



Technische Universität Hamburg-Harburg

Institut  
für  
Zuverlässiges Rechnen

---

**Modellierung und Regelung dynamischer  
Kickbewegungen humanoider Roboter auf Basis des  
NAO-Robotiksystems**

**Bachelorarbeit**

vorgelegt von  
Nathapon Olaf Lüders

Hamburg, den 4. Februar 2013

---

Erstprüfer : Prof. Dr.-Ing. Sven-Ole Voigt

Zweitprüfer: Prof. Dr. Herbert Werner

Betreuer: Dipl.-Ing. Stefan Kaufmann

---

### **Erklärung**

Hiermit versichere ich, Nathapon Olaf Lüders, diese Arbeit im Rahmen der an der Technischen Universität Hamburg-Harburg üblichen Betreuung selbstständig angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel eingesetzt zu haben.

Nathapon Olaf Lüders

Hamburg, den 4. Februar 2013

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
<b>2. NAO-Robotiksystem</b>	<b>3</b>
2.1. Hardware . . . . .	3
2.1.1. Sensoren . . . . .	3
2.2. Software . . . . .	5
2.2.1. Simulationsumgebung Webots . . . . .	5
2.2.2. Choregraphe . . . . .	6
<b>3. Regelstrategien für humanoide Roboter</b>	<b>8</b>
3.1. Statische Balance . . . . .	8
3.2. Dynamische Balance . . . . .	9
3.2.1. Zero Momentum Point (ZMP) . . . . .	9
3.2.2. Cart-table Modell . . . . .	11
3.2.3. 3D Linear Inverted Pendulum Mode (3DLIPM) . . . . .	11
3.2.4. Spezifisches dynamisches Modell . . . . .	13
3.2.5. Evaluation der Modelle . . . . .	13
<b>4. Roboterspezifisches dynamisches Modell</b>	<b>14</b>
4.1. Mechanische Grundlagen . . . . .	14
4.1.1. Massenträgheitsmomente und Steineranteile . . . . .	14
4.1.2. d'Alembertsches Moment . . . . .	16
4.2. Berechnung der d'Alembertschen Momente im NAO . . . . .	17
4.2.1. Annahmen . . . . .	17
4.2.2. Arme . . . . .	19
4.2.3. Schussbein . . . . .	22
4.3. Moment auf Grund von Spiel in den Gelenken . . . . .	23
4.4. Statisches Moment . . . . .	25
4.5. Zusammenfassung . . . . .	25
<b>5. Implementierung</b>	<b>26</b>
5.1. Verwendete Module . . . . .	26
5.1.1. Blackboard . . . . .	26
5.1.2. ForwardKinematics . . . . .	26
5.1.3. InverseKinematics . . . . .	27
5.1.4. Com . . . . .	27
5.1.5. BodyPartCom . . . . .	27
5.1.6. VelocitiesAccelerations . . . . .	28

5.1.7. KickMotion . . . . .	28
5.1.8. Torque . . . . .	29
5.1.9. Control . . . . .	30
5.2. Prozessarchitektur . . . . .	32
<b>6. Evaluation</b>	<b>33</b>
6.1. Simulation . . . . .	33
6.1.1. Testbedingungen . . . . .	34
6.1.2. Simulationsspezifische Parameter . . . . .	35
6.1.3. Realitätsspezifische Parameter . . . . .	37
6.1.4. Auswertung . . . . .	40
6.1.5. Bewertung der Modellkomponenten . . . . .	41
6.2. Realversuch . . . . .	42
6.2.1. Testbedingungen . . . . .	42
6.2.2. Auswertung . . . . .	47
6.2.3. Bewertung der Modellkomponenten . . . . .	48
6.3. Zusammenfassung und Bewertung . . . . .	49
<b>7. Zusammenfassung und Ausblick</b>	<b>50</b>
<b>A. Hardwareübersicht</b>	<b>52</b>
A.1. Abmessungen des NAO . . . . .	52
A.2. Massen, Schwerpunktpositionen und Inertialmatrizen der Bauteile . .	53
A.3. Gelenkwinkel . . . . .	60
A.4. Reglerparameter . . . . .	62
<b>B. Inhalte der CD</b>	<b>63</b>

# Abbildungsverzeichnis

1.1.	NAO-Roboter [Ald13]	1
2.1.	Resistive Kraftsensoren [Ald13]	4
2.2.	Webots: Benutzeroberfläche	6
2.3.	Choregraphe: Workflow	7
2.4.	Choregraphe: Echtzeitsteuerung	7
3.1.	Supportfläche [Kau11]	8
3.2.	Zero Momentum Point [VB04]	9
3.3.	Zusammenhang CoP, ZMP, FZMP [VB04]	10
3.4.	Cart-table Modell [SK08]	11
3.5.	3DLIPM [Kau11]	12
4.1.	Steineranteil [DD09]	14
4.2.	NAO: Koordinatensystem [Ald13]	17
4.3.	NAO: Einzelteile des Arms [Ald13]	19
4.4.	NAO: Arm [Ald13]	20
4.5.	NAO: Einzelteile des Beins [Ald13]	22
4.6.	Auswirkung des Spiels in den Gelenken	24
5.1.	Inverse Kinematik [Kau11]	27
5.2.	Webots: Schussphasen	28
5.3.	Veranschaulichung des Regelungsbereichs im Fuß	30
5.4.	PID-Regler mit Anti-Windup (nach [Wer11])	31
5.5.	Prozessarchitektur	32
6.1.	Webots: Ausgangsposition kurz vor dem Schuss	34
6.2.	Plot: ZMP ohne Regelung, Simulation mit simulationsspezifischen Parametern	35
6.3.	Plot: ZMP mit Regelung, Simulation mit simulationsspezifischen Parametern	36
6.4.	Plot: Stellgrößen, Simulation mit simulationsspezifischen Parametern	36
6.5.	Plot: Moment im Fußgelenk ohne Regelung, Simulation mit realitätspezifischen Parametern	37
6.6.	Plot: ZMP ohne Regelung, Simulation mit realitätsspezifischen Parametern	38
6.7.	Plot: ZMP mit Regelung, Simulation mit realitätsspezifischen Parametern	39
6.8.	Plot: Stellgrößen, Simulation mit realitätsspezifischen Parametern	40

6.9. Plot: ZMP ohne und mit Regelung, Simulation mit realitätsspezifischen Parametern . . . . .	40
6.10. Plot: Momentenverläufe der einzelnen Modellkomponenten, Simulation . . . . .	41
6.11. Foto: Ausgangsposition kurz vor dem Schuss . . . . .	42
6.12. Plot: Moment im Fußgelenk ohne Regelung, Realversuch . . . . .	44
6.13. Plot: ZMP ohne Regelung, Realversuch . . . . .	45
6.14. Plot: ZMP mit Regelung, Realversuch . . . . .	46
6.15. Plot: Stellgrößen, Realversuch . . . . .	46
6.16. Plot: ZMP mit und ohne Regelung, Realversuch . . . . .	47
6.17. Plot: Momentenverläufe der einzelnen Modellkomponenten, Realversuch . . . . .	48
A.1. NAO: Abmessungen [Ald13] . . . . .	52
A.2. NAO: Kopf V. 4.0 [Ald13] . . . . .	53
A.3. NAO: Genick [Ald13] . . . . .	53
A.4. NAO: Torso [Ald13] . . . . .	54
A.5. NAO: Schulter [Ald13] . . . . .	54
A.6. NAO: Oberarm [Ald13] . . . . .	55
A.7. NAO: Ellenbogen [Ald13] . . . . .	55
A.8. NAO: Unterarm [Ald13] . . . . .	56
A.9. NAO: Hand [Ald13] . . . . .	56
A.10.NAO: Becken [Ald13] . . . . .	57
A.11.NAO: Hüfte [Ald13] . . . . .	57
A.12.NAO: Oberschenkel [Ald13] . . . . .	58
A.13.NAO: Schienbein [Ald13] . . . . .	58
A.14.NAO: Fußgelenk [Ald13] . . . . .	59
A.15.NAO: Fuß [Ald13] . . . . .	59
A.16.NAO: Gelenkwinkelbegrenzungen beider Arme. [Ald13] . . . . .	60
A.17.NAO: Gelenkwinkelbegrenzungen des Torsos [Ald13] . . . . .	60
A.18.NAO: Gelenkwinkelbegrenzungen des linken Beins. [Ald13] . . . . .	61
A.19.NAO: Gelenkwinkelbegrenzungen des rechten Beins [Ald13] . . . . .	61
A.20.NAO: Gelenkwinkelbegrenzungen des Kopfes [Ald13] . . . . .	61

# 1. Einleitung

Es wird das Jahr 2050 geschrieben: *"Die Technische Universität Hamburg-Harburg blickt auf das Jahr 2013 zurück, in dem der Grundstein für das Erfolgsprojekt RobotING gelegt wurde. Heute Abend tritt der amtierende Fußballweltmeister Deutschland, gegen das Team der TUHH an. Dieses besteht jedoch nicht aus Menschen sondern aus humanoiden Robotern."* So in etwa könnte eine Schlagzeile aus dem Jahre 2050 lauten, wenn der Plan des *RoboCup* Veranstalters aufgeht. humanoide Roboter sollen bis 2050 gut genug Fußball spielen können, um dann gegen den Fußballweltmeister anzutreten und zu gewinnen. Bis dahin ist es jedoch noch ein langer Weg.

Bevor die Autonomie und Bewegung humanoider Roboter soweit ist, um sich mit Menschen zu messen, muss noch viel Grundlagenforschung auf dem Gebiet der humanoiden Robotik betrieben werden. Um diese Forschungsfelder weiterzuentwickeln, sind Fußballturniere wie der *RoboCup* ideal. In der *RoboCup - Standard Platform League* treten ausschließlich NAO Roboter gegeneinander an. Dadurch wird der Schwerpunkt auf die Softwareentwicklung gelegt, da alle Teams die gleiche Hardwareder Grundlage besitzen. Die Roboter dürfen während des Spiels nicht gesteuert werden, weshalb eine komplett autonome Handlungsweise von großer Bedeutung ist. Außerdem müssen die Bewegungsabläufe möglichst stabil, präzise und schnell erfolgen. Die Bewegungen, die für Menschen selbstverständlich und natürlich wirken, sind für humanoide Roboter nicht so einfach nachzuahmen, wie man vielleicht glauben mag. So erfährt man bei der Forschung an humanoiden Robotern auch oft etwas über die Funktionsweise der eigenen Bewegungen und Verhaltensmuster.

Es ist von großem Interesse, möglichst viele Informationen über Kräfte zu haben, die auf den NAO während einer Bewegung wirken, um einen ausbalancierten und



**Abb. 1.1.:** Der NAO. [Ald13]

stabilen Bewegungsablauf zu erhalten. Es ist im Allgemeinen nicht zufriedenstellend, wenn man in der Dynamik auf die Rückgabewerte von Sensoren angewiesen ist. Arbeitet man mit dem NAO stellt sich außerdem schnell heraus, dass die Drucksensoren an den Füßen sehr unzuverlässige Werte zurückgeben. Daher ist es nötig ein Modell zu entwickeln, welches die Kräfte, die auf den Fuß wirken, möglichst gut nachbildet und ggf. sogar vorausberechnet. In dieser Arbeit wird deshalb ein robotspezifisches dynamisches Modell entwickelt und implementiert, um möglichst genaue Informationen über Kräfte und Momente während der Bewegung des NAOs zu erhalten. Auf diesen Informationen aufbauend, wird ein Regler implementiert und anschließend anhand einer einfachen Schussbewegung getestet. Die Implementierung erfolgt in der Programmiersprache C++ und wird sowohl in einer Simulation als auch auf dem NAO selbst evaluiert.

