

# Entwicklung eines dynamischen Aufstehens unter Vermeidung inkorrektter Bewegungszustände

Bachelorarbeit

Philip Reichenberg

Matrikelnummer: 4119829

28. November 2018



Fachbereich Mathematik / Informatik  
Studiengang Bachelor Informatik

1. Gutachter: Dr. Tim Laue
2. Gutachter: Prof. Dr. Rolf Drechsler

## Erklärung

Ich versichere, die Bachelorarbeit ohne fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen sind, sind als solche kenntlich gemacht.

Oyten, den 28. November 2018

.....

(Philip Reichenberg)

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Ziele . . . . .	2
1.3	Aufbau . . . . .	2
<b>2</b>	<b>Hintergrund</b>	<b>3</b>
2.1	RoboCup . . . . .	3
2.2	Der NAO . . . . .	3
2.3	B-Human Framework . . . . .	4
<b>3</b>	<b>Grundlagen</b>	<b>5</b>
3.1	Keyframe Interpolation . . . . .	5
3.1.1	Grundlagen der Aufstehbewegungen . . . . .	5
3.2	PID-Regler . . . . .	6
3.3	ZMP-Balancer . . . . .	6
3.4	Invertiertes Pendel . . . . .	6
<b>4</b>	<b>Probleme</b>	<b>7</b>
4.1	Gelenkwinkel werden nicht erreicht . . . . .	7
4.2	Statische Bewegungen . . . . .	10
4.3	Schwer zu balancierende Zustände . . . . .	10
<b>5</b>	<b>Entwicklung</b>	<b>15</b>
5.1	Die Aufstehbewegungen . . . . .	15
5.1.1	Die verwendeten Aufstehbewegungen . . . . .	15
5.1.2	Zustandsabhängigkeiten . . . . .	16

---

5.1.2.1	Umändern der Zustandsabhängigkeiten . . . . .	16
5.1.3	Die angepassten Bewegungen . . . . .	18
5.2	Pausieren in der Bewegung . . . . .	23
5.2.1	Evaluation der Pausierung in der Bewegung . . . . .	24
5.3	Abbrechen von Bewegungen . . . . .	24
5.4	Bedingungen und optionale Keyframes . . . . .	25
5.5	Balancierung . . . . .	27
5.5.1	Die Regelgröße: Der Schwerpunkt in der Fußebene . . . . .	27
5.5.2	Das Balancieren . . . . .	28
5.5.3	Einstellen der Parameter . . . . .	31
5.5.3.1	Ergebnisse vom Einstellen der Parameter . . . . .	33
5.6	Gelenkkompensierung . . . . .	34
5.7	Programmablauf . . . . .	35
<b>6</b>	<b>Evaluation</b>	<b>39</b>
6.1	Ziele . . . . .	39
6.2	Versuchsaufbau Balancierung . . . . .	40
6.3	Realistischer Test . . . . .	43
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>47</b>
7.1	Zusammenfassung . . . . .	47
7.2	Ausblick . . . . .	47
	<b>Abbildungsverzeichnis</b>	<b>50</b>
	<b>Literaturverzeichnis</b>	<b>52</b>

# Kapitel 1

## Einleitung

### 1.1 Motivation

Beim Fußballspiel ist es wichtig Vorteile zu erreichen, um damit mehr Tore zu schießen als das gegnerische Team. Dazu zählen das Blockieren von anderen Spielern, Annahmen von Bällen und sich neu auf dem Feld zu positionieren. Ebenfalls ist die Gesundheit der Spieler wichtig, denn Spieler mit Brüchen oder Sehnenrissen können nicht teilnehmen und fallen somit aus. Da beim Roboterfußball die Roboter manchmal umfallen, geht viel Zeit verloren und es entstehen strategische Nachteile, weil die genannten Vorteile nicht mehr umsetzbar sind. Zusätzlich ist es für einen gefallenen Roboter schwer, sich zu orientieren und so zu wissen, wo er auf dem Feld hinterher nach dem Aufstehen steht. Es ist also wichtig, die Häufigkeit des Umfallens sowie die Zeit, bis sich der Roboter aufgerichtet hat, zu minimieren. Da Aufstehversuche bei Robotern häufig fehlschlagen können, erhöht sich der Zeitaufwand des Aufstehens unweigerlich. Dazu kommt, dass sich mit jedem Fallen die Hardwarezustände verschlechtern und die Häufigkeit, dass der Roboter erneut umfällt oder das Aufstehen fehlschlägt, dementsprechend steigt. Die Roboter sind nämlich nicht in der Lage, sich wie Menschen abzustützen oder abzurollen.

Ebenso kommt es durch erhöhtes Umfallen dazu, dass Gelenke vollständig kaputt gehen können und somit repariert werden müssen. So gab es beispielsweise beim Team B-Human in den letzten 12 Monaten 38 defekte Gelenke, indem entweder Zahnräder brachen, Motoren ausfielen oder Sensoren versagten. Folge sind Roboterausfälle, welche im schlimmsten Fall zu Unterbesetzungen des Teams führen können. Ebenfalls verändern sich durch Reperaturen die Hardwarezustände so stark, dass danach roboterspezifische Einstellungen geändert werden müssen, was sehr zeitaufwendig ist und sich die Roboter nur noch mehr abnutzen.

Das aktuelle Aufstehverfahren vom Team B-Human [vgl. B-Human, 2018] hatte auf dem RoboCup 2017 eine Erfolgsrate von ca. 70% und unter Verwendung des Verfahrens aus Abschnitt 5.4 und Anpassungen der Aufstehbewegungen auf dem RoboCup 2018 eine Erfolgsrate von 80%. Es existierten mehrere Aufstehbewegungen für jeweils vorne und hinten, angepasst an die verschiedenen *NAOs*. Die schnellste Aufstehbewegung für vorne lag bei ca. 3.5 Sekun-

den, die langsamste bei ca. 8 Sekunden. Für hinten betrug die Schnellste ca. 4.7 Sekunden und die Langsamste ca. 7 Sekunden.

Zum Vergleich hatte das Nao-Team HTWK Leipzig [vgl. Nao-Team HTWK, 2018] auf dem RoboCup 2018 eine Erfolgsrate von ungefähr 80% . HTWK Leipzig verwendet jeweils nur eine Aufstehbewegung für vorne und hinten, welche zwischen 2.5 und 3.5 Sekunden brauchen.

Es besteht also noch viel Luft nach oben für eine Verbesserung in der Qualität des aktuellen Aufstehens. Sei es die Entwicklung, damit jeder *NAO* das gleiche Aufstehen verwendet, das Aufstehen häufiger funktioniert, schnell oder hardwarechonend ist.

Für diese Statistiken wurden die Live-Stream Aufnahmen vom Team Nao Devils [vgl. Nao Devils, 2018] ausgewertet.

## 1.2 Ziele

Ziel dieser Arbeit ist es, ein einfaches Framework für Bewegungen zu schaffen, um damit gezielt Aufstehbewegungen für die Roboter des Typs *NAO* zu erstellen, welche in Normalfällen, aber auch unter Extrembedingungen stabil funktionieren. Dabei sollen inkorrekte Bewegungszustände erkannt und behoben werden, indem diese entweder ausbalanciert oder durch alternative Bewegungen ausgeglichen werden. Im Fokus der Entwicklung soll hierbei ein möglichst optimiertes Aufstehen sein, welches schnell, stabil und zuverlässig ist, sich in jeder Situation identisch verhält und auch auf neueren, zukünftigen Versionen des *NAOs* funktioniert.

## 1.3 Aufbau

In dieser Arbeit wird ein Framework erarbeitet, mit dem es möglich sein soll, Roboter des Typs *NAO* mit schlechten sowie guten Hardwarezuständen konsistent, schnell und zuverlässig aufstehen zu lassen. Nachdem dafür zuerst der Hintergrund und die Grundlagen erklärt werden, folgt daraufhin die Darstellung der grundsätzlichen Probleme, welche die Umsetzung von Aufstehbewegungen erschweren. Darauf aufbauend werden dann verschiedene Teilbereiche untersucht und entwickelt, um die Ziele dieser Arbeit zu erfüllen. Die Evaluation am Ende soll dann zeigen, inwiefern die Ziele erreicht wurden, welche entwickelten Problemlösungen erfolgreich waren und welche Probleme weiterhin existieren. Abschließend folgt dann noch eine Zusammenfassung und ein Ausblick darüber, was verbessert werden kann und ob die Ergebnisse in anderen Bereichen eingesetzt werden können.

## Kapitel 2

# Hintergrund

### 2.1 RoboCup

Der RoboCup ist Teil der *RoboCup Federation*, welche das Ziel hat, die Wissenschaft der Robotik mithilfe von weltweiten Wettbewerben, Konferenzen und Messen zu fördern. Der RoboCup ist eine Wettbewerbsveranstaltung, in der Forschungsgruppen gegeneinander antreten. Einer der Wettbewerbe beinhaltet die Standard Platform League, kurz *SPL*, in der alle teilnehmenden Teams mit Robotern des Typs *NAO* (siehe Abschnitt 2.2) von Softbank [vgl. Softbank Robotics, 2018] spielen müssen, sodass dies ein reiner Softwarewettbewerb ist. In der *SPL* müssen jeweils fünf gegen fünf *NAOs* auf einem  $9\text{m} \times 9\text{m}$  großen Kunstrasen mit angepassten Regeln Fußball spielen.

Ziel des RoboCups ist es, im Jahre 2050 mit einem Robotikteam im Fußball den amtierenden menschlichen FIFA Fußballweltmeister zu schlagen. Um dieses Ziel zu erreichen, werden die Regeln, welche nicht vollständig dem echten Fußball gleichen, jährlich angepasst. So wurde in der *SPL* noch vor paar Jahren mit farbigen Toren und Bällen gespielt. Diese sind aber mittlerweile durch weiße Tore und schwarz-weiße Bälle ausgetauscht.

Wichtig im Kontext dieser Arbeit sind auch die Zeitstrafen. Wann immer ein *NAO* einen Regelverstoß begeht, bekommt dieser eine Zeitstrafe und wird für diese Zeit aus dem Spiel genommen. Für jeden weiteren Verstoß erhöht sich die Zeitstrafe. Die Zeitstrafen sind dabei geltend für das ganze Team und nicht für die einzelnen *NAOs*. Eine der Regeln besagt, dass ein gefallender *NAO* innerhalb von 5 Sekunden anfangen muss aufzustehen und nach 20 Sekunden dies geschafft haben muss [vgl. Committee, 2017].

### 2.2 Der NAO

Die aktuell verwendeten *NAOs* der Generation V5 (siehe Abbildung 2.1) haben 25 Freiheitsgrade, ein Gewicht von 5.4kg und eine Höhe von 57.4cm. Im Torso sind ein Gyrometer und ein Accelerometer verbaut, mit denen in allen drei Dimensionen die Geschwindigkeiten und

Beschleunigungen gemessen und damit unter anderem die Neigungen berechnet werden.

Die Motoren der Gelenk vom *NAO V5* können alle 10ms angesteuert werden. In der Arbeit von Böckmann [2015] wurde experimentell ermittelt, dass es eine größere zeitliche Verzögerung gibt, bis angesteuerte Gelenkwinkel umgesetzt werden.

So dauert es zwischen 30ms und 40ms von der Übertragung bis zur Messung einer Änderung der Gelenkwinkel. Diese Verzögerung ist so lang, dass in der Zeit, bis die Auswirkungen gemessen werden, bereits zwei weitere Signale an die Motoren gesetzt wurden.

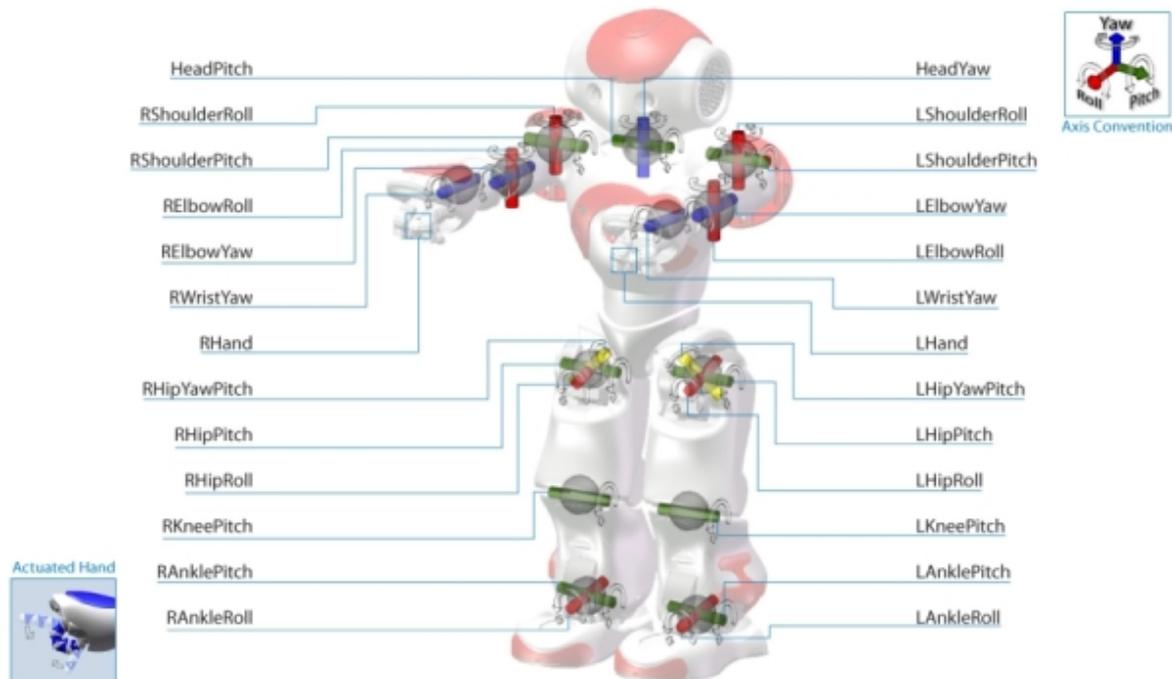


Abbildung 2.1 Die Darstellung des NAO V5 [vgl. Aldebaran SoftBank Group, 2015]

## 2.3 B-Human Framework

B-Human ist ein Roboterfußballteam, welches an den jährlichen RoboCup Wettbewerben in der *SPL* teilnimmt. Die Implementierung dieser Arbeit wird in die von B-Human entwickelte Software integriert, welche den *NAO* autonom Fußball spielen lässt. In der Software werden dabei 100 Mal in der Sekunde Gelenkwinkel berechnet und den Motoren übergeben. Eine einzelne dieser 100 Berechnungen wird im Folgenden als Frame bezeichnet.

Aktuell gibt es verschiedene Bewegungssteuerungen, abhängig davon, was vom *NAO* verlangt wird. Dazu zählen das Stehen, Laufen, Schießen und Aufstehen. Alle Bewegungsteile basieren auf unterschiedlichen Ansätzen zum Berechnen und Umsetzen von Bewegungen und sind für sich jeweils gekapselt. Das Aufstehen hat somit keine Verbindung zum Laufen und Schießen, außer dass zwischen diesen gewechselt wird, falls der *NAO* erkennt, dass er gefallen ist oder wieder steht.

## Kapitel 3

# Grundlagen

### 3.1 Keyframe Interpolation

Ein übliches Verfahren für die Planung und Umsetzung von Bewegungen von Robotern ist die Verwendung von Keyframes. Ein Keyframe beschreibt einen Zustand zu einem bestimmten Zeitpunkt. Bei Robotern können die Keyframes explizite Gelenkwinkel angeben aber auch einzelne definierte Punkte im Raum relativ zum Roboter mit Rotationen, mit denen dann Gelenkwinkel berechnet werden. Zwischen diesen Keyframes wird interpoliert, um zwischen zwei Keyframes zu jedem Zeitpunkt Gelenkwinkel zu berechnen. Diese Zeitpunkte beschreiben dann die Frames. Mit dieser Grundlage wurden unter anderem erfolgreich Schüsse für humanioide Roboter entwickelt [vgl. Depinet u. a., 2014; Müller u. a., 2011].

In dem verwendeten B-Human System wird in dem vorhandenen Aufstehverfahren eine Keyframe Interpolation verwendet. Dafür sind mehrere Zwischenposen, die der *NAO* ansteuern soll, angegeben. Diese Zwischenposen beinhalten die Gelenkwinkel aller Gelenke sowie eine Dauer, in der diese Gelenkwinkel erreicht werden sollen. Die Interpolation ist eine lineare Berechnung, die die Gelenkwinkel der Start- und Zielposition nimmt und einen linearen Verlauf zwischen diesen Werten unter Berücksichtigung der angegebenen Dauer berechnet.

#### 3.1.1 Grundlagen der Aufstehbewegungen

Für diese Arbeit werden die Aufstehbewegungen verwendet, die bereits im B-Human System vorhanden sind. Diese sind die Bewegungen *FrontFast*, *FrontFreeJoints*, *BackFast*, *BackFreeJoints* und *SitDownFreeJoints*, welche für den RoboCup 2018 verwendet wurden. Eine Darstellung kann aus dem *Code Release 2018* vom Team B-Human entnommen werden [vgl. Röfer u. a., 2018]. Die Bewegung *FrontFast* ist eine schnelle Aufstehbewegung für vorne, *FrontFreeJoints* eine langsame für vorne. *BackFast* ist eine Schnelle für hinten und *BackFreeJoints* eine etwas Langsamere für hinten. *SitDownFreeJoints* ist eine Nachfolgebewegung für *FrontFreeJoints* und *BackFreeJoints*.

## 3.2 PID-Regler

Ein *Proportional-Integral-Differential-Regler*, kurz *PID-Regler* genannt, dient dazu, Regelgrößen zu beeinflussen, um deren Sollwert trotz Störungen zu erreichen und zu halten [vgl. Simon B., 2015]. Dafür beschreibt der *P-Anteil* das proportionale Verhalten, der *I-Anteil* das integrale Verhalten und der *D-Anteil* das differentiale Verhalten. Zum Balancieren in der Robotik werden PID-Regler bereits erfolgreich verwendet [vgl. Müller u. a., 2011]. In Kapitel 5.5 wird der in dieser Arbeit entwickelte PID-Regler vorgestellt.

## 3.3 ZMP-Balancer

Der *Zero-Moment-Point*, kurz *ZMP*, beschreibt den Punkt in der Fußebene, in der sich alle horizontalen Kräfte des *NAOs* aufheben und ist eine übliche Grundlage zum Balancieren in der Robotik [vgl. Sugihara u. a., 2002; Kuffner u. a., 2001]. In dem aktuellen B-Human System existiert bereits ein *ZMP-Balancierverfahren* seit 2017 [vgl. Röfer u. a., 2017]. Dieses wird in dem aktuellen Aufstehverfahren aber nur zu dem Zeitpunkt verwendet, zu dem der *NAO* bereits eine sitzende Position erreicht hat und sich nur noch aufrecht hinstellen muss. Dies hat den Grund, dass zu jedem Zeitpunkt davor der *NAO* eine breitbeinige Körperpose einnimmt (siehe Abschnitt 3.1.1) und die Füße nie einen vollständigen Bodenkontakt haben. Dieser Bodenkontakt, beziehungsweise die Planarität der Füße, wird aber vorausgesetzt.

