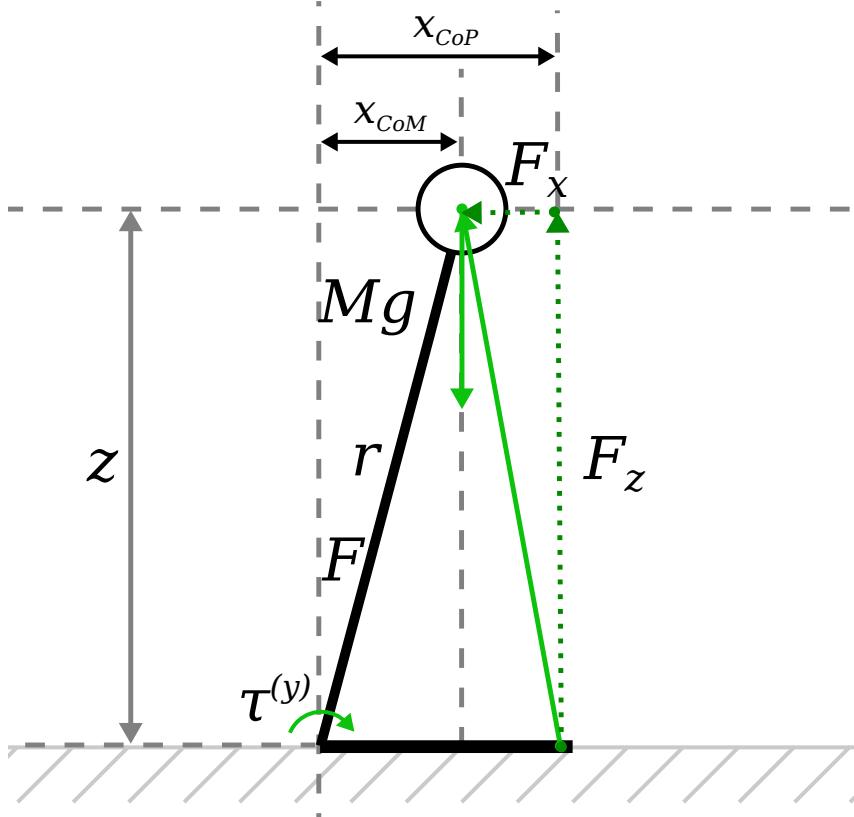


Abbildung 4.1: Darstellung der Bodenreaktionskraft des AMPM. Die Bodenreaktionskraft greift im ZMP an.

Die Voraussetzungen des LIPM aus (2.7) werden nun eingesetzt und der Impulssatz in x -Richtung angewendet. Damit wird (4.1) zu

$$\ddot{x} = \frac{g}{z} (x_{\text{CoM}} - x_{\text{CoP}}) - \frac{\tau_h}{Mz} \quad (4.2)$$

Aus Gründen der Vereinfachung wird die Position des Schwerpunkts x_{CoM} im weiteren Verlauf dieser Arbeit mit x benannt.

Wendet man nun die Definition des dynamischen Gleichgewichts auf dieses Differentialgleichung, kann man Grenzen für die Zustandsgrößen des Modells ermitteln, in denen das Modell mit verschiedenen Regelungsstrategien stabilisierbar ist [8].

4.2 Stabilisierbarkeit mit der Ankle-Strategie

Da das maximal mögliche Moment, welches im Fußgelenk des Roboters wirken kann durch die Ausdehnung des Supportpolygons beschränkt ist, lassen sich Grenzen für die Zustände des Roboters berechnen, für die das LIPM-System gerade noch stabilisierbar ist [8]. Hierzu wird eine Energiebetrachtung durchgeführt, in welcher das Konzept des *Capture Points* eingeführt wird. Am Ende dieses Abschnittes steht eine Entscheidungsregion, mit der

bewertet werden kann, ob die *Ankle-Strategie* für eine gegebene Belastung ausreicht, um den Roboter zu stabilisieren.

Orbitale Energie des LIPM und Capture Point

Wird die Gleichung (2.11) genauer betrachtet, kann man Ähnlichkeiten zu einem Masse-Feder-System feststellen. Die Bewegungsgleichung eines Masse-Feder-Systems lautet:

$$M\ddot{x} = -kx \quad (4.3)$$

Das LIPM ist anschaulich betrachtet also ein solches Masse-Feder-System mit einer Masse $M = 1$ und der Federkonstante $k = -\frac{g}{z}$ [5, 9]. Von einem solchen System lässt sich die Gesamtenergie ausrechnen.

$$E_{\text{LIPM}} = \underbrace{\frac{1}{2}\dot{x}^2}_{E_{\text{kin}}} - \underbrace{\frac{g}{2z}x^2}_{E_{\text{pot}}} \quad (4.4)$$

Man spricht bei einem System von einem Gleichgewichtszustand, wenn der Energiegehalt dieses Systems am geringsten ist. In diesem Fall muss also gelten:

$$E_{\text{LIPM}} \stackrel{!}{=} 0 \quad (4.5)$$

Für das LIPM lässt sich nun ein Punkt im Zustandsraum berechnen, für den es sich im Gleichgewicht befindet. Pratt et al. [9] nennen diesen Punkt den *Capture Point*.

$$E_{\text{LIPM}} = \frac{1}{2} \left(\dot{x}^2 - \frac{g}{z}x^2 \right) \stackrel{!}{=} 0 \quad (4.6)$$

$$\Leftrightarrow x_{\text{capture}} = \pm \dot{x} \sqrt{\frac{z}{g}} \quad (4.7)$$

Bewegt sich der CoM auf den CoP zu, so ist das System stabil. Bewegt sich der CoM jedoch vom CoP weg (negative Lösung), dann kann das System nicht stabil sein. Demzufolge muss die positive Lösung aus (4.6) verwendet werden, um die Stabile Region des Modells berechnen zu können.

Mit der Substitution $\omega^2 = \frac{g}{z}$ folgt dann:

$$x_{\text{capture}} = \frac{\dot{x}}{\omega} \quad (4.8)$$

Die Position des *Capture Points* ist eine wichtige Kenngröße der dynamischen Balance humanoider Roboter. Durch die Analyse von verschiedenen Balancestrategien werden im weiteren Verlauf dieses Kapitels verschiedene Entscheidungshorizonte hergeleitet. So wird es in der Regelung des Systems möglich sein, in verschiedenen Situationen unterschiedliche Verfahren anzuwenden, die den Roboter stabilisieren können.

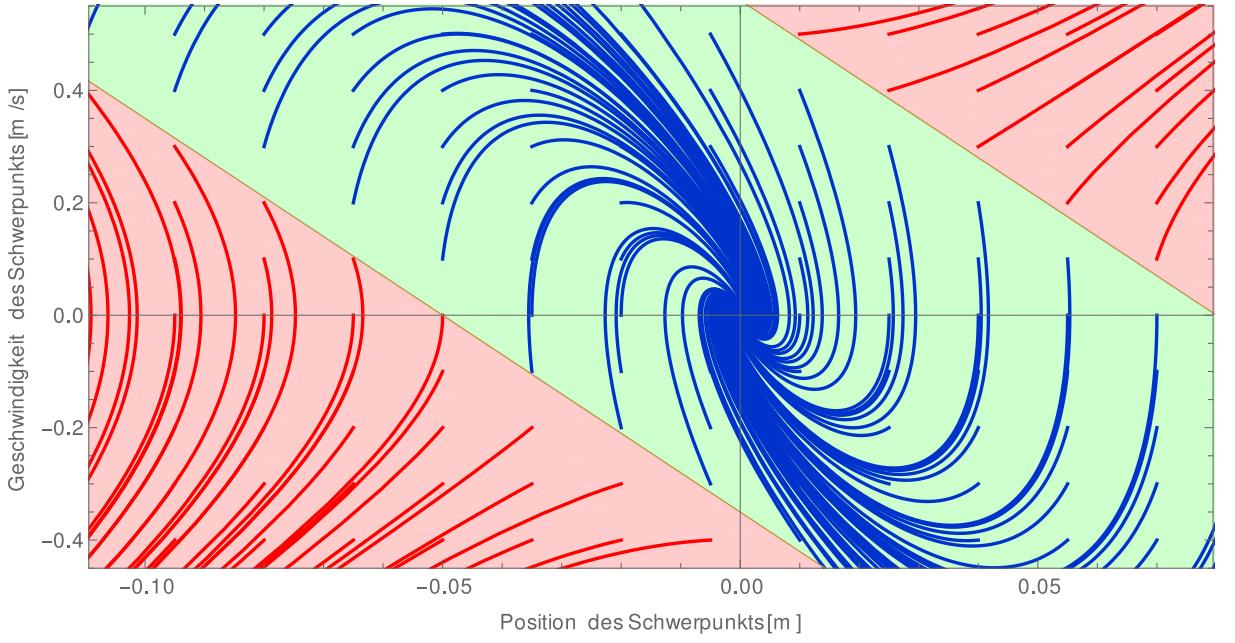


Abbildung 4.2: Ergebnisse der Simulation eines LIPM, welches mit der *Ankle-Strategie* stabilisiert wird. Als Anfangszustandsvektor des Pendels wurden Punkte im hier dargestellten Bereich der Zustandsebene verwendet.

Anforderungen an die Position der Capture Points

Wie in Abb. 4.2 zu sehen, kann die *Ankle-Strategie* den Roboter nicht für beliebige Situationen stabilisieren. Die Grenzen des Bereichs, in dem die *Ankle-Strategie* ausreicht, um das Gleichgewicht des LIPMs sicherzustellen, werden in diesem Abschnitt hergeleitet.

Die Position des *Capture Points* muss, um die Balance des Roboters sicherzustellen, so liegen, dass die aktuatorbedingten Grenzen des NAO-Robotiksystems nicht verletzt werden. Die Beschränkung des Supportpolygons ist für den Fall, dass nur ein Fuß des Roboters Kontakt mit dem Boden hat durch die Geometrie des Roboterfußes beschrieben. Aus den Abmaßen der Roboterfußes resultiert ein maximales Moment, welches durch die Bodenreaktionskraft aufgebracht werden kann, die Bodenreaktionskraft selbst ist durch die Gewichtskraft des Roboters (2.7) vorgegeben [21].

Betrachtet man zunächst nur das Verhalten des NAOs in der xz -Ebene, so kann durch die Ausdehung des Supportpolygons ein Intervall I_S auf der x -Achse angegeben werden (Abb. 4.3).

$$I_S := [\delta^-, \delta^+] \quad (4.9)$$

Für die Position des *Capture Points* erhält man durch das Anwenden dieser Grenzen mit

$$\delta^- < \frac{\dot{x}}{\omega} + x < \delta^+ \quad (4.10)$$

eine Region im Zustandsraum des LIPM, für den der Roboter unter alleiniger Anwendung

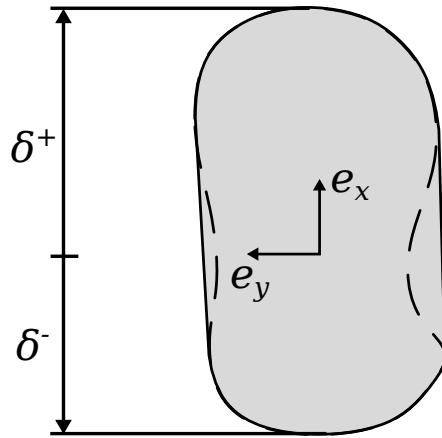


Abbildung 4.3: Darstellung der Grenzen der stabilen Region für den Fall, dass nur ein Fuß des Roboters Bodenkontakt hat.

der Ankle-Strategie stabilisierbar ist [8]. Verlässt der Zustand des Roboters die Region (4.10) ist es erforderlich, eine andere Regelstrategie, zum Beispiel das Aufbringen eines Moments durch die Arme oder die Hüfte des Roboters oder das gezielte Setzen eines Ausfallschrittes anzuwenden, welche die stabile Region entsprechend erweitert (siehe Kapitel 4.3).

4.3 Stabilisierbarkeit unter Verwendung der Flywheel-Strategie

Die Verwendung des Oberkörpers des NAOs als Schwungmasse stellt eine Erhöhung der Anzahl der Eingangsgrößen des betrachteten Regelungssystems dar. Mit Methoden der Regelungstechnik für Multiple Input Single Output (MISO)-Systeme wäre es möglich, diese beiden Eingangsgrößen gleichzeitig zu nutzen, um das dynamische Gleichgewicht des Roboters zu regeln. Im Rahmen dieser Arbeit soll das Gesamtsystem jedoch als Single Input Single Output (SISO)-System betrachtet werden, weswegen der Einsatz der Schwungmasse nur dann erfolgt, wenn der Roboter unter alleiniger Verwendung der Ankle-Strategie, wie in Kapitel 4.2 gezeigt, nicht mehr stabilisierbar ist.

Das Ziel der Verwendung des Flywheels ist, den Roboter durch die Aktuation des Oberkörpers wieder in einen Zustand zu bringen, in dem er mit der Ankle-Strategie stabilisiert werden kann [8]. Die exakte Position in der stabilen Region ist zunächst nicht weiter relevant, wichtig ist lediglich, dass der Zustand des Roboters in diese Region gebracht wird. Aus diesem Grund kann eine erhebliche Verbesserung der gesamten Stabilisierbarkeit des Roboters erwartet werden, wenn zur Steuerung des Roboteroberkörpers ein *Bang-Bang Controller*² verwendet wird.