Im folgenden Kapitel wird zunächst auf das NAO-Robotiksystem eingegangen, welches die Grundlage für diese Arbeit darstellt. Dazu werden kurz die notwendigen Informationen zur Hardware und Software für dieses Thema vorgestellt. Danach werden verschiedene Modellvorstellungen, die für die Entwicklung eines dynamischen Modells und eine Regelung wichtig sind, angesprochen und anschließend evaluiert. In Kapitel 4 wird das dynamische Modell komponentenweise aufgestellt und erläutert. Anschließend wird auf die Implementierung des dynamischen Modells, der Schussbewegung und der Regelung eingegangen. Außerdem werden die Module vorgestellt, die für die Implementierung von grundlegender Bedeutung. Um die implementierten Module auf ihre Tauglichkeit zu prüfen, werden in Kapitel 6 Simulationen und Realversuche durchgeführt und hinsichtlich der Aufgabenstellung dieser Arbeit evaluiert. Zum Schluss wird ein Ausblick auf weitere Entwicklungsoptionen gegeben.

## 2. NAO-Robotiksystem

Der NAO ist ein programmierbarer humanoider Roboter des französischen Herstellers *Aldebaran Robotics*. Da er sehr vielseitig ist, wird er bereits an vielen Universitäten und Forschungsinstitutionen zu Forschungszwecken eingesetzt. Seit 2007 wird der NAO in der *Standard Platform League (SPL)* des *RoboCups* verwendet.

### 2.1. Hardware

Bei dem hier verwendeten NAO handelt es sich um einen NAO H25 V. 3.3 mit dem Kopf des NAO V. 4.0. Er ist daher mit einem 1,6 GHz Prozessor und 1 GB RAM ausgestattet, wodurch seine Datenverarbeitungsgeschwindigkeit im Vergleich zum Vorgängermodell mit 500 MHz Prozessor und 256 MB SDRAM stark erhöht wurde. Die maximale Akkuladung beträgt 2,15 Ah. Laut Hersteller kann der NAO 60 Minuten in aktiver Benutzung und 90 Minuten in normaler Benutzung betrieben werden. Er ist 57,3 cm groß, hat ein Gewicht von 5,2 kg und besitzt 25 Freiheitsgrade. Die Gelenke sind zudem in ihrer Bewegungsfreiheit den menschlichen Gelenken angepasst. Es ist also nicht möglich den NAO in eine nicht humanoide Haltung zu bewegen.

Der Datentransfer sowie die Steuerung und Programmierung des NAOs können entweder über W-LAN oder ein kabelgebundenes Netzwerk stattfinden. Aus praktikablen Gründen wird meist die Verbindung über W-LAN genutzt.

#### 2.1.1. Sensoren

Der NAO besitzt eine Vielzahl von Sensoren, die jedoch in dieser Arbeit größtenteils nicht von Bedeutung sind. In diesem Abschnitt werden deshalb die drei relevanten Sensorarten vorgestellt.

##### Magnetic Rotary Encoders (MRE)

Die Winkelpositionen der einzelnen Gelenke werden über magnetische Drehwinkelgeber, die auf dem Hall-Effekt basieren, gemessen. Es wird laut Hersteller mit einer Präzision von 0,1 ° gemessen.

##### Inertial Unit

Die *Inertial Unit* befindet sich im Torso und besitzt drei Komponenten - das Accelerometer und zwei Gyroskope. Sie haben einen gemeinsamen vom Kopf unabhängigen Prozessor und dienen zur Erfassung der Orientierung und des Bewegungszustands des NAOs.

Das Gyroskop misst jeweils mit einer Präzision von 5% und registriert Winkelgeschwindigkeiten von bis zu  $500 \text{ }^{\circ} \frac{1}{\text{s}}$ .

Das Accelerometer misst Beschleunigungen in alle drei Raumrichtungen. Die Präzision beträgt laut Hersteller 1% und es kann eine bis zu zweifache Erdbeschleunigung erfasst werden. Zu beachten ist, dass die Beschleunigung mittlerweile in zwei verschiedenen Einheiten bzw. Größen aus dem Accelerometer abgerufen werden kann. Zum einen eine Angabe in  $\frac{\text{m}}{\text{s}^2}$  und zum anderen in einer von Aldebaran Robotics angegeben einheitenlosen Größe. Laut Dokumentation entspricht eine Ausgabe von 56 ungefähr der einfachen Erdbeschleunigung. Messungen auf dem NAO, um diesen Wert zu bestätigen, führen allerdings zu dem Ergebnis das die Ausgabe von 65 der Erdbeschleunigung entspricht.

### Force Sensitive Resistors (FSRs)

In den Füßen befinden sich jeweils vier druckempfindliche Sensoren, die die elektrische Widerstandsänderung in Abhängigkeit der Druckkraft messen. Sie besitzen ein Messspektrum von 0 bis 25 N und dienen im späteren Verlauf als Referenz für das dynamische Modell, welches die im Fußgelenk wirkenden Momente annähern soll. Anzumerken ist, dass die Ausgabe dieser Sensoren kg entspricht. Zudem verlieren sie sehr schnell ihre Kalibrierung und fallen gelegentlich für eine gewisse Zeit komplett aus.

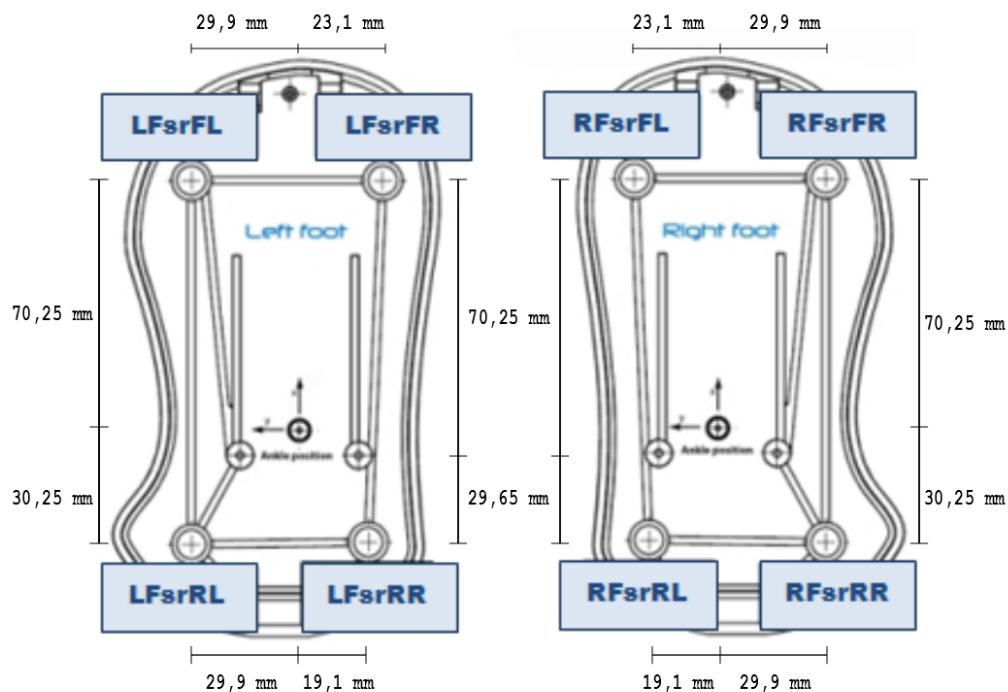


Abb. 2.1.: FSRs an der Fußsohle des NAOs. [Ald13]

## 2.2. Software

In diesem Abschnitt wird die Software vorgestellt, die die Grundlage für diese Arbeit bildet und benötigt wird. Der NAO wird in der Sprache C++ oder Python programmiert, welche über das *NAOqi* Framework mit dem NAO kommuniziert, da auf dem NAO die vom Hersteller entwickelte *NAOqi* Software läuft. *NAOqi* und das *NAOqi* Framework werden im Folgenden nicht weiter erläutert, da sie sich nicht direkt auf das Ergebnis dieser Arbeit auswirken und schon in vorherigen wissenschaftlichen Arbeiten der Technischen Universität Hamburg-Harburg beschrieben wurden (s. [Kau11], [Hol12]).

Durch Drücken auf den Hauptschalter des eingeschalteten NAOs am Brustkorb, gibt dieser seine IP und den Zustand seines Akkus aus. Über die IP kann man sich im Browser mit einem Passwort in den *Web Client* einloggen. Dieser gibt Auskunft über die geladenen Module mit zugehöriger Dokumentation. Des Weiteren können noch softwaretechnische Einstellungen, wie z.B die Erlaubnis zur Deaktivierung des *Fallmanagers*, gemacht werden. Der *Fallmanager* ist bei Bewegungen stets aktiv und lässt den NAO durch das Accelerometer ein Fallen frühzeitig erkennen. Wird ein Fall registriert, bricht der NAO alle Programmabläufe ab und versucht sich in eine möglichst geschützte Haltung zu bringen.

Für die Datenübertragung wird ein FTP-Client genutzt.

### 2.2.1. Simulationsumgebung Webots

Bevor ein neuer Bewegungsablauf auf dem NAO implementiert wird, ist es immer sinnvoll, jenen vorher in einer Simulation zu testen. Auf diese Weise können unerwartete Bewegungsabläufe die den NAO beschädigen würden unter idealen Bedingungen beobachtet und im Nachhinein vermieden werden.

Als Simulationsumgebung wird *Webots 7.0.1* gewählt. Dies ist die aktuelle vom Hersteller unterstützte Simulationsumgebung, welche die von *Aldebaran Robotics* entwickelte Simulationsumgebung *NAOSim* abgelöst hat. *Webots* liefert eine dreidimensionale zeitlich synchronisierte Simulation, welche jedoch auf Grund des hohen Rechenaufwands nur selten in Echtzeit abläuft.



**Abb. 2.2.:** Webots 7.0.1 als Simulationsumgebung mit NAO und Fußballfeld der TUHH.

Es gibt eine Vielzahl an Einstellmöglichkeiten. Dies betrifft zum einen den Roboter selbst und zum anderen auch die Umgebung. In der mitgelieferten Prototyp Datei des Herstellers ist der NAO bereits für die NAO Version 3.3 vordefiniert. In dieser Datei stehen alle für die Simulation benötigten Abhängigkeiten, damit das Verhalten des Roboters möglichst gut simuliert werden kann. Der NAO kann daher auch in der Simulation nicht seine realen maximalen Geschwindigkeiten und Winkelstellungen übersteigen. Da der für diese Arbeit verwendete NAO den Kopf der Version 4.0 hat, wurden bereits im Vorfeld die Daten für den Kopf angepasst. Mit der Prototyp Datei ist es folglich auch möglich mehrere NAOs dieses Typs auf dem Spielfeld gleichzeitig zu simulieren, was allerdings eine Erhöhung des Rechenaufwands zur Folge hat.

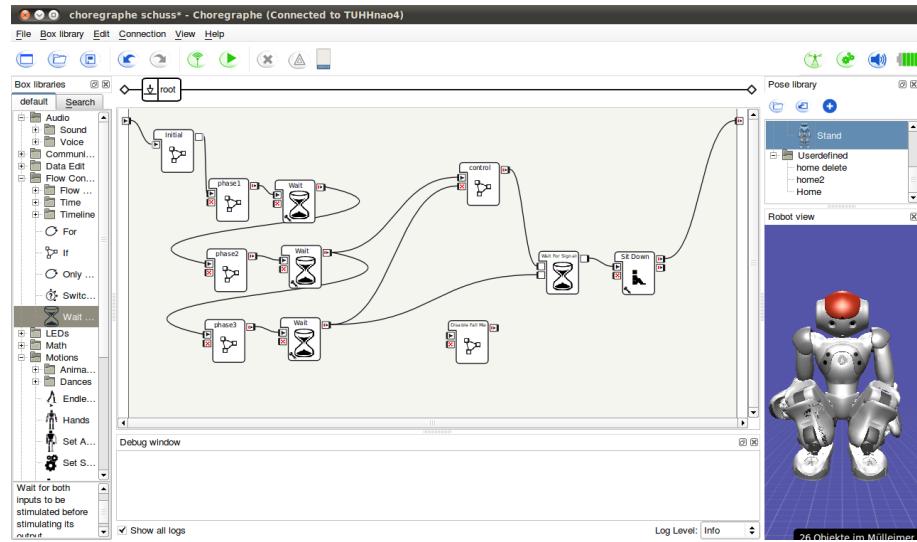
Als Umgebungswelt wird ein Fußballfeld mit einem Ball verwendet. Sowohl Fußballfeld als auch Ball entsprechen den Normen des *RoboCups*, an dem die TUHH zukünftig teilnehmen wird. In *Webots* ist es auch möglich die Kameraposition, Ballposition sowie die Position und Ausrichtung des NAOs über die *sidebar* zu verstehen (s. Abb. 2.2). Man kann unter anderem auch eine Kraft am NAO angreifen lassen. Gesteuert wird der NAO über einen *Controller*. In dieser Arbeit wird ein in Matlab geschriebener *Controller* verwendet. Anstelle von Matlab kann auch C, C++, Java, Python, ROS oder URBI verwendet werden.