Aufgrund dieser Tatsache wird weder dieses *ZMP-Balancierverfahren* für die Arbeit verwendet noch wird ein Neues entwickelt, da dies den zeitlichen Rahmen sprengen würde.

## 3.4 Invertiertes Pendel

Ein übliches Balancierverfahren besteht darin, den Roboter als ein invertiertes Pendel zu modellieren [vgl. Graf u. Röfer, 2011]. Damit ist es möglich, den Roboter energiesparend zu balancieren. Da beim Aufstehen des *NAOs* der Hauptfokus darauf liegt, nicht umzufallen und sich der *NAO* die meiste Zeit in einer tieferen Körperhaltung befindet und immer auf zwei Beinen gestützt sein soll, wird für diese Arbeit kein Ansatz des invertierten Pendels umgesetzt. Außerdem wäre ein solcher Ansatz sehr schwer zu realisieren, da der *NAO* für die meiste Zeit des Aufstehens nicht balancierfähig ist und sich nicht wie ein Pendel in der Realität verhält. Genauso ist es je nach Design des Aufstehens gewollt, dass der *NAO* Fall- oder Schwungphasen hat, um von einer liegenden Pose in eine aufrechten Pose zu gelangen.

## Kapitel 4

# Probleme

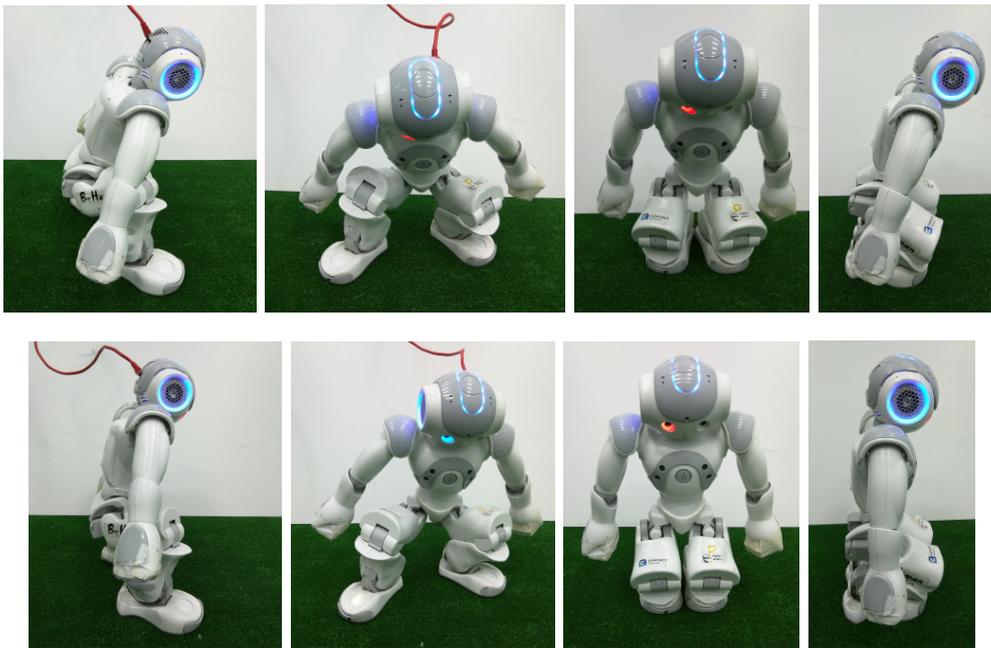
In diesem Kapitel werden die Probleme dargestellt, welche während des Aufstehens auftreten können. Für die Ausarbeitung wurde dafür das Aufstehen von B-Human aus dem Jahr 2017 und 2018 analysiert und versucht, die Probleme zu kategorisieren.

### 4.1 Gelenkwinkel werden nicht erreicht

Die Gelenkmotoren der *NAOs* können nur eine begrenzte Kraft erzeugen, um die Gelenke zu bewegen. Steht ein anderer *NAO* also auf dem eigenen Fuß oder liegt zwischen den Beinen ein Ball, so kann nicht genug Kraft erzeugt werden, um diese Einflüsse zu überwinden. Oft ist aber kein äußerer Einfluss das Problem, sondern die Motoren selber. So konnte auf dem RoboCup 2017 und 2018 beobachtet werden, dass hier die meisten Aufstehversuche für die Rückenbewegungen und der langsamen Vorderbewegungen das Problem hatten, dass sich das rechte *HipPitch*-Gelenk, im Folge *RHipPitch*-Gelenk genannt, nicht bewegen konnte (siehe Abbildung 4.2) und somit über zwei Keyframes lang die angesteuerte Körperposition komplett falsch war und das Aufstehen nur knapp funktioniert hat.

In Abbildung 4.1 ist der Unterschied zwischen korrekten und inkorrekten eingenommenen Körperposen beim *NAO* zu sehen. Bei den falsch eingenommenen Posen ist der *NAO* deutlich mehr nach vorne geneigt und somit kurz vor dem Umfallen.

Zusätzlich haben die Gelenke unterschiedlich viel Spiel. In Abbildung 4.3 bis Abbildung 4.7 ist der Verlauf des Winkels vom Knies eines *NAOs* zu sehen. Hier wurde ein Sollwinkel von null Grad angesteuert und erfolgreich gehalten. Da das Knie aber ein Spiel von bis zu sechs Grad hat, kann dieses frei bewegt werden. In Abbildung 4.3 wurde das Knie bis auf Anschlag leicht per Hand gedrückt. Der Motor regelt gegen die Verschiebung, sodass das Gelenk wieder null Grad nach ca. 0.4 Sekunden erreicht. Dadurch verschiebt sich das Gelenkspiel vollständige in die andere Richtung. Das Gelenk wird nun bis auf Anschlag in die andere Richtung bewegt und leicht gehalten. Der Motor steuert erneut an, um den Sollwinkel von null Grad zu erreichen, wodurch sich das Gelenkspiel erneut vollständig in die gegensätzliche Richtung

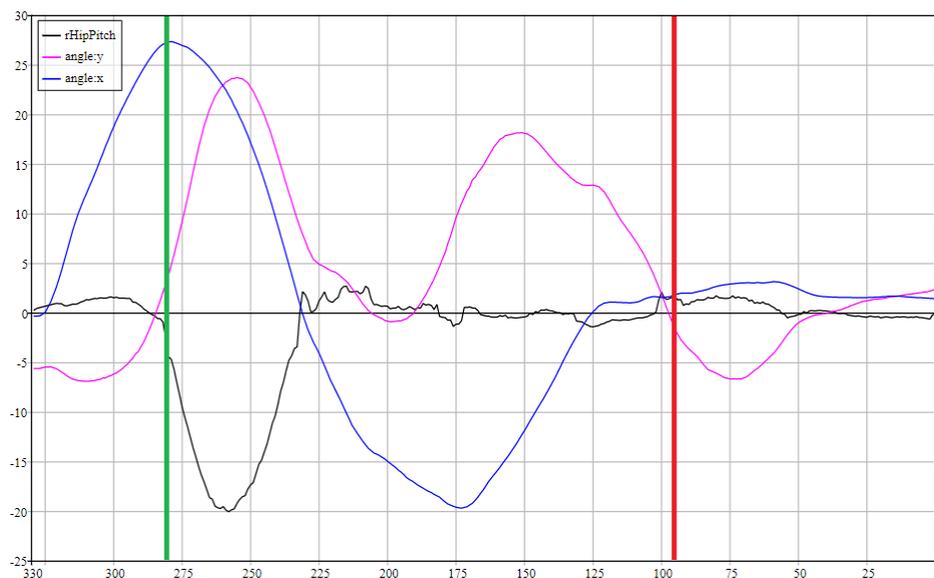


**Abbildung 4.1** Vergleich von je zwei Posen aus zwei verschiedenen Perspektiven. Oben mit und unten ohne erreichten Gelenkwinkeln. Jeweils zwei Bilder links mit und ohne nicht erreichten Gelenkwinkeln vom *RHipPitch*-Gelenk. Jeweils zwei Bilder rechts mit nicht erreichten Gelenkwinkeln für das *HipYawPitch*-Gelenk

verschiebt.

In Abbildung 4.4 wurde das Knie sehr weich gestellt, sodass das Gelenkspiel vorhanden war, aber der Motor zu schwach, um gegen die Verschiebung zu regeln. Dieser Test diente dazu zu kontrollieren, ob das Gelenkspiel und das Nachregeln eine Auswirkung auf den gemessenen Winkel haben. Eine solche Auswirkung kann aber nicht bestätigt werden. In Abbildung 4.5 wurde das Gelenk wie in Abbildung 4.3 auf Anschlag in die positive Richtung verschoben. Nach dem darauf folgenden Nachregeln des Motors wurde das Gelenk im Gelenkspiel von Anschlag zu Anschlag per Hand mehrmals bewegt. Es ist zu beobachten, dass sich der Motor trotz der dauerhaften Bewegung korrekt nachregelte, sodass sich das Gelenkspiel nach paar Sekunden wieder korrekt gleichmäßig um den angesteuerten Zielwinkel von null Grad bewegte.

In Abbildung 4.6 wurde der Ablauf von Abbildung 4.3 wiederholt. Hier wurde das Knie nicht bis auf Anschlag des Spiels gedrückt, sondern innerhalb des Spiels gehalten. Es ist zu beobachten, dass der Motor dadurch deutlich länger brauchte, bis dieser nachregelt. In Abbildung 4.7 wurde der Roboter vom Sitzen ins Stehen gesetzt (der erste Peak am Anfang). Sobald die Gelenke ihre Zielwinkel erreicht hatten, kam das Knie direkt am Anfang in sein Spiel und der Motor regelte dagegen an. Danach wurde der *NAO* hochgehoben. Das Knie geriet nur sehr leicht in sein Gelenkspiel, wodurch erst nach einer Sekunde der Motor anfang nach zu regeln. Dadurch ist das Gelenkspiel erneut in eine Richtung verschoben und sobald der *NAO* wieder hingestellt wird, geriet das Knie an den Anschlag seines Gelenkspiels und der Motor regelte wieder dagegen an.



**Abbildung 4.2** Der Ablauf der Aufstehbewegung vom Rücken ab den Zeitpunkt, wenn der *NAO* eine breitbeinige Pose erreicht hat. Dargestellt ist der Unterschied von gesetzten und erreichten Gelenkwinkeln des *RHipPitch*-Gelenkes (schwarz), die Torstoneigung nach vorne (magenta) und die Torstoneigung zur Seite (blau). Eine Neigung nach rechts ist dabei positiv. Grün markiert den Zeitpunkt, zu dem der *NAO* anfangen will, sein *RHipPitch*-Gelenk zu bewegen. Rot markiert den Zeitpunkt, zu dem der *NAO* in einer sitzenden Position ankam. Die Y-Achse ist in Grad, die X-Achse die Anzahl der vergangenen Frames.

Dieses Nachregeln kommt nicht von dem verwendeten B-Human System sondern von der Herstellersoftware. Dieser Effekt kann nicht ausgestellt werden, soll aber auch nicht in dieser Arbeit weiter untersucht werden und wird somit ignoriert. Die Verfahren, die in dieser Arbeit entwickelt werden, müssen daher mit diesem Effekt umgehen können. Mit diesen Tests wurde aber sichergestellt, dass die gemessenen Gelenkwinkel denen der Realität entsprechen und keine Fusion mehrere Messungen oder Sensoren ist.

Zusätzlich zu den genannten Problemen sind die Untergründe, auf denen Wettbewerbe ausgetragen werden, sowie zum Testen vor Ort bei B-Human, unterschiedlich. So ist der Boden bei B-Human ein plattgetretener Kunstrasen, während auf den Wettbewerben meist ein neuer Kunstrasen verwendet wird. Dadurch ergibt sich ein Unterschied im Bodenwiderstand, welcher einen Einfluss auf das Aufstehen nehmen kann. Denn je höher der Bodenwiderstand ist, desto schwerer haben es die Beinmotoren, sich zu bewegen.

Ein weiterer Effekt durch das Eingreifen der Herstellersoftware ist, dass diese erkennt, falls Gelenke zu viel und zu lange beansprucht wurden. Jedem Gelenk wird somit eine berechnete Temperatur und ein Temperaturstatus zugeordnet. Je schlechter, also heißer dieser ist, desto stärker wird die Festigkeit der betroffenen Gelenke reduziert. Dieser Effekt ist nicht deaktivierbar und sorgt in Wettkampfspielen dazu, dass die *NAOs* bei längeren Spielen um

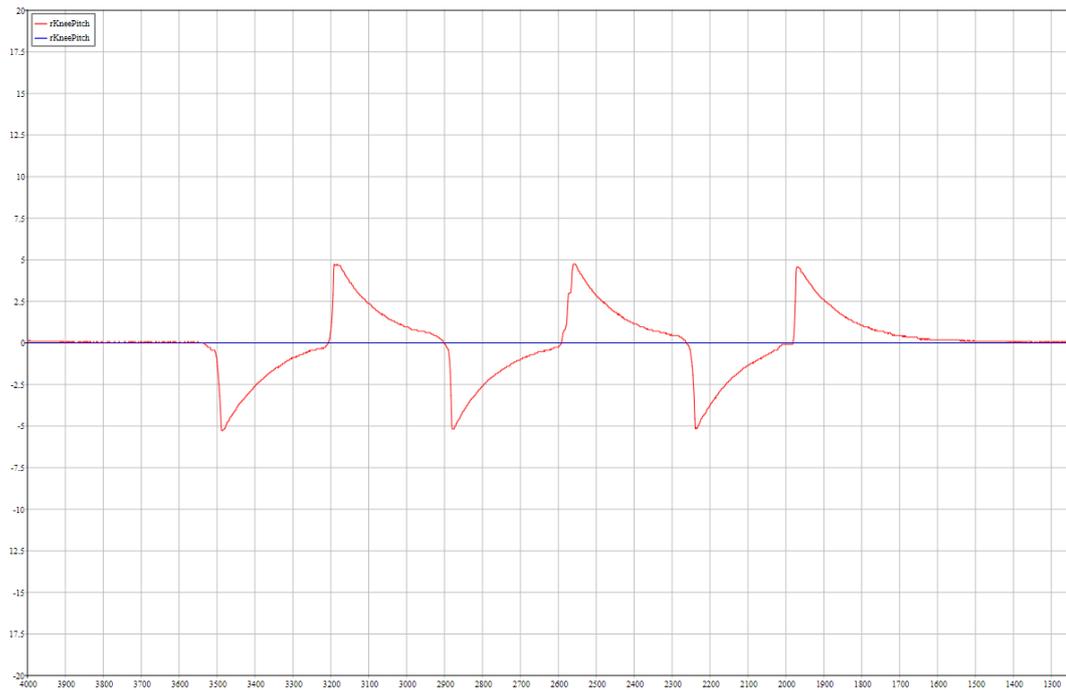
so öfter umfallen, da die Gelenke nachgeben. Die so weich gestellten Gelenke nehmen dann auch Einfluss auf das Aufstehen, welches dann potenziell häufiger scheitert.

## 4.2 Statische Bewegungen

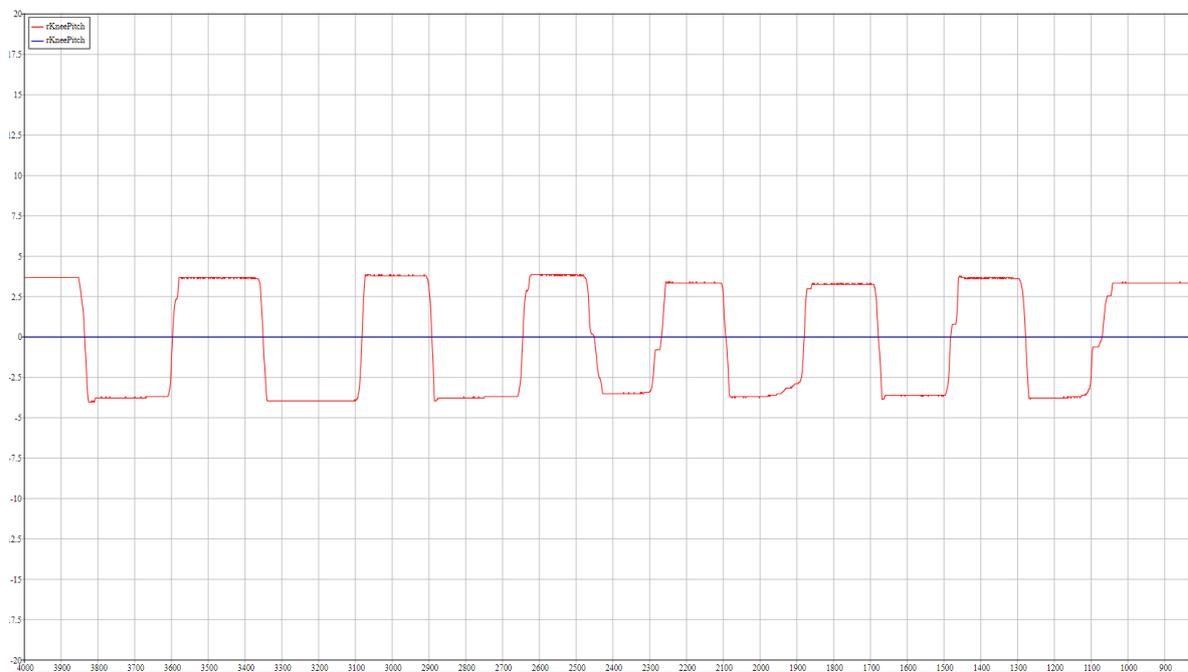
Jeder *NAO* ist, wie bereits erwähnt, unterschiedlich stark abgenutzt, wodurch die Gelenke unterschiedlich viel Kraft und Spiel haben. Auch ist nicht jede Position auf dem Spielfeld gleich. So gab es auf dem RoboCup 2018 Unebenheiten, auf denen die *NAOs* einige Zentimeter in den Boden eingesackt sind, sowie Kanten im Feld, da der Kunstrasen auf Holzplatten aufgebaut wurde, welche keinen gleichmäßigen Übergang hatten. All diese Faktoren sorgen dafür, dass jeder Aufstehversuch unterschiedlich ist. Es kann also keine perfekte statische Aufstehbewegung erstellt werden, die immer unter allen Bedingungen perfekt funktioniert.