²dt.: Zweipunktregler

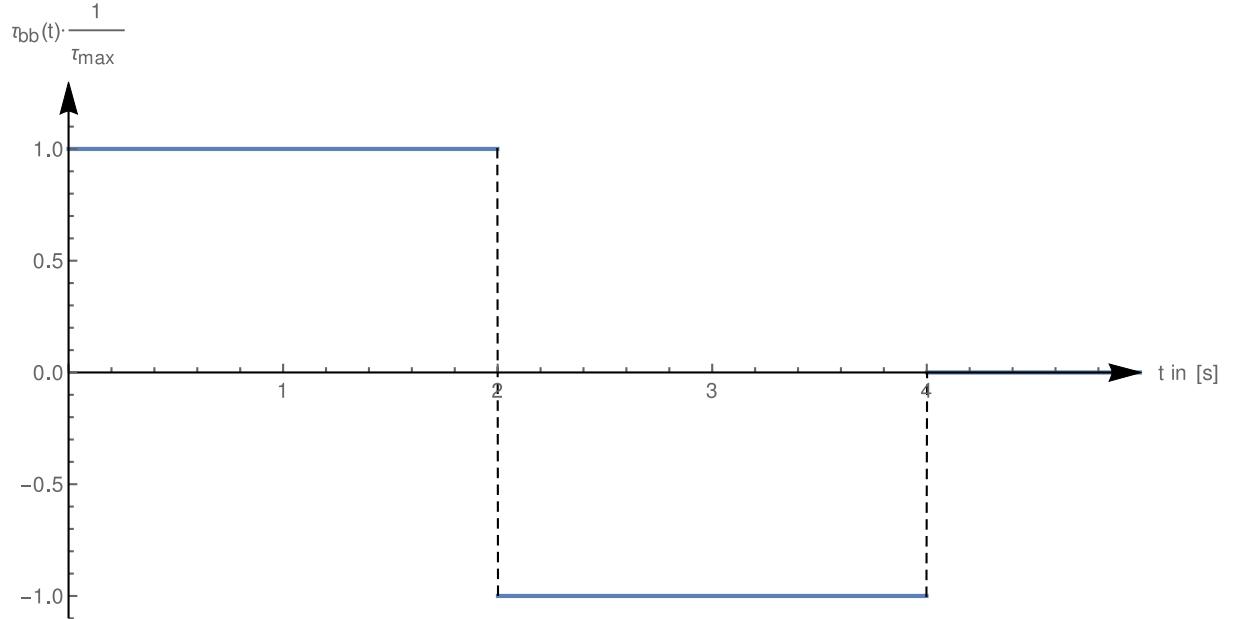


Abbildung 4.4: Zeitlicher Verlauf des Steuersignals der Zweipunktregelung des Flywheels mit den Parametern $T_1 = 2\text{s}$ und $T_2 - T_1 = 2\text{s}$.

Limitierungen des Flywheels

Die Anwendung der *Flywheel-Strategie* auf dem NAO ist auf verschiedene Art und Weise limitiert. So kann der Oberkörper nur bis zu einem maximalen Winkel von $\pm 25^\circ$ ausgelenkt werden, ohne die Stabilität des Roboters negativ zu beeinträchtigen. Auch die Masse des Schwungrades kann nicht verändert werden. Es ist ferner zwingend erforderlich, dass nach dem Einbringen eines positiven Moments über den Oberkörper ein betragsmäßig gleich großes, negatives Moment folgt, um die Rotation des Oberkörpers wieder zu stoppen.

Der Bang-Bang Controller

Ein *Bang-Bang Controller* ist ein Regler, der, einmal aktiviert, zunächst eine maximale Aussteuerung in die eine Richtung für eine Zeit T_1 hat, dann die gleiche maximale Aussteuerung in die andere Richtung bis zum Zeitpunkt T_2 hält und danach wieder in die Nullposition zurückspringt (siehe Abb. 4.4) [19, 21]. Der *Bang-Bang Controller* kann so dimensioniert werden, dass er die durch das Robotiksystem gegebenen Begrenzungen des Flywheels einhält.

$$\tau_{bb}(t) = \tau_{\max} \sigma(t) - 2\tau_{\max} \sigma(t - T_1) + \tau_{\max} \sigma(t - T_2) \quad (4.11)$$

Die Wahl eines solchen Eingangssignals für die Flywheelbewegung wurde sowohl von [8], als auch von [9] erfolgreich eingesetzt.

Stabile Region unter Verwendung des Flywheels

Die Bewegungsgleichungen für das AMPM wurden bereits in Kapitel 2.2 hergeleitet. Analog zum Vorgehen bei der Bestimmung des *Capture-Points* für die Ankle-Strategie lässt sich nun auch ein Ausdruck für einen solchen Punkt unter Anwendung der Flywheel-Strategie bestimmen.

Löst man die Differentialgleichung (2.15) zum Beispiel, indem man die Übertragungsfunktion im Frequenzbereich bildet:

$$\ddot{x} = \frac{g}{z}x - \frac{1}{Mz}\tau_h \quad \circ\bullet \quad s^2X(s) = \frac{g}{z}X(s) - \frac{1}{Mz}\tau_h(s) \quad (4.12)$$

$$G(s) = \frac{X(s)}{\tau_h(s)} = -\frac{1}{Mz} \left(\frac{1}{s^2 - \omega^2} \right) \quad (4.13)$$

wobei $\omega^2 = \frac{g}{z}$.

Die *Zero-State-Antwort* des Systems für den Eingang (4.11) lässt sich durch eine Faltung des Eingangssignals mit der Übertragungsfunktion im Zeitbereich, also eine Multiplikation beider Signale im Frequenzbereich ermitteln [19]. Hierzu wird zunächst (4.11) in den Frequenzbereich transformiert:

$$\tau_{bb}(t) \quad \circ\bullet \quad \tau_{bb}(s) = \left(\frac{e^{-T_2 s}}{s} - \frac{2e^{-T_1 s}}{s} + \frac{1}{s} \right) \cdot \tau_{\max} \quad (4.14)$$

Die Multiplikation im Frequenzbereich

$$X_{zsr}(s) = G(s) \cdot \tau_{bb}(s) \quad (4.15)$$

Rücktransformation (4.16) lässt sich nun mit z.B. *Wolfram Mathematica* symbolisch lösen (siehe Anhang A).

$$X_{zsr}(s) \quad \bullet\circ \quad x_{zsr}(t) \quad (4.16)$$

Das Ergebnis lautet

$$x_{zsr}(t) = -\frac{\tau_{\max}}{Mz\omega^2} [(\cosh(\omega t) - 1)\sigma(t) - 2(\cosh(\omega(t - T_1)) - 1)\sigma(t - T_1) \\ + (\cosh(\omega(t - T_2))\sigma(t - T_2))] \quad (4.17)$$

und durch Ableiten:

$$\dot{x}_{zsr}(t) = -\frac{\tau_{\max}}{Mz\omega} [(\sinh(\omega t) - 1)\sigma(t) - 2(\sinh(\omega(t - T_1)) - 1)\sigma(t - T_1) \\ + (\sinh(\omega(t - T_2))\sigma(t - T_2))] \quad (4.18)$$

Die Gleichungen (4.21) und (4.22) ergeben zusammen mit der *Zero-Input-Response* des Systems [4, 9]:

$$x_{zir}(t) = x_0 \cosh(\omega t) + \frac{1}{\omega} \dot{x}_0 \sinh(\omega t) \quad (4.19)$$

$$\dot{x}_{\text{zir}}(t) = \omega x_0 \sinh(\omega t) + \dot{x}_0 \cosh(\omega t) \quad (4.20)$$

Ein Gleichungssystem für das Verhalten des Schwerpunktzzustands für den Eingangswert (4.11). Da das Ziel der Flywheel-Strategie ist, den Zustand des Roboters nach der Aktuation des Oberkörpers, also zum Zeitpunkt T_2 , in einen Bereich zu bringen, der mit der *Ankle-Strategie* stabilisierbar ist, wird diese Funktion hier auch nur zum Zeitpunkt T_2 ausgewertet.

$$\begin{aligned} x(T_2) = & \frac{1}{\omega} \dot{x}_0 \sinh(\omega T_2) - \frac{\tau_{\max}}{M z \omega^2} [\cosh(\omega T_2) - 2 \cosh(\omega(T_2 - T_1)) + 1] \\ & + x_0 \cosh(\omega T_2) \end{aligned} \quad (4.21)$$

$$\begin{aligned} \dot{x}(T_2) = & \dot{x}_0 \cosh(\omega T_2) - \frac{\tau_{\max}}{M z \omega} [\sinh(\omega T_2) - 2 \sinh(\omega(T_2 - T_1)) + 1] \\ & + x_0 \omega \sinh(\omega T_2) \end{aligned} \quad (4.22)$$

Zur Bestimmung des *Capture-Points* müssen die gleichen Bedingungen gelten, wie schon bei der Berechnung des *Capture-Points* für das LIPM in (4.8). Den Ausdruck für den *Capture Point* für das AMPM leiten Pratt et al. [9] durch Einsetzen von (4.21) für x_{capture} und (4.22) für \dot{x} in die Gleichung (4.8) und das anschließende Auflösen nach x_0 her.

$$x_{\text{capture}} = \frac{1}{\omega} \dot{x} - \frac{\tau_{\max}}{M g} \left[\frac{(e^{\omega T_2} - 2e^{\omega(T_2 - T_1)} + 1)}{e^{\omega T_2}} \right] \quad (4.23)$$

Der neue *Capture-Point* kann nun in (4.10) eingesetzt werden. Um die maximal mögliche Erweiterung der stabilisierbaren Region in der Zustandsebene zu erhalten, wird angenommen, dass die vorgegebenen Grenzen des Flywheels voll ausgenutzt werden, d.h. dass die Flywheelbewegung mit maximalem Moment τ_{\max} einen Zeitraum von $2 \cdot T_2 = T_{\max}$ in Anspruch nimmt [8]. Damit sichergestellt ist, dass die Orientierung des Roboteroberkörpers zum Zeitpunkt $t = T_2$ gleich der Orientierung zum Zeitpunkt $t = 0$ ist, muss $2 \cdot T_1 = T_2$ gelten. Man erhält dann

$$\delta^- - \frac{\tau_{\max}}{M g} (e^{2\omega T_{\max}} - 2e^{\omega T_{\max}} + 1) < x + \frac{\dot{x}}{\omega} < \delta^+ + \frac{\tau_{\max}}{M g} (e^{2\omega T_{\max}} - 2e^{\omega T_{\max}} + 1) \quad (4.24)$$

und unter Anwendung der zweiten binomischen Formel:

$$\delta^- - \frac{\tau_{\max}}{M g} (e^{\omega T_{\max}} - 1)^2 < x + \frac{\dot{x}}{\omega} < \delta^+ + \frac{\tau_{\max}}{M g} (e^{\omega T_{\max}} - 1)^2 \quad (4.25)$$

eine anschauliche Ungleichung, die die Erweiterung der stabilen Region abbildet (siehe Abb. 4.5).

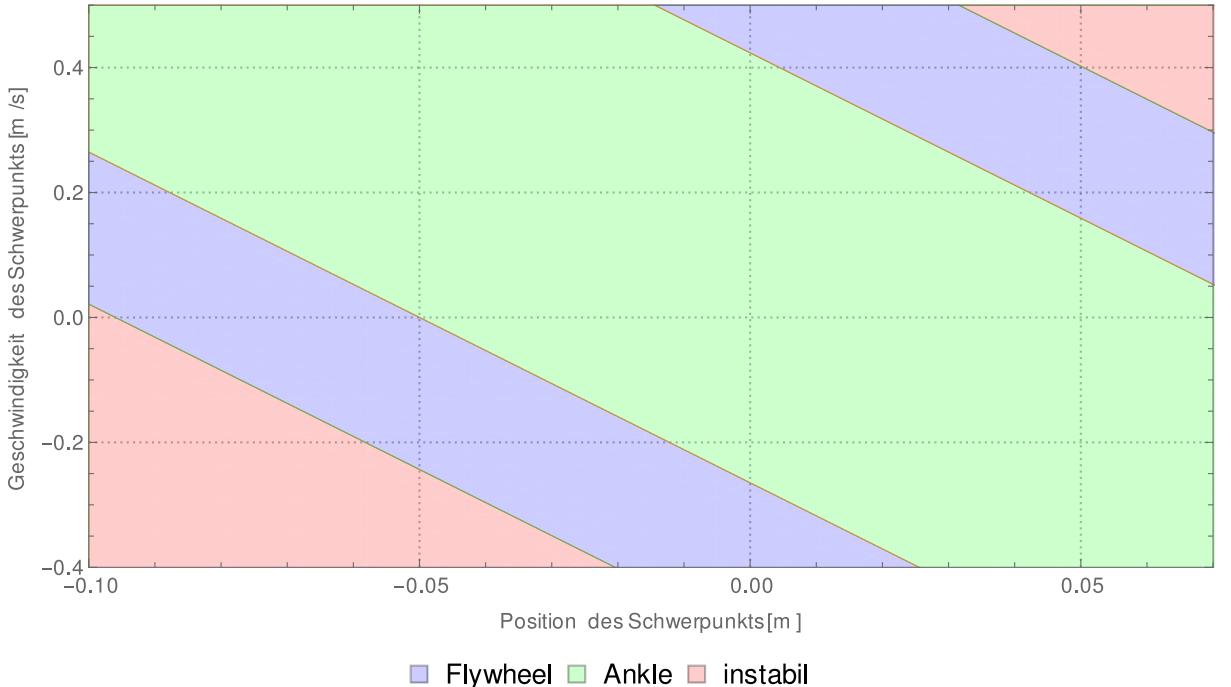


Abbildung 4.5: Die erweiterte Region in der der Roboter stabilisierbar ist, wenn die *Flywheel-Strategie* zusätzlich zur gesättigten *Ankle-Strategie* angewendet wird ist hier blau gekennzeichnet.