## 2.2.2. Choregraphe

Choregraphe ist eine grafische Oberfläche zum Programmieren des NAOs. Sie arbeitet objektorientiert und funktioniert über *drag-and-drop*. Jedes Modul wird in einer eigenen Box angezeigt und kann in Reihe oder parallel zu anderen Modulen ausgeführt werden. Es ist auch möglich ein Modul durch ein Ereignis auszulösen.

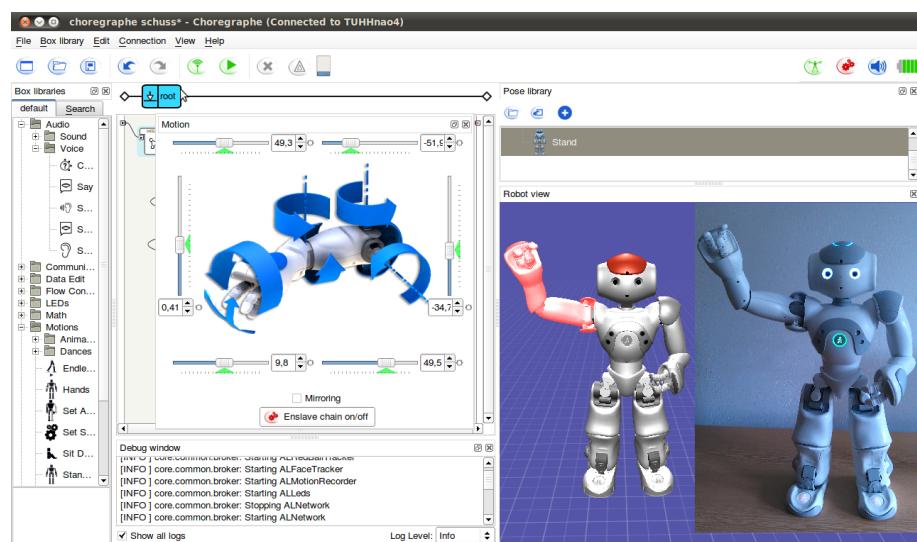
Beim Start von Choregraphe ist schon eine Vielzahl an vorgefertigten Modulen vorhanden. Der NAO kann z. B. laufen, sich hinsetzen oder versuchen seine Position auf dem Boden zu erkennen und dann aufzustehen, falls er nicht schon steht. In gleicher Weise sind auch bereits Abfragen verschiedener Sensoren, Timer für Verzögerungen

und logische Funktionen implementiert. Die Algorithmen für Bewegungsabläufe sind jedoch nicht perfekt, sondern dienen eher dem Zweck der Demonstration. Es ist daher durchaus interessant und zweckmäßig eigene verbesserte Bewegungsmodule zu erstellen, welche dann als eigene Box hinzugefügt werden und in den Programmfluss mit eingebunden werden können.



**Abb. 2.3.:** Workflow eines Programms auf dem NAO in Choregraphe.

Choregraphe bietet zudem eine Echtzeitsteuerung und -synchronisation des NAOs. Somit ist es möglich den echten NAO in eine gewünschte Position zu bringen, diese in Choregraphe zu speichern und später über Choregraphe wieder anzufahren. Dies ist durchaus zweckmäßig, um den NAO vor einer Schussbewegung in eine standardisierte Ausgangsposition zu bringen, die auch in der Simulation oder in realen Tests verwendet wird.



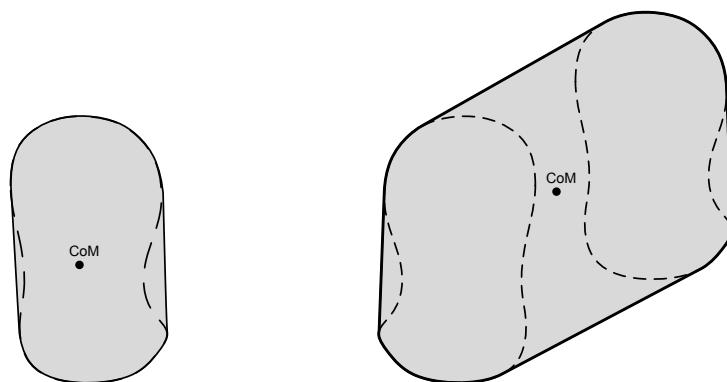
**Abb. 2.4.:** Echtzeitsteuerung des rechten Arms des NAOs.

# 3. Regelstrategien für humanoide Roboter

In diesem Kapitel werden verschiedene Regelstrategien für die Balance humanoider Roboter vorgestellt und anschließend hinsichtlich ihres Nutzens für diese Arbeit bewertet. Vor allem in der Dynamik treten oft unvorhersehbare Störungen auf, die nur durch eine geeignete Regelung wieder ausgeglichen werden können. Im Allgemeinen basieren alle Modelle auf den Kräften und Momenten, die im Standfuß oder den Standfüßen wirken, da sie während der Fortbewegung der einzige Angriffspunkt für reaktive Kräfte sind. Um während einer Bewegung das Risiko umzufallen möglichst gering zu halten, ist eine stabile Ausgangslage wichtig, welche durch das Modell der statischen Balance beschrieben werden kann.

## 3.1. Statische Balance

Die statische Balance eines Roboters ist gewährleistet, so lange sich der Schwerpunkt des Roboters in dem von den Füßen aufgespannten Polygon befindet. Steht der Roboter auf einem Bein, muss sich der Schwerpunkt innerhalb der projizierten Fläche des Standfußes befinden. Diese Flächen werden im Folgenden als Supportfläche bezeichnet.



**Abb. 3.1.:** Supportfläche bei statischer Balance auf einem Bein und bei statischer Balance auf zwei Beinen. [Kau11]

Würde man den Roboter mechanisch freischneiden, wäre zu sehen, dass der Kraftangriffspunkt unter dem Fuß auf einer Linie mit dem Angriffspunkt der Gewichtskraft des Roboters liegt. Die Kraftpfeile sind vom Betrag her gleich und in entgegengesetzter Richtung orientiert. Wirken auf den Roboter zusätzlich äußere Kräfte, müssen

diese durch Reibkräfte ausgleichbar sein oder innerhalb der Supportfläche in Richtung der Gewichtskraft wirken.

## 3.2. Dynamische Balance

Für die dynamische Balance reicht das Modell der statischen Balance nicht mehr aus, da während einer Bewegung der Schwerpunkt des Roboters zwischenzeitlich aus der Supportfläche hinaustreten kann. Es herrscht in Folge dessen ein kurzzeitiges Kräfteungleichgewicht. Dies führt wiederum zu resultierenden Momenten, die bei ausreichender Größe ein Kippen verursachen können. Neben den genannten resultierenden Momenten wirken sich auch die d'Alembertschen Momente (s. Abschnitt 4.1.2), die in Folge beschleunigter Bewegungen entstehen, stark auf die dynamische Balance aus. Es ist also ein Modell nötig, welches die Momente um den Standfuß oder die Standfüße berechnet, um entsprechende Gegenbewegungen einzuleiten. Ein solches Modell existiert bereits und ist unter der Abkürzung ZMP *Zero Momentum Point* bekannt und wird im folgenden Abschnitt näher erläutert.

### 3.2.1. Zero Momentum Point (ZMP)

Der ZMP wurde erstmalig 1968 von Miomir Vukobratović eingeführt (s. [SK08]).

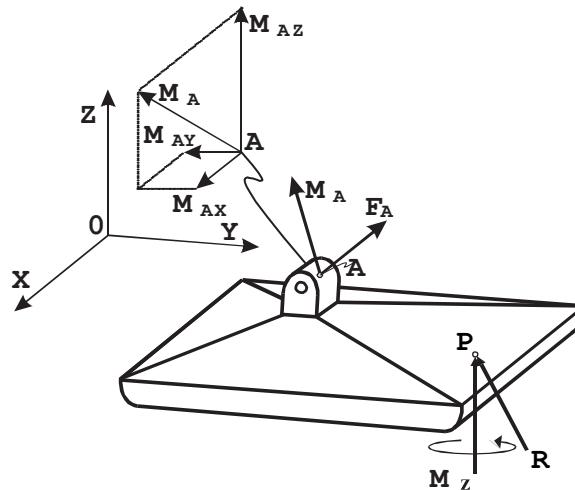
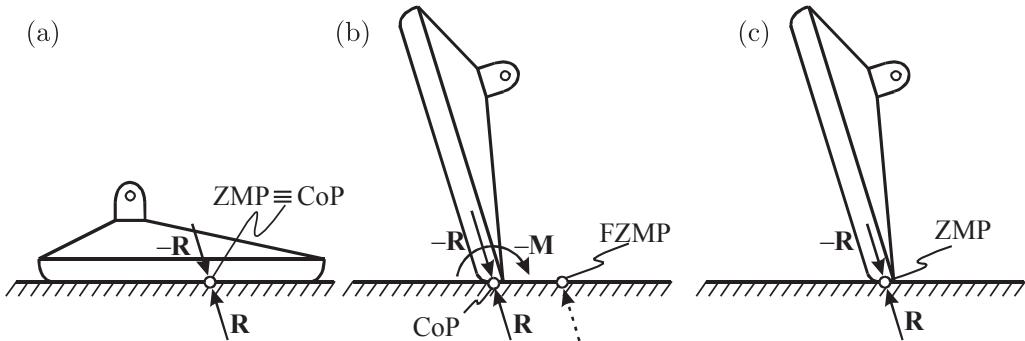


Abb. 3.2.: ZMP im Punkt P schematisch dargestellt. [VB04]

Ausgegangen wird von dem reibungsbehafteten Modell eines Fußes in Abbildung 3.2.  $M_A$  und  $F_A$  repräsentieren eine resultierende Kraft und ein resultierendes Moment, welche durch die Bewegung und das Eigengewicht hervorgerufen werden. Punkt P ist der Angriffspunkt der reaktiven Kraft  $R$  und des reaktiven Moments  $M_z$ . Die Kraftkomponenten in x- und y-Richtung der Kraft  $F_A$  sowie die Momentenkomponente  $M_{Az}$  werden durch die vorherrschende Reibung ausgeglichen. Die Kompensation durch Reibung ist durch das Moment  $M_z$  und die Kraftkomponenten in x- und y-Richtung von  $R$  dargestellt. Betrachtet man dieses Modell ohne Reibung, würden

diese Kräfte einen Roboter bisher nur in ein Gleiten über den Boden und eine Rotation um die vertikale Achse versetzen. Da der Fuß selbst keine Kraft aufbringen kann, wird sich der Angriffspunkt P der reaktiven Kraft  $R$ , je nach Größe von  $M_{Ax}$  und  $M_{Ay}$  verschieben, um ein Momentengleichgewicht anzustreben.  $F_{Az}$  führt nur zu einer Betragsänderung von  $R_z$ . Der ideale Angriffspunkt für die Reaktionskraft  $R$  ist P und demnach der ZMP. Befindet sich der ZMP nicht mehr innerhalb der Supportfläche, wird der Roboter kippen. Das ursprüngliche ZMP Modell ist also nur für die Supportfläche definiert. Liegt P außerhalb der Supportfläche wird dieser als FZMP (*Fictitious Zero Momentum Point*) bezeichnet und ist somit nicht direkt messbar.