## 4.3 Schwer zu balancierende Zustände

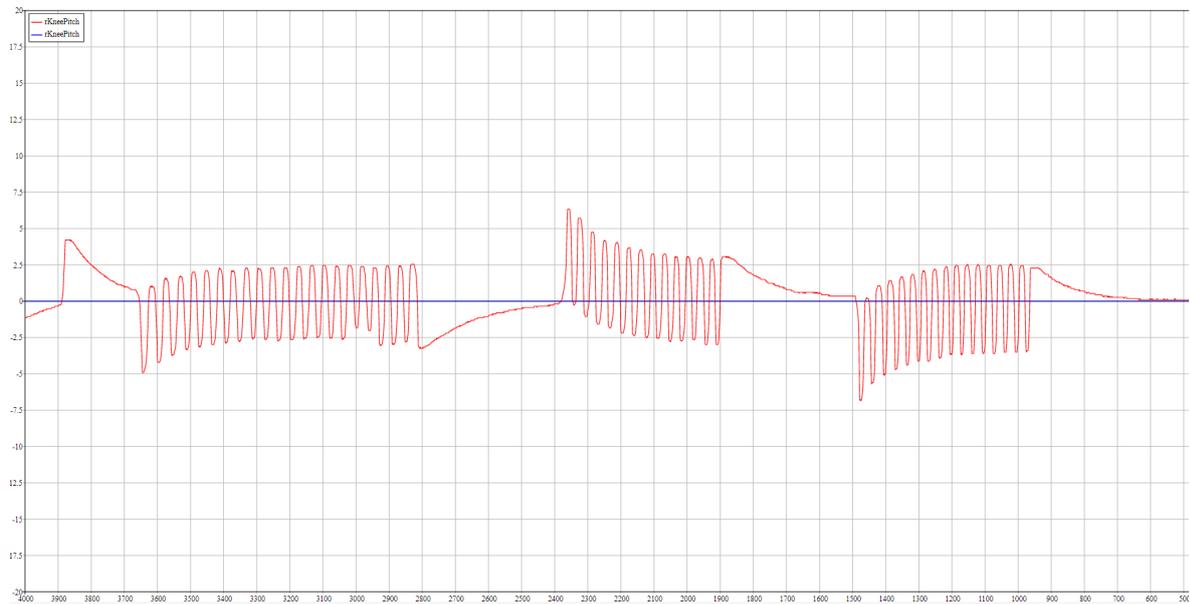
Da das Aufstehen nicht zu jedem Zeitpunkt balanciert werden kann, beispielweise wenn der *NAO* noch auf dem Boden liegt, müssen diese Fälle zusätzlich behandelt werden. Solche Fälle sollen defekte Gelenke, aber auch falsche Körperneigungen, behandeln, damit nachfolgende Keyframes korrekt ablaufen können. Ein Beispiel aus dem RoboCup 2018 ist, dass einer der *NAOs* einen defekten Magnetsensor im Ellbogen hatte. Dadurch konnte er bei der Rückenbewegung oft nicht seinen Arm hinter den Rücken bekommen und blieb damit auf dem Boden liegen. Abgesehen von seinem Arm war der *NAO* in einem sehr guten Zustand, weshalb dieser Fall um so schlimmer war, da dadurch der *NAO* unter anderem in einer Situation nicht rechtzeitig aufstehen konnte und aufgrund der Regeln aus dem Spiel genommen werden musste.



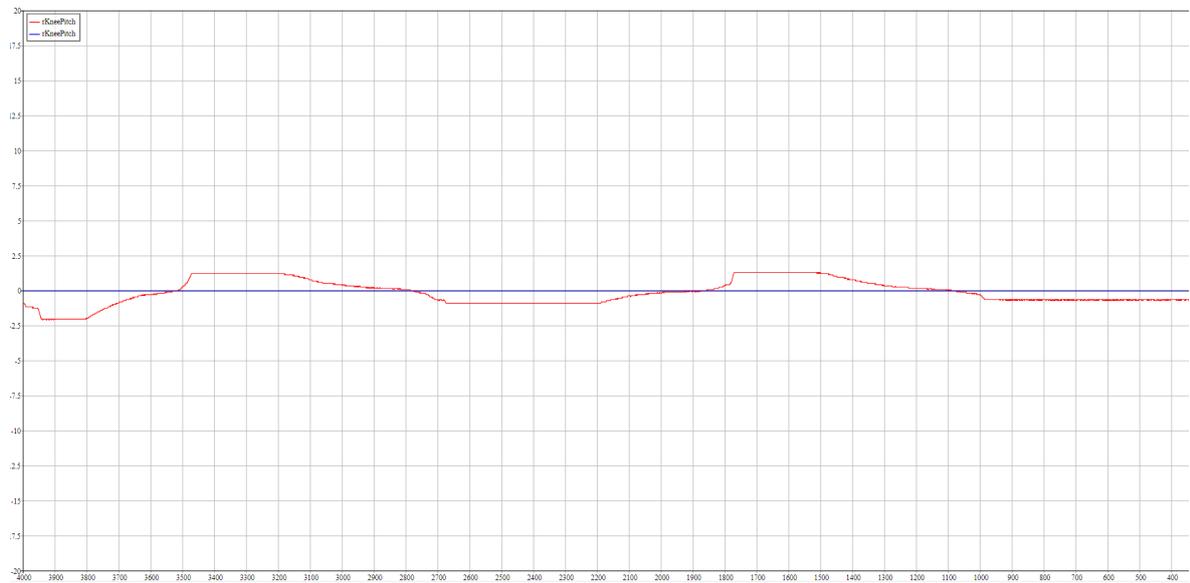
**Abbildung 4.3** Ein Gelenk unter leichter Belastung. Die Y-Achse beschreibt die Abweichung von Soll-Ist in Grad, die X-Achse die Anzahl der vergangenen Frames.



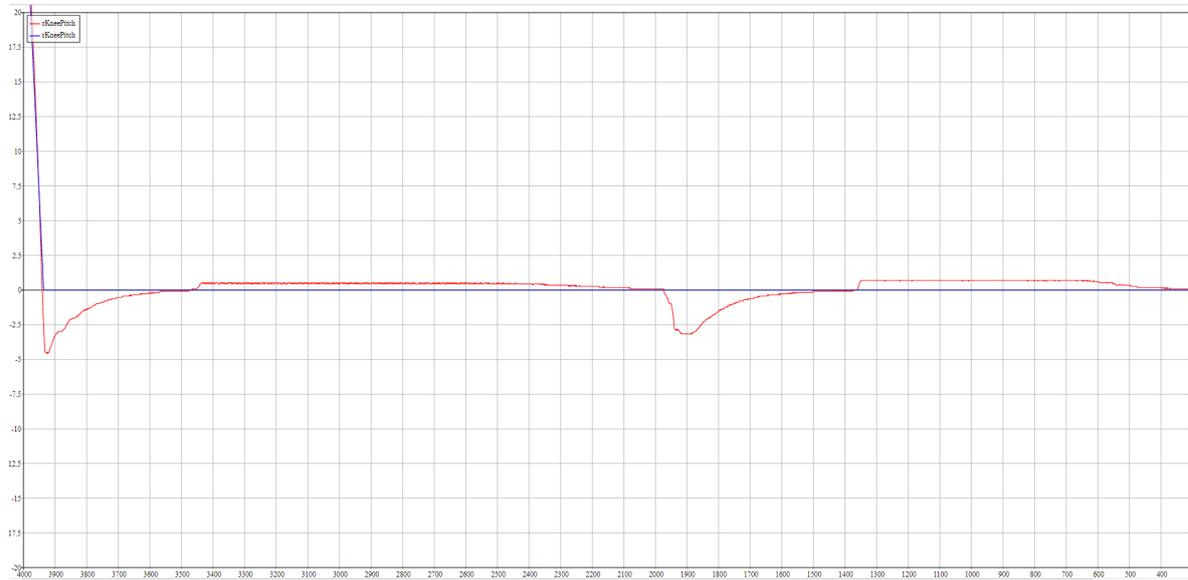
**Abbildung 4.4** Ein weich gestelltes Gelenk unter leichter Belastung. Die Y-Achse beschreibt die Abweichung von Soll-Ist in Grad, die X-Achse die Anzahl der vergangenen Frames.



**Abbildung 4.5** Ein Gelenk welches von Anschlag zu Anschlag im Gelenkspiel bewegt wird. Die Y-Achse beschreibt die Abweichung von Soll-Ist in Grad, die X-Achse die Anzahl der vergangenen Frames.



**Abbildung 4.6** Ein Gelenk welches innerhalb des Gelenkspiels gehalten wird. Die Y-Achse beschreibt die Abweichung von Soll-Ist in Grad, die X-Achse die Anzahl der vergangenen Frames.



**Abbildung 4.7** Ein Gelenk unter Be- und Entlastung, während der *NAO* hingestellt und hochgehoben wird. Die Y-Achse beschreibt die Abweichung von Soll-Ist in Grad, die X-Achse die Anzahl der vergangenen Frames.



## Kapitel 5

# Entwicklung

Damit die Probleme, wie in Kapitel 4 beschrieben, vermieden werden können, werden verschiedene Systeme entwickelt, welche als separate Aufgabe haben, diese zu verhindern.

### 5.1 Die Aufstehbewegungen

Das Ziel der zu entwickelnden Aufstehbewegungen ist, dass diese möglichst schnell sind und auf allen *NAOs* funktionieren. Ebenfalls sollten sie bereits ohne weitere Balanciermaßnahmen und Fehlerzustandsvermeidungen häufig funktionieren. Die Bewegungsabläufe müssen also berücksichtigen, dass Gelenkwinkel um einige Grad abweichen und dass das Spiel der Gelenke unterschiedliche Körperneigungen erzeugen kann.

#### 5.1.1 Die verwendeten Aufstehbewegungen

Im Rahmen dieser Bachelorarbeit werden keine vollständig neuen Aufstehbewegungen entwickelt, sondern deren Abläufe werden weniger fehleranfällig gestaltet und optimiert. Dafür wird mit den Aufstehbewegungen gearbeitet, die bereits im B-Human System vorhanden sind. Wie bereits im Abschnitt 3.1.1 erwähnt, können die alten Aufstehbewegungen im Code Release 2018 von B-Human [vgl. Röfer u. a., 2018] nachgelesen werden.

Ebenfalls werden vorab alle Keyframes, die doppelt sind, entfernt. Diese hatten nur die Auswirkung, dass der *NAO* kurzzeitig in der Aufstehbewegung pausiert hat, um sicherzustellen, dass die Gelenke ihre Zielwinkel aus dem letzten Keyframe erreicht haben. Da das aber nur unnötig Zeit kostet, sollen Fehlstellungen durch die Verfahren aus Abschnitt 5.5 und Abschnitt 5.6 ausgeglichen werden.

### 5.1.2 Zustandsabhängigkeiten

Es ist zu beachten, dass in einigen Körperpositionen bestimmte Gelenkwinkel nicht erreichbar sind. So kann ein *NAO*, der auf seinen Armen liegt, diese nicht einfach ausstrecken, da die Armgelenke nicht stark genug sind. Ebenfalls ist es nur selten möglich, die Beingelenke über den Boden zu schleifen, wenn viel Gewicht auf dem Bein liegt (siehe Abschnitt 4.1). Die Bewegung muss also so ablaufen, dass eine Maximallast nicht überschritten wird, wenn sich die Beingelenke bewegen müssen. Da diese Maximallast bei den *NAOs* unterschiedlich ist, wird experimentell ermittelt, wie zeitlich verzögert betroffene Gelenke sich erst bewegen sollten, um diesem Problem entgegen zu wirken. Dabei bin ich folgendermaßen vorgegangen: Ein *NAO* wurde hingelegt, damit dieser das Aufstehen ausführt. Danach wurde im aufgezeichneten Log, welches vom B-Human Framework aufgezeichnet wird, untersucht, wie die Torsoneigung und der Unterschied der erreichten und gesetzten Gelenkwinkel war. Ist die Torsoneigung zu stark abweichend zu der aus der Ruhelage oder ist der Unterschied der erreichten und gesetzten Gelenkwinkel zu groß, wurde mit drei weiteren Versuchen geprüft, ob dieses Problem häufiger auftritt oder nur einmalig war. Tritt es mehrmals auf, so wurden die Keyframes insofern umgeändert, dass diese das betroffene Gelenk auf die Winkel setzt, die auch erreicht werden. Da so der Bewegungsablauf verändert wird, wurden mit nicht betroffenen Gelenken die Roboterpose so verändert, dass die Torsoneigung in der Ruhephase dieselbe ist.

Sobald ein Zustand erreicht wurde, in dem ich nichts mehr ändern musste, wurde auf mehreren *NAOs* untersucht, ob das Problem universell gelöst ist.

Da zum Beispiel die Arme ungeeignet sind, um eine falsche Roboterpose auszugleichen und planare Füße zum Boden am besten für eine stabile Körperhaltung sind, wurden folgende Regeln aufgestellt, mit der die Torsoneigung und falsche Gelenkwinkel ausgeglichen werden sollten:

Die *AnklePitch*-Gelenke können immer zum Ausgleich verwendet werden, wenn die Torsoneigung zu weit nach vorne ist.

Die *HipPitch*-Gelenke können verwendet werden, sofern diese keiner zu großen Last ausgesetzt sind und falls Schwungphasen bei dem Umändern der *AnklePitch*-Gelenke nicht mehr funktionieren.

Die *AnkleRoll*-Gelenke werden verwendet, um die Planarität der Füße zum Boden zu sichern.

#### 5.1.2.1 Umändern der Zustandsabhängigkeiten

Da in dieser Arbeit möglichst wenige Aufstehbewegungen existieren sollen, wird für das Aufstehen vom Rücken nur die Bewegung *BackFast* übernommen, da die Bewegung *BackFreeJoints* zu dieser zu ähnlich ist.

Die *FrontFast* Bewegung wird so abgeändert, dass die Arme nicht erst nach vorne bewegt werden, wenn die Beine sich zusammenziehen, sondern bereits im ersten Keyframe. In der

originalen Bewegung sollte die Bewegung des Zusammenziehens der Beine den *NAO* so über den Boden schieben, dass die Arme automatisch nach vorne bewegt werden. Die Arme sind, so wie sie verbaut sind und aufgrund der Kraft, die die Motoren ausüben können, ohne dieses Schieben nicht in der Lage, sich nach vorne zu bewegen. Indem die Arme aber bereits so positioniert werden, dass sie nicht mehr auf das über den Boden schieben warten müssen, wird die Wahrscheinlichkeit erhöht, dass sich die Arme nach vorne bewegen können, sobald der *NAO* seine Beine zusammenzieht. Diese Änderung funktionierte in den Tests so gut, dass unter anderem die Festigkeit der Armgelenke von 100% auf 40% reduziert werden konnte.

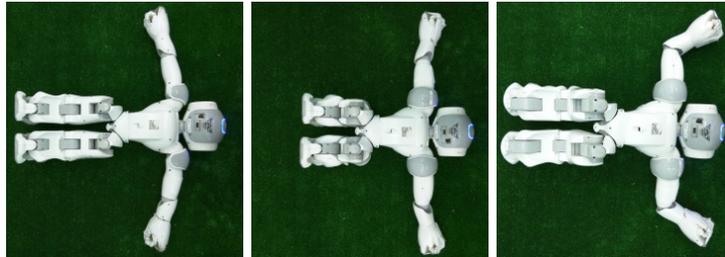
Die zweite Änderung ist die Umgehung des Problems, dass sich das *RHipPitch*-Gelenk bei der Bewegung von *BackFast* fast nie bewegen kann und der *NAO* somit eine viel zu stark nach vorne geneigte Torsolage bekommt. Hierfür wurde der Keyframe, in dem sich der *NAO* auf sein zweites Bein, dem rechten, bewegt, so geändert, dass das *RHipPitch*-Gelenk vorerst geschlossen bleibt und sich nicht ändert. Die *RAnklePitch*- und *RAnkleRoll*-Gelenke wurden so umgeändert, dass die Planarität des Fußes zum Boden sichergestellt ist und die geplante Torsoneigung beibehalten wird. Die linken Beingelenke wurden leicht abgeändert, sodass der *NAO* etwas mehr nach links geneigt ist und somit potenziell weniger Gewicht auf dem rechten Bein liegt.

Um nun die Bewegung nach dieser Änderung zu vervollständigen, wurde ein Keyframe hinzugefügt, in dem sich das *RHipPitch*-Gelenk nun öffnet. Die anderen Beingelenke, abgesehen von den Knien, werden angepasst, sodass die Torsoneigung gleich bleibt.

Mit diesen Änderungen wurden mehrere *NAOs* aufstehen gelassen und als einziges Problem konnte beobachtet werden, dass das *HipYawPitch*-Gelenk manchmal bis zu 15 Grad zu weit offen blieb. Dies kann zwei Gründe haben. Zum einen besteht der Zusammenhang, wenn der Torso zu weit nach vorne geneigt ist, so ist der Motor vom *HipYawPitch*-Gelenk nicht stark genug, um gegen das Gewicht anzuarbeiten. Zum anderen kann das *HipYawPitch*-Gelenk stark abgenutzt sein und zu viel Spiel besitzen. Da der neueste *NAO* bereits 1.5 Jahre alt ist und an zwei Wettbewerben teilgenommen hat, kann diese Vermutung nicht getestet werden. Dieses Problem wird daher in Abschnitt 5.2, Abschnitt 5.4, Abschnitt 5.5 und Abschnitt 5.6 untersucht.