4.4 Anwendung der Entscheidungsregionen auf verschiedene Posen

Die in diesem Kapitel hergeleiteten Entscheidungsregionen lassen sich durch geeignete Parametrisierung zunächst auf beliebige Posen anwenden. Sollen die Parameter dynamisch für beliebige Posen bestimmt werden, ist es erforderlich zu wissen, welche Änderungen sich auf welche Parameter auswirken.

In den Entscheidungshorizont (4.10) gehen die geometrischen Grenzen des Supportpolygons und die Höhe des Schwerpunkts ein. Beide dieser Größen können während einer Bewegung berechnet werden. Die Höhe des Schwerpunkts über dem Boden kann auf dem NAO-Robotiksystem, wie in Kapitel 3 erläutert, über die Winkel aller Motoren berechnet werden.

Die Grenzen des Supportpolygons (4.9) können ebenfalls bestimmt werden, wenn bekannt ist, welche Körperteile des Roboters mit dem Boden in Kontakt sind. Vereinfachend wird hier die Annahme getroffen, dass eine Stabilisierung des Roboters nur erforderlich ist, wenn der Roboter ausschließlich mit den Füßen den Boden berührt. Der Abstand beider Füße zueinander lässt sich nun über eine Vektordifferenz berechnen. Hierbei ist \vec{p}_l der Positionsvektor des linken Fußes in Torsokoordinaten und \vec{p}_r die Position des rechten Fußes in den gleichen Koordinaten.

Der Abstand beider Füße kann dann aus der Differenz beider Positionsvektoren berechnet werden.

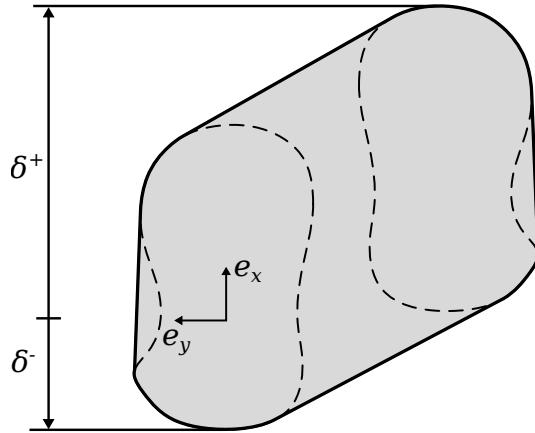


Abbildung 4.6: Darstellung des Supportpolygons für beide Füße. Die Grenzen werden relativ zum hinteren der beiden Füße angegeben.

$$\vec{p}_\Delta = \vec{p}_r - \vec{p}_l \quad (4.26)$$

Die Grenzen des Supportpolygons können nun durch den Versatz der beiden Fußmittelpunkt in der jeweiligen Koordinatenrichtung erweitert werden. In Gleichung (4.27) ist die Erweiterung des Entscheidungshorizontes (4.10) für die x -Richtung dargestellt. $\langle \cdot, \cdot \rangle$ ist hier das Standardskalarprodukt.

$$\delta_{\text{neu}}^+ = \langle (1, 0, 0)^T, p_\Delta \rangle + \delta^+ \quad (4.27)$$

$$\delta_{\text{neu}}^- = \delta^- \quad (4.28)$$

Die Grenzen für das Supportpolygon werden hier stets relativ zum hinteren der beiden Füße angegeben (siehe Abb. 4.6).

Für den Fall, dass der Roboter mit nur einem Fuß Kontakt zum Boden hat, entspricht das Supportpolygon der konvexen Hülle dieses einzelnen Fußes (siehe Abb. 1.2a). Zur Bestimmung des Kontaktzustandes beider Füße ist die Präzision der FSRs hinreichend genau.

5 Modellverifizierung

Um abschätzen zu können, wie der Kompromiss aus Modellgenauigkeit und -komplexität konditioniert ist, ist es notwendig das Verhalten des Modells in der Simulation mit dem Verhalten des realen Roboters unter gleichen Rahmenbedingungen zu vergleichen. Dazu müssen Experimente durchgeführt werden, deren Ergebnisse gemessen und aufgezeichnet werden. Hierzu werden die Messsysteme des NAO-Robotiksystems verwendet, die in Kapitel 3 vorgestellt und bewertet wurden. Zum Vergleich des Modells mit dem realen Roboter wird die Messung einer Sprungantwort des Systems durchgeführt und diese mit dem Verlauf der Sprungantwort des LIPMs verglichen.

Die Messung einer Sprungantwort ist zum Vergleich zweier Systeme geeignet, da das Spektrum einer Sprungfunktion $\sigma(t)$ nicht bandbegrenzt ist [19].

5.1 Betrachtung der Übertragungsfunktion des LIPM

Im ersten Schritt der Modellverifizierung soll das LIPM, welches in Kapitel 2.1 eingeführt wurde, mit dem tatsächlichen dynamischen Verhalten des NAO-Roboters verglichen werden. Die Simulation des Modells wird in *Simulink* durchgeführt. Es gibt zahlreiche Möglichkeiten, ein Modell in *Simulink* umzusetzen. Für die Verifikation mithilfe einer Sprungantwort wurde hier die Darstellung als Übertragungsfunktion gewählt. Eine Übertragungsfunktion bildet das dynamische Verhalten eines Systems ab und hat eine Eingangs- und eine Ausgangsgröße. In Kapitel 1.1 wurde das Konzept des ZMP eingeführt und die Differentialgleichung (4.2) hergeleitet, die die Position des CoP und des CoM miteinander in Verbindung bringt. Um nun ausschließlich das Verhalten des LIPM zu betrachten, kann man wie in Kapitel 2.2 bereits gezeigt, den Term der das Hilfsmoment, welches das AMPM auszeichnet gleich null setzt. Es folgt die Differentialgleichung:

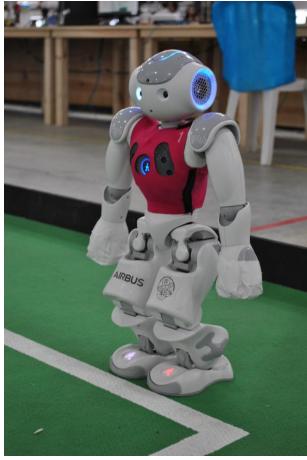
$$\ddot{x} = \frac{g}{z} (x - x_{CoP}) \quad (5.1)$$

Um eine Übertragungsfunktion für eine Differentialgleichung, wie sie in (5.1) angegeben ist, herzuleiten, bedient man sich der Laplace-Transformation und betrachtet das System im Frequenzbereich.

Für die Durchführung der Messung einer Sprungantwort wird angenommen, dass das System zum Zeitpunkt $t = 0$ in Ruhe ist und sich der CoM über dem Koordinatenursprung befindet. Durch die Transformation von (5.1) in den Frequenzbereich

$$\ddot{x} = \frac{g}{z} (x - x_{CoP}) \quad \circ \bullet \quad s^2 X = \frac{g}{z} (X - x_{CoP}) \quad (5.2)$$

erhält man nach Umstellen die Übertragungsfunktion $G(s)$ die das Verhalten des LIPM mit der Position des CoP als Eingang und der Position des CoM als Ausgang beschreibt:



(a) NAO in *Ready-Pose*



(b) NAO in *Penalized-Pose*

Abbildung 5.1: Verschiedene Posen des Roboters führen zu unterschiedlichen Modell- und Reglerparametern.

$$G(s) = \frac{X(s)}{x_{CoP}(s)} = \frac{\omega^2}{s^2 + \omega^2} \quad (5.3)$$

$$\omega^2 = -\frac{g}{z} \quad (5.4)$$

Da die Position des CoP, solange man vollständigen Bodenkontakt voraussetzt, direkt proportional zu dem Moment ist, welches im Fußgelenk des Roboters wirkt, kann man hier von einem momentenbasiert zu regelnden System sprechen.

5.2 Vergleich der Eingangssprungantwort von LIPM und NAO

Um zu verifizieren, dass das LIPM eine realistische Annäherung an das dynamische Verhalten des NAOs sein kann, wird die Sprungantwort des Systems in *Simulink* mit Messungen der Sprungantwort des realen Systems verglichen.

Zur Durchführung des Versuches wird der NAO manuell in eine Position gebracht, in der die Höhe des Schwerpunkts über dem Boden $z = 0,3$ m beträgt und der Zustand x_{com} des Schwerpunkts im Ursprung des Zustandsraums liegt. Diese Pose wird im Nachfolgenden *Ready-Pose* (siehe Abb. 5.1a) genannt. Um den Roboter genau positionieren zu können, wird die Möglichkeit des in Kapitel 3.3 vorgestellten Debugprotokolls genutzt, Messwerte in Echtzeit anzeigen lassen zu können. Durch die Darstellung der Schwerpunktsposition und -geschwindigkeit in einem Diagramm kann der NAO exakt ausgerichtet werden.

Aus diesem Zustand heraus wurde nun der Fußmotor mit maximal möglichem Moment eingeschaltet und die Schwerpunktsposition fortlaufend gemessen. Der NAO beginnt, wie durch die Instabilität des Systems zu erwarten, umzufallen und wird im letzten Moment aufgefangen. Der Verlauf einer Messkurve der Schwerpunktsposition ist für eine Messung exemplarisch in Abb. 5.2 dargestellt.

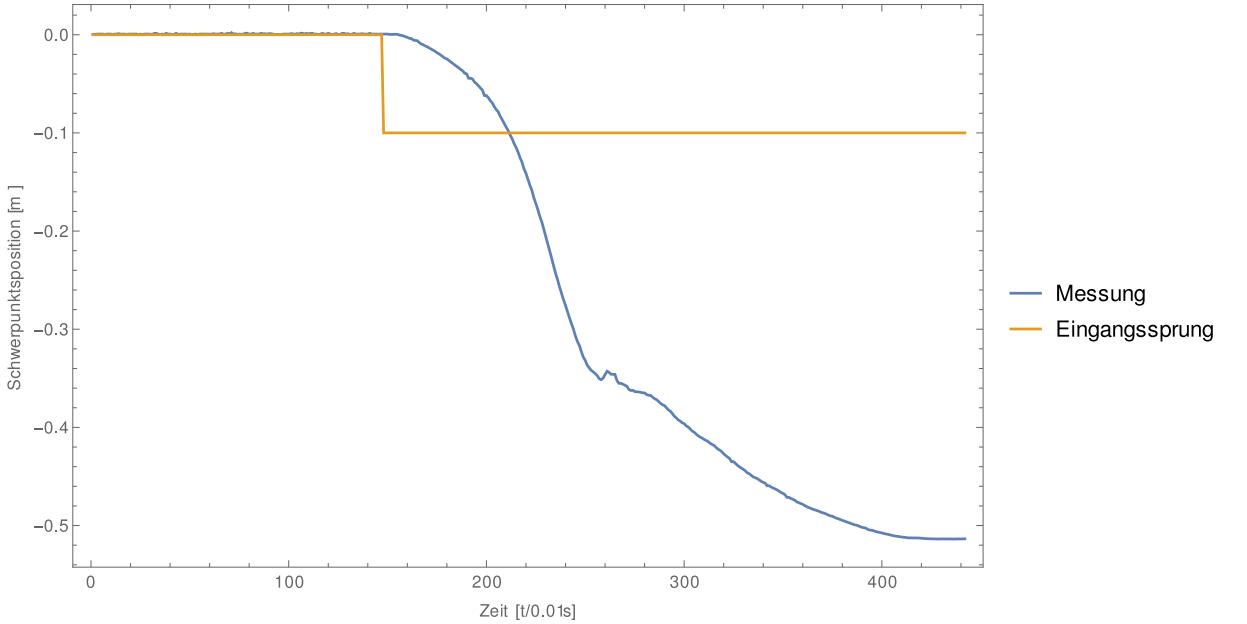


Abbildung 5.2: Einzelne Messung der Eingangssprungantwort des NAOs. Der Eingangssprung ist nicht maßstäblich dargestellt und gibt nur den zeitlichen Verlauf des Eingangssignals wieder. Der NAO wurde nach ca. 250 Messpunkten aufgefangen und abgelegt.

Zur Modellverifikation wurden mehrere Messungen der Sprungantwort des NAOs durchgeführt und diese statistisch auf Standardabweichung und Mittelwert untersucht um eine breite Datengrundlage zur Abstimmung der Parameter des LIPM zu schaffen. Die Daten wurden stets so aufbereitet, dass die Auswertung der Sprungantwort mit der Flanke des Eingangssignals beginnt und nach einer 150 Messpunkten endet. Es gibt zwei Gründe für das Kürzen der Messreihen. Erstens ist das LIPM nicht in der Lage kann, das Auffangen des NAOs vor dem Ende des Falls abzubilden. Zweitens wird durch die Linearisierung des Modells eine Abhängigkeit des Modellfehlers vom Winkel des Pendels eingeführt, wodurch die Genauigkeit des Modells für große Auslenkungen des Schwerpunkts verringert wird.