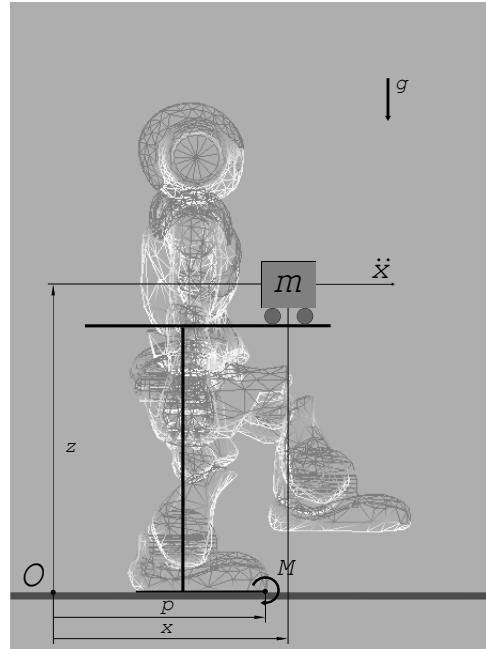


**Abb. 3.3.:** CoP (*Center of Pressure*) und FZMP (*Fictitious Zero Momentum Point*) schematisch dargestellt. [VB04]

In Abbildung 3.3 (a) ist zu sehen, dass der ZMP dem CoP entspricht, solange sich der ZMP innerhalb der Supportfläche befindet. Es kann jedoch der Fall eintreten, dass ein horizontales Moment zu groß ist und nur noch ein FZMP existiert (s. Abb. 3.3 (b)).

### 3.2.2. Cart-table Modell

Das *Cart-table Modell* ist eine Möglichkeit die, auf die Supportfläche wirkenden, Momente zu berechnen und folglich den ZMP zu bestimmen. Abbildung 3.4 veranschaulicht die Funktionsweise dieses Modells.



**Abb. 3.4.:** *Cart-table Modell* eines Roboters mit vollem Kontakt zum Boden.[SK08]

Der Wagen der Masse  $m$  repräsentiert die Masse des Roboters und die Bewegung des Schwerpunktes in x-Richtung.  $z$  entspricht dem Abstand zwischen Schwerpunkt und Boden. Die Länge des Tischfußes entspricht der Länge der Supportfläche des Roboters. Berechnet man den ZMP mit Hilfe des *Cart-table Modells* gilt Gleichung 3.2. Der berechnete ZMP kann auch außerhalb der Supportfläche liegen und wäre nach Abschnitt 3.2.1 ein FMZP. (Herleitung s. [SK08]):

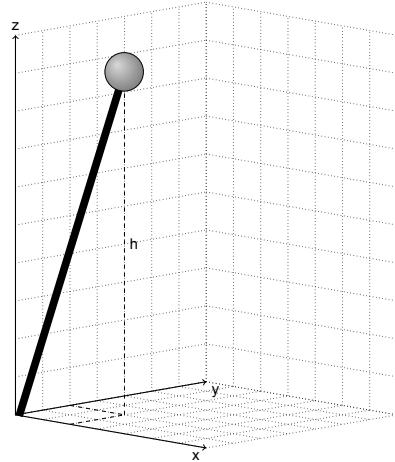
$$p = x - \frac{z}{g} \cdot \ddot{x} \quad (3.1)$$

Diese Formel für den ZMP setzt voraus, dass der Fuß stets mit der gesamten Fläche auf dem Boden aufliegt, was veranschaulicht einer Verklebung mit dem Boden gleichkommen würde.  $z$  kann also als Konstante angesehen werden.

Über vordefinierte Fußtrajektorien ist es möglich den Bereich zu berechnen, für den der ZMP in der Supportfläche liegt. Über Gleichung 3.2 ist es dann möglich die Schwerpunktsbewegung an diesen Bereich anzupassen, sodass sich der ZMP permanent innerhalb der Supportfläche bewegt.

### 3.2.3. 3D Linear Inverted Pendulum Mode (3DLIPM)

Der 3DLIPM kann als vereinfachte Darstellung eines zweibeinigen Roboters angesehen werden. Die Bewegung des Schwerpunktes wird durch ein inverses Pendel mit masselosem Stab und Punktmasse beschrieben.



**Abb. 3.5.:** Veranschaulichung des 3DLIPM in einem dreidimensionalen Koordinatensystem. [Kau11]

Die Punktmasse entspricht dem Gewicht des Roboters und befindet sich auf der Höhe des Roboterschwerpunktes. Vereinfachend wird in der Regel angenommen, dass sich das Pendel in einer Ebene parallel zum Boden mit einer konstanten Höhe  $h$  bewegt. Auf unebenem Terrain kann diese vereinfachende Annahme nicht mehr getroffen werden. Die Bewegung des Pendelkopfes bzw. des Roboterschwerpunktes auf der Ebene parallel zum Untergrund lässt sich durch folgende Gleichungen beschreiben:

$$x(t) = x_0 \cdot \cosh(k \cdot t) + \dot{x}_0 \cdot \frac{1}{k} \cdot \sinh(k \cdot t) \quad (3.2)$$

$$\dot{x}(t) = x_0 \cdot k \cdot \sinh(k \cdot t) + \dot{x}_0 \cdot \cosh(k \cdot t) \quad (3.3)$$

Dabei gilt  $k = \sqrt{\frac{g}{h}}$ .  $x_0$  und  $\dot{x}_0$  stehen für den Abstand des Schwerpunktes zum Ursprung des Pendels und die Geschwindigkeit des Schwerpunktes zum Zeitpunkt  $t = 0$ . Im Gegensatz zum *Cart-table Modell* wird beim 3DLIPM nicht versucht den Schwerpunkt so zu steuern, dass der ZMP in der Supportfläche bleibt. Es wird versucht die Beinbewegungen des Roboters an die pendelförmige Bewegung des Schwerpunktes anzupassen. Bei einem Laufalgorithmus heißt das, dass die Fußpositionen von der Pendelbewegung abhängen. Um eine möglichst stabile Ausgangsposition beim Fußwechsel zu haben, sollte beim Wechsel der Pendelrichtung der Schwerpunkt über dem Zentrum des neuen Supportfußes liegen. Berechnet man aus Gleichung 3.3 die Beschleunigung und setzt diese mit Gleichung 3.2 in die ZMP Gleichung für das *Cart-table Modell* 3.1 ein, sieht man, dass die optimale Position des Pendelursprungs im ZMP liegt. Jeder Schritt sollte den Fuß und somit die Supportfläche also in den Pendelursprung bzw. den ZMP setzen (Herleitung und Beweis s. [Kau11]). Bezogen auf eine Schussbewegung kann beispielsweise die Pendelbewegung des Schwerpunktes nach links ausgenutzt werden, um das rechte Bein zu heben und zu schießen, um in der rückwärtigen Pendelbewegung nach rechts wieder mit dem Schussbein aufzusetzen.

### 3.2.4. Spezifisches dynamisches Modell

Einen weiteren Ansatz für die Regelung eines humanoiden Roboters bietet das roboterspezifische dynamische Modell. Es reduziert den Roboter nicht mehr ausschließlich auf eine Punktmasse im Schwerpunkt, sondern bezieht das gesamte dynamische und statische Verhalten ein. Dies erfordert jedoch Informationen über das dynamische Verhalten der Gelenke und über die mechanischen Eigenschaften der Einzelteile. Über Vorwärtsskinematik und inverse Kinematik kann eine Vorausbestimmung des dynamischen Verhaltens bzw. des ZMP erfolgen.

### 3.2.5. Evaluation der Modelle

Die Regelung über das *Cart-table Modell* erfolgt über eine Anpassung des ZMP, da er innerhalb des vordefinierten Supportbereichs bleiben muss. Es ist jedoch nicht möglich die zukünftige Position des ZMP über das *Cart-table Modell* zu bestimmen. Es kann also nur eine Regelung erfolgen.

Der 3DLIPM Modell beschränkt das mechanische Verhalten des Roboters auf eine Punktmasse im Schwerpunkt, welche sich wie ein Pendel verhält. Wie in Abschnitt 3.2.3 beschrieben, wird bei Bewegungsabläufen in einer flachen Umgebung oft die Vereinfachung gemacht, dass sich der Schwerpunkt immer auf gleicher Höhe in einer Ebene parallel zum Boden bewegt. Abgesehen von der Eliminierung dieser Vereinfachung ist dieses Modell nicht erweiterbar.

Das roboterspezifische dynamische Modell beschreibt die Dynamik des Roboters je nach Genauigkeit der Inertialmatrizen, Massenangaben und Winkelpositionen und deren Ableitungen. Es kann vereinfacht aufgebaut werden und ist danach erweiterbar, bis es das dynamische Verhalten den Anforderungen entsprechend beschreibt.

Die beste Approximation für den ZMP kann das roboterspezifische dynamische Modell liefern. Da für eine Regelung von Bewegungsabläufen, der ZMP von großer Bedeutung ist, führt eine genaue Bestimmung dieses Punktes zu einer gut funktionierenden Regelung. Es wird deshalb im nächstens Kapitel ein roboterspezifisches dynamisches Modell aufgestellt und anschließend auf dem NAO implementiert.

# 4. Roboterspezifisches dynamisches Modell

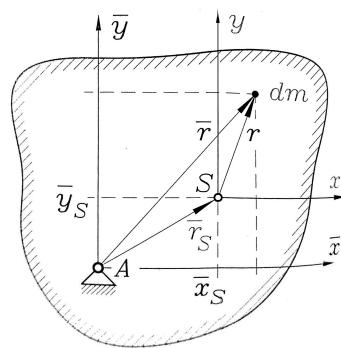
## 4.1. Mechanische Grundlagen

Dieser Abschnitt gibt einen kurzen Überblick über die mechanischen Grundlagen, die notwendig sind, um das im Folgenden erstellte dynamische Modell nachvollziehen zu können. Da sich dieses Modell auf die im Fuß wirkenden Momente während einer Schussbewegung bezieht, sind die in Abschnitt 4.1.1 beschriebenen Massenträgheitsmomente von besonderer Bedeutung.

### 4.1.1. Massenträgheitsmomente und Steineranteile

Massenträgheitsmomente sind ein Maß für die Drehträchtigkeit eines Körpers, um eine bestimmte Achse und werden in der Einheit  $\text{kg}\cdot\text{m}^2$  angegeben. Abgesehen von einer Kugel ist das Massenträgheitsmoment eines jeden Körpers abhängig von der Drehachse. Ein Stab lässt sich z. B. leichter um seine Längsachse drehen als senkrecht zu seiner Längsachse. Daher werden Massenträgheitsmomente stets mit ihrer Drehachse indiziert.

Des Weiteren werden Massenträgheitsmomente in der Regel mit Bezug zum Schwerpunkt oder mit Bezug zu einem Fix- bzw. Drehpunkt des Körpers angegeben. Wird eine Drehung abseits des Schwerpunktes ausgeführt, muss zusätzlich der Satz von Steiner angewandt werden, um das Massenträgheitsmoment mit Bezug zu einem neuen Drehpunkt zu bestimmen (Herleitung s. [DD09]):



**Abb. 4.1.:** Berechnung des Steineranteils. [DD09]

$$I_{zz}^A = I_{zz}^S + \underbrace{m \cdot \bar{r}_S^2}_{\text{Steineranteil}} \quad (4.1)$$

$I^S$  ist hier das Massenträgheitsmoment mit Bezug zum Schwerpunkt und wird zum Steineranteil, der sich aus der Masse des Körpers und dem Quadrat des Abstandes vom Drehpunkt zum Schwerpunkt zusammensetzt, addiert. Der Abstand  $\bar{r}_S$  wird senkrecht zur Drehachse gemessen. Ist das Massenträgheitsmoment mit Bezug zum Schwerpunkt des Körpers gegeben ( $I_{zz}^S$ ), muss der Steineranteil addiert werden. Ist es mit Bezug zu einem Koordinatensystem im Punkt A gegeben ( $I_{zz}^A$ ) und die Drehachse verläuft nicht durch A, sollte das Massenträgheitsmoment zuerst auf den Schwerpunkt bezogen werden. Es wird dazu der Steineranteil vom Körperschwerpunkt zum Punkt A ausgerechnet und von  $I_{zz}^A$  abgezogen.

$$I_{zz}^S = I_{zz}^A - \underbrace{m \cdot \bar{r}_S^2}_{\text{Steineranteil}} \quad (4.2)$$

Wenn man ein Massenträgheitsmoment eines komplexeren Bauteils bestimmen will, lässt sich dies unter anderem durch Aufteilen in kleine Körper mit bekannten Massenträgheitsmomenten und anschließendes Addieren der einzelnen Massenträgheitsmomente erreichen. Zu beachten sind dann jedoch die unterschiedlichen Steineranteile jedes Einzelkörpers, da die Schwerpunkte der Einzelkörper verteilt im Gesamtkörper liegen. Dreht sich ein beliebiger unsymmetrischer Körper nicht um eine seiner Hauptachsen<sup>1</sup>, tritt eine dynamische Unwucht auf, da der Körper danach bestrebt ist seine Rotationsachse in Richtung einer Hauptachse zu ändern. Diese auftretenden Nebenträgheitsmomente werden als Deviationsmomente oder Zentrifugalmomente bezeichnet. Sie werden mit den Indizes  $xy$ ,  $xz$  und  $yz$  gekennzeichnet.