### 5.1.3 Die angepassten Bewegungen

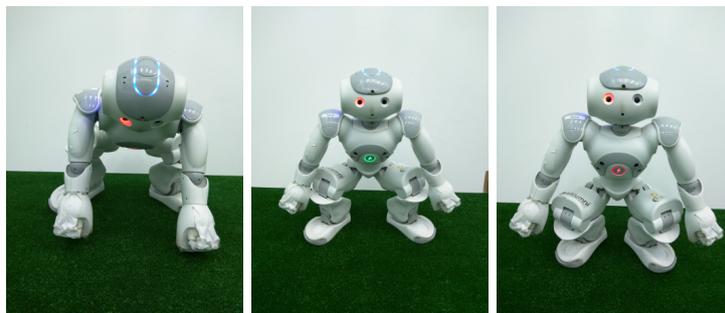
Die schnelle Vorderbewegung *FrontFast* ist nun wie in Abbildung 5.1, Abbildung 5.2 und Abbildung 5.3 dargestellt und braucht ungefähr 2.8 Sekunden:



**Abbildung 5.1** *FrontFast* Teil 1  
Von links nach rechts: Armposition vorbereiten

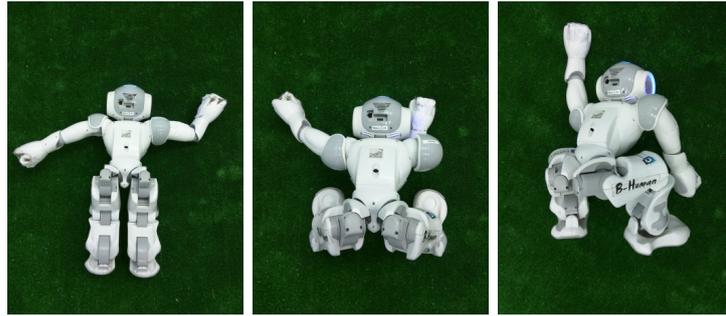


**Abbildung 5.2** *FrontFast* Teil 2  
(Der selbe Keyframe) Auf die Arme stützen



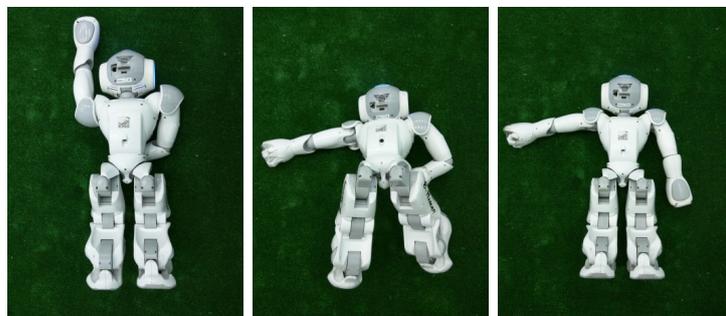
**Abbildung 5.3** *FrontFast* Teil 3  
Von links nach rechts: Aufrichten  
Auf diese Bewegung folgt *FromSplitt* (siehe Abbildung 5.13)

Die Positionierung der Arme in Abbildung 5.1 soll dabei deren Fehlposition, wie in Abbildung 5.4 zu sehen, verhindern.



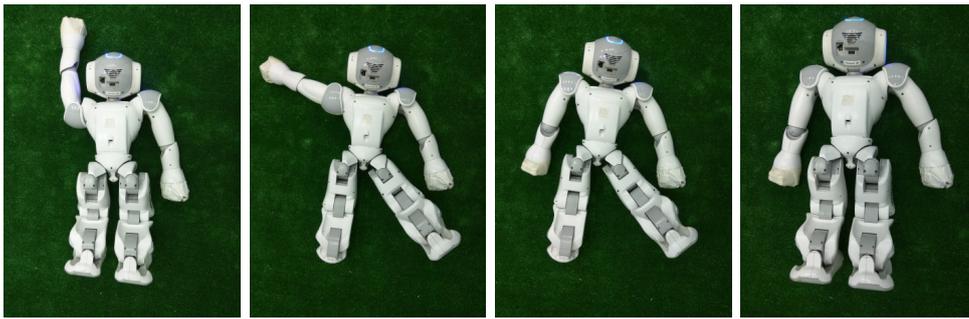
**Abbildung 5.4** Fehler bei *FrontFast*  
 Von links nach rechts: Linker Arm bleibt hängen und dient nicht mehr als Stütze

Da es nach einem Torwertsprung vorkommen kann, dass der *NAO* auf einen seiner Arme liegt, soll er, bevor die Bewegung *FrontFast* ausführt wird, die Bewegung *LyingOnArm* Abbildung 5.5 ausführen, um den Arm zu befreien. Diese Bewegung dauert 2.1 Sekunden.



**Abbildung 5.5** Bewegung *LyingOnArm*. Nach einem Torwertsprung bleibt der *NAO* auf dem Arm liegen  
 Von links nach rechts: Hüfte wird hochgehoben, um den Arm frei zu bekommen

Nachdem die Bewegung *LyingOnArm* ausgeführt wurde, ist weiterhin ein Arm noch ausgestreckt. Da diese Fehlposition bei der Bewegung *Front* Abbildung 5.7 zu einem eingeklemmten Arm führen würde, soll dieser vorher ebenfalls mit der Bewegung *ArmStretched* Abbildung 5.6 neben den Körper bewegt werden. Ein eingeklemmter Arm, wie mehrfach auf dem RoboCup 2018 beobachtet, führt jedes Mal zu einem Fehlversuch. Die Bewegung dauert 2.4 Sekunden.



**Abbildung 5.6** Bewegung *ArmStretched*  
 Von links nach rechts: Gegengesetztes Bein ausstrecken und durch die Kippbewegung den Arm von vorne nach hinten holen

Der Ablauf der langsamen Vorderbewegung *Front* ist in Abbildung 5.7 dargestellt (sie hat weiterhin als Nachfolger eine Bewegung, diese ist die Bewegung *textitFromSplitt* Abbildung 5.12). Änderung ist eine Wartezeit in der Pose im 3. Bild, bis er übergekippt ist. Dies hat den Grund, wie mehrmals auf dem RoboCup 2018 zu beobachten war, dass die *NAOs* manchmal am Boden hängen und auf dem Bauch liegen blieben. Mit dem Ansatz aus Abschnitt 5.4 wird nach dem 3. Keyframe gewartet, bis der *NAO* eine Mindesttorsionseigung hat, um sicherzustellen, dass er überkippt und nicht auf dem Bauch liegen bleibt. Die Bewegung dauert 3.4 Sekunden ohne Wartezeit.



**Abbildung 5.7** Bewegung *Front*  
 Von links nach rechts: Arme nach hinten bewegen und aufrichten  
 Auf diese Bewegung folgt *FromSplitt* (siehe Abbildung 5.12)

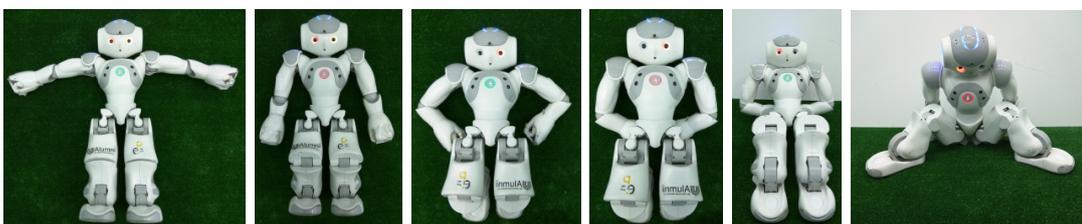
Da es, wie bereits erwähnt, vorkommen kann, dass die Bewegung *Front* mit einem anfangs ausgestreckten Arm ausgeführt wird, soll sichergestellt werden, dass, falls die Bewegung *ArmStretched* (siehe Abbildung 5.6) fehlschlägt, die Bewegung *Front* nicht automatisch scheitert.

Wird also ein fehlpositionierter Arm durch das Verfahren aus Abschnitt 5.4 erkannt, so wird die Bewegung abgewandelt, wie sie in Abbildung 5.8 zu sehen ist. Bis der *NAO* in diesem Fall mit der Bewegung *Front* fertig ist dauert es 4.5 Sekunden.



**Abbildung 5.8** Back up für die Bewegung *Front*  
 Von links nach rechts: Arm vorne behalten, die Beine möglichst weit weg bewegen und den Arm um die Beine herum bewegen

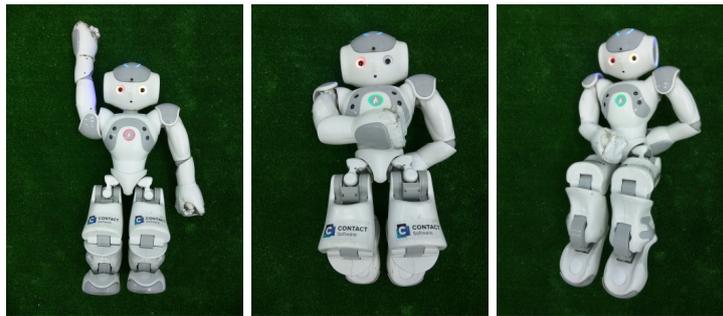
Die Rückenbewegung ist wie in Abbildung 5.9 gestaltet. In dem Keyframe, in dem der *NAO* seine Beine nach vorne austreckt, um nach vorne zu kippen, wird das Verfahren aus Abschnitt 5.4 verwendet. Dadurch wartet dieser in diesem Keyframe, bis er genügend nach vorne gekippt ist. So wird verhindert, dass der *NAO* bei unebenen Böden wieder nach hinten fällt. Die Bewegung dauert ca. 1.7 Sekunden ohne die Wartezeit am Ende.



**Abbildung 5.9** Bewegung *Back*  
 Von links nach rechts: Arme hinter den Rücken bewegen und aufrichten  
 Auf diese Bewegung folgt *FromSplitt* (siehe Abbildung 5.12)

Auf dem RoboCup 2018 konnte bei der Bewegung *Back* ein Problem beobachtet werden: wenn einer der Arme eine Beschädigung hat, schafft es der *NAO* meistens nicht, seinen Arm rechtzeitig hinter seinen Rücken zu bekommen. Dadurch funktioniert die Kippbewegung nicht mehr und der Roboter bleibt auf dem Boden liegen.

Ebenfalls kann es vorkommen, dass der Torwart nach einem Sprung mit einem ausgestreckten Arm liegen bleibt. Dabei kann es vorkommen, dass der Arm nicht schnell genug hinter den Rücken bewegt werden kann (siehe Abbildung 5.10). Um dies zu verhindern, wird ebenfalls mit dem im Abschnitt 5.4 vorgestellten Ansatz ein Back up für diesen Fall hinzugefügt. Wird in der Bewegung *Back* festgestellt, dass ein Arm zu stark von seinen gesetzten Gelenkwinkeln abweicht, so werden zusätzlich 3 Keyframes ausgeführt, in denen der Arm hochgehoben wird (siehe Abbildung 5.11). Dieses Back up dauert zusätzlich 0.7 Sekunden.



**Abbildung 5.10** Falsch positionierter Arm in der Bewegung *Back*  
 Von links nach rechts: Ein Arm ist ausgestreckt, kann nicht schnell genug bewegt werden und bleibt am Torso hängen



**Abbildung 5.11** Back up in der Bewegung *Back*  
 Von links nach rechts: Arm wird hochgehoben und danach wieder hinter den Rücken bewegt

Zuletzt gibt es noch die Bewegung *FromSplitt*, die aus der breitbeinigen Pose startet, wie in Abbildung 5.12 zu sehen. Sie ist der Nachfolger für die Bewegungen *Front* und *Back*, sowie für das Aufstehen aus der im B-Human System verwendeten breitbeinigen Pose, welche zum Annehmen von Bällen verwendet wird. Sie ist sehr ähnlich zu der Pose, die der *NAO* als letzten Keyframe in den Bewegungen *Front* und *Back* einnimmt. Diese Bewegung dauert 1.4 Sekunden.



**Abbildung 5.12** Bewegung *FromSplit*  
 Von links nach rechts: Bewegung auf das linke Bein, dann auf das rechte Bein,  
 dann Öffnen des *RHipPitch*-Gelenkes  
 Auf diese Bewegung folgt das Sitzen und Stehen (siehe Abbildung 5.13)

Die Bewegungen *FrontFast* und *FromSplit* gehen alle in dieselbe sitzende Körperpose über, nachdem diese beendet sind. Diese ist in Abbildung 5.13 dargestellt. Das Sitzen und Hochdrücken ins Stehen dauert zusammen 0.7 Sekunden.



**Abbildung 5.13** Vom Sitzen ins Stehen

Insgesamt dauert das Aufstehen von vorne im Schnitt 3.6 Sekunden und von hinten im Schnitt 4 Sekunden. Im Vergleich dazu dauert das vorherige Aufstehen für vorne zwischen 3 (*FrontFast*) und 6.8 Sekunden (*FrontFreeJoints*) und für hinten 4.7 Sekunden (*BackFast*).

## 5.2 Pausieren in der Bewegung

Um nicht erreichte Gelenkwinkel sowie zu starke oder zu schwache Schwungphasen während des Aufstehens besser unter Kontrolle zu bringen, sollen zu stark abweichende Bewegungszustände pausieren und ausbalanciert werden. Ziel ist dabei, dass der *NAO* aufgrund von Schwankungen oder nicht umgesetzten Gelenkwinkeln in einer aufrechten Körperlage bleibt, bis entschieden wird, dass die Bewegung fortgeführt werden kann.

Die Umsetzung erfolgt, indem jedem Keyframe eine Torsoneigung für vorne und hinten zugewiesen wird. Ebenfalls werden diese Torsoneigungen zwischen den Keyframes linear interpoliert. Wenn nun während des Aufstehens diese Torsoneigungen überschritten werden, wird das Aufstehen pausiert und so lange balanciert, bis der *NAO* wieder innerhalb des zugewiesenen

Neigungsbereiches ist.

### 5.2.1 Evaluation der Pausierung in der Bewegung

Für die Evaluation wurde kein Balancierverfahren verwendet. Es wurde die alte Aufstehbewegung *Back* zum Evaluieren verwendet und der *NAO* mit der Hand leicht angestupst. Ebenfalls wurde das Pausieren nur aktiviert, nachdem der *NAO* bereits eine breitbeinige aufrechte Körperlage erreicht hat. Für die Evaluation wurde ein *NAO* in echt sowie in der Simulation 5 mal aufstehen gelassen.

In allen Versuchen konnte dasselbe Problem beobachtet werden. Dadurch, dass die Aufstehbewegung keine stationäre Bewegung ist, in der der *NAO* dauerhaft eine feste Körpeneigung hat, sondern ein Momentum besitzt, ist die Folge, dass die gesamte Aufstehbewegung nicht mehr funktioniert, wenn die Schwungphase aus dem Rhythmus kommt. Wenn der *NAO* ein Momentum nach vorne hat und nicht umfällt, bekommt er ein Momentum nach hinten und fällt dadurch meistens hin. Dies gilt auch für den umgekehrten Fall.

Ebenfalls können weitherhin Gelenke, wenn sie sich nicht bewegen können, nicht ihre Zielwinkel erreichen, sodass insgesamt nur Zeit beim Aufstehen verschwendet wird.

Das Pausieren in der Bewegung wirkt sich demzufolge nicht positiv auf das Verhalten des *NAOs* aus, sondern bringt ihn lediglich in einen anderen inkorrekten Zustand. Ein Balancierverfahren müsste gegen das Pausieren sowie gegen die Aufstehbewegung arbeiten. Ich habe mich deshalb dazu entschieden, ein Pausieren in der Bewegung nicht zu verwenden. Stattdessen soll mit dem Verfahren aus Abschnitt 5.4 die Zustandsübergänge stabilisiert werden, sofern der *NAO* noch auf dem Boden liegt oder sich in einer sitzenden oder stehenden Pose befindet.

## 5.3 Abbrechen von Bewegungen

Da sich die *NAOs*, wenn sie die Aufstehbewegung nicht schaffen können, Schaden zufügen, wenn sie die Bewegung weiter ausführen, ist es wichtig, in dem frühestmöglichen Moment die Aufstehbewegung abbrechen. Außerdem sorgt ein frühes Erkennen und Abbrechen von gescheiterten Aufstehbewegungen für eine Zeitersparnis, sodass der *NAO* schnell wieder steht.

Da jeder Keyframe eine unterschiedliche maximale Torsoneigung und Schwerpunktverlagerung zulässt, ist es sehr zeitaufwendig eine frühestmögliche Fallerkennung umzusetzen. Zudem soll das Aufstehen auch abbrechbar sein, wenn der *NAO* noch in Körperpositionen wie in Abbildung 5.2 zu sehen ist. Dafür wäre zwar eine Erkennung wie in der Arbeit von Krause [2018] beschrieben, indem der Schwerpunkt in der Fußebene modelliert wird, theoretisch umsetzbar. Für einen Großteil der Zeit ist aber beim Aufstehen der Schwerpunkt weit außerhalb der Fußebene, wodurch eine Modellierung erschwert wäre, weshalb ich mich für eine einfa-

chere Implementierung entschieden habe. So wird dasselbe Verfahren wie in Abschnitt 5.2.1 übernommen. Die gesetzten Torstoneigungen sind hier höher und so gewählt, dass ein Aufstehen für den *NAO* unmöglich ist, um False-Positive-Fälle zu verhindern. Um nun Schäden zu minimieren, wird dasselbe Vorgehen verwendet, welches bereits nach einer Fallerkennung im B-Human System umgesetzt ist. So werden alle Gelenke beim Erkennen einer gescheiterten Aufstehbewegung von der Festigkeit auf 20% gesetzt und der Kopf so geneigt, dass dieser nicht als erstes auf dem Boden aufkommt. Das heißt bei einem Fall nach vorne wird der Kopf nach hinten geneigt, bei einem Fall nach hinten nach vorne. Zusätzlich wird zwei Sekunden gewartet, bevor ein neuer Aufstehversuch gestartet wird, um Blockkaden mit anderen *NAOs* zu verhindern.

## 5.4 Bedingungen und optionale Keyframes

Wie bereits in Kapitel 4 und Abschnitt 5.1.3 erwähnt, ist bei statischen Bewegungen mit dem Auftreten von diversen Problemen zu rechnen. So können andere Bewegungen vorher ausgeführt worden sein, welche dazu führen, dass am Anfang der Aufstehbewegungen keine Standardpose eingenommen werden kann. Dies ist der Fall nach einem Torwartsprung, nach einem gescheiterten Aufstehversuch oder aufgrund von beschädigten Gelenken.

Um diese Fälle unter Kontrolle zu bringen, werden vordefinierte Bedingungen implementiert. Mit diesen soll entweder einmalig vor oder nach einem Keyframe abgefragt werden, ob ein Eingreifen erforderlich ist, um ein erkanntes Problem beim Aufstehen zu lösen. Die Bedingungen können die Torstoneigung, fehlpositionierte Gelenke oder das Abbrechen des Aufstehversuches sein.

Wird nun vor einem Keyframe eine solche Bedingung abgefragt und als positiv ausgewertet, so wird dieser Keyframe sowie alle weiteren zu diesem Keyframe gehörigen Keyframes ausgeführt, ansonsten werden sie übersprungen.

Damit soll unter anderem geprüft werden, ob in der Bewegung *Back* (siehe Abbildung 5.9) das Back up für einen fehlpositionierten Arm (siehe Abbildung 5.11) beziehungsweise ob in der Bewegung *Front* (siehe Abbildung 5.7) das Back up für einen fehlpositionierten Arm (siehe Abbildung 5.8) ausgeführt werden muss.

Wird eine solche Bedingung am Ende eines Keyframes abgefragt, so soll der aktuelle Aufstehversuch pausiert werden, bis die Bedingung erfüllt, eine maximale Wartezeit vergangen ist oder erkannt wurde, dass die Bedingung nicht erfüllbar ist.

Diese Vorgehensweise lässt die Keyframes warten, bis der *NAO* eine bestimmte Körperlage erreicht hat. So können die Aufstehbewegungen in jedem Versuch identischer ausgeführt werden. Diese dynamische Wartezeit wird am Ende der Bewegung *Back* (siehe Abbildung 5.9) verwendet. Dadurch wird gewartet, bis sich der *NAO* aufgerichtet hat und er nicht, während

er nach vorne kippt und sich aufrichtet, die Aufstehbewegung zu früh fortführt und riskiert, auf dem Boden liegen zu bleiben.