Eine Auswahl von fünf exemplarischen Messungen sowie die Standardabweichung und der Mittelwert sind in Abb. 5.3 dargestellt. Die grüne Linie zeigt den Verlauf der Reaktion des in *Simulink* simulierten LIPM, welches entsprechend dem Versuchsparameter $z = 0,3\text{ m}$ konfiguriert ist. Die Varianz der Messungen nimmt zu, je größer der Abstand des CoM zum Ursprung ist.

Auswertung der Messergebnisse

Der Vergleich der Messergebnisse mit dem Verlauf der Sprungantwort des Modells zeigt, dass das LIPM gerade für kleine Auslenkungen des Roboters eine gute Annäherung an die Realität ist. Der Fehler zwischen Modell und Roboter wird für größere Auslenkungen des Pendels konsequent größer. Dies liegt daran, dass die Höhe des Roboterschwerpunkts entgegengerichteter Annahme des LIPM nicht konstant ist, sondern bei einem Sturz gegen Null

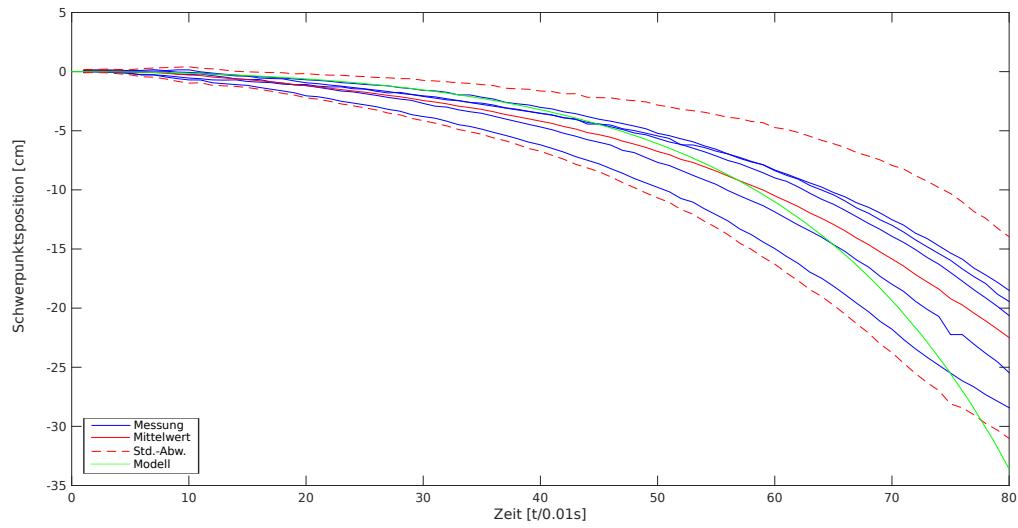


Abbildung 5.3: Die Sprungantwort des NAOs stimmt mit der des Systems insbesondere für kleine x_{CoM} überein. Der Fehler, der durch die Linearisierung entsteht, wird für große x_{CoM} größer. Das Intervall $\pm 2\sigma$ ist durch die gestrichelten Linien dargestellt.

geht. Trotzdem stimmt die Bewegung des Pendels mit der des Roboters qualitativ bis zum Ende des Sturzes überein.

Im Hinblick auf die Verwendung des LIPMs in dieser Arbeit ist besonders zu berücksichtigen, dass gerade ein Fallen des Roboters verhindert werden soll. Ferner ist auch davon auszugehen, dass sich der Zustand des Roboterschwerpunkts in einer Stoßsituation in der Umgebung vom Ursprung des Zustandsraums befindet.

6 Dynamische Stabilisierung des Roboters

Wie in den vorhergehenden Kapiteln theoretisch gezeigt wurde, lässt sich die Stabilität eines humanoiden Roboters durch die Anwendung zweier verschiedener Regelungsstrategien, der *Ankle-Strategie* und der *Flywheel-Strategie*, verbessern.

Die tatsächliche Implementierung der beiden Regelstrategien auf dem NAO-Robotiksystem im *tuhhSDK* wurde im Rahmen dieser Arbeit durchgeführt und soll im folgenden Kapitel vorgestellt werden. Zunächst wird erläutert, welche Rahmenbedingungen durch die Firmware des NAO-Roboters entstehen und wie eine Steuerung des Fußmomentes für dieses System ermöglicht wird. Danach werden zwei Ansätze zur Regelung mit der *Ankle-Strategie* vorgestellt und die Auswirkungen der Regelung, sowie die in Kapitel 4 betrachteten Stabilitätskriterien verifiziert.

Danach wird die Implementierung der *Flywheel-Strategie* erläutert und verifiziert, ob diese Strategie tatsächlich die Stabilität des realen Roboters verbessern kann.

Am Ende der Betrachtungen wird beschrieben, wie die entwickelten Reglerparameter in Echtzeit angepasst werden können, um eine Anwendung der beiden hier vorgestellten Regelungsstrategien in beliebigen aufrechten Posen zu ermöglichen.

6.1 Steuerung des Motormoments im Knöchelgelenk

Stiffness

Die Motoren des NAOs sind durch den Hersteller bereits softwareseitig durch einen Device Communication Manager (DCM) abstrahiert [4] und werden durch drei Parameter angesprochen:

- α die gewünschte Winkelstellung des Gelenks
- t die Zeit, in der diese Stellung erreicht werden soll
- s der normierte Strom, der durch den Motor fließt

Der letzte dieser Parameter ist direkt proportional zum Drehmoment, welches der Motor aufbringt. Dieser Parameter nennt sich Stiffness [17].

$$s = \left| \frac{I}{I_{max}} \right| \in [0, 1] \quad (6.1)$$

Aufgrund der Betragsbildung in (6.1) lässt sich diesem Parameter keine Information über die Drehrichtung des Motors entnehmen.

Der übliche Anwendungsfall für die Steuerung der Motoren des NAOs ist das Anfahren oder Halten definierter Positionen. Die Einstellung einer Stiffness ermöglicht dem Roboter, eine Position mit einem gewissen maximalen Moment zu halten. Die Richtung, in die das Motormoment wirkt, wird durch die von außen wirkenden Momente vorgegeben.

Steuerung des Drehmoments

Wie aus (6.1) ersichtlich, kann der Betrag des Motormoments durch Steuerung des Motorstroms direkt beeinflusst werden. Um die Richtung des Drehmoments steuern zu können, muss eine Funktion entwickelt werden, die von der Betragsbildung in (6.1) unabhängig ist.

Hierzu wird die Zielposition des Motors auf einen Wert gesetzt, der – entsprechend der gewünschten Drehrichtung – entweder größer oder kleiner ist, als der aktuell gemessene Motorwinkel. Gleichzeitig wird der zweite Parameter t des Treiberaufrufs auf 0 gesetzt, was einen infiniten Geschwindigkeitsfehler des Motorreglers zur Folge hat. Der fließende Motorstrom I wird weiterhin durch den Parameter s vorgegeben.

Durch das Wählen der Parameter

$$\begin{aligned}\alpha &\rightarrow \alpha \pm \alpha \\ t &\rightarrow 0 \\ s &\rightarrow s\end{aligned}$$

erhält man für die Steuerung des Motordrehmoments eine Funktion, die von einem Parameter $s' \in [-1, 1]$ abhängt. Der Parameter α muss größer als 25° sein, damit die Reaktion des Motorreglers maximal ist. Diese Funktion wird durch eine Methode in C++ umgesetzt.

Steuerung des CoP über das Moment im Fußgelenk

Aufgrund der Annahme, dass sich der Schwerpunkt des Roboters nicht in z -Richtung bewegt, lässt sich das Moment, welches im Fußgelenk wirkt leicht in die Position des CoP umrechnen. Aus der Annahme folgt analog zu (2.7):

$$F_z = Mg \quad (6.2)$$

Aus dem Momentengleichgewicht um das Fußgelenk und der Definition des ZMP (siehe auch Abb. 4.1) erhält man dann:

$$-x_{\text{CoP}} \cdot Mg = M_{\text{Ankle}}^{(y)} \quad (6.3)$$

Setzt man nun entsprechende Werte [17] für M und g ein, folgt mit

$$x_{\text{CoP}} = \frac{\tau_{\text{Ankle}}^{(y)}}{5,2 \text{ kg} \cdot 9,81 \text{ m s}^{-2}} \approx \frac{\tau_{\text{Ankle}}^{(y)}}{51 \text{ N}} \quad (6.4)$$

ein Ausdruck, mit dem das Motormoment mit der Position des CoP in Beziehung gesetzt wird.

Der Zusammenhang zwischen dem normierten Moment s' und der Position des CoP lässt sich aus den Herstellerangaben zum Motor im Fußgelenk und dessen Getriebeübersetzung

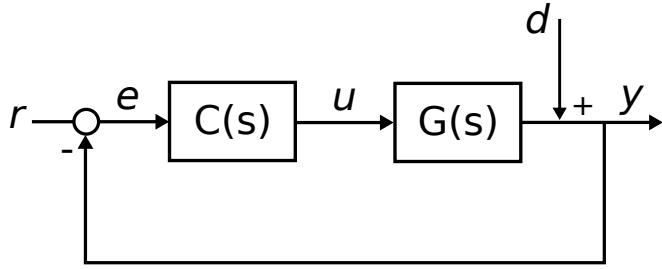


Abbildung 6.1: Geschlossener PD-Regelkreis.

berechnen [17]. Es gilt dann mit

$$s' = \frac{\tau_{\text{Ankle}}^{(y)}}{\underbrace{16,1 \text{ mN m}}_{\text{max. Motormoment}} \cdot \underbrace{130,85}_{\text{Übersetzung}}} \quad (6.5)$$

und (6.4) die Proportionalität

$$x_{\text{CoP}} = s' \cdot \frac{16,1 \text{ mN m} \cdot 130,85}{5,2 \text{ kg} \cdot 9,81 \text{ m s}^{-2}} \approx s' \cdot 0,0413 \text{ m} \quad (6.6)$$

Der Zahlenwert, der hier für den Proportionalitätsfaktor berechnet wurde, ist in der Realität sehr wahrscheinlich nicht korrekt, da die Motoren des NAOs durch Verschleiß und Betriebstemperatur erfahrungsgemäß sehr starke Leistungsschwankungen aufweisen. Dennoch ermöglicht dieser Wert, wie in Kapitel 6.5 gezeigt, eine Umrechnung der Reglerparameter vom Modell zum echten Roboter.

6.2 Regelung des Fußmomentes

Im Rahmen der Ankle-Strategie soll das Motormoment im Fußgelenk zur Kontrolle der Position des CoP softwareseitig geregelt werden. Die Übertragungsfunktion des Systems (5.3) hat zwei reellwertige Pole. Durch die Nutzung eines geeigneten Reglers sollen die Pole des Gesamtsystems aus Regler und LIPM jetzt so positioniert werden, dass das Verhalten des Systems später definierten Eigenschaften genügt. Der geschlossene Regelkreis aus Abb. 6.1 wird unter der Verwendung von *Matlab* und dem Wurzelortskurvenwerkzeug *rltool* getuned.

Open-Loop Eigenschaften des LIPM

Das LIPM mit der Übertragungsfunktion $G(s)$ (5.3), die hier noch einmal dargestellt wird, hat zwei Pole und ist daher ein System zweiter Ordnung [19].

$$G(s) = \frac{\omega^2}{s^2 + \omega^2} \quad \text{mit} \quad \omega = \sqrt{\frac{g}{z}}$$

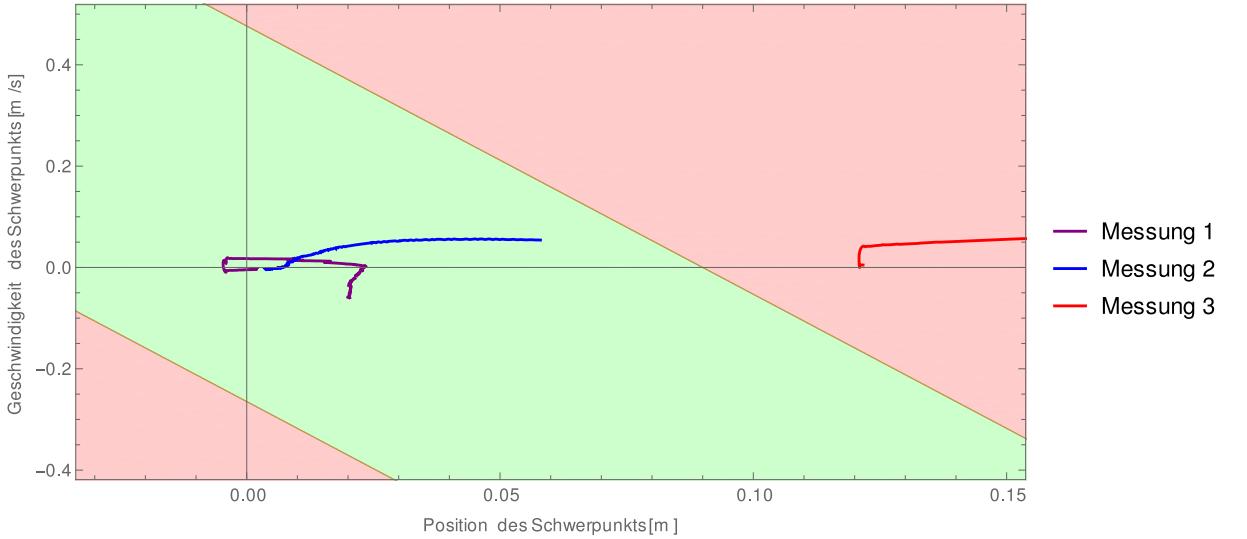


Abbildung 6.2: Phasendiagramm für den NAO bei alleiniger Regelung des Fußmomentes mit einem PD-Regler.