## Inertialmatrizen

Inertialmatrizen beschreiben das rotatorische Trägheitsverhalten eines Körpers um alle Achsen. Sie sind folgendermaßen aufgebaut:

$$I_{xyz}^S = \begin{pmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{pmatrix} \quad (4.3)$$

Auf der Diagonalen stehen dabei die Massenträgheitsmomente um die drei Hauptachsen des Körpers. Auf den Nebendiagonalen sind die Deviationsmomente symmetrisch zur Diagonalen angeordnet. Multipliziert man den Trägheitstensor von rechts mit einem Vektor, in dem die Winkelbeschleunigungen um die  $x$ -,  $y$ -, und  $z$ -Achse eingetragen sind, erhält man das Drehmoment um die jeweilige Achse in Form eines Vektors. In der Dokumentation des NAOs sind die Inertialmatrizen mit Bezug zum körperteiligen Koordinatensystem angegeben.

Unterscheidet sich das körperteilige Koordinatensystem vom Koordinatensystem der Drehachse, muss die Inertialmatrix transformiert werden. Dies erfolgt nach der folgenden Transformationsregel, wobei C die Rotationsmatrix darstellt:

$$I'_{xyz} = C^T I_{xyz} C \quad (4.4)$$

---

<sup>1</sup>Bei einer Drehnung um eine Hauptachse des Körpers ist das Deviationsmoment Null.

Dies wird am besten anhand einer Beispielrechnung deutlich. Diese bezieht sich in ihren Annahmen schon auf ein später auftauchendes Problem (Abschnitt 4.2.2). Angenommen die Inertialmatrix eines Körpers bezüglich seiner Hauptachsen sei mit 4.5 gegeben.

$$I_{xyz}^S = \begin{pmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{pmatrix} \quad (4.5)$$

Es erfolgt eine Drehung um die  $y$ -Achse. Ist dieser Körper dabei um seine eigene  $x$ -Achse gedreht, kann 4.5 mit Hilfe von 4.4 transformiert werden.

$$C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\delta_x) & -\sin(\delta_x) \\ 0 & -\sin(\delta_x) & \cos(\delta_x) \end{pmatrix} \quad (4.6)$$

$$C^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\delta_x) & \sin(\delta_x) \\ 0 & -\sin(\delta_x) & \cos(\delta_x) \end{pmatrix} \quad (4.7)$$

Einsetzen von  $C$ ,  $C^T$  und  $I_{xyz}^S$  in 4.4 liefert:

$$I'_{xyz} = \begin{pmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} \cdot \cos^2(\delta_x) + I_{zz} \cdot \sin^2(\delta_x) & (-I_{yy} + I_{zz}) \cdot \cos(\delta_x) \sin(\delta_x) \\ 0 & (-I_{yy} + I_{zz}) \cdot \cos(\delta_x) \sin(\delta_x) & I_{yy} \cdot \cos^2(\delta_x) + I_{zz} \cdot \sin^2(\delta_x) \end{pmatrix} \quad (4.8)$$

$C$  kann auch Rotationen um mehrere Achsen enthalten.

### 4.1.2. d'Alembertsches Moment

Um die Geschwindigkeit eines Körpers zu erhöhen oder zu verringern, muss er beschleunigt bzw. gebremst werden, wobei das Bremsen einer Beschleunigung entgegen der Bewegungsrichtung entspricht. Betrachtet wird nun ein starres Pendel mit der Länge  $l = 0,5$  m, an dessen Ende eine Kugel der Masse  $m = 2$  kg mit dem Radius  $R = 0,1$  m im Schwerpunkt aufgehängt ist. Im Fixpunkt des Pendels befindet sich ein Antrieb, um das Pendel um die  $y$ -Achse auszulenken. Der Pendelarm wird als masselos angesehen.

Das Massenträgheitsmoment einer Kugel wird Gleichung 4.9 berechnet (s. [DD09]).

$$I_{xx, \text{Kugel}}^S = I_{yy, \text{Kugel}}^S = I_{zz, \text{Kugel}}^S = \frac{2}{5} \cdot m \cdot R^2 = 0,008 \text{ kgm}^2 \quad (4.9)$$

Im nächsten Schritt wird nach Gleichung 4.1 der Steineranteil addiert.

$$I^{\text{Pendel}} = I_{yy, \text{Kugel}}^S + 2 \text{ kg} \cdot (0,5 \text{ m})^2 = 0,508 \text{ kgm}^2 \quad (4.10)$$

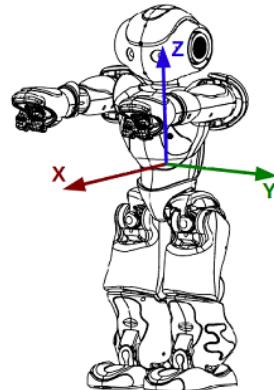
Um die Geschwindigkeit des Pendels zu ändern, muss durch den Antrieb ein Drehmoment im Fixpunkt aufgebracht werden. Dieses berechnet sich nach:

$$M = I \cdot \ddot{\alpha} \quad (4.11)$$

Auf den Antrieb wirkt dann ein gleich großes Moment mit umgekehrtem Vorzeichen, da ein massebehafteter Körper träge ist. Dieses rückwirkende Moment wird als d'Alembertsches Moment bezeichnet. Der Ausgleich des d'Alembertschen Moments findet durch den Kontakt zum festen Untergrund statt. Wäre dieser nicht vorhanden, würde sich der Motor in entgegengesetzter Richtung drehen. Beispielsweise kann man sich einen Hubschrauber, dessen Heckrotor in der Luft ausfällt, vorstellen. Wirkt dem d'Alembertschen Moment, welches auf den Antrieb des Rotors ausgeübt wird, keine Kraft entgegen, drehen sich sowohl Rotor als auch Hubschrauber in entgegengesetztem Drehsinn.

## 4.2. Berechnung der d'Alembertschen Momente im NAO

In diesem Abschnitt wird die Berechnung der d'Alembertschen Momente im NAO besprochen. Da dieses Modell roboterspezifisch ist, wird mit Bezug auf das Koordinatensystem des NAOs gerechnet. Abbildung 4.2 zeigt die *Nullposition* des NAOs. In dieser Position haben die Koordinatensysteme aller Körperteile die gleiche Ausrichtung, wie das eingezeichnete Torso-Koordinatensystem. Im NAO selbst werden standardmäßig alle Positionen der Gelenke relativ zum Torso ausgegeben.



**Abb. 4.2.:** Ausrichtung der Koordinatensysteme in *Nullposition*. [Ald13]

Die für Berechnung benötigten Informationen über Inertialmatrizen, Massen, Schwerpunktpositionen und Winkelstellungsbeschränkungen, befinden sich im Anhang (s. A.).

### 4.2.1. Annahmen

Das Drehmoment berechnet sich über die Ableitung des Drehimpulses nach der Zeit:

$$M = I \cdot \ddot{\varphi} = I \cdot \dot{\omega} = \dot{L} \quad (4.12)$$

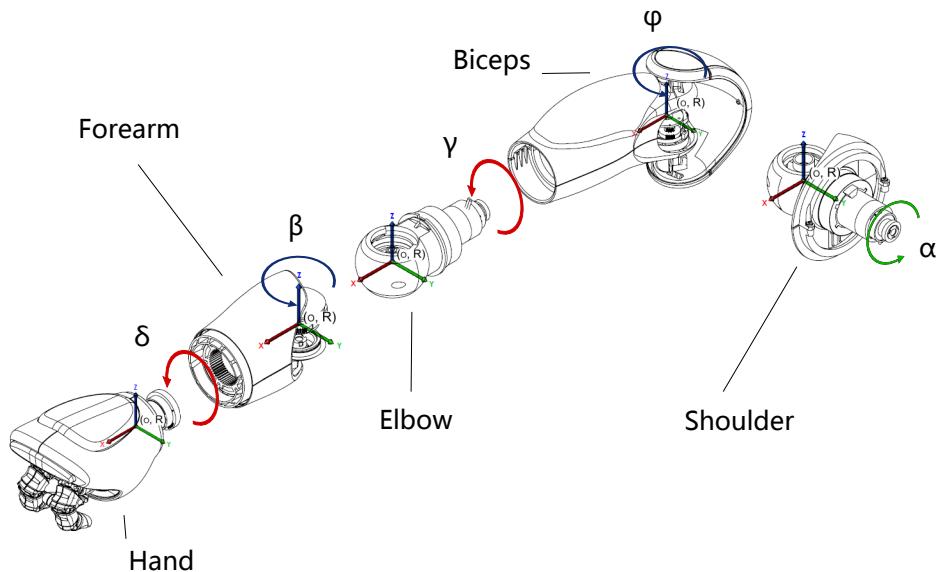
Da die d'Alembertschen Momente von Interesse sind, werden die Drehmomente mit umgekehrtem Vorzeichen berechnet. Die Schwierigkeit in der Berechnung des d'Alembertschen Moments um das Fußgelenk des Standbeins besteht darin, dass

der NAO Roboter sehr viele Freiheitsgrade besitzt. Ziel der Arbeit ist es ein möglichst genaues dynamisches Modell zu erstellen. Es werden dennoch folgende Annahmen getroffen, um keine Berechnungen auszuführen, die nur einen geringfügigen oder keinen Effekt auf das zum Schluss berechnete Moment haben:

1. In diesem Modell wird davon ausgegangen, dass der NAO auf einem Bein steht. Steht der NAO auf zwei Beinen, teilen sich die wirkenden Momente auf beide Fußgelenke auf, sodass die Berechnungen im Fußgelenk des Standbeins verfälscht werden.
2. Da der NAO bei einer Schussbewegung vorwiegend Gefahr läuft nach vorne oder nach hinten zu kippen, sind die d'Alembertschen Momente um die  $y$ -Achse von besonderer Bedeutung. Deshalb werden nur Winkelgeschwindigkeiten um parallele Drehachsen zur  $y$ -Achse des Standfußes in die Berechnung einbezogen.
3. Die Abspreizung der Arme vom Körper werden in den Berechnungen noch nicht berücksichtigt. Eine Abspreizung des Arms hat zur Folge, dass sich ein Körperteil des Arms nicht mehr nur um die körperteileigene  $y$ -Achse dreht, sondern sich zum Teil um seine  $y$ -Achse und zum Teil um seine  $x$ -Achse dreht. Die Steineranteile werden unabhängig davon korrekt berechnet. Während einer Schussbewegung wird der sich bewegende Arm jedoch möglichst nahe und gestreckt am Körper gehalten, um ein möglichst hohes d'Alembertsches Moment mit einer kleinen Bewegung zu erzeugen.
4. Die Deviationsmomente werden gleich Null gesetzt, da die Drehmomente um die  $y$ -Achse untersucht werden und Drehungen um andere Achsen im Vergleich zu den Drehungen um die  $y$ -Achse unbedeutend klein sind.
5. Der Kopf ist starr, da das Trägheitsmoment bei einer Drehung um die eigene  $y$ -Achse sehr gering ist und keine Bewegung des Kopfes beim Schuss nötig ist.

### 4.2.2. Arme

Das d'Alembertsche Moment, welches an der Schulter wirkt und durch die Bewegung des Armes hervorgerufen wird, wird im Folgenden erläutert. Es ist zu beachten, dass mit  $\alpha$  indizierte Trägheitsmomente in Bezug zur Bewegung des Schultergelenks stehen und mit  $\beta$  indizierte Trägheitsmomente in Bezug zur Bewegung des Ellenboengelenks stehen.  $\bar{r}_{a-b}$  gibt den Abstand zwischen Drehachse (a) und dem Ursprung des körperteileigenen Koordinatensystems (b) an.  $\bar{r}_{S, a-b}$  gibt den Abstand zwischen Drehachse (a) und dem Schwerpunkt des Körperteils (b) an.