Bei der Bewegung *FrontFast* wird in der Pose, in der der *NAO* im ersten Keyframe ist (siehe Abbildung 5.3), gewartet, bis eine Torstoneigung von 70 Grad erreicht wurde. Grund dafür ist, dass die *NAOs* ohne diese Wartezeit mit dem nachfolgenden Keyframe fortfahren würden, während sie noch eine Torstoneigung zwischen 130 und 100 Grad haben. Durch das Warten positionieren sich die Gelenke besser und der Schwerpunkt des *NAOs* wird weiter nach hinten verschoben, sodass er sich gezielter auf die Beine hochdrücken kann und weniger vom Schwung und den Bodenverhältnissen abhängig ist. Nach dem darauf folgenden Keyframe wird erneut gewartet, bis der *NAO* eine Torstoneigung von 45 Grad hat. Ansonsten besäßen die *NAOs* nach diesem Keyframe eine Torstoneigung zwischen 60 und 40 Grad. Beide Wartezeiten sollen helfen, dass sich die *NAOs* in jedem Versuch identischer verhalten und dass das entstehende Momentum mithilfe des Balancierverfahrens aus Abschnitt 5.5 besser unter Kontrolle gebracht werden kann.

Ob die Wartezeiten vollständig ausgeführt werden müssen, falls sie unerfüllbar sind, wird in jedem Frame entschieden, indem die Bedingungen jedes Mal überprüft werden. Da die Torstoneigung und die fehlpositionierten Gelenke nur Zahlen repräsentieren, wird deren Abstand zum abgefragten Schwellwert berechnet. Für jeden Frame, in dem dieser Abstand größer geworden ist, wird die Häufigkeit mitgezählt. Insofern die Häufigkeit einen Schwellwert übersteigt, wird die Wartezeit abgebrochen.

## 5.5 Balancierung

Beim Aufstehen ist es das Ziel, dass der *NAO* nicht umfällt und am Ende, wenn er in das Stehen übergeht, eine möglichst geringe Schwankung hat. Ebenfalls haben die Keyframes unterschiedliche Aufgaben beim Aufstehen, welche sich unterbrechen lassen auf Schwung- und Ruhephasen.

Es wird daher ein PID-Regler, welcher den Schwerpunkt projiziert auf die Fußebene des *NAOs* als Regelgröße bekommt (siehe Abschnitt 5.5.1), entwickelt.

Ebenfalls sollen die Keyframes unterschiedliche Parameter für den PID-Regler sowie unterschiedliche Sollwerte für den projizierten Schwerpunkt bekommen.

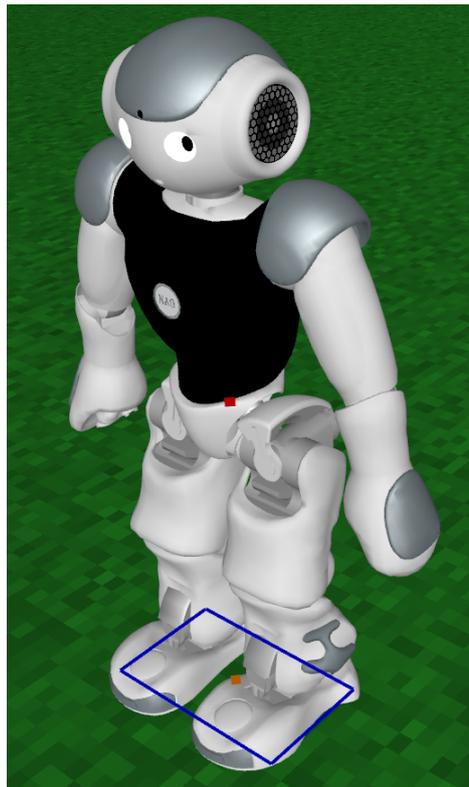
### 5.5.1 Die Regelgröße: Der Schwerpunkt in der Fußebene

Um den projizierten Schwerpunkt des *NAOs* verwenden zu können, soll dieser in Relation zum Stützpolygon betrachtet werden. Das Stützpolygon ist die konvexe Hülle der Fußfläche. Ziel ist es, den Schwerpunkt innerhalb dieser Hülle zu halten. Ist der Schwerpunkt in der Ruhephase außerhalb des Stützpolygons, so fällt der *NAO*. Dieser Ansatz wurde in der Arbeit von Krause [2018] verwendet, um zu erkennen, ob ein *NAO* fällt. In Abbildung 5.14 ist ein vereinfachtes Stützpolygon in blau und der aktuelle Schwerpunkt im Torso in rot zu sehen.

Im B-Human System existiert bereits eine Schnittstelle zur Berechnung des Schwerpunktes. Die Positionen und Gewichte der Körperteile sind bereits bekannt. Die Positionen der Körperteile werden durch deren Masse gewichtet und miteinander addiert, um den aktuellen Schwerpunkt des *NAOs* zu erhalten.

Die Position des Schwerpunktes wird nun auf den Boden projiziert. Die Differenz des Mittelpunktes zum Stützpolygon ist dann die Regelgröße.

Um diese Werte zu berechnen, werden vorhandene Schnittstellen im B-Human System verwendet. Im B-Human System existieren verschiedene Koordinatensysteme. Davon ist eines das Torsokoordinatensystem, welches im Torso verankert ist und dort seinen Ursprung besitzt. Die 3D-Koordinaten der Füße können vorberechnet in Torsokoordinaten abgefragt werden. Um damit nun korrekt zu arbeiten, muss das Koordinatensystem parallel zum Boden rotiert werden, damit der Schwerpunkt korrekt auf die Fußebene projiziert wird. Dafür wird eine Sensorfusion der Gyro- und Accelerometer verwendet. Aus dieser berechnet sich die Torsoneigung des *NAOs*, mithilfe dieser dann eine Rotationsmatrix definiert wird. Mit der Rotationsmatrix können dann abschließend die Koordinaten der Füße und des Schwerpunktes rotiert werden, sodass deren Koordinaten parallel zum Boden sind. Aus den Fußkoordinaten lassen sich dann die jeweils kleinsten und größten X- und Y-Koordinaten bestimmen, mit denen ein vereinfachtes Stützpolygon ermittelt wird, sowie dessen Mittelpunkt, wie es in Abbildung 5.14 zu sehen ist.



**Abbildung 5.14** Das Stützpolygon (blau) mit dessen Mittelpunkt (orange) und der Schwerpunkt des NAOs (rot)

### 5.5.2 Das Balancieren

Der entwickelte PID-Regler bekommt als Regelgröße die Differenz vom Schwerpunkt in der Fußebene zum Mittelpunkt des Stützpolygons, welcher im Folgenden als  $CoM_{Foot}$  bezeichnet wird. Zusätzlich bekommt der PID-Regler noch einen Sollwert, welcher am Ende des aktuellen Keyframes erreicht werden soll, sowie den Sollwert des letzten Keyframes. Die Stellgröße, die aus dem PID-Regler resultiert, wird dann auf die Gelenkwinkel des aktuellen Frames auf vordefinierte Gelenke addiert.

Der aktuelle Sollwert ist hierbei eine lineare Interpolation zwischen dem Sollwert des letzten Keyframes und des aktuellen, um somit den Sollwert zu jedem Zeitpunkt zwischen den Keyframes zu erhalten.

Für den PID-Regler gilt nun, dass der P-Anteil die Differenz des aktuellen Sollwertes und dem gemessenen Istwert vom  $CoM_{Foot}$  ist, im Folge dargestellt als  $\mathbf{C}$ . Der I-Anteil ist das Integral der P-Anteile und der D-Anteil ist die Differenz des letzten und aktuellen  $CoM_{Foot}$ ,

im Folge dargestellt als  $\vec{C}$ .

$$\begin{aligned}
 P_{\Delta t} &= C \\
 D_{\Delta t} &= \vec{C} \\
 I_{\Delta t} &= I_{\Delta t-1} + P_{\Delta t} \\
 Value_{\Delta t} &= P_{\Delta t} + D_{\Delta t} + I_{\Delta t}
 \end{aligned}$$

Wie in Abschnitt 5.5 bereits erwähnt, gibt es beim Aufstehen unterschiedliche Phasen, weshalb unterschiedlich stark geregelt werden muss. Jeder Anteil des Reglers bekommt daher einen Parameter, welcher in jedem Keyframe anders sein kann.

$$\begin{aligned}
 P_{\Delta t} &= p \cdot C \\
 D_{\Delta t} &= d \cdot \vec{C} \\
 I_{\Delta t} &= i \cdot (I_{\Delta t-1} + P_{\Delta t})
 \end{aligned}$$

Ebenfalls es ist wichtig, die einzelnen Anteile im Kontext der Zeit zu setzen. Da jede Framedauer, in der Gelenkwinkel gesetzt und Sensordaten ausgelesen werden, im Schnitt bei 10ms liegt (in der Formel als 0.01 repräsentiert), wird der D-Anteil damit gewichtet und die Ausgabe des Reglers damit verringert.

$$D_{\Delta t} = d \cdot \frac{\vec{C}}{0.01}$$

$$Value_{\Delta t} = (P_{\Delta t} + D_{\Delta t} + I_{\Delta t}) \cdot 0.01$$

Beim Aufstehen ist der Bereich, in dem der Roboter nicht umfällt, sehr groß. Damit ist gemeint, dass der *NAO* sehr weit nach vorne und hinten geneigt sein kann, ohne dabei umzufallen. In einer stehenden Pose ist dieser Bereich deutlich geringer, da der Schwerpunkt sich weiter nach oben verschiebt, je aufrechter und mit den Beinen ausgestreckter der *NAO* steht. Es soll deshalb in Abschnitt 5.5.3 untersucht werden, ob es sinnvoll ist, die einzelnen Parameter zu verstärken, je mehr der  $CoM_{Ist}$  vom  $CoM_{Soll}$  abweicht. Die Verstärkung soll hierbei für alle Anteile einzeln eingestellt werden. Da ein Regler eigentlich dazu dient, eine Regelgröße trotz Störungen einzuhalten, wird die Verstärkung als Funktion definiert, damit bei geringen Abweichungen fast keine und nur bei starken Abweichungen eine Verstärkung existiert. Ansonsten würde die Verstärkung als Störung agieren und der Regler könnte seine Aufgabe nicht erfüllen.

Für die Verstärkung werden mehrere Parameter definiert. Es gibt eine maximale Verstärkung für jeweils eine positive und negative  $CoM_{Soll}$  Abweichung,  $\Psi^+$  und  $\Psi^-$ . Es soll auch einen Abweichungswert von  $CoM_{Ist}$  zu  $CoM_{Soll}$  geben, bei dem die Verstärkungsfunktion 1 ausrechnet und somit den jeweiligen PID-Anteil verdoppeln würde,  $\Phi^+$  und  $\Phi^-$ .

$$Faktor = \min\left(\frac{1}{\Phi} \cdot C, \Psi\right)$$

Da, wie bereits erwähnt, eine Verstärkung unerwünscht ist, sofern  $C < \Psi$ , wird der Bruch  $\frac{1}{\Phi}$  durch eine  $f(x) = x^m$  Funktion ersetzt. Der Grad der Funktion wird dabei ebenfalls von den Keyframes vorgegeben. Dabei wird dieser wieder jeweils für eine positive und negative CoM-Abweichung eingestellt, mit den Parametern  $\Theta^+$  und  $\Theta^-$ .

$$\Gamma = \begin{cases} \min\left(\frac{1}{\Phi^{+\Theta^+}} \cdot C^{\Theta^+}, \Psi^+\right), & \text{wenn } C > 0 \\ \min\left(\frac{1}{\Phi^{-\Theta^-}} \cdot C^{\Theta^-}, \Psi^-\right), & \text{sonst } C \leq 0 \end{cases}$$

Diese Verstärkung kann für alle drei PID-Regler-Anteile einzeln eingestellt werden. Wir erhalten also:

$$\begin{aligned}
 P &= (p + \Gamma_p) \cdot C \\
 D &= (d + \Gamma_d) \cdot \vec{C} \cdot 0.01 \\
 I &= (i + \Gamma_i) \cdot (I_{\Delta t-1} + P_{\Delta t})
 \end{aligned}$$

Dadurch gibt es kaum eine Verstärkung, wenn die Abweichung gering ist, aber dafür eine hohe, wenn die Abweichung groß ist. Da eine große Abweichung bedeutet, dass der *NAO* potenziell umfallen würde, wird dies durch die Verstärkung verhindert. Ebenfalls wird ein Aufpendeln verhindert, da sich die Verstärkung schnell abbaut. Dazu kommt, dass dadurch, dass sich der *NAO* in einer dauerhaften, nicht pendelnden Bewegung befindet, sich ein mögliches Aufpendeln erst am Ende der Aufstehbewegung zeigt. Damit ist die Bewegung aus Abbildung 5.13 gemeint. Dieses mögliche Aufpendeln soll durch den PID-Regler verhindert werden, in dem in dieser Bewegungsphase am Ende die Parameter genau darauf eingestellt sind. Um dies zu erreichen, werden die Parameter wie in Abschnitt 5.5.3 erklärt, ermittelt.

In Kapitel 4 wurde bereits erwähnt, dass sich die Gelenke nicht jederzeit bewegen können. Deshalb werden nicht in jedem Keyframe, in dem der PID-Regler aktiv ist, die PID-Regler Ausgabe auf die gleichen Gelenke addiert, sondern es wird für jeden Keyframe eine Liste von Gelenken definiert, welche verwendet werden sollen.

Zusätzlich wird jedem Gelenk noch ein weiterer Faktor definiert, sodass es möglich ist, zum Beispiel auf die Fußgelenke den vollen Balancierwert zu addieren und auf die Hüftgelenke nur den halben. Diese Parameter sollen ebenfalls in Abschnitt 5.5.3 ermittelt werden, um zu untersuchen, ob eine Balancierung mit Gelenken, die näher am Körperschwerpunkt sind, sinnvoll ist und ob eine zu starke Balancierung den *NAO* mehr ins Schwingen bringt als es zu verhindern.

### 5.5.3 Einstellen der Parameter

Um gute Parameter für den PID-Regler zu ermitteln, wird ein ähnlicher Versuchsaufbau verwendet, wie in der Arbeit von Müller [2010] für ein Experiment beim Schießen unter Einfluss eines Stoßes. Es wurde an einem Tor eine Wasserflasche aufgehängt und unter einem Winkel fallen gelassen, sodass diese den *NAO* beim Schießen auf Höhe des Torsos trifft. Das Gewicht von 350g in der Arbeit von Müller [2010] wurde damals so gewählt, dass der *NAO* mit Balancierung in 10 von 10 Fällen nicht umfiel.

Da in dieser Arbeit die Folgeversion V5 der *NAOs* verwendet wird, wird ein Gewicht von 520g benutzt, da sich das Gewicht der *NAOs* erhöht hat. Das Gewicht wurde in der Arbeit von Böckmann [2015] verwendet, unter Umsetzung desselben Versuchsaufbaus. In Abbildung 5.15 ist dieser zu sehen.

Um nun passende Parameter zu finden, wird der *NAO* in die Körperpose gebracht wie sie

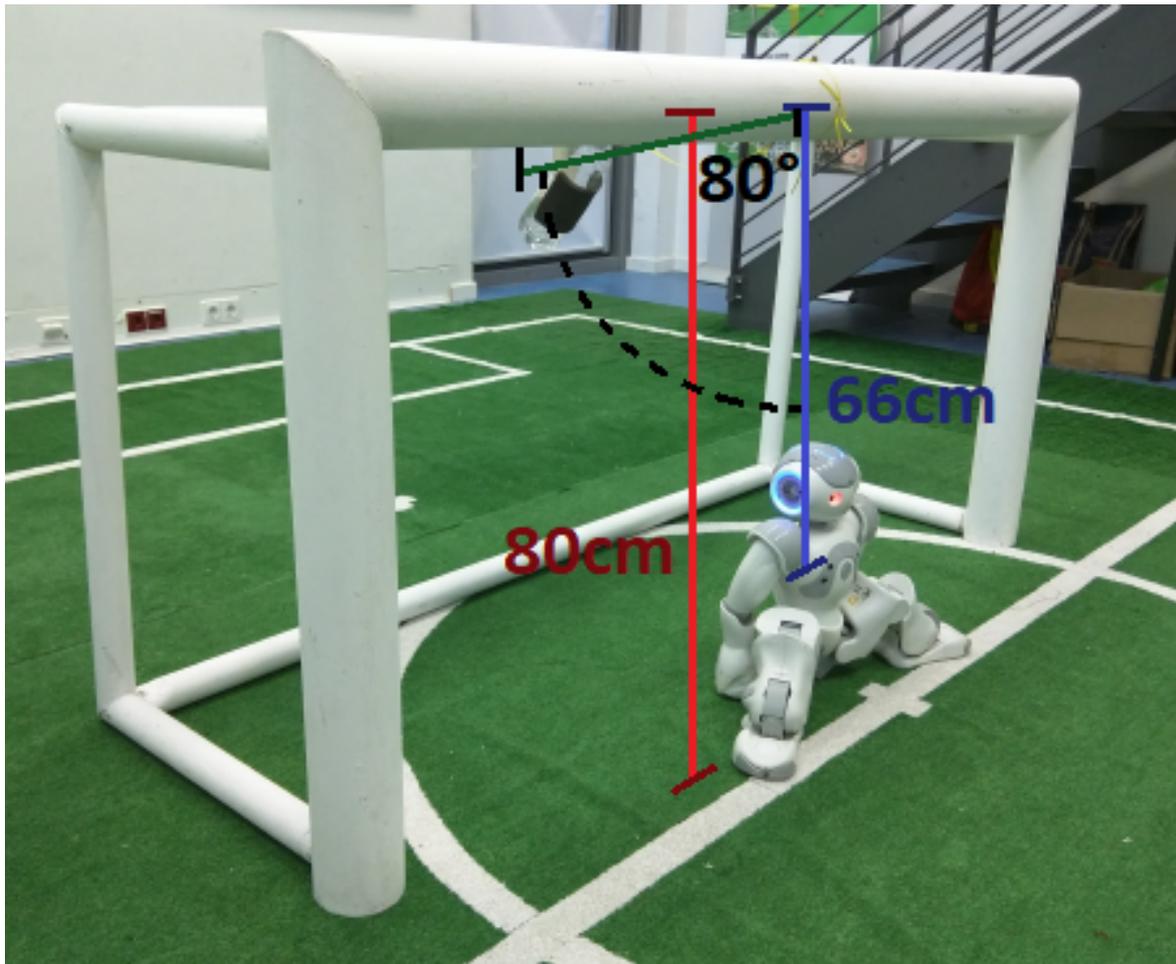


Abbildung 5.15 Versuchsaufbau zum Ermitteln der PID-Regler Parameter

in Abbildung 5.15 zu sehen ist. Diese Körperpose wird im aktuellen B-Human System verwendet, um Bälle in der eigenen Feldhälfte anzunehmen. Da aus dieser Pose auch im echten Spiel aufgestanden werden muss, werden die Parameter unter realistischen Bedingungen eingestellt. Aus dieser Stellung führt der *NAO* unter anderem die Bewegung *FromSplitt* aus Abbildung 5.12, welche in dieser Arbeit modifiziert wurde.