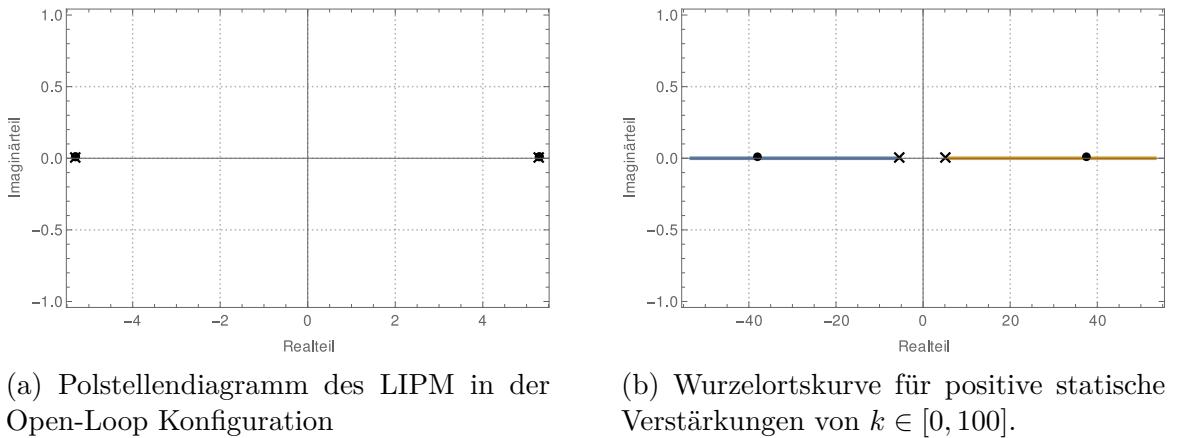


Abbildung 6.3: Positionen der Pole für das ungeregelte lineare invertierte Pendelmodell.

Die Pole sind symmetrisch zur imaginären Achse angeordnet (Abb. 6.3a), woraus unmittelbar resultiert, dass das System nicht von sich aus stabil ist und auch nicht durch eine beliebige statische Verstärkung stabilisiert werden kann (siehe Abb. 6.3b).

Zur Stabilisierung des Systems soll ein Regler dimensioniert werden, der die Pole des LIPM in einem entsprechenden Bereich so platziert, dass das System ein im folgenden spezifiziertes dynamisches Verhalten aufweist.

Das Dämpfungsmaß eines Systems bestimmt, wie stark das Überschwingen eines Systems ist und beeinflusst indirekt auch die Reaktionsgeschwindigkeit des Systems. Bei der Wahl eines geeigneten Dämpfungsmaßes muss also ein Kompromiss aus möglichst kurzer Reaktionszeit und gleichzeitig geringem Überschwingen getroffen werden. Üblich ist hier eine Wahl des Dämpfungsmaßes von mindestens $\zeta = \frac{1}{\sqrt{2}}$ [19]. Um dieses Dämpfungskriterium

einhalten zu können, muss für alle Pole p_i des geregelten Modells gelten (siehe Abb. 6.4):

$$\arg(p_i) \in [-\sin^{-1}(\zeta), \sin^{-1}(\zeta)] \quad (6.7)$$

Um die Bandbreite und die Ausregelzeit eines Regelungssystems zu spezifizieren, müssen die Pole des Systems in einem Intervall auf der negativen reellen Halbachse liegen. Die Bandbreite des geregelten Systems ist insbesondere für die Diskretisierung des Reglers von Belang. Hier ist darauf zu achten, dass die Bandbreite ω_b des geregelten Systems klein genug ist, dass die maximale Abtastrate mit der auf dem NAO-Robotiksystem gesampled werden kann ausreichend ist, das heißt [22]:

$$\frac{1}{T_s} \stackrel{!}{\approx} 30 \cdot \omega_b \quad (6.8)$$

Aus der maximalen Abtastrate des NAOs von $\omega_s = \frac{2\pi}{T_s} = 200\pi \text{ rad s}^{-1}$ ergibt sich also eine maximale regelbare Bandbreite des Systems von $\omega_b \approx 20,73 \text{ rad s}^{-1}$.

Die gewünschte Ausregelzeit t_s des Systems setzt den möglichen Polpositionen eine Grenze nach rechts, die nicht überschritten werden darf. Die folgende Schätzformel für Systeme zweiter Ordnung [19] gibt einen ungefähren Rahmen für die Begrenzung der Systempole vor.

$$\zeta \omega_n \approx \frac{4,6}{t_s} \quad (6.9)$$

6.3 Verschiedene Regleransätze

In diesem Abschnitt wird der Einsatz eines idealen PD-Reglers mit dem Einsatz eines idealen PID-Reglers zur Regelung des Fußgelenkes im LIPM verglichen. Da die Regelung auf dem NAO-Robotiksystem digital implementiert wird (siehe Kapitel 6.5) können ideale Reglerparameter verwendet werden. Diese werden im weiteren Verlauf des Kapitels dann in digitale Reglerparameter umgerechnet.

Zunächst werden beide Regler für das Pendelmodell unter der Verwendung von *Matlab* automatisch getuned. Hierbei wird das Pendel so parametrisiert, dass es sich wie der NAO verhält (6.10).

$$\omega = \sqrt{-\frac{g}{z}} = \sqrt{\frac{-9,81 \text{ m s}^{-2}}{0,35 \text{ m}}} \approx 5,2942 \text{ rad s}^{-1} \quad (6.10)$$

Sowohl ein idealer PID- als auch ein idealer PD-Regler lassen sich über die Übertragungsfunktion (6.11) darstellen. Im Falle des PD-Reglers wird der Parameter $K_i = 0$ gesetzt.

$$C(s) = K_p + \frac{K_i}{s} + K_d s \quad (6.11)$$

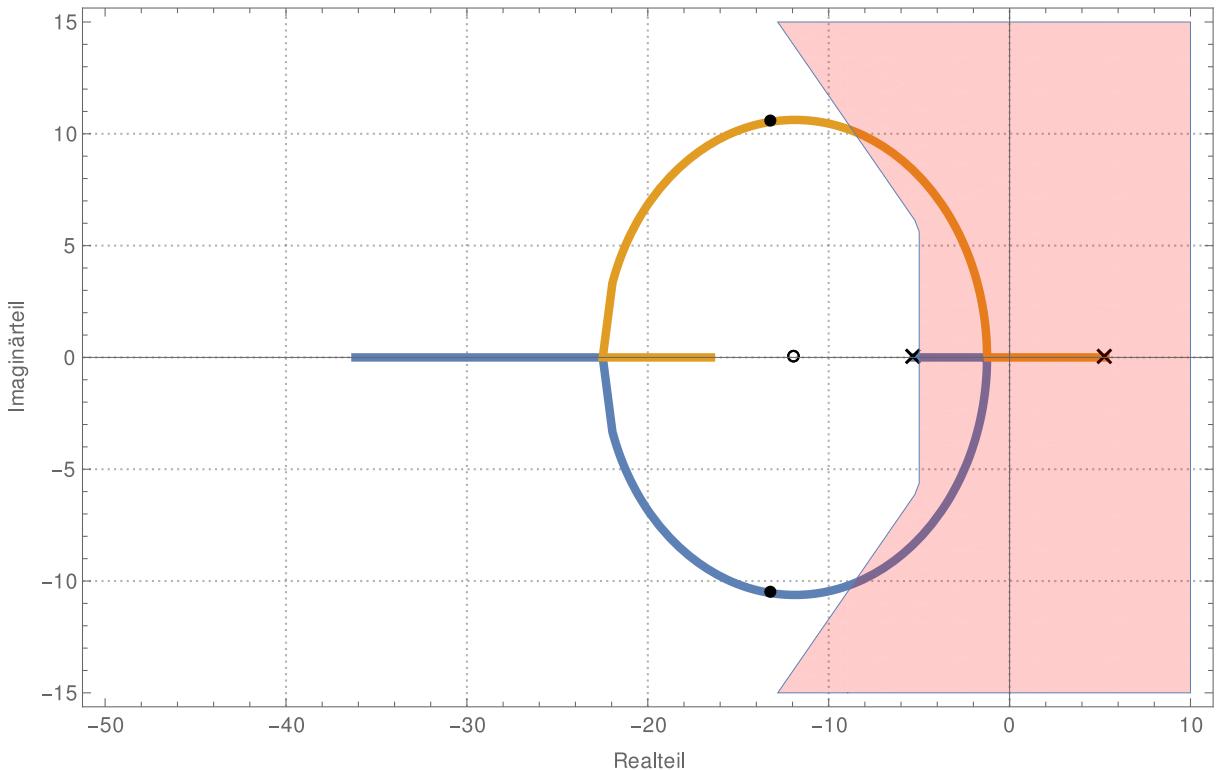


Abbildung 6.4: Wurzelortskurve des PD-geregelten Systems. Eine geeignete Regelung kann für einen großen Bereich der statischen Verstärkung erreicht werden. Die aus den Regleranforderungen hervorgehenden Grenzen sind, bis auf die Bandbreitenlimitierung, rot markiert.

Die Wurzelortskurven des PD und PID geregelten Systems ergeben sich wie in Abb. 6.4 und 6.6 dargestellt. Die statische Verstärkung des in Abb. 6.6 abgebildeten PID-Reglers ist ca. dreimal so hoch, wie die des PD-Reglers. Diese Verstärkung ist erforderlich, da der Verlauf der Wurzelortskurve des imaginären Polpaars sehr lange außerhalb des durch die Regleranforderungen vorgegebenen Bereichs verläuft. Die hohe Verstärkung hat zu Folge, dass der reelle Pol des Systems weit in negative reelle Richtung verschoben wird.

Es ist also zu erwarten, dass die Bandbreite des PID geregelten Systems signifikant höher ist, als die des PD geregelten Systems, welches wegen der fehlenden dritten Polstelle dieses Problem nicht aufweist. Aus Abb. 6.5 kann die Bandbreite beider Systeme bestimmt werden. Die Bandbreite des PD-geregelten Systems ergibt sich zu $27,54 \text{ rad s}^{-1}$, während die Bandbreite des PID-geregelten Systems in dieser Konfiguration $55,72 \text{ rad s}^{-1}$ beträgt.

Ferner ist zu beobachten, dass im Falle einer PD-Regelung ein größerer Spielraum für die Wahl der statischen Verstärkung besteht, in dem die Ziele für das geregelte Systems erfüllt werden können. Ungenauigkeiten und Abweichung in der Modellierung oder durch Hardwareunterschiede über verschiedene Roboter hinweg spielen also im Falle einer PD-Regelung eine kleinere Rolle, da mit einem einmal definierten Regler ein relativ großer Bereich an statischen Verstärkungen noch zu einer ausreichenden Performance führt. Der

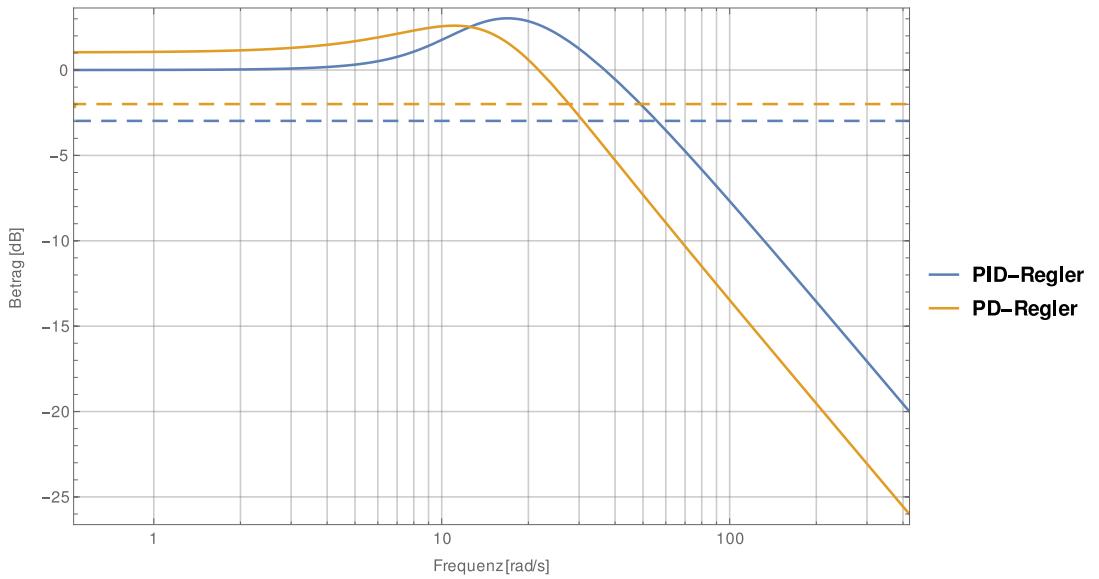


Abbildung 6.5: Bodediagramm der beiden Regleransätze.