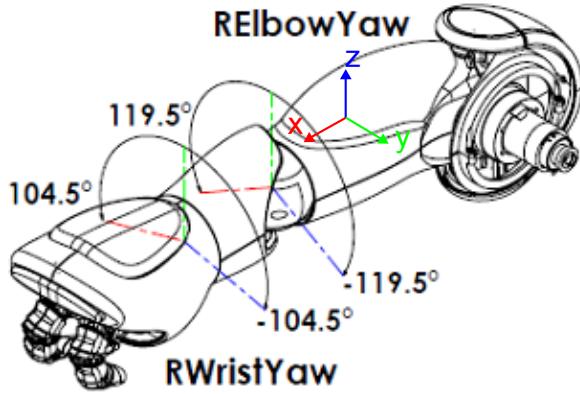
**Abb. 4.3.:** Körperteile des NAO Arms. [Ald13]

Die Drehachse des Oberarms und des Schultergelenks ist zugleich die  $y$ -Achse der körperteileigenen Koordinatensysteme und es kann für diese beiden Teile das von *Aldebaran Robotics* angegebenen Massenträgheitsmoment bezüglich der  $y$ -Achse verwendet werden.

$$I_{yy, \text{Shoulder}}^{\alpha} = I_{yy, \text{Shoulder}}^S + m_{\text{Shoulder}} \cdot \bar{r}_{S, \text{Shoulder-Shoulder}}^2 \quad (4.13)$$

$$I_{yy, \text{Biceps}}^{\alpha} = I_{yy, \text{Biceps}}^S + m_{\text{Biceps}} \cdot \bar{r}_{S, \text{Biceps-Shoulder}}^2 \quad (4.14)$$

Für den Ellenbogen, den Unterarm und die Hand wird das auf den Schwerpunkt bezogene Massenträgheitsmoment benutzt. Je nach Verdrehung des Unterarms oder der Hand um die  $x$ -Achse ändert sich das Massenträgheitsmoment, da nicht mehr nur um die  $y$ -Achse gedreht wird, sondern auch um die  $z$ -Achse. Abbildung 4.4 veranschaulicht dieses Verhalten.



**Abb. 4.4.:** Rechter Arm des NAOs. [Ald13]

In der Abbildung ist der rechte Arm in seiner *Nullposition* dargestellt. D. h., dass alle Gelenkwinkel  $0^\circ$  betragen. Die gestrichelte grüne Linie markiert diese Position. Die gestrichelten blauen und roten Linien stellen die Winkelbegrenzungen für Drehungen dieser Gelenke dar. Das eingezeichnet Koordinatensystem wird hier als fest angenommen, da es nur zur Erläuterung dient. Die den körpereigenen Koordinatensysteme drehen sich bei Bewegungen mit. In der *Nullposition* entspricht die Ausrichtung des eingezeichneten Koordinatensystems der Ausrichtung der körperteileigenen Koordinatensysteme. Bei einer Drehung um das Schultergelenk dreht sich der Unterarm und die Hand nur um die  $y$ -Achse. Ist das *ElbowYaw* Gelenk um  $90^\circ$  verdreht, würde sich der Unterarm mit der Hand um die körperteileigene  $z$ -Achse drehen.  $I_{Elbow}^\alpha$ ,  $I_{Forearm}^\alpha$  und  $I_{Hand}^\alpha$  sind also winkelabhängig. Das gleiche Verhalten überträgt sich demnach auch auf die Schwerpunkte der einzelnen Körperteile. Die Verdrehung des Unterarms und der Hand werden im Folgenden mit  $\gamma$  bzw.  $\delta$  bezeichnet. Die  $\cos^2$  und  $\sin^2$  Terme lassen sich durch das in 4.8 berechnete Beispiel erklären.

$$I_{yy, Elbow}^\alpha = \cos^2(\gamma) \cdot I_{yy, Elbow}^S + \sin^2(\gamma) \cdot I_{zz, Elbow}^S + m_{Elbow} \cdot \bar{r}_{S, Shoulder-Elbow}^2 \quad (4.15)$$

$$I_{yy, Forearm}^\alpha = \cos^2(\gamma) \cdot I_{yy, Forearm}^S + \sin^2(\gamma) \cdot I_{zz, Forearm}^S + m_{Forearm} \cdot \bar{r}_{S, Shoulder-Forearm}^2 \quad (4.16)$$

$$I_{yy, Hand}^\alpha = \cos^2(\gamma + \delta) \cdot I_{yy, Hand}^S + \sin^2(\gamma + \delta) \cdot I_{zz, Hand}^S + m_{Hand} \cdot \bar{r}_{S, Shoulder-Hand}^2 \quad (4.17)$$

Anzumerken ist hier, dass der Abstand zwischen Unterarmschwerpunkt und Drehachse abhängig von der Stellung des Ellenbogengelenkes (*ElbowRoll*) variieren kann. Sofern der Unterarm um  $90^\circ$  verdreht ist, sodass das Ellenbogengelenk eine Rotation um die  $y$ -Achse des in Abbildung 4.4 eingezeichneten Koordinatensystems

vollführt, muss zusätzlich das durch diese Bewegung erzeugte Drehmoment einbezogen werden. Dabei dreht sich der Unterarm um seine eigene  $z$ -Achse, weshalb hier das Massenträgheitsmoment bzgl. der  $z$ -Achse gewählt wird.

$$I_{zz, \text{Forearm}}^{\beta} = I_{zz, \text{Forearm}}^S \quad (4.18)$$

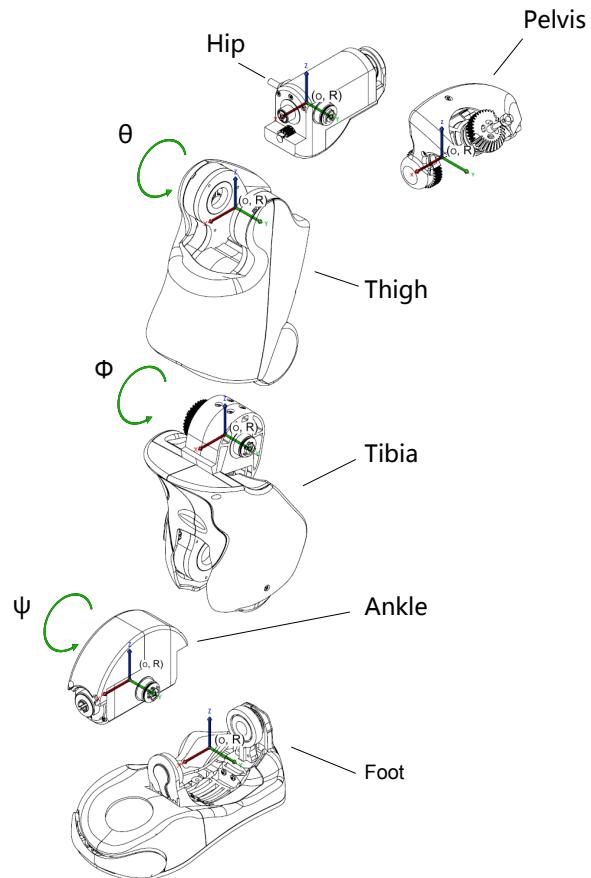
$$\begin{aligned} I_{yy, \text{Hand}}^{\beta} &= \cos^2(\delta) \cdot I_{yy, \text{Hand}}^S + \sin^2(\delta) \cdot I_{zz, \text{Hand}}^S \\ &\quad + m_{\text{Hand}} \cdot \bar{r}_{S, \text{Elbow-Hand}}^2 \end{aligned} \quad (4.19)$$

Das gesamte durch Armbewegungen hervorgerufene d'Alembertsche Moment um die  $y$ -Achse der Schulter ergibt sich durch Addition der zusammengehörigen Trägheitsmomente und anschließende Multiplikation mit der jeweiligen Winkelbeschleunigung.  $\cos \varphi$  und  $\sin \beta$  werden mit  $-\ddot{\beta}$  multipliziert, da die Rotation um die  $y$ -Achse der Schulter mit steigendem  $\varphi$  abnimmt, aber erst bei zunehmendem  $\beta$  ins Gewicht fallen darf, da der Unterarm sonst senkrecht zur Drehachse (Schulter) rotiert.

$$\begin{aligned} M_{y, \text{Arm}} &= (I_{yy, \text{Shoulder}}^{\alpha} + I_{yy, \text{Biceps}}^{\alpha} + I_{yy, \text{Elbow}}^{\alpha} + I_{yy, \text{Forearm}}^{\alpha} \\ &\quad + I_{yy, \text{Hand}}^{\alpha}) \cdot (-\ddot{\alpha}) \\ &\quad + (I_{yy, \text{Forearm}}^{\beta} + I_{yy, \text{Hand}}^{\beta}) \cdot (-\ddot{\beta}) \cdot \cos(\varphi) \cdot \sin(\beta) \end{aligned} \quad (4.20)$$

### 4.2.3. Schussbein

Das durch die Schussbewegung erzeugte d'Alembertsche Moment berechnet sich analog zum d'Alembertschen Moment an der Schulter. Das Beckengelenk und das daran befestigte Hüftgelenk können keine Drehung um die  $y$ -Achse vollführen. Sie sind demnach bei der Schussbewegung im Schussbein nicht weiter von Bedeutung. Die Indizierung mit  $\theta$ ,  $\phi$  und  $\psi$  bezieht sich hier auf die Drehungen des Hüft-, Knie- und des Fußgelenks. Die Verwendung und Indizierung von  $\bar{r}$  wird aus Abschnitt 4.2.2 übernommen.



**Abb. 4.5.:** Körperteile des NAO Beins. [Ald13]

Der Oberschenkel dreht sich um seine eigene  $y$ -Achse. Diese ist gleichzeitig die  $y$ -Achse des Hüftgelenks, weshalb der Index *Hip* für die Drehung um das Hüftgelenk steht.

$$I_{yy, \text{Thigh}}^\theta = I_{yy, \text{Thigh}}^S + m_{\text{Thigh}} \cdot \bar{r}_{S, \text{Hip-Thigh}}^2 \quad (4.21)$$

Der Abstand des Schienbeinschwerpunktes ist von der Stellung des Kniegelenks abhängig.

$$I_{yy, \text{Tibia}}^\theta = I_{yy, \text{Tibia}}^S + m_{\text{Tibia}} \cdot \bar{r}_{S, \text{Hip-Tibia}}^2 \quad (4.22)$$

$$I_{yy, \text{Ankle}}^\theta = I_{yy, \text{Ankle}}^S + m_{\text{Ankle}} \cdot \bar{r}_{S, \text{Hip-Ankle}}^2 \quad (4.23)$$

$$I_{yy, \text{Foot}}^\theta = I_{yy, \text{Foot}}^S + m_{\text{Foot}} \cdot \bar{r}_{S, \text{Hip-Foot}}^2 \quad (4.24)$$

Die Trägheitsmomente in Bezug auf die Drehung des Kniegelenks lauten:

$$I_{yy, \text{Tibia}}^\phi = I_{yy, \text{Tibia}} \quad (4.25)$$

$$I_{yy, \text{Ankle}}^\phi = I_{yy, \text{Ankle}}^S + m_{\text{Ankle}} \cdot \bar{r}_{S, \text{Knee-Ankle}}^2 \quad (4.26)$$

$$I_{yy, \text{Foot}}^\phi = I_{yy, \text{Foot}}^S + m_{\text{Foot}} \cdot \bar{r}_{S, \text{Knee-Foot}}^2 \quad (4.27)$$

Sowohl Fußgelenk als auch Fuß vollführen eine Drehung um ihre eigene  $y$ -Achse. Daher ist es hier sinnvoll das von Aldebaran Robotics angegebene Massenträgheitsmoment mit Bezug zum körpereigenen Koordinatensystem zu verwenden.

$$I_{yy, \text{Ankle}}^\psi = I_{yy, \text{Ankle}} \quad (4.28)$$

$$I_{yy, \text{Foot}}^\psi = I_{yy, \text{Foot}} \quad (4.29)$$

Durch Addition der zusammengehörigen Trägheitsmomente und Multiplizieren mit der jeweiligen Winkelbeschleunigung erhält man das durch die Bewegung des Schussbeins verursachte d'Alembertsche Moment.