Zunächst werden die Parameter, um sie einzustellen, auf 0 gestellt. Da Soll-Werte benötigt werden, wird zu erst mehrmals ein *NAO*, welcher vom Hardwarezustand gut ist, ohne Einflüsse unter Idealbedingungen normal aufstehen gelassen. Danach werden die durchschnittlichen  $CoM_{Foot}$ -Werte am Ende der Keyframes berechnet und als Soll-Werte verwendet. Die Soll-Werte werden hier aber bereits leicht angepasst, sodass in der Phase, in der der *NAO* sich nach vorne bewegt, der Soll-Wert leicht verringert wird und in der Phase, in der er sich nach hinten bewegt, leicht vergrößert. Dadurch wird der gewünschte  $CoM_{Soll}$  leicht nach vorne beziehungsweise nach hinten geschoben, um gezielt und besser das Momentum durch den PID-Regler ab- beziehungsweise aufzubauen.

Für das Ermitteln der Parameter wird nun der *NAO* aus der breitbeinigen Pose aufstehen

gelassen. Während des Aufstehversuches wird die Wasserflasche in einem 80 Grad Winkel auf den *NAO* fallen gelassen. Nach dem Versuch werden die berechneten  $CoM_{Ist}$  betrachtet. Wird gegen den Stoß vom D-Anteil nicht sofort stark genug gegengeregelt, so wird dieser leicht erhöht. Ist die Differenz  $CoM_{Soll}$  und  $CoM_{Ist}$  kurz nach dem Stoß zu groß, so wird der P-Anteil erhöht. Haben beide Anteile keinen Erfolg, so wird der I-Anteil angepasst. Musste der P-Anteil oder D-Anteil zu stark erhöht werden, so werden die Verstärkungen erhöht.

### 5.5.3.1 Ergebnisse vom Einstellen der Parameter

Es wurden Parameter gefunden, mit denen mehrere *NAOs* beim Aufstehen mit Stößen auf den Rücken nicht umfallen und sich nicht aufgrund des PID-Reglers aufschaukeln.

Bei Stößen auf den Bauch konnten keine Parameter gefunden werden, mit denen die *NAOs* nicht umfielen und sich auch nicht aufschaukelten. Dies liegt an der Anatomie der Füße. Dadurch, dass die Fußflächen nach vorne wie beim Menschen verlängert sind, sind Stöße von hinten eher abfangbar als Stöße von vorne. Um dennoch Parameter zu ermitteln, wurde die Wasserflasche aus verschiedenen Winkeln auf den *NAO* fallen gelassen, um kleinere Stöße zu simulieren.

Da in einem echten Spiel Fremdeinwirkungen wie Stöße nur von anderen *NAOs* kommen können und diese, aufgrund der Füße, nicht in der Lage sind, gegen den Torso eines aufstehenden *NAOs* von vorne zu laufen, werden die ermittelten Parameter als ausreichend bewertet.

Ebenfalls wurden hier nur Parameter für die Bewegungen entwickelt, welche die Bewegung *FromSplitt* (siehe Abbildung 5.12) verwenden. Für die Vorderbewegung *FrontFast* (siehe Abbildung 5.1 bis Abbildung 5.3) werden die Parameter mit dem Wissen aus dem Stoßexperiment eingestellt. Dadurch, dass die Bewegung *FrontFast* aus einer Schwungphase (siehe Abbildung 5.3) besteht, gefolgt von einer Ruhephase (siehe Abbildung 5.13), wurden die Parameter von den Keyframes der Bewegung *FromSplitt* verwendet, in denen sich der *NAO* auf seine Beine bewegt (für die Schwungphase) sowie sein Gleichgewicht hält (für die Ruhephase). Danach wurden mehrere *NAOs* aufstehen gelassen und mit den berechneten  $CoM_{Ist}$  wurde entschieden, welche Parameter noch angepasst werden mussten. Die Parameter für die Bewegung *FrontFast* sind somit mehr nach Gefühl und zum Teil durch Experimente ermittelt worden.

Ebenfalls wurde herausgefunden, dass eine Balancierung mit den Fußflächen sinnvoll ist, sofern der *NAO* nicht zu weit nach vorne geneigt ist und somit die Fußflächen relativ planar zum Boden liegen. Falls dies nicht gewährleistet werden kann, ist es sinnvoll, mit verringerten Werten mit den *HipPitch*-Gelenken zu balancieren, sofern sich diese im betroffenen Keyframe bewegen können. Es wird daher bei der Bewegung *FromSplitt* nur mit den Füßen balanciert, bis der *NAO* den Keyframe erreicht hat, in dem er sein *RHipPitch*-Gelenk bewegen kann. In diesem Keyframe wird die Stellgröße des PID-Reglers vollständig auf die Füße und die Hüfte aufgeschlagen. Sobald der *NAO* ins Sitzen und Stehen übergeht, wird nur auf die Füße die

volle Stellgröße addiert und die Halbe auf die Hüfte.

Aus Gesprächen mit dem RoboCup Team Nao-Team HTWK [Nao-Team HTWK, 2018] konnte ich in Erfahrung bringen, dass deren *NAOs* in der Aufstehphase vom Sitzen ins Stehen auch mit den Kniegelenken balancieren. Die Stellgröße wird daher mit halben Wert auf die Kniegelenke addiert. Dadurch konnte beobachtet werden, dass die *NAOs* in dieser Phase sehr viel gerader und deutlich weniger schwankend aufstehen.

Bei starken Schwungphasen, wie sie bei *FrontFast* vorkommen, konnte beobachtet werden, dass eine volle Balancierung mit der Hüfte und den Füßen sinnvoll ist, um das Momentum des *NAOs* zu kontrollieren.

Ebenfalls ist aufgefallen, dass der I-Anteil des PID-Reglers mit hohen Parametern sehr oft die *NAOs* zum Schwanken und Umfallen gebracht hat. Deshalb wurde dieser universell für alle Keyframes, in denen balanciert wird, mit einem festen Parameter in Höhe von 0.02 ersetzt.

## 5.6 Gelenkkompensierung

Aus Abschnitt 4.1 ist bereits bekannt, dass es vorkommen kann, dass Gelenke nicht ihre Soll-Winkel erreichen. Dieser Umstand soll mit anderen Gelenken ausgeglichen werden. Zwar wurden die Bewegungen im Abschnitt 5.1.3 dahin gehend verändert, dass genau dieses Problem nicht mehr auftreten sollte, dennoch reicht das nicht für eine 100%ige Prevention aus. Denn bereits leichte Unebenheiten des Feldes haben einen großen Einfluss auf das Aufstehen und es können einzelne Gelenke zu jedem Zeitpunkt zusätzlich unterschiedlich stark belastet sein, was selbst die noch so beste Aufstehbewegung nicht ausgleichen kann, da diese nur statisch ist.

Da die verwendeten Keyframes mit expliziten Gelenkwinkeln arbeiten, wird das Framework erweitert, sodass jedem Keyframe eine Liste zugeordnet werden kann, welche definiert, welches Gelenk wie mit welchen Gelenken ausgeglichen werden soll.

Es kann nicht vorausgesetzt werden, dass sich die Gelenke perfekt linear verhalten. Insbesondere die Gelenke, die nicht ihre Soll-Werte erreichen, bewegen sich nicht zwingend in die richtige Richtung. Es muss daher verhindert werden, dass durch den Gelenkausgleich die Gelenke, aufgrund dieses Verhaltens, anfangen zu zittern, um ein verstärktes Abnutzen der Gelenke zu vermeiden. Dafür wird mit angegeben, um wie viel Grad das betroffene Gelenk vom Soll-Wert abweichen muss, bevor ein Ausgleich stattfindet. Ebenfalls wird ein Maximum angegeben, um Schäden vorzubeugen. Dies hat den Grund, dass bei der Gelenkkompensierung von der geplanten Bewegung zu stark abgewichen werden könnte, was zu unerwarteten Posen führen kann, welche den *NAO* mehr schaden könnten als ein gescheiterter Versuch. Ebenfalls soll nicht für jeden Frame, in dem Gelenkwinkel gesetzt werden, der Ausgleich stattfinden, sondern es sollen die Gelenkziele des aktuellen Keyframe verändert werden.

Durch die Maxima, wie stark ausgeglichen werden darf, und der Anpassung der Zielwinkel statt der aktuellen Winkel, sollen Überanpassungen verhindert werden. So soll verhindert werden, dass der Balancer und die Kompensierung die inkorrekten Zustände ausgleichen wollen und der *NAO* deshalb nach hinten überkippt.

Es wird ebenfalls davon ausgegangen, dass mit jedem Grad, mit dem ein Gelenk vom Soll-Wert abweicht, diese Differenz mit einem Faktor auf andere Gelenke addiert werden kann, um dieselbe Torsoneigung zu erhalten. Es existiert also ein linearer Zusammenhang, welcher experimentell geprüft wurde. Dafür habe ich einen *NAO* in verschiedene Keyframe geknetet und die Torsoneigung aufgenommen. Danach wurde das *HipYawPitch*-Gelenk um 5, 10, 15 und 20 Grad verstellt und mit den restlichen Beingelenken so ausgeglichen, dass dieselbe Torsoneigung erreicht wurde. Die Verhältnisse waren identisch.

Um nun ein Gelenk auszugleichen, werden den Keyframes, in denen bekannt ist, dass ein Gelenk stark abweicht, eine Liste von Gelenken zugeordnet mit jeweils eigenen Faktoren, auf denen dann die Abweichung vom Soll- und Ist-Winkel aufaddiert werden.

Dieses einfache Verfahren wird im Kapitel 6 evaluiert, ob damit mehr Schaden entsteht oder die Aufstehversuche verbessert werden.

Wie in Kapitel 4 dargelegt, existiert eine Verzögerung von 30ms bis 40ms, bis Gelenkwinkel umgesetzt werden. Es werden deshalb die letzten drei Gelenkwinkelanfragen zwischengespeichert, sodass die aktuell erreichten Gelenkwinkel mit denen von vor 3 Frames verglichen werden können.

## 5.7 Programmablauf

Der Programmablauf ist nun zum Teil wie in den folgenden UML-Diagrammen vereinfacht dargestellt. Zu sehen ist der Ablauf, um am Anfang zu entscheiden, wie im aktuellen Zustand vorgegangen werden soll (siehe Abbildung 5.16). Zusätzlich ist auch der Ablauf zu sehen, wann welche Bewegung ausgeführt wird (siehe Abbildung 5.17).

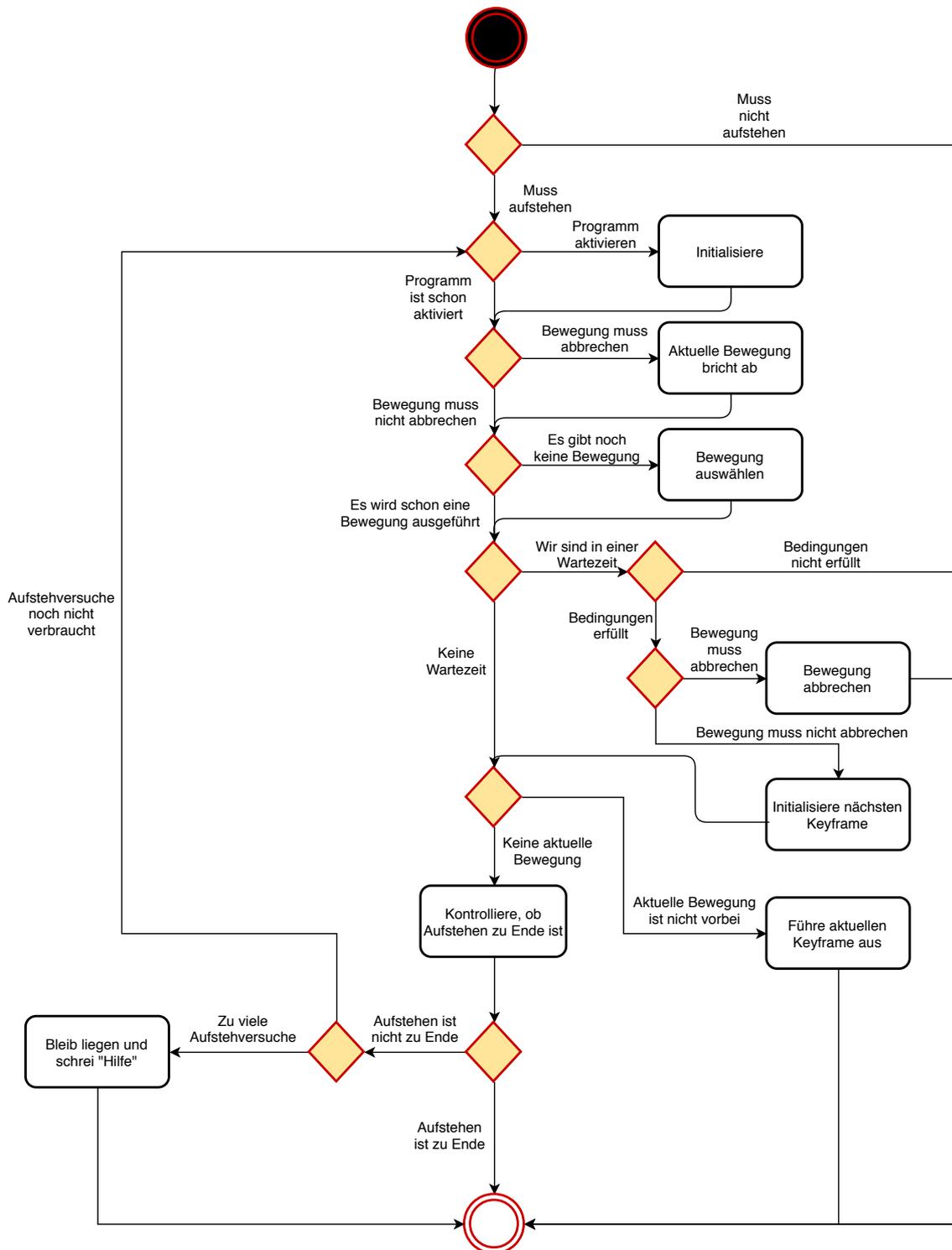


Abbildung 5.16 Vereinfachter Programmablauf zum Entscheiden, wie im aktuellen Zustand vorgegangen werden soll

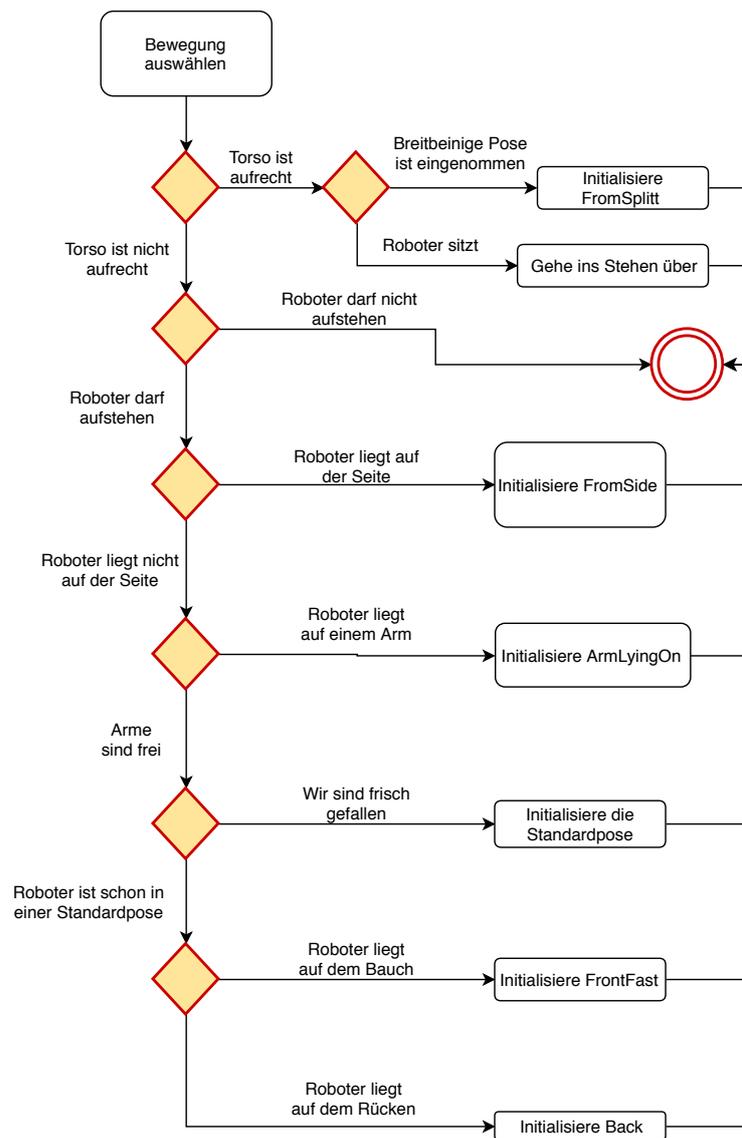


Abbildung 5.17 Vereinfachter Ablauf, wann welche Bewegung ausgeführt wird



## Kapitel 6

# Evaluation

### 6.1 Ziele

In dieser Evaluation wird das entwickelte Aufstehverfahren unter verschiedenen Bedingungen untersucht. Da das Aufstehverfahren aus dem Jahr 2017 jeweils drei verschiedene Aufstehbewegungen für vorne und hinten beinhaltete, hätten die besten funktionierenden Aufstehbewegungen für die einzelnen *NAOs* experimentell ermittelt werden müssen. Dafür würde jedem *NAO* das für ihn damals verwendete Aufstehen zugeordnet und ausprobiert werden. Bei denen, wo das Aufstehen entweder nicht funktionieren würde oder sehr wackelig wäre, würden die anderen Aufstehbewegungen ausprobiert und das jeweils beste ausgesucht werden. Dies würde zu einer unnötigen Belastung und Abnutzung für die *NAOs* führen, weshalb für die Evaluation entschieden wurde, nur die Auswertung aus den Videoaufnahmen aus dem RoboCup 2017 zu nehmen.

Ebenfalls möchte ich erwähnen, dass die Aufstehbewegung *Front* nicht evaluiert wurde. Dies hat den Grund, dass die Versuche erst mit den Bewegungen *FrontFast* und *Back* durchgeführt wurden und aufgrund der Ergebnisse bestand kein Bedarf mehr, die Ergebnisse der Bewegung *Front* zu ermitteln, um die *NAOs* nicht weiter abzunutzen.