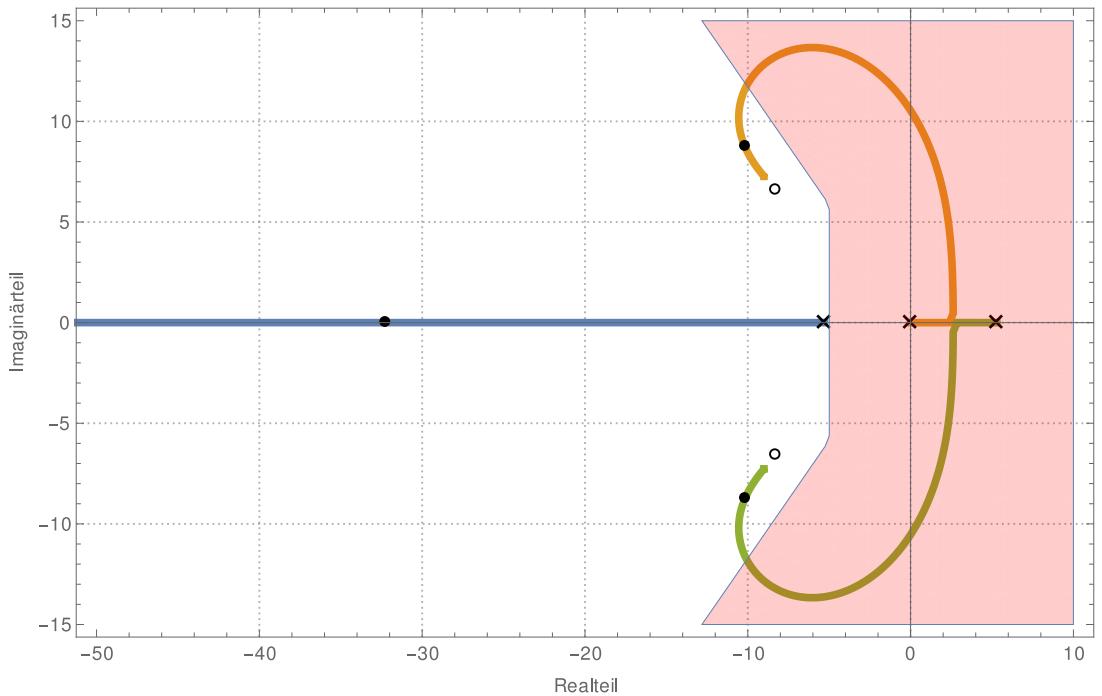


Abbildung 6.6: Wurzelortskurve des PID-geregelten Systems. Der Pol auf der reellen Achse entfernt sich bei der nötigen statischen Verstärkungen vom Ursprung. Diese ist aber erforderlich um das komplexe Polpaar in einen performanten Bereich zu bringen.

PID-Regler muss dafür relativ genau getuned werden, was sich gerade in der realen Anwendung auf dem Roboter als schwierig erweisen kann.

Die niedrigere Bandbreite, die der PD-Regler wegen der fehlenden Integratorpolstelle erreichen kann, wird durch Nachteile im Zeitbereich erkauft. In dieser Arbeit wird das

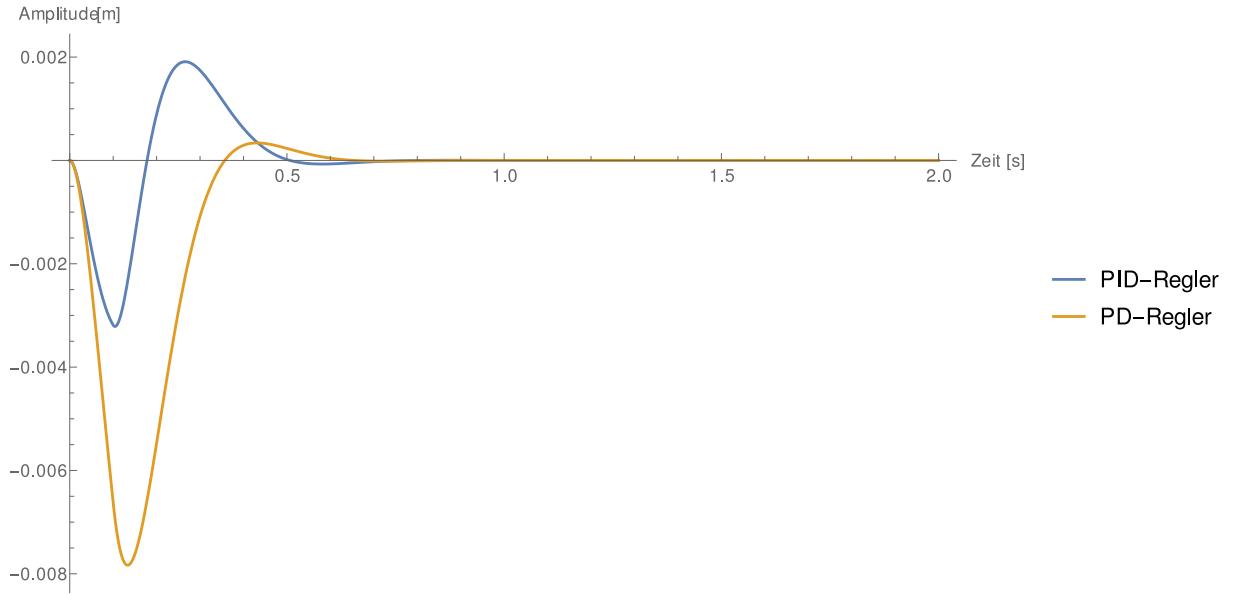


Abbildung 6.7: Verlauf der Systemantwort für eine rechteckförmige Eingangsstörung.

Verhalten des Roboters nach der Einwirkung einer Störung von außen betrachtet, die am Systemausgang eingebracht wird. Im Verhältnis zum PID-Regler ist es mit einem PD-Regler nicht immer möglich, die Stellgröße nach einer Eingangsstörung wieder auf den Sollwert zurückzuregeln, es stellt sich eine bleibende Regelabweichung ein. Diese Verhalten ist jedoch im hier betrachteten Anwendungsfall nicht problematisch, da im Bereich des Roboterfußball davon ausgegangen werden kann, dass alle Störungen Impulsweise, zum Beispiel durch die Kollision mit einem anderen Roboter, auftreten und somit nach einer kurzen Zeit wieder vollständig verschwinden (siehe Abb. 6.7). .

Die Performance der beiden Regler kann in Abb. 6.7 gut verglichen werden. Die Grafik zeigt den Verlauf der Schwerpunktsposition des simulierten Pendels für folgendes Eingangssignal.

$$d(t) = 0,1 \text{ m} [\sigma(t) - \sigma(t - 0,1 \text{ s})] \quad (6.12)$$

Der PD-Regler reagiert langsamer als der PID-Regler auf die Störung, deswegen hat dieser eine vier mal so große Auslenkung von 0,008 m in negative Richtung. Jedoch schwingt der PID-Regler in der fallenden Flanke des Stoßes in die andere Richtung über, da der Integrator noch geladen ist und somit ein stärkeres Regelsignal erzeugt wird. Beachtenswert ist, dass beide Regler in dieser Konfiguration die gleiche Zeit benötigen, die Störung vollständig auszugleichen.

Reglerparameter

Die Reglerparameter, die für die Regler hier verwendet wurden, sind in der Tabelle 1 angegeben. Diese Parameter wurden mithilfe des `rltools` bestimmt.

	K_p	K_i	K_d
PID-Regler:	-24.7578	-166.881	-1.50337
PD-Regler:	-8.92085	0	-0.751683

Tabelle 1: Reglerparameter für das simulierte LIPM.

Es ist zu beachten, dass in der Simulation die Position des CoP geregelt wird, während auf dem echten Roboter der CoP über das Fußgelenkmoment direkt gesteuert wird. Die Umrechnung hierfür ist in Kapitel 6.1 erläutert, der proportionale Zusammenhang zwischen den beiden Stellgrößen führt dazu, dass die Reglerparameter bei der Implementierung auf dem NAO skaliert werden müssen, um ein korrektes Verhalten zu erzielen.

6.4 Regelung des Oberkörpers

Um die Stabilsierbarkeit des Roboters auf eine größere Region im Zustandsraum zu erweitern wurde in Kapitel 4.3 die Verwendung eines *Bang-Bang-Controllers* zur Steuerung des Oberkörpermoments besprochen.

Diese Bewegung wird ausgeführt, sobald der Zustand des Roboters den durch die *Ankle-Strategie* stabilisierbaren Bereich in Abb. 4.5 (grün) verlässt. Dazu wird die Bedingung (4.6) in jedem *Cycle* ausgewertet und überprüft. Je nachdem, ob die untere oder obere Grenze der Bedingung verletzt wurde, wird eine *Flywheel-Motion* in die eine, oder in die andere Richtung ausgeführt.

Abschaltung der CoM-Regelung

Die Regelung des Roboteroberkörpers versucht, den Schwerpunkt wieder in einen stabilisierbaren Zustand zu bewegen. Der Regler der Ankle-Strategie kann währenddessen aus zwei Gründen abgeschaltet werden: Erstens, wäre keine Verbesserung in der Stabilisierung des Roboters mit der Ankle-Strategie mehr zu erwarten, da der Motor im Fußgelenk des Roboters zu diesem Zeitpunkt bereits in Sättigung betrieben werden muss. Zweitens kommt es durch die steile Flanke des eingebrachten Moments durch die Flywheel-Strategie zu einem hohen Wert für den differentiellen Fehleranteil in der Regelstrecke, welches gerade am Ende der Flywheelregelung zu Problemen mit einer hohen Amplitude am Eingang der Regelstrecke führen kann. Im Idealfall würde man durch einen *Feed-Forward* Aufbau des Regelungssystems genau diesen Effekt ausgleichen, es sollte in diesem speziellen Fall jedoch keinen signifikanten Unterschied machen, den PD-Regler für den Zeitraum der Flywheelbewegung zu deaktivieren und ihn unmittelbar nach dem Ende des *Bang-Bang*-Signals wieder zu aktivieren. Die Verwendung eines *Feed-Forward* Aufbaus hätte hier insbesondere deshalb keine Auswirkungen, da bei der Wahl des Entscheidungshorizonts zum Beginn der *Flywheel-Strategie* nach (4.10) der Regler der *Ankle-Strategie* bereits in Sättigung betrieben wird.

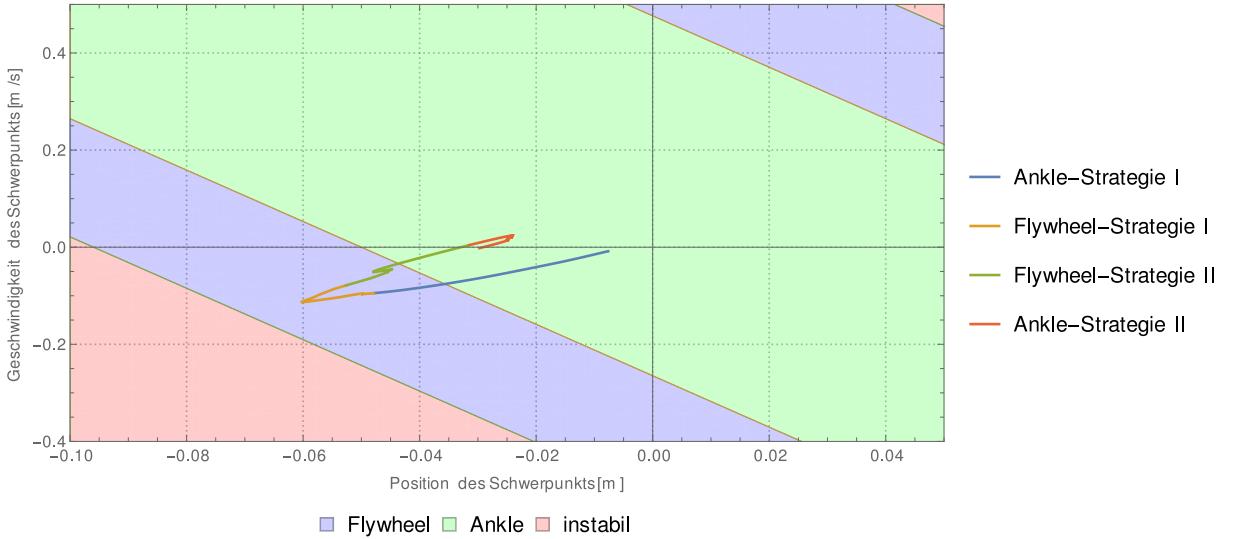


Abbildung 6.8: Phasenverlauf unter Verwendung der *Flywheel-Strategie*. Der Roboter reagiert erst verspätet auf das Verlassen der grünen Region und balanciert sich dann durch zwei Flywheel-Aktuationen, die die Stabilisierung des Roboters herstellen. Die *Bang-Bang-Regelung* hat einen ruckartigen Phasenverlauf zur Folge.

Anwendung der Flywheel-Strategie auf dem realen Roboter

Zur Anwendung eines *Bang-Bang-Controllers* zur Steuerung der *Flywheel-Motion* ist es erforderlich, die Grenzen der Region, in der die Ankle-Strategie noch zu einer Stabilisierung führen kann, zu bestimmen. Die Bestimmung der Parameter ergeben sich durch die Dimensionen der Füße des NAO-Roboters, sie können jedoch auch experimentell bestimmt werden, indem die Gleichung (4.10) in Echtzeit ausgewertet und der Roboter durch Stöße zum Umfallen bewegt wird. Praktikable Werte für den NAO-Roboter sind $\delta^- = -0,04 \text{ m}$ und $\delta^+ = 0,09 \text{ m}$.