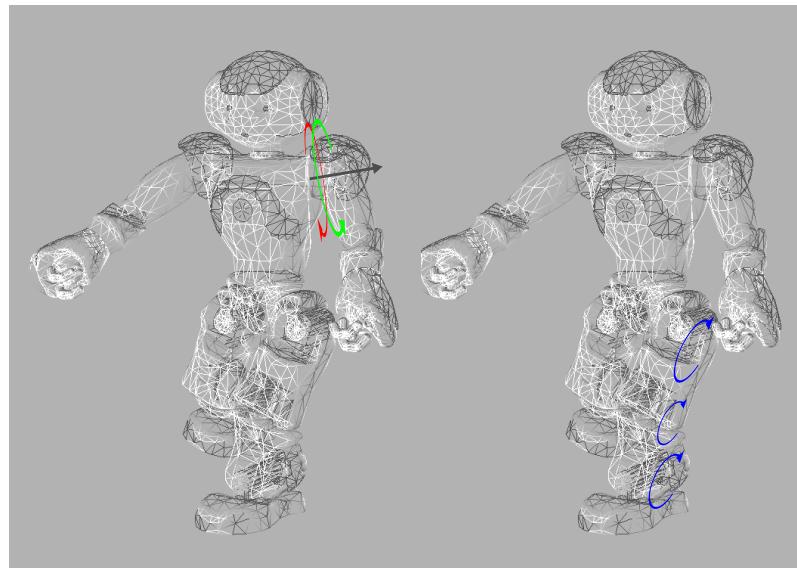
$$\begin{aligned} M_{y, \text{Leg}} = & (I_{yy, \text{Thigh}}^\theta + I_{yy, \text{Tibia}}^\theta + I_{yy, \text{Ankle}}^\theta + I_{yy, \text{Foot}}^\theta) \cdot (-\ddot{\theta}_{\text{Leg}}) \\ & + (I_{yy, \text{Tibia}}^\phi + I_{yy, \text{Ankle}}^\phi + I_{yy, \text{Foot}}^\phi) \cdot (-\ddot{\phi}_{\text{Leg}}) \\ & + (I_{yy, \text{Ankle}}^\psi + I_{yy, \text{Foot}}^\psi) \cdot (-\ddot{\psi}_{\text{Leg}}) \end{aligned} \quad (4.30)$$

Grundsätzlich entsteht durch das Abspreizen eines Beins, die im vorigen Abschnitt erwähnte Situation, welche dazu führt, dass das auf das Hüftgelenk wirkende Massenträgheitsmoment nicht mehr ausschließlich durch  $I_{yy}$  beschrieben wird. Des Weiteren wird nicht mehr ausschließlich eine Drehung um die  $y$ -Achse vollzogen. Dieser Winkel ist bei einer normalen Schussbewegung relativ gering, sodass er vernachlässigt wird.

### 4.3. Moment auf Grund von Spiel in den Gelenken

Kein Gelenk im NAO ist perfekt starr. Ursache dafür kann zum einen Verschleiß an den Zahnrädern sein, zum anderen kann es auch an der Elastizität der Bauteile liegen. Viele Bauteile des NAOs sind aus Kunststoff, weil Kunststoff günstiger und leichter ist. Zudem leitet er keinen Strom. Ein weiterer Grund kann

auch eine beabsichtigte lose Aufhängung einiger Gelenke sein, um Stöße abzufangen und somit Schäden an empfindlichen Teilen innerhalb der Gelenke wie z. B. Sensoren, zu vermeiden. Es kann demnach nicht von einer verlustfreien Übertragung der d'Alembertschen Momente über mehrere Gelenke ausgegangen werden. Das Spiel in den Gelenken ist jedoch sehr klein, sodass sich die Reibung nur geringfügig auf diesen Verlust auswirkt. Die Auswirkungen des Spiels werden anhand eines Beispiels erläutert (s. Abb. 4.6)



**Abb. 4.6.:** Auswirkung des Spiels in den Gelenken bei Bewegung des Arms.

Der NAO steht auf einem Bein und bewegt seinen linken Arm in einer beschleunigten Bewegung in positiver Drehrichtung (mit grünem Momentenpfeil gekennzeichnet). Auf Grund des in Abschnitt 4.1.2 beschriebenen mechanischen Verhaltens tritt ein entgegengesetztes d'Alembertsches Moment an der Schulter des Torsos auf, welches in rot dargestellt ist. Würde eine verlustfreie Übertragung des d'Alembertschen Momentes möglich sein, würde es in den Fuß weitergeleitet und dann durch eine Gegenkraft im Boden ausgeglichen werden. Tatsächlich dreht sich der Torso um das Hüftgelenk in kleinem Maße nach hinten. Der gleiche Vorgang wiederholt sich folglich für das Knie und das Fußgelenk, was durch die blauen Momentenpfeile dargestellt ist. Diese Bewegung führt dazu, dass der NAO nach hinten kippt. Durch die Bewegung der gesamten Masse des Körpers entsteht also ein ausschlaggebendes Moment, welches abgefangen werden muss und demnach mit in die Berechnung des ZMP einfließen sollte. Da die gesamte Masse des NAOs beschleunigt wird, ist es praktikabel den eingebauten Beschleunigungssensor im Torso des NAOs zu verwenden. Es wird davon ausgegangen, dass der NAO sich in kleinem Maße durch das Spiel bewegt. Daraus wird ausschließlich der Abstand vom Fußgelenk zum Schwerpunkt in  $z$ -Richtung betrachtet und die Orientierung des Accelerometers außer Acht gelassen. Nach dem zweiten Newtonschen Axiom ergibt sich das resultierende Moment:

$$M_{y, Body} = -m_{NAO} \cdot \ddot{x} \cdot z_{S, NAO} - m_{NAO} \cdot \ddot{z} \cdot x_{S, NAO} \quad (4.31)$$

Das berechnete Moment geht negativ ein, um im späteren Verlauf dieser Arbeit die Drucksensoren in den Fußsohlen als Referenz verwenden zu können.

## 4.4. Statisches Moment

Auch wenn ein Modell für die dynamische Balance gewählt wird, muss die statische Balance weiterhin beachtet werden. Es ist daher wichtig, das statische Moment mit in die Berechnung des Gesamtmomentes einfließen zu lassen. Das in dieser Arbeit erstellte Modell betrachtet nur die um die  $y$ -Achse wirkenden Momente.

$$M_{y, stat.} = -x_{CoM-Foot} \cdot m_{NAO} \cdot g \quad (4.32)$$

$m_{NAO}$  entspricht hier der Gesamtmasse des NAOs abzüglich des Gewichts eines Fußes mit Gelenk:  $m_{NAO} = 4,78$  kg.  $g$  ist die Erdbeschleunigung mit einem Wert von  $g = 9,81 \frac{\text{m}}{\text{s}^2}$ . Der Abstand des Fußkoordinatensystems zum Schwerpunkt geht negativ in die Berechnung ein, da das vom Boden auf den Fuß wirkende Moment angenähert werden soll, wobei die Drucksensoren in den Füßen im späteren Verlauf als Referenz dienen.

## 4.5. Zusammenfassung

Aus den beschriebenen Zusammenhängen und den aufgestellten Momenten lässt sich das gesamte dynamische Modell für die im Fußgelenk des Standbeins wirkenden Momente aufstellen. Es setzt sich aus der Addition folgender Terme zusammen:

1.  $M_{y, LArm}$  [4.20]
2.  $M_{y, RArm}$  [4.20]
3.  $M_{y, Leg}$  [4.30]
4.  $M_{y, Body}$  [4.31]
5.  $M_{y, stat.}$  [4.32]

Dieses Prinzip lässt sich auf die Drehung um die  $x$ - oder  $z$ -Achse übertragen. Zusätzlich können dann noch die Deviationsmomente berücksichtigt werden.

# 5. Implementierung

In diesem Kapitel werden zunächst die Module beschrieben, welche für die Implementierung einer Schussbewegung, eines dynamischen Modells und einer Regelung, die auf dem ZMP basiert, notwendig sind. Darauf wird die Prozessarchitektur erläutert.

## 5.1. Verwendete Module

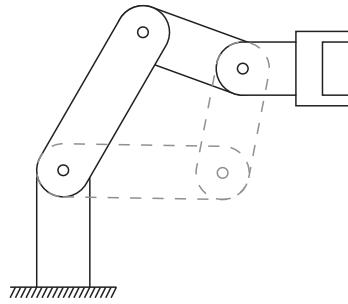
Die folgende Auflistung und Erklärung der verwendeten Module besteht zum einen aus Modulen, die als Grundlage für die Berechnung dienen, und zum anderen aus Modulen, die für die Momentenberechnung und Regelung erstellt und implementiert werden mussten. Die Module *Blackboard*, *ForwardKinematics*, *InverseKinematics*, *Com* und *KinematicMatrix* sind zu Beginn dieser Arbeit schon vorhanden gewesen (Entwicklung und Implementierung s. [Kau11]).

### 5.1.1. Blackboard

Das *Blackboard* dient als Speicherschnittstelle für Sensordaten und kinematische Informationen, wie z. B. der aktuellen Position des Schwerpunktes oder anderer Körperteile. Zudem können auch Roboterkonstanten wie z. B. das Gewicht einzelner Körperteile oder maximale Winkelstellungen der Gelenke über das *Blackboard* abgerufen werden. Es wird in jedem Zyklus mit den aktuellen Werten beschrieben, sodass man einen einfachen und zentralen Zugriff auf diese Informationen hat.

### 5.1.2. ForwardKinematics

Das Modul *ForwardKinematics* berechnet bei Eingabe von Winkelstellungen einer Aktorkette die translatorische Position und Orientierung von Gelenken relativ zum Koordinatensystem des Torsos. Es ist zum einen auf diese Weise möglich die aktuelle Position eines Gelenks abzufragen und zum anderen zukünftige Positionen eines Gelenks vorauszuberechnen. Im ersten Fall liest *ForwardKinematics* die aktuellen Winkelpositionen aus dem *Blackboard* aus und gibt eine kinematische Matrix zurück, die aus einer  $3 \times 3$  Rotationsmatrix und einem  $3 \times 1$  Positionsvektor besteht. Für die Erstellung dieser Matrizen ist bereits ein weiteres Modul Namens *KinematicMatrix* vorhanden.



**Abb. 5.1.:** Die Endposition des Krans ist durch zwei verschiedene Gelenkpositionen erreichbar. [Kau11]

### 5.1.3. InverseKinematics

Die inverse Kinematik berechnet invers zur Vorwärtsskinematik die Winkelstellungen einer Gelenkkette bei Eingabe einer kinematischen Matrix, wie sie in *ForwardKinematics* beschrieben wurde. So ist es möglich, bestimmte Positionen durch die einfache Änderung der kartesischen Koordinaten des Positionsvektors anzufahren. Die inverse Kinematik ist jedoch nicht eindeutig, denn die Positionsangabe bezieht sich immer auf das Ende einer Aktorkette wie z. B. den Fuß oder die Hand. Es kann der Fall eintreten, dass es zwei verschiedene Kombinationen an Gelenkwinkelstellungen gibt, die das Ende der Aktorenkette in die gleiche Position bringen. Abbildung 5.1 veranschaulicht dieses Problem.