Im ersten Teil wurde das Aufstehen ohne die Gelenkkompensierung aus Abschnitt 5.6 durchgeführt, um herauszufinden, welche Gelenke diese Kompensierung benötigen und ob die Kompensierung mehr Nachteile bringt.

Im zweiten Teil wurde das Aufstehen unter realistischen Bedingungen getestet. Dafür wurde das Aufstehen nach dem ersten Teil der Evaluation angepasst, um unter Extrembedingungen ebenfalls zu funktionieren. Es wurde die Gelenkkompensierung insofern angepasst, dass aus den Daten der ersten Evaluation entschieden wurde, das HipYawPitch-Gelenk mit den restlichen Beingelenken auszugleichen, hauptsächlich mit den HipPitch-Gelenken. Die anderen Bereiche, um das Aufstehen zu verbessern, blieben unverändert.

Die realistischen Bedingungen waren dabei die Veranstaltung Deep Dive BBM Strategy Meeting von Bosch in Stuttgart. Auf dieser Veranstaltung haben ein anderer Student aus dem Projekt B-Human und ich für zwei Tage mit zwei *NAOs* mehrere Demos vorgeführt. Hier sind die *NAOs* von selber sehr oft umgefallen und mussten wieder aufstehen, was für die Evaluation ausgewertet wurde.

## 6.2 Versuchsaufbau Balancierung

Es wurden drei Versuche durchgeführt, um die Stabilität der Aufstehbewegungen sowie der Balancierung und der inkorrekten Bewegungszustandserkennung zu evaluieren.

Im ersten Versuch wurden die *NAOs* ohne Einflüsse unter normalen Bedingungen auf dem Spielfeld aufstehen gelassen. Dafür kam das bei uns im Projektraum liegende Feld, ein neues, nicht plattgetretenes Feld sowie ein Feldstück, welches für Testspiele verwendet wird, zum Einsatz. Vier *NAOs* mussten auf jedem Feldstück aufstehen, jeweils dreimal auf dem Projektraumfeld auf verschiedenen Stellen und jeweils einmal auf den anderen beiden Feldstücken. Die Versuchsreihe wurde einmal mit Balancierung und optionalen Keyframes sowie einmal ohne Balancierung und ohne optionalen Keyframes durchgeführt. Die Gelenkkompensierung wurde hierbei nicht verwendet.

In Tabelle 6.1 sind die Ergebnisse zum ersten Versuch zu sehen. In den Versuchen mit Balancierung und Fehlererkennung haben alle Aufstehversuche auf den vier *NAOs* funktioniert. Es ist lediglich anzumerken, dass nur auf einem *NAO* die Vorderbewegung *FrontFast* in zwei Versuchen knapp war und dieser fast umgefallen ist. Dies hat den Grund, dass das HipYawPitch-Gelenk kurzzeitig bis zu 10 Grad hinterher hing.

In Abbildung 6.1 sind die Verläufe der beiden Bewegungen in beiden Fällen dargestellt. So ist zu sehen, dass bei der Bewegung *FrontFast* ohne aktivierte Verfahren die Bewegung mehr Momentum hat als mit. Dies ist daran zu erkennen, dass die *NAOs* ohne aktivierte Verfahren bei dem x-Achsenwert 125 im Schnitt eine Torsoneigung von knapp 31 Grad eingenommen haben und mit aktiviertem Verfahren eine mit 37 Grad. Ebenfalls sorgt der PID-Regler, dass am Ende der Bewegung der *NAO* mehr nach vorne geneigt bleibt (Torsoneigung bleibt zwischen 8 und -1 Grad) statt, aufgrund des Momentums, nach hinten über zu schwanken (Torsoneigung sinkt bis auf ca. -7 Grad).

Bei der Bewegung *FromSplitt* sorgt der PID-Regler dafür, dass der *NAO* mehr nach vorne geneigt bleibt. Folge dessen ist, dass die Reglergröße  $CoM_{Foot}$  stark nach oben ausreißt.

Insgesamt ist zu beobachten, dass der PID-Regler zwar dafür sorgt, dass wie gewünscht eine Schwerpunktverschiebung entsteht, der *NAO* nicht zu weit nach hinten geneigt ist und seine Fußfläche effektiver ausnutzt, dafür ist aber der Verlauf ungleichmäßiger.

Im zweiten Versuch wurde der Aufbau aus Abschnitt 5.5.3 verwendet und durchgeführt. Hier-

Fall	FrontFast	Back
Mit Austehverfahren	20/20	20/20
Ohne Aufstehverfahren	11/20	13/20

**Tabelle 6.1** Erfolgsraten des Aufstehens Versuch 1

bei wurde auf dem Projektraumfeld an jeweils vier *NAOs* der Stoßtest zweimal durchgeführt. Die *NAOs* sind dabei in allen acht Fällen nicht umgefallen und ohne weitere Schwankungen aufgestanden. Es ist lediglich anzumerken, dass durch den Aufprall die Füße zuckten, da der PID-Regler kurzzeitig stark reagiert hatte.

Im dritten Versuch wurde unter dem Feld eine Holzplatte gelegt und diese um 0.7 Grad sowie 2.1 Grad angewinkelt. Der Versuchsaufbau ist in der Abbildung 6.2 zu sehen. Diese Konstellation sollte auf einem Wettkampffeld eigentlich nicht vorkommen, dennoch können Felder oder einzelne Feldbereiche von der Norm abweichen. Die Aufstehversuche wurden nur mit dem neuen Feldstück durchgeführt. Es wurden drei *NAOs* jeweils zweimal bergab sowie bergauf liegend aufstehen gelassen.

In der Tabelle 6.2 sind die Ergebnisse bei einer Bodenneigung von 0.7 Grad zu sehen. Es ist zu erkennen, dass die Vorderbewegung weiterhin nicht in der Lage ist, bei einer Gewichtsverlagerung nach vorne zu funktionieren. Die *NAOs* waren zwar in der Lage, sich korrekt auf ihre Arme abzustützen, das HipYawPitch-Gelenk blieb aber zu weit offen und konnte sich nicht schließen. Daraufhin konnten die *NAOs* keine aufrechte Pose erreichen und fielen nach vorne um.

Fall	FrontFast	Back
Bergab liegend	1/6	6/6
Bergauf liegend	6/6	6/6

**Tabelle 6.2** Erfolgsraten des Aufstehens bei einer Bodenneigung von 0.7 Grad



**Abbildung 6.2** Die geneigte Holzplatte mit 2.1 Grad Neigung

Für den Versuch bei einer Bodenneigung von 2.1 Grad wurde nur die Rückenbewegung *Back*

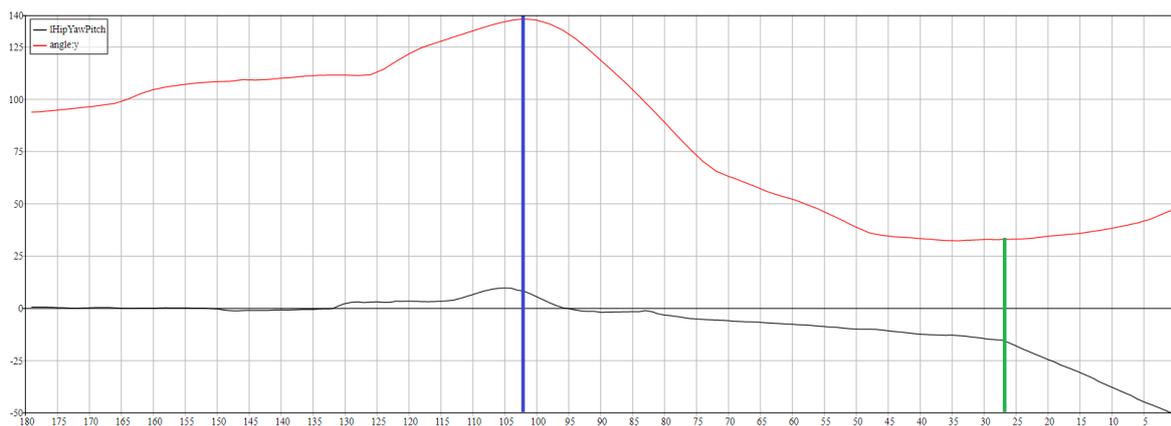
evaluiert, weil die Vorderbewegung *FrontFast* schon bei 0.7 Grad nicht gut funktioniert hat, die Rückenbewegung *Back* aber schon. Ebenfalls habe ich für die Bewegung *FrontFast* untersucht, ob die Bewegung funktionieren könnte, wenn sich der NAO beim Fallen auf die Arme stützt und den Aufstehversuch von da an weiterführt. Dieses Back up hat zwar auf dem Boden mit einer 0.7 Grad Neigung funktioniert, aber weder auf einem ebenen Boden noch auf einem 2.1 Grad geneigten Boden.

Dieser Versuch wurde auf vier *NAOs* jeweils bergab und bergauf liegend durchgeführt. In der Tabelle 6.3 sind die Ergebnisse zu sehen. Es ist nur einer von acht Versuchen bergab liegend gescheitert. In diesem Versuch hatte der *NAO* zu viel Gewicht nach vorne, sodass sich das HipYawPitch-Gelenk nicht bewegen konnte. Bei den bergauf liegenden Versuchen war das Problem, dass die Bewegung von Grund auf eine sehr aufrechte Bewegung ist, sodass eine starke Neigung nach hinten nicht ausgleichbar ist, da die *NAOs* ihren Rücken nicht wie Menschen nach vorne krümmen können und das Heranziehen der Beine nicht ausreichte, um ein Fallen nach hinten zu vermeiden.

Fall	Ergebnis
Bergab liegend	7/8
Bergauf liegend	3/8

**Tabelle 6.3** Erfolgsraten des Aufstehens bei einer Bodenneigung von 2.1 Grad

Bei mehreren *FrontFast* Bewegungen, die gescheitert sind, konnte beobachtet werden, dass sich das HipYawPitch-Gelenk nicht bewegte (siehe Abbildung 6.3).



**Abbildung 6.3** Ablauf der Bewegung *FrontFast*.

In rot die Torsoneigung (Grad), in schwarz die Abweichung Soll-Ist des HipYawPitch-Gelenkes (Grad).

Blau markiert den Keyframe, in dem der *NAO* sich auf die Arme stützt (siehe Abbildung 5.3).

Grün markiert den Keyframe, in dem der *NAO* anfängt in das Sitzen über zu gehen (siehe Abbildung 5.13).

Die X-Achse beschreibt die Anzahl der vergangenen Frames.

### 6.3 Realistischer Test

Das Aufstehen hatte eine Erfolgsrate von 91.4%, dabei haben 74 von 84 Versuchen von *FrontFast* und 78 von 82 Versuchen von *Back* funktioniert. Es ist anzumerken, dass die *NAOs* bei der Demovorführung nur sehr wenige Pausen hatten, sodass wichtige Gelenke, wie die AnklePitch-Gelenke oder die Knie, die meiste Zeit auf über 100 Grad Celsius geschätzt wurden. Über 40% der Spielzeit betrug die Temperatur der genannten Gelenke über 70°C. Dadurch bedingt gaben die Gelenke nach und die *NAOs* konnten weder gut laufen noch schießen, ohne dabei umzufallen. Trotz dieses Problems sind nur 10 Aufstehversuche gescheitert. Das Problem der heiß werdenen Gelenke aus Kapitel 4 scheint somit beim Aufstehen nicht aufzutreten.

Drei Versuche sind gescheitert, weil die Knie oft nicht in der Lage sind, sich zu bewegen, wenn sie zu weich sind, was bei Temperaturen von über 70 Grad Celcius der Fall ist. Dieses Problem trat in der letzten Phase der Aufstehbewegungen auf, als die *NAOs* in einer sitzenden Position ankamen und sich aufrichten wollten. In den 3 Fällen, wo das Aufrichten nicht funktionierte, hatten die *NAOs* ein Momentum nach hinten und waren bereits um über 5 Grad nach hinten geneigt. In Verbindung mit den weichen Knien führte das zu den gescheiterten Versuchen.

In zwei Versuchen schafften es die *NAOs* bei der Vorderbewegung nicht, am Anfang ihre Arme richtig nach vorne zu positionieren. Dadurch wurde ein Arm eingeknickt und der Schwerpunkt war zu weit vorne. Folglich konnten sich die *NAOs* nicht mehr aufrichten und sie fielen nach vorne um.

Vier Versuche sind bei der Rückenbewegung gescheitert, weil der Torwart nach seinem Sprung auf dem Rücken landete und noch einen Arm ausgestreckt hatte. Da die Bewegung zu schnell abläuft, war der Arm nicht in der Lage, sich schnell genug hinter den Rücken zu bewegen und blieb vorne am Torso hängen. Der beobachtete Fehlerfall wurde für den 2. Tag der Demo erst korrigiert, da dieser vorher nicht erkannt worden war. Da die ShoulderPitch-Gelenke in diesem inkorrekten Bewegungszustand um mehr als 40 Grad falsch lagen, kann dieser Unterschied gemessen werden, um das entsprechende Back up (siehe Abbildung 5.11), welches in Abschnitt 5.1.3 vorgestellt wurde, anzuwenden. Am zweiten Demo Tag trat dieser Fehlerfall noch zwei weitere Male auf und das BackUp wurde erfolgreich ausgeführt.

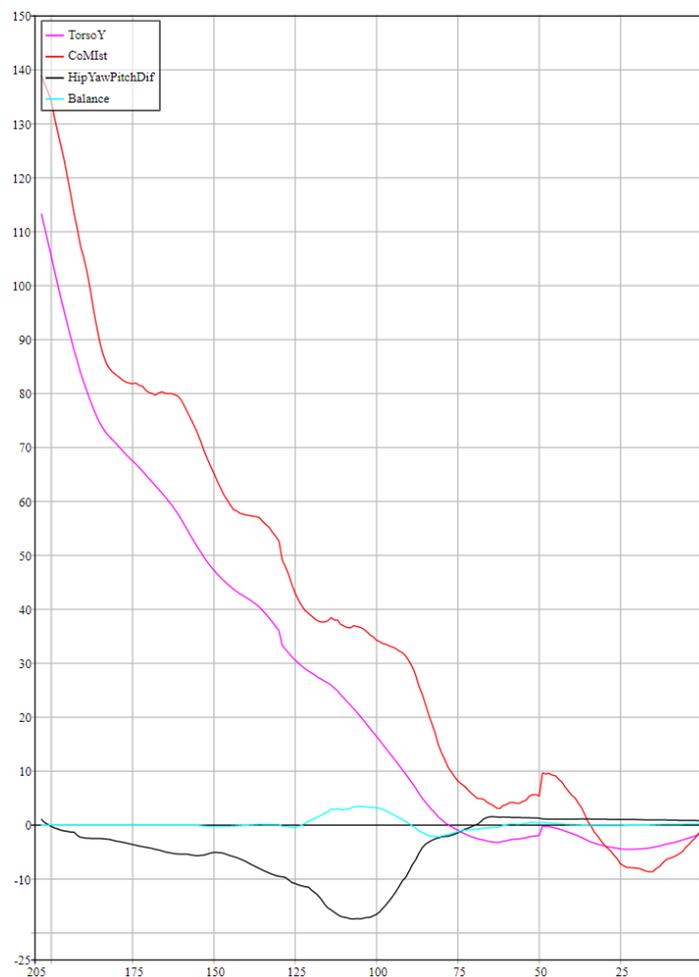
Ein Versuch scheiterte, als der *NAO* nach dem Aufstehen stand, aber noch zu viel Restmomentum hatte und nach hinten fiel.

In drei Versuchen lag der Torwart nach seinem Sprung auf der Seite und die Ausführung der Bewegung, um ihn von der Seite herunter zu bekommen, schlug fehl, sodass der *NAO* weiterhin auf der Seite liegen blieb. In allen 3 Fällen führte er danach die *FrontFast* Bewegung aus. Diese schlug fehl und erst dann blieb der *NAO* korrekt auf einer Seite liegen.

Letztlich kam es einmal vor, dass der *NAO* das Aufstehen erfolgreich schaffte aber das Aufstehen mit Restschwankung beendete. Dadurch kam es zu einer Fehlkennung eines Fallens des *NAOs*, wodurch das Aufstehmodul wieder aktiviert wurde. Aufgrund der aufrechten Tor-

soneigung setzte sich der *NAO* langsam hin und ging langsam wieder in das Stehen über, um dann das Aufstehen erneut zu beenden. Dies ist ein Fehler in der Fallerkennung und nicht im Aufstehen.

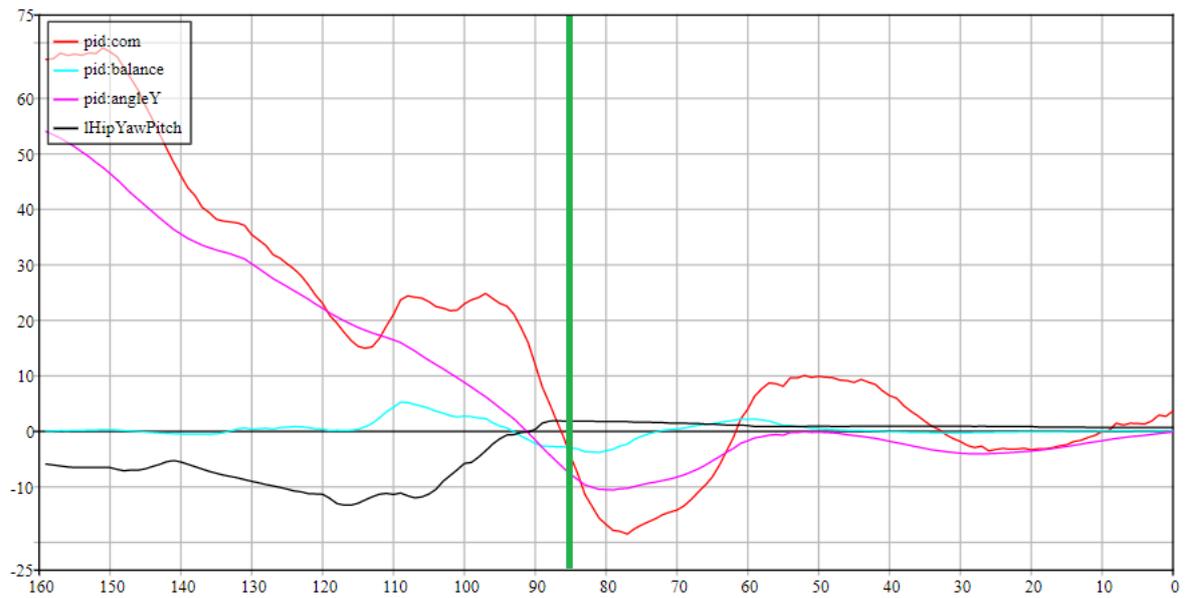
Die Gelenkkompensierung vom *HipYawPitch*-Gelenk wurde neunmal benötigt und hat in allen neun Fällen die Aufstehbewegung gerettet. In Abbildung 6.4 ist einer dieser Fälle veranschaulicht. In einem der neun Fälle kippte der *NAO* durch den Ausgleich fast nach hinten über (siehe Abbildung 6.5), ohne diesen Ausgleich hätte er es aber gar nicht geschafft und wäre nach vorne umgefallen. Grund dafür könnte ein Programmierfehler des Ausgleichens sein, bei dem die Berechnung der Ausgleichswerte für die *HipRoll*- und *AnkleRoll*-Gelenke jeweils ein falsches Vorzeichen hatte. Dieser Fehler trat in drei der neun Fälle auf und nur dieser eine führte zu einem zu starken Ausgleich.