Solange keine weitere Regelstrategie implementiert ist, die eingreift, wenn die Zustände des Roboters die Region (4.25) verlassen, sind Zahlenwerte für die Parameter der Flywheelbewegung (4.11) nur von theoretischem Interesse, um die exakte Ausdehnung der erweiterten Gleichgewichtsregion zu bestimmen. Zur Balancierung des Roboters mit der *Flywheel-Strategie* lässt sich eine geeignete Bewegung durch das Setzen von plausiblen Grenzen für die Oberkörperneigung implementieren.

Im Verlauf dieser Arbeit wurden Grenzen von $\pm 25^\circ$ für die Neigung des Roboteroberkörpers verwendet. Das Aufbringen der jeweiligen Momente dauerte stets $T_1 = T_2 = 0,1 \text{ s}$. Eine Berechnung von τ_{\max} wurde in dieser Arbeit nicht durchgeführt, die grundsätzliche Betrachtung des Flächenträgheitsmoments des Oberkörpers kann jedoch in [15] nachgelesen werden.

Der Phasenverlauf der durch die Anwendung der *Flywheel-Strategie* nach der Einwirkung eines starken Stoßes gegen den Roboter zustande kommt, ist in Abb. 6.8 dargestellt.

Die Anwendung der *Flywheel-Strategie* kann, wie in Abb. 6.8 zu sehen ist, zu einem Erfolg führen. Der hier dargestellte Phasenverlauf des Roboterzustands zeigt, wie der Roboter

nach einer starken Störung zunächst versucht, sich mit der *Ankle-Strategie* zu stabilisieren (blau). Nachdem der Zustand des Roboters verlassen hat, wird eine Aktuation des Oberkörpers durchgeführt. Hier ist zu erkennen, dass vom Zeitpunkt des Verlassens des grünen Bereichs bis zum Beginn der *Flywheel-Motion* eine gewisse Zeit verstreicht. Die erste Bewegung des Oberkörpers treibt den NAO noch nicht vollständig in die grüne Zone zurück. Dies ist damit zu erklären, dass die hier angewendete *Flywheel-Bewegung* nicht perfekt parametrisiert ist. Da der Zustand des Roboters noch immer die Bedingung (4.10) verletzt, wird eine zweite *Flywheel-Motion* durchgeführt (grün). Diese reicht aus, um den Zustand des Roboters in den grünen Bereich zurück zu treiben. Dort setzt die *Ankle-Strategie* erneut ein und kann den Roboter in einen Ruhezustand versetzen. Da die *Ankle-Strategie* einen PD-Regler verwendet, stellt sich eine bleibende Regelabweichung ein.

Die Anwendung der *Flywheel-Strategie* kann den Roboter in manchen Situationen stabilisieren. Dennoch zeigt sich in vielen Versuchen, dass die Ruckartigkeit der Flywheelbewegung oft einen Gegenteiligen Effekt hat. Die relativ große Masse des Roboteroberkörpers führt dann zu einem Kippen der Füße, was einen sanften Übergang zur Anwendung der *Ankle-Strategie* unmöglich macht.

Stephens wendet das Flywheel in [21] in einem optimalen Regler an. Die Regelverläufe des optimalen Reglers ähneln weniger dem Verlauf einer Zweipunktregelung, sondern haben viel mehr eine dreieckige Trajektorie. Soll die *Flywheel-Strategie* in weiteren Arbeiten verwendet werden, empfiehlt sich, andere Methoden der Regelungstechnik auf diese anzuwenden.

6.5 Digitale Reglerimplementierung

In diesem Kapitel wurden anhand des LIPM zwei Regelungsansätze zur Anwendung der *Ankle-Strategie* vorgestellt. Auf dem NAO-Robotiksystem muss eine solche Regelungsstrategie in Software implementiert werden und stellt demzufolge eine digitale Regelung dar.

Die Betrachtung des kontinuierlichen Modells zur Entwicklung einer Regelungsstrategie und die anschließende Umwandlung des kontinuierlichen Reglers in ein diskretes System ist eine valide Methode zum Entwurf einer Softwaregestützten Regelung [22].

Durch die Verwendung des NAO-Robotiksystems sind die Methoden der Digital-Analog Wandlung der Messsignale sowie der umgekehrten Transformation von digitalen Steuersignalen in analoge Motorsignale bereits vorgegeben und können nicht verändert werden.

Tustin-Approximation zur Bestimmung der digitalen Reglerparameter

Eine mögliche Methode zur Bestimmung der digitalen Reglerparameter aus den kontinuierlichen Reglerparametern ist die Tustin-Approximation [22, 23]. Die Tustin-Approximation ermöglicht eine Umwandlung der Reglerfunktion $C(s)$ in eine diskrete Reglerfunktion

	K_p	K_i	K_d
PID-Regler:	-1,0225	-0,1378	-1,2418
PD-Regler:	-0,3684	0	-0,6209

Tabelle 2: Approximierte Reglerparameter für den realen Roboter.

$D(z)$ durch die Substitution:

$$C(s) \rightarrow D(z) : s \rightarrow \frac{2}{T_s} \cdot \frac{z - 1}{z + 1} \quad (6.13)$$

Die Übertragungsfunktion des Reglers (6.11) lässt sich nun wie folgt annähern [19]:

$$D(z) = K_p + K_i \cdot \frac{T_s}{2} \cdot \frac{z + 1}{z - 1} + K_d \cdot \frac{2}{T_s} \cdot \frac{z - 1}{z + 1} \quad (6.14)$$

Für die diskreten Reglerparameter wird eine Skalierung der kontinuierlichen Reglerparameter um den Faktor $\frac{T_s}{2}$ für K_i und um $\frac{2}{T_s}$ für den Parameter K_d angenommen.

Unter Berücksichtigung der Skalierung (6.6) ergeben sich also die Reglerparameter für die digitale Regelung des NAOs, wie in Tabelle 2 angegeben.

Die Parameter, die auf diese Weise bestimmt wurden, können in der Realität tatsächlich zur Regelung des NAOs verwendet werden. Die PID-Reglerparameter ermöglichen eine Stabilisierung des NAOs jedoch nur für sehr kleine Störungen, bei großen Störungen wird das Gesamtsystem schnell instabil und schwingt sich auf, was unweigerlich zu einem Sturz führt.

Die Performance des PD-Reglers ist mit den so bestimmten Parametern besser als die des PID-Reglers. Jedoch lässt sich durch manuelles Feinabstimmen der Reglerparameter noch eine etwas bessere Performance mit einer kürzeren Reaktionszeit erreichen. Gute, manuell bestimmte Parameter für den PD-Regler sind $K_p = -0.05$ und $K_d = -0.4$.

Die Parametrisierung eines Reglers zur Anwendung der *Ankle-Strategie* in y -Richtung wird im Rahmen dieser Arbeit aus Zeitgründen nicht durchgeführt. Diese Reglerparameter sollten sich aber von den Parametern des x -Reglers nur durch einen konstanten Faktor unterscheiden, da das verwendete Modell lediglich durch die Höhe des Schwerpunkts verändert wird. Der Skalierungsfaktor ergibt sich aus der Tatsache, dass ein anderer Motor für die Neigung des Fußgelenks um die x -Achse verbaut ist.

6.6 Dynamische Adaption der Reglerparameter an verschiedene Roboterposen

Da das der Regelung zugrunde liegende Modell von der Schwerpunktsposition des Roboters abhängig ist, ist zu erwarten, dass für verschiedene Posen auch verschiedene Reglerparameter erforderlich sind.

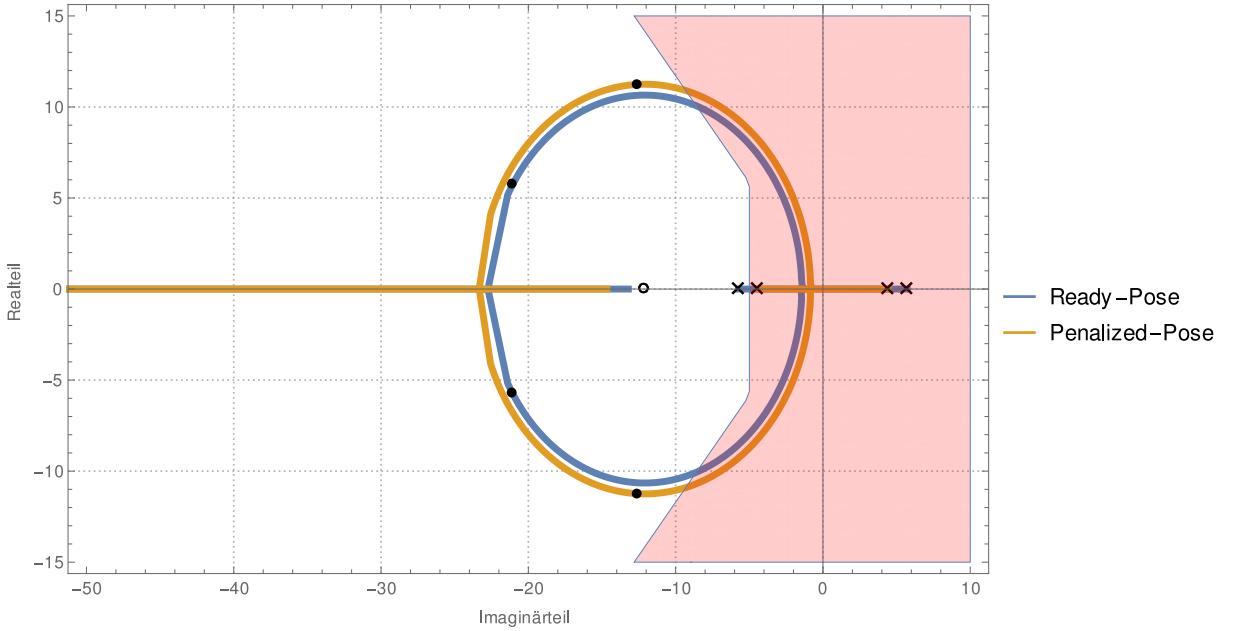


Abbildung 6.9: Wurzelortskurven der *Ready-Pose* und der *Penalized-Pose*. Durch den erhöhten Schwerpunkt sind die Reglerparameter für die *Penalized-Pose* nicht mehr ideal.

Zunächst soll untersucht werden, ob die in Tabelle 2 verwendeten Reglerparameter auch für andere Posen als die *Ready-Pose* verwendet werden können. Experimentell lässt sich zeigen, dass der Regler auch in einer Pose mit einer anderen Schwerpunktshöhe, der sog. *Penalized-Pose* (siehe Abb. 5.1b) die dynamische Balance des Roboters halten kann. Trotzdem besteht noch Verbesserungspotential bei der Ausregelzeit des Systems (siehe Abb. 6.9).

Eine einfache Möglichkeit, die Parameter des hier verwendeten PD-Reglers dynamisch an die Pose des Roboters anzupassen, ist eine Veränderung der statischen Verstärkung des Reglers in Abhängigkeit vom Modellparameter, der Höhe des Schwerpunktes über dem Boden.

$$C'(s) = k \cdot \left(K_p + \frac{K_i}{s} + K_d s \right) \quad (6.15)$$

Durch eine geeignete Wahl des Parameters k können die Positionen der Pole des geschlossenen Reglekreises für beliebige Schwerpunktshöhen an der nahezu gleichen Stelle in der komplexen Ebene gehalten werden. Die Reglerparameter aus Tabelle 1 wurden für die *Ready-Pose* dimensioniert, die einen konstanten Abstand von Boden und Schwerpunkt von ca. 0,3 m annimmt.

Wird k auf die Modellparameter des dimensionierten Reglers normiert, folgt:

$$k(z) := \frac{z}{0,3 \text{ m}} \quad (6.16)$$

Durch Anwendung dieser Verstärkung auf die Reglerparameter, wie in (6.15) darge-

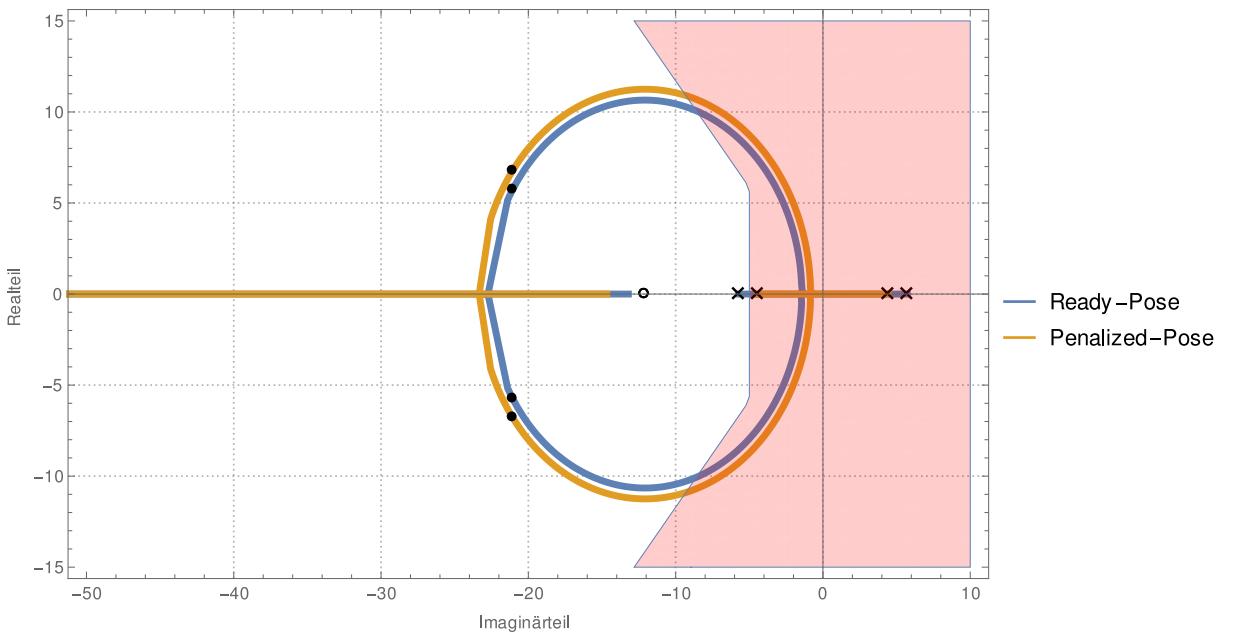


Abbildung 6.10: Die Skalierung der Reglerparameter mit einer geeigneten Verstärkung führt zu einer ähnlichen Verteilung der Pole bei verschiedenen Modellparametern.

stellt, erhält man die gewünschten Polpositionen des geschlossenen Regelkreises (siehe Abb. 6.10).