**Abbildung 6.4** Ablauf der Bewegung *FrontFast* mit starker Abweichung von Soll-Ist des *HipYawPitch*-Gelenkes.

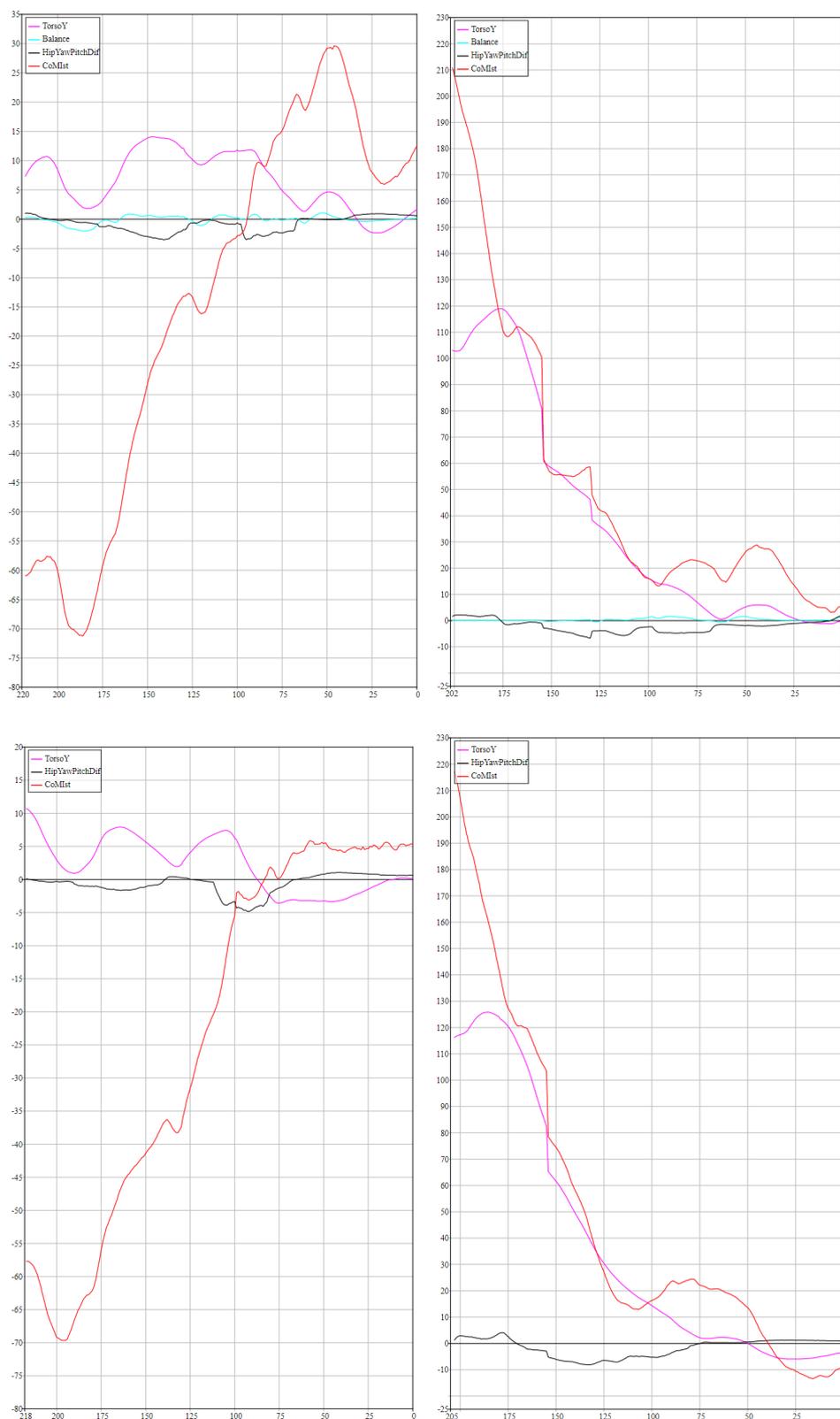
In magenta die Torsoneigung (Grad), in schwarz die Abweichung Soll-Ist des *HipYawPitch*-Gelenkes (Grad), in cyan die Stellgröße und in rot der  $CoM_{Foot}$  (mm).

Die X-Achse beschreibt die Anzahl der vergangenen Frames.



**Abbildung 6.5** Ähnliche wie in Abbildung 6.4. Hier hat der Ausgleich des HipYawPitch-Gelenkes dazu geführt, dass der *NAO* das Aufstehen zwar geschafft hat, aber als er im Sitzen ankam, eine Torsioneigung von -10 Grad erreichte und somit kurz vorm Unfällen war.

Die X-Achse beschreibt die Anzahl der vergangenen Frames.



**Abbildung 6.1** Verlauf der Bewegungen *Back* (oben links) und *FrontFast* (oben rechts) mit aktivierten Verfahren. Verlauf der Bewegungen *Back* (unten links) und *FrontFast* (unten rechts) ohne die zusätzlichen Verfahren.

In magenta die Torsoneigung (Grad), in rot der  $CoM_{Foot}$  (mm), in schwarz die Abweichung von Soll-Ist für das HipYawPitch-Gelenk (Grad) und in cyan die Stellgröße vom PID-Regler.

Die X-Achse beschreibt die Anzahl der vergangenen Frames.

## Kapitel 7

# Zusammenfassung und Ausblick

### 7.1 Zusammenfassung

In dieser Arbeit wurde ein dynamisches Aufstehen unter Vermeidung inkorrektter Bewegungszustände entwickelt und evaluiert und kann unter [Reichenberg, 2018] aufgerufen werden. Dafür wurden die vorhandenen Aufstehbewegungen und deren grundsätzliche Probleme analysiert und dahingehend angepasst, dass die Gelenke weniger häufig überlastet werden. Aufbauend darauf wurden drei Bereiche entwickelt, um das Aufstehen zu stabilisieren und so zu normen, dass sich möglichst jeder Aufstehversuch gleich verhält. Dafür wurden zum einen in die Keyframeinterpolation Bedingungen hinzugefügt, um auf unerwartete Ereignisse reagieren zu können, wozu ein Balancierverfahren nicht in der Lage wäre. Zusätzlich wurde damit der zeitlichen Ablauf optimiert. Zum anderen wurde die Keyframeinterpolation erweitert, um auf überbelastete Gelenke zu reagieren. Zusätzlich wurde ein PID-Regler basiertes Balancierverfahren, auf Basis des Schwerpunktes projiziert auf die Fußebene, neu entwickelt.

In der Evaluation wurde anschließend das Aufstehen, unter verschiedensten Bedingungen, getestet, um die verschiedenen entwickelten Methoden zu untersuchen, mit dem Ergebnis, dass alle drei Teilbereiche zur deutlichen Besserung der Aufstehversuche beitragen. Ebenfalls konnte in Erfahrung gebracht werden, dass das Nichterreichen von Gelenkwinkeln erfolgreich durch andere Gelenke ausgleichbar ist. Die Erfolgsrate des Aufstehens von 2017 und 2018 in Höhe von 70%-80% konnte somit erfolgreich unter Extrembedingungen mit noch stärker abgenutzten *NAOs* auf über 91% erhöht werden.

### 7.2 Ausblick

In dieser Arbeit wurden einfache Lösungsansätze umgesetzt, um inkorrekte Zustände (siehe Abschnitt 5.6) von humanoiden Robotern des Typs *NAO* auszugleichen. Die Umsetzung in dieser Arbeit ist aber sehr pflegebedürftig. So müssen inkorrekte Bewegungszustände zuerst

einmal auftreten und von einem Menschen erkannt werden, welcher diese dann dem Framework hinzugefügt.

Des Weiteren treten einige inkorrekte Bewegungszustände nur selten auf und sind schwer reproduzierbar, wodurch nur schwer sicherzustellen ist, ob eine Lösungsumsetzung statt einen Fehlerzustand zu lösen lediglich einen anderen auslöst. Da dieser Ausgleich der inkorrekten Bewegungszustände trotz Programmierfehler unter Extrembedingungen sehr gut funktioniert, besteht zumindest für Roboter des Typs *NAO* nicht die Notwendigkeit, einen anderen Ansatz im aktuellen Framework umzusetzen. Eine Implementierung, in der humanoide Roboter inkorrekte Zustände selbstständig messen und eine Lösung finden, wäre somit zwar besser und stabiler, die in dieser Arbeit entwickelte Lösung scheint aber auszureichen. Für andere Bewegungsbereiche, wie das Laufen oder das Schießen, könnte untersucht werden, ob ein Gelenkausgleich, wie in Abschnitt 5.6 entwickelt wurde, sinnvoll ist.

Genauso könnte für Robotikanwendungen mit schlechter Hardware der in dieser Arbeit verwendete Ansatz untersucht werden, ob damit die Erhaltung der Stabilität des Roboters im Allgemeinen verbessert werden kann.

Ebenfalls muss das Balancierverfahren aus Abschnitt 5.5 überarbeitet werden. Die Verstärkung sorgt zwar dafür, dass in Grenzfällen die Aufstehversuche häufiger erfolgreich sind, dennoch zappelt der *NAO* in den Fällen, in denen er kurz vor dem Umfallen ist, noch zu stark. In der gesamten Evaluation kam dies aber nur zweimal vor. Dazu geht aus den Ergebnissen der Evaluation aus Kapitel 6 hervor, dass der PID-Regler zwar hilft, um den Schwerpunkt des *NAOs* erfolgreich zu verschieben, dies führt aber zu leichten Ungleichmäßigkeiten im Ablauf der Bewegung.

Zuletzt ist die Erkennung des Abbrechens im Aufstehen ausbaufähig. Wie bereits in Abschnitt 5.3 erwähnt, ist eine frühestmögliche Erkennung wichtig, um möglichen Schäden vorzubeugen. Zwar scheint die in dieser Arbeit entwickelte Erkennung auszureichen, da im gesamten Zeitraum der Entwicklung bei keinem *NAO* ein Gelenk gebrochen ist, dennoch könnte eine Implementierung für die Erkennung eines fallenden *NAOs* aus der Arbeit von Krause [2018] verwendet werden, um in jedem Keyframe das Aufstehen früher abzurechnen.

# Abbildungsverzeichnis

2.1	Darstellung vom NAO V5 . . . . .	4
4.1	Vergleich von zwei Posen . . . . .	8
4.2	Problem vom RHipPitch-Gelenk . . . . .	9
4.3	Ein Gelenk unter leichter Belastung . . . . .	11
4.4	Ein weich gestelltes Gelenk unter leichter Belastung . . . . .	11
4.5	Ein Gelenk welches von Anschlag zu Anschlag im Gelenkspiel bewegt wird . . . . .	12
4.6	Ein Gelenk welches innerhalb des Gelenkspiels gehalten wird . . . . .	12
4.7	Ein Gelenk unter Be- und Entlastung, während der <i>NAO</i> hingestellt und hochgehoben wird. Die Y-Achse beschreibt die Abweichung von Soll-Ist in Grad, die X-Achse die Anzahl der vergangenen Frames. . . . .	13
5.1	FrontFast Teil 1 . . . . .	18
5.2	FrontFast Teil 2 . . . . .	18
5.3	FrontFast Teil 3 . . . . .	18
5.4	Fehler bei FrontFast . . . . .	19
5.5	Bewegung LyingOnArm . . . . .	19
5.6	Bewegung ArmStretched . . . . .	20
5.7	Bewegung Front . . . . .	20
5.8	Back up für die Bewegung Front . . . . .	21
5.9	Bewegung Back . . . . .	21
5.10	Falsch positionierter Arm in der Bewegung <i>Back</i> . . . . .	22
5.11	Back up in der Bewegung <i>Back</i> . . . . .	22
5.12	Bewegung FromSplitt . . . . .	23
5.13	Vom Sitzen ins Stehen . . . . .	23
5.14	Das Stützpolygon . . . . .	28

---

5.15	Versuchsaufbau zum Ermitteln der PID-Regler Parameter . . . . .	32
5.16	Vereinfachter Programmablauf zum Entscheiden, wie im aktuellen Zustand vorgegangen werden soll . . . . .	36
5.17	Vereinfachter Ablauf, wann welche Bewegung ausgeführt wird . . . . .	37
6.2	Die geneigte Holzplatte mit 2.1 Grad Neigung . . . . .	41
6.3	Ablauf der Bewegung <i>FrontFast</i> . . . . .	42
6.4	Ablauf der Bewegung <i>FrontFast</i> mit starker Abweichung von Soll-Ist des HipYawPitch- Gelenkes . . . . .	44
6.5	Sonderfall: Ablauf der Bewegung <i>FrontFast</i> mit starker Abweichung von Soll- Ist des HipYawPitch-Gelenkes . . . . .	45
6.1	Vergleich mit und ohne entwickelte Verfahren . . . . .	46

# Literaturverzeichnis

- [Aldebaran SoftBank Group 2015] ALDEBARAN SOFTBANK GROUP: *NAO Documentation*. 2015. – [http://doc.aldebaran.com/2-1/family/nao\\_dcm/actuator\\_sensor\\_names.html](http://doc.aldebaran.com/2-1/family/nao_dcm/actuator_sensor_names.html)  
Abgerufen am: 20.11.2018
- [B-Human 2018] <https://www.b-human.de/index-de.html> Abgerufen am: 22.11.2018
- [Böckmann 2015] BÖCKMANN, Arne: *Entwicklung einer dynamischen Schussbewegung mit dem humanoiden Roboter NAO*. Deutschland, Universität Bremen, Masterarbeit, 2015
- [Committee 2017] COMMITTEE, RoboCup T.: *RoboCup Standard Platform League (NAO) Rule Book*. <http://spl.robocup.org/wp-content/uploads/downloads/2017/01/RuleBook2017.pdf> Abgerufen am: 27.11.2018, 2017
- [Depinet u. a. 2014] DEPINET, Mike ; MACALPINE, Patrick ; STONE, Peter: *Keyframe Sampling, Optimization, and Behavior Integration: Towards Long-Distance Kicking in the RoboCup 3D Simulation League*. <https://pdfs.semanticscholar.org/ea40/9f8a80ed7e28bcc3db8a02077d21f8e75297.pdf>, 2014
- [Graf u. Röfer 2011] GRAF, Colin ; RÖFER, Thomas: *A Center of Mass Observing 3D-LIPM Gait for the RoboCup Standard Platform League Humanoid*. Deutschland : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.299.390&rep=rep1&type=pdf>, 2011
- [Krause 2018] KRAUSE, Daniel: *Fallerkennung durch Vorhersage der Schwerpunkttrajektorie für den humanoiden Roboter NAO*. Deutschland, Universität Bremen, Bachelorarbeit, 2018.  
– Nicht veröffentlicht
- [Kuffner u. a. 2001] KUFFNER, James ; NISHIWAKI, Koichi ; KAGAMI, Satoshi ; INABA, Masayuki ; INOUE, Hirochika: Motion Planning for Humanoid Robots Under Obstacle and Dynamic Balance Constraints. In: *International Conference on Robotics & Automation*, 2001
- [Müller 2010] MÜLLER, Judith: *Roboter spielen Fussbälle - Modellierung komplexer, dynamischer Bewegungen humanoider Roboter*. Deutschland, Universität Bremen, Diplomarbeit, 2010

- [Müller u. a. 2011] MÜLLER, Judith ; LAUE, Tim ; RÖFER, Thomas: *Kicking a Ball – Modeling Complex Dynamic Motions for Humanoid Robots*. Deutschland : <http://www.b-human.de/downloads/RC-Mueller-etal-11.pdf>, 2011
- [Nao Devils 2018] [https://www.youtube.com/channel/UCp\\_3raHDiDfTqcIygMrg-0w](https://www.youtube.com/channel/UCp_3raHDiDfTqcIygMrg-0w), Youtubekanal, Abgerufen am: 20.11.2018
- [Nao-Team HTWK 2018] <http://robocup.imn.htwk-leipzig.de/index.php?lang=de> Abgerufen am: 21.11.2018
- [Reichenberg 2018] REICHENBERG, Philip: *Die entwickelten Bewegungen in der Bachelorarbeit von Philip Reichenberg*. 2018. – [https://www.youtube.com/playlist?list=PLv1U39oLmWzZNoAdvmQX81\\_n7-2wfWQU2](https://www.youtube.com/playlist?list=PLv1U39oLmWzZNoAdvmQX81_n7-2wfWQU2) Erstellt am 23.11.2018
- [Röfer u. a. 2017] RÖFER, Thomas ; LAUE, Tim ; BÜLTER, Yannick ; KRAUSE, Daniel ; KUBALL, Jonas ; MÜHLENBROCK, Andre ; POPPINGA, Bernd ; PRINZLER, Markus ; POST, Lukas ; ROEHRIG, Enno ; SCHRÖDER, René ; THIELKE, Felix: *B-Human Team Report and Code Release 2017*. Deutschland : <http://www.b-human.de/downloads/publications/2017/coderelease2017.pdf>, 2017
- [Röfer u. a. 2018] RÖFER, Thomas ; LAUE, Tim ; BÜLTER, Yannick ; KRAUSE, Daniel ; KUBALL, Jonas ; MÜHLENBROCK, Andre ; POPPINGA, Bernd ; PRINZLER, Markus ; POST, Lukas ; ROEHRIG, Enno ; SCHRÖDER, René ; THIELKE, Felix: *B-Human Team Report and Code Release 2018*. Deutschland : <http://www.b-human.de/downloads/publications/2018/coderelease2018.pdf>, 2018
- [Simon B. 2015] SIMON B.: *Balancing Robot Tutorial*. <http://wired.chillibasket.com/2015/03/pid-controller/>, 2015. – Abgerufen am 16.03.2015
- [Softbank Robotics 2018] <https://www.softbankrobotics.com/emea/en/company> Abgerufen am: 20.11.2018
- [Sugihara u. a. 2002] SUGIHARA, Tomomichi ; NAKAMURA, Yoshihiko ; INOUE, Hirochika: Realtime Humanoid Motion Generation through ZMP Manipulation based on Inverted Pendulum Control. In: *International Conference on Robotics & Automation*, 2002