Die Anwendung dieses Verfahrens zur Anpassung der Reglerparameter ermöglicht die Verwendung der *Ankle-Strategie* zur Stabilisierung des NAO-Roboters in verschiedenen Posen.

7 Zusammenfassung und Ausblick

Das Ziel dieser Arbeit ist, die Stabilität von Bewegungen humanoider Robotern durch geeignete Modellierung bewerten und durch eine Regelungsstrategie verbessern zu können. Die unterschiedlichen hierzu vorgestellten Modelle und Modellgruppen kamen bereits bei anderen Forschungsprojekten zum erfolgreichen Einsatz. Zwei Balancierungsstrategien, die *Ankle-Strategie* und die *Flywheel-Strategie* wurden anhand von Erkenntnissen der Biomechanik von Menschen ausgewählt. Da auch Menschen als Auswahlkriterium für die Wahl einer Balancierungstechnik den Zustand ihres Schwerpunkts beobachten, wird das gleiche Entscheidungskriterium für den humanoiden Roboter gewählt. Diese Entscheidungen und die daraus resultierenden Anforderungen führen dazu, dass zur Modellierung des Robotiksystems zwei Modelle, das LIPM und das darauf aufbauende AMPM, ausgewählt werden.

Mit der Messung einer Eingangssprungantwort wird verifiziert, dass das LIPM das Verhalten des realen Roboters ausreichend genau abbildet und als Grundlage für die weitere Arbeit dienen kann. Die messtechnischen Methoden des verwendeten NAO-Robotiksystems unterliegen einem Messrauschen und einer Verzögerung, deren Auswirkungen untersucht werden. Hier ist es erforderlich eine Filtermethode zu auszuwählen, um das Signalrauschen zu verringern. Das Problem der großen Messverzögerung der Sensorik des NAOs verstärkt sich dadurch.

Im darauf folgenden Teil der Entwicklungen wird gezeigt, wie die Stabilität eines humanoiden Roboters anhand des Zustands seines Schwerpunkts bewertet werden kann und eine Entscheidungsfunktion bestimmt, die es ermöglicht eine geeignete Regelungsstrategie in Echtzeit auszuwählen.

Die mathematische Herleitung von zwei Regelungsstrategien, sowie deren Implementierung und Test im Kontext des `tuhhSDK` dienen der praktischen Verifizierung des theoretischen Teils der Arbeit. Auch das dynamische, zustandsabhängige Umschalten zwischen beiden Regelungsstrategien wird in diesem Zusammenhang implementiert. Es wird experimentell überprüft, dass die Anwendung der *Flywheel-Strategie*, wie vorher theoretisch gezeigt, den Bereich im Zustandsraum erweitert, für den der NAO stabilisiert werden kann. Die entwickelte Regelungsstrategie lässt sich auf verschiedene Positionen des NAOs anwenden, wenn die Reglerparameter dynamisch angepasst werden. Die Adaption der Reglerparameter funktioniert jedoch nur unter der Voraussetzung, dass ausschließlich die Füße des Roboters den Boden berühren.

Im Verlauf der praktischen Arbeit hat sich gezeigt, dass eine Regelung mit klassischen Reglern wie PID-Reglern nicht ideal geeignet sind, humanoide Roboter zu stabilisieren. Hierfür sollen im folgenden mehrere Gründe angeführt werden.

Durch eine Messverzögerung bei den Motorgelenkwinkeln liegen Informationen über den Zustand des Schwerpunkts bei alleiniger Verwendung der Winkeldaten erst verspätet vor. Wird ein klassischer geschlossener Regelkreis verwendet, erfolgt die Regelung in Abhängigkeit dieser Messdaten und ist deshalb ebenfalls verspätet, was die Robustheit des Systems stark beeinträchtigt. Die notwendige Filterung der Messdaten verstärkt diesen Effekt weiter. Dieses Problem könnte durch eine modellprädiktive Regelung adressiert werden, wobei

hier reaktivere Sensordaten wie die IMU als Messgrößen verwendet werden sollten. Zum jetzigen Zeitpunkt führt die Verwendung der im Rahmen dieser Arbeit implementierten PID-Regelung unter echten Bedingungen keine wesentliche Verbesserung der Stabilität des Roboters.

Ferner ist es schwierig, bei der Parametrisierung eines PID-Reglers beliebige Kriterien zur Optimierung der Parameter zu verwenden. Ein wichtiges Kriterium bei der Regelung des Fußmomentes ist, die Aktuatorgrenzen zu berücksichtigen um die Notwendigkeit einer *Anti-Windup-Konfiguration* zu eliminieren. Ferner sollte eine Multiple Input Multiple Output (MIMO)-Regelung des Systems in Erwägung gezogen werden, die in der Lage ist, Oberkörper- und Fußgelenksregelung gleichzeitig einzusetzen, um beide Regelstrategien zur Stabilisierung des Roboters optimal ausnutzen zu können.

Eine deutliche Erweiterung der Gleichgewichtszone des Roboters ließe sich erreichen, wenn zusätzlich zu den beiden vorgestellten Regelungsstrategien das Setzen eines Ausfallschrittes berücksichtigt werden würde. Die in dieser Arbeit vorgestellte Methode zur Bewertung des Roboterzustands liefert durch die Bestimmung der Grenzen der *Flywheel-Strategie* bereits einen Entscheidungshorizont für ein solches Agieren. Zur Modellierung der Dynamik eines Ausfallschrittes könnte eine Adaption des zu Beginn der Arbeit vorgestellten Variable Impedance Inverted Pendulum Modells verwendet werden, wobei die Limitierungen des NAO-Robotiksystems auf momentenbasierte Regelung hier zu Problemen führen könnte.

Die in dieser Arbeit vorgestellten Ansätze zur Stabilisierung eines humanoiden Roboters während verschiedener Bewegungsabläufe zeichnen sich vor allem durch ihre Modularität und Erweiterbarkeit aus. Das den Berechnungen zu Grunde liegende LIPM ist, wie das AMPM und das VIIP zeigen, durch gezielte Erweiterung um bestimmte Phänomene anpassbar und behält dabei eine unerreicht geringe Komplexität, die die theoretische Betrachtung der dynamischen Balance anschaulicher macht. Die Bewertung der dynamischen Balance durch eine Beobachtung des Schwerpunktszustands ermöglicht es, weitere Regelungsstrategien für bestimmte Bereiche der Zustandsebene hinzuzufügen und so das Balanceverhalten des Roboters natürlicher und besser zu gestalten.

Das Ziel weiterführender Arbeiten sollte es sein, modernere Regelungsmethoden der optimalen und robusten Regelung auf das hier vorgestellte Regelungsmodell anzuwenden und somit eine Verbesserung der Robustheit des Roboters zu erreichen. Auch die Erweiterung um ein geeignetes Schrittmodell ist als Aufgabe für eine weiterführende Betrachtung der dynamischen Balance von humanoiden Robotern denkbar.

Betrachtet man das Thema der dynamischen Balance vor dem Hintergrund einer eigenen Entwicklung eines Robotiksystems, entwächst dem Verlauf dieser Arbeit, dass eine momentenbasierte Regelung eines Robotiksystems die Umsetzung mancher Regelungsstrategien unmöglich macht. Ein neu zu entwickelndes Robotiksystem sollte daher kräftebasiert geregelt werden können.

Anhang

A Zero State Response der Flywheel-Strategie

Dieser *Wolfram Mathematica* Code löst die Faltung zur Bestimmung der Zero-State Antwort für die Flywheel-Strategie symbolisch.

```
In[1]:= tau[t_] := tauMax - 2*tauMax*UnitStep[t-T1] + tauMax*UnitStep[t-T2]

In[2]:= lTau[s_] := LaplaceTransform[tau[t], t, s]

In[3]:= G[s_] := -1/(M*z)*(1/(s^2 - omega^2))

In[4]:= zsr[s_] := G[s] * lTau[s]

In[5]:= FullSimplify[InverseLaplaceTransform[zsr[s], s, t]]
```

B Inhalte der CD

1. Diese Arbeit in digitaler Form
2. Quellcode der Implementierung der Regelungsstrategien
3. *Mathematica*-Code

Literatur

- [1] F. Poppinga, F. Bergmann, and S. Kaufmann, “HULKs-Team Research Report 2014,” 2015.
- [2] B. E. Maki and W. E. McIlroy, “The role of limb movements in maintaining upright stance: the “change-in-support” strategy,” *Physical therapy*, vol. 77, no. 5, pp. 488–507, 1997.
- [3] Y.-C. Pai and J. Patton, “Center of mass velocity-position predictions for balance control,” *Journal of biomechanics*, vol. 30, no. 4, pp. 347–354, 1997.
- [4] S. Kaufmann, “Implementierung und Analyse verschiedener Regelstrategien für dynamische Laufbewegungen humanoider Roboter auf Basis des NAO-Robotiksystems,” Diplomarbeit, Technische Universität Hamburg-Harburg, 2011.
- [5] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi, *Introduction to Humanoid Robotics*, ser. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2014.
- [6] M. Vukobratović and B. Borovac, “Zero-moment point—thirty five years of its life,” *International Journal of Humanoid Robotics*, vol. 1, no. 01, pp. 157–173, 2004.
- [7] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, “The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation,” in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 1. IEEE, 2001, pp. 239–246.
- [8] B. Stephens, “Humanoid push recovery,” in *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*. IEEE, 2007, pp. 589–595.
- [9] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, “Capture point: A step toward humanoid push recovery,” in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*. IEEE, 2006, pp. 200–207.
- [10] T. Koolen, T. De Boer, J. Rebula, A. Goswami, and J. Pratt, “Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models,” *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1094–1113, 2012.
- [11] T. Sugihara and Y. Nakamura, “Variable impedance inverted pendulum model control for a seamless contact phase transition on humanoid robot,” in *IEEE International Conference on Humanoid Robots (Humanoids2003)*, 2003, pp. 0–0.
- [12] ——, “Contact phase invariant control for humanoid robot based on variable impedance inverted pendulum model,” in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 1. IEEE, 2003, pp. 51–56.

- [13] A. Goswami, “Kinematic and dynamic analogies between planar biped robots and the reaction mass pendulum (RMP) model,” in *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on.* IEEE, 2008, pp. 182–188.
- [14] S.-H. Lee and A. Goswami, “Reaction mass pendulum (RMP): An explicit model for centroidal angular momentum of humanoid robots,” in *Robotics and Automation, 2007 IEEE International Conference on.* IEEE, 2007, pp. 4667–4672.
- [15] N. O. Lüders, “Modellierung und Regelung dynamischer Kickbewegungen humanoider Roboter auf Basis des NAO-Robotiksystems,” Bachelorarbeit, Technische Universität Hamburg-Harburg, 2013.
- [16] C. Collette, A. Micaelli, C. Andriot, and P. Lemerle, “Robust balance optimization control of humanoid robots with multiple non coplanar grasps and frictional contacts,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on.* IEEE, 2008, pp. 3187–3193.
- [17] *Nao Software Documentation*, Aldebaran Robotics, 2014.
- [18] J. J. Alcaraz-Jiménez, M. Missura, H. Martínez-Barberá, and S. Behnke, “Lateral disturbance rejection for the nao robot,” in *RoboCup 2012: Robot Soccer World Cup XVI.* Springer, 2013, pp. 1–12.
- [19] H. Werner, “Introduction to Control Systems,” 2013, Lecture Notes.
- [20] T. Bray, “The JavaScript Object Notation (JSON) Data Interchange Format,” *Internet Engineering Task Force (IETF)*, 2014.
- [21] B. Stephens, “Push Recovery Control for Force-Controlled Humanoid Robots,” Ph.D. Dissertation, Carnegie Mellon University, Pittsburgh, Pennsylvania USA, 2011.
- [22] H. Werner, “Control System Theory and Design,” 2014, Lecture Notes.
- [23] A. Tustin, “A method of analysing the behaviour of linear systems in terms of time series,” *Journal of the Institution of Electrical Engineers-Part IIA: Automatic Regulators and Servo Mechanisms*, vol. 94, no. 1, pp. 130–142, 1947.