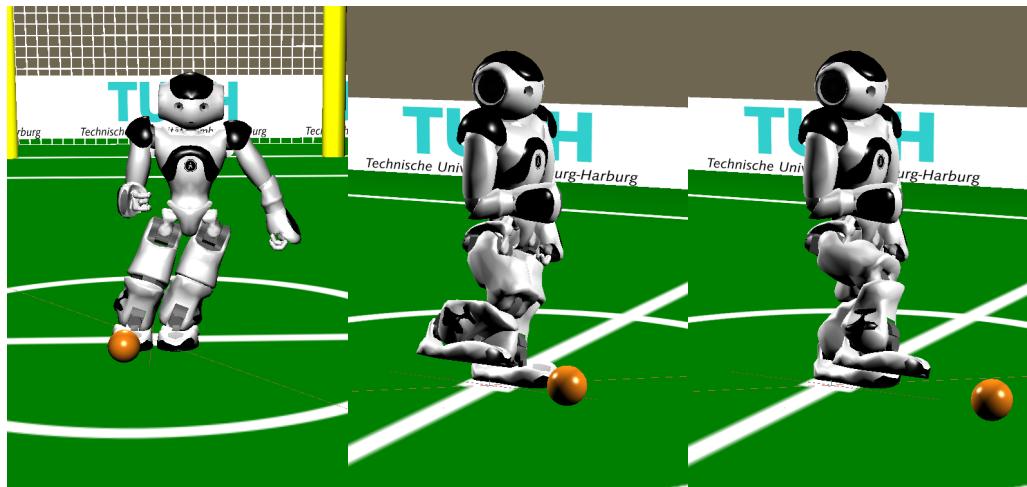
Die reale Bewegung der Aktoren ist dann nicht vorauszusehen. humanoide Roboter haben jedoch in der Regel Gelenkwinkelbeschränkungen, die den menschlichen Beschränkungen entsprechen. Somit treten solche Sonderfälle bei normalen Bewegungsabläufen sehr selten auf.

### 5.1.4. Com

*Com* (*Center of mass*) nutzt das Modul *ForwardKinematics* und das *Blackboard*, um zu Beginn jedes Zyklus die Schwerpunktposition des gesamten Körpers auszurechnen und im *Blackboard* abzulegen. Die Position wird im Koordinatensystem des Torsos beschrieben. Es ist auch möglich die Schwerpunktposition einzelner Gelenkketten zu bestimmen.

### 5.1.5. BodyPartCom

Das Modul *BodyPartCom* stellt eine kleine Ergänzung zum *Com* Modul dar. Es berechnet über die kinematischen Matrizen und die Schwerpunktangaben jedes Körperteils den senkrechten Abstand zwischen Drehachse und Körperteil. Als Drehachsen werden bisher nur die Schultergelenke, die Ellenbogengelenke, die Gelenke an den Oberschenkeln und die Kniegelenke verwendet, da diese für die Momentenberechnung um die *y*-Achse von größter Bedeutung sind.



**Abb. 5.2.:** Simulation der Enzustände der drei Phasen in Webots.

### 5.1.6. VelocitiesAccelerations

Für das Aufstellen eines dynamischen Modells ist es nötig, neben den Informationen über die Position der Gelenke auch Informationen über deren Geschwindigkeit und Beschleunigung haben. Das Modul *VelocitiesAccelerations* berechnet diese, indem es zyklisch die aktuellen Gelenkposition aus dem *Blackboard* ausliest und über den Differenzenquotienten die Winkelgeschwindigkeit bestimmt und im *Blackboard* ablegt. Die aktuellen Winkelpositionen und Geschwindigkeiten werden in einer globalen Variable abgespeichert, um sie im nächsten Zyklus im Differenzenquotienten verwenden zu können. Die Zyklusdauer  $\Delta t$  beträgt 10 ms. Die Berechnung der Winkelbeschleunigungen erfolgt analog zu den Winkelgeschwindigkeiten über den Differenzenquotienten.

### 5.1.7. KickMotion

Die Schussbewegung wird in drei Phasen aufgeteilt. In der ersten Phase bringt sich der NAO in eine stabile Ausgangsposition. In der zweiten Phase wird das Schussbein nach hinten ausgeholt und in der dritten Phase erfolgt der Schuss. Alle Bewegungen werden über Manipulation des Positionsvektors der Gelenke erreicht. Die dadurch veränderte kinematische Matrix wird über *InverseKinematics* in die entsprechenden Gelenkwinkelstellungen umgerechnet. Die Gelenkwinkel können dann als Befehl an den NAO gesendet werden.

#### Phase 1

Um beim Anheben des Schussbeins nicht zur Seite zu fallen, muss die Projektion des Gesamtschwerpunktes in die Supportfläche der zukünftigen Haltung gebracht werden. Dies wird über Schwerpunktverschiebung erreicht. In einem iterativen Berechnungsverfahren wird zunächst über die Module *Com* und *ForwardKinematics* der Schwerpunkt des Körpers virtuell über den Ursprung des Koordinatensystems im Standfuß geschoben. Der Schwerpunkt hat sich zu Anfang der Bewegung in der

Mitte des Torsos befunden, was nach der Verschiebung allerdings nicht mehr gültig wäre. Es ist eine erneute Berechnung des Schwerpunktes notwendig, worauf eine erneute virtuelle Verschiebung über *ForwardKinematics* erfolgt. Nach vier Zyklen ist eine ausreichend hohe Genauigkeit ( $< 1 \text{ mm}$ ) erreicht. Es folgt die Verschiebung des Schwerpunktes über den Ursprung des Standfuß-Koordinatensystems. Der Arm auf der Seite des Schussbeins wird zusätzlich angewinkelt, um den Schwerpunkt möglichst weit auf die Körperseite des Standbeins zu bringen. Je weiter der Schwerpunkt auf der Seite des Schussbeins ist, desto weiter muss der Oberkörper über das Standbein gefahren werden. Dies führt zu einer sehr schrägen Haltung, die Instabilität hervorruft.

## Phase 2

Der NAO hebt das Bein an und holt mit dem Bein leicht nach hinten aus, um einen längeren Beschleunigungsweg zu erhalten. Der Ball kann dann mit einer höheren Geschwindigkeit getroffen werden, sodass der Schuss insgesamt stärker ist.

## Phase 3

Das Bein wird mit möglichst hoher Geschwindigkeit nach vorne gefahren, sodass eine Schussbewegung entsteht. Durch die hohe Beschleunigung entsteht eine Instabilität im Standfuß.

### 5.1.8. Torque

*Torque* ist dafür zuständig aus den kinematischen Informationen das Moment auszurechnen, welches am Fußgelenk des NAOs angreift. Es werden dazu die Winkelstellungen und -beschleunigungen, sowie die Abstände der Körperteile zu den Drehachsen eingelesen. Zusätzlich sind noch die Massenträgheitsmomente und Gewichtsangaben nötig.

Das Moment, welches durch das Spiel der Gelenke verursacht wird, wird über das Accelerometer berechnet. Dabei wird über die Beschleunigung in  $x$ -Richtung, das Gewicht des NAOs und den Abstand des Schwerpunktes vom Fuß das wirkende Moment berechnet. Die Sensorwerte sind dabei über das *Blackboard* abrufbar. Anschließend wird die Ausgabe mit einem Tiefpassfilter erster Ordnung gefiltert ( $\tau = 0,65$ ). Das durch die Arme und Beine verursachte d'Alembertsche Moment wird grundsätzlich über die Winkelbeschleunigungen, Massenträgheitsmomente und Abstände von Drehachse zum Körperteil-Schwerpunkt berechnet. Des Weiteren sind die Winkelstellungen bestimmter Gelenke nötig, da sie die Ausrichtung eines Körperteils relativ zur Drehachse beeinflussen können. Diese Momente werden mit einem Tiefpassfilter erster Ordnung gefiltert ( $\tau = 0,55$ ).

Das statische Moment lässt sich über das *Com* Modul und die kinematische Matrix des Fußes bestimmen. Anschließend werden alle Momente addiert und ausgegeben.

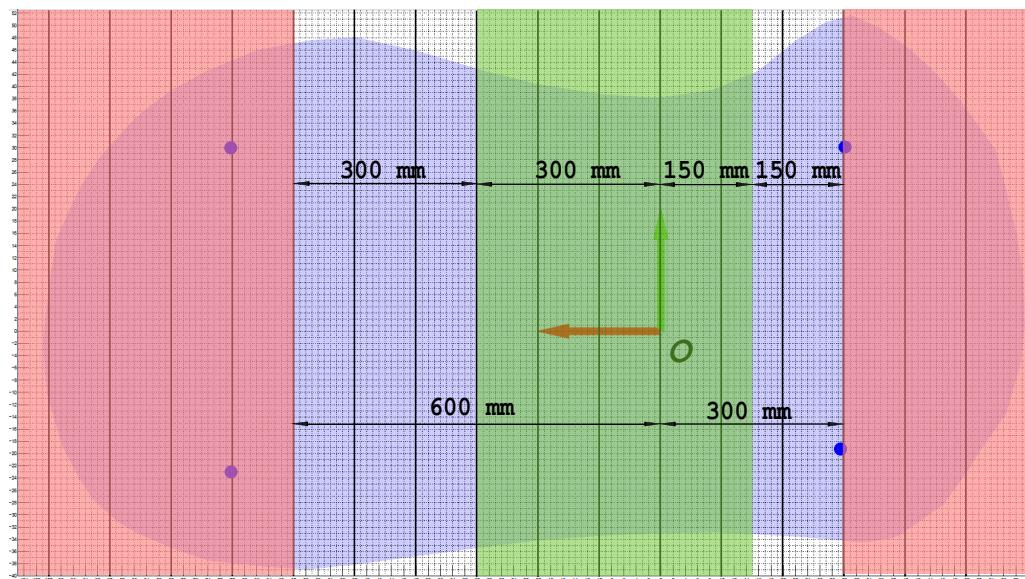
### 5.1.9. Control

Der Momentenausgleich wird ausschließlich über die Bewegung der Arme und die damit verbundenen d'Alembertschen Momente und Schwerpunktverschiebung geregelt. Es wird immer nur der Arm auf der Seite des Standbeins bewegt.

$$ZMP = -\frac{M_{gesamt}}{m_{NAO} \cdot g} \quad (5.1)$$

Der ZMP berechnet sich über das im Fuß wirkende Moment, welches vom Modul *Torque* geliefert wird, wobei zu beachten ist, dass das Moment negativ in die weiteren Berechnungen einfließt. Es spiegelt, das vom Boden auf den Fuß wirkende Moment wieder, welches bei negativem Vorzeichen ein Fallen nach vorne impliziert, was wiederum zu einem postiven  $x$ -Wert des ZMP führen muss. Es wird vereinfachend davon ausgegangen, dass die Reaktionskraft des Bodens immer der Gewichtskraft des NAOs mit  $m_{NAO} = 5,08$  kg entspricht.  $g$  entspricht der Erdbeschleunigung mit  $9,81 \frac{\text{m}}{\text{s}^2}$ .

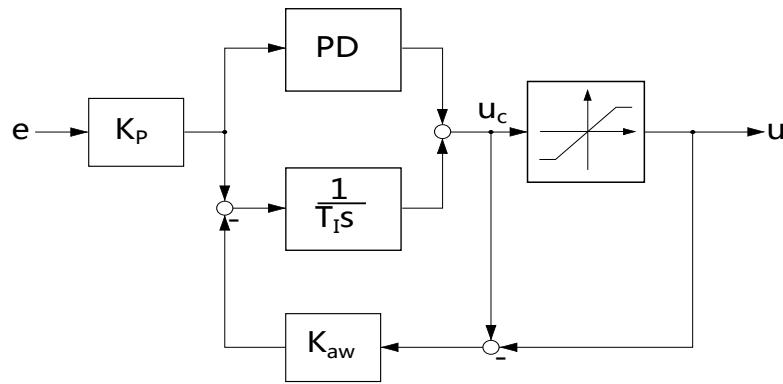
Nach der Berechnung des ZMP wird eine Fallunterscheidung getroffen. Bei negativem ZMP werden die Arme nach vorne bewegt und bei positivem ZMP nach hinten. Ist der ZMP jedoch sehr nahe am Koordinatenursprung, wird der Fehler gleich Null gesetzt, damit die Arme nicht unnötig in Situationen bewegt werden, in denen der Roboter auch ohne Regelung nicht ernsthaft gefährdet ist, zu fallen. Zudem würde ohne den Bereich, in dem der Fehler gleich Null gesetzt wird, die Gefahr vergrößert werden, dass der NAO durch seine eigene Bewegung in eine Dauerschwingung gerät. Ergänzend werden auch maximale Grenzen festgelegt, an denen der NAO die stärkste Armbewegung zum Gegensteuern verwenden soll. Abbildung 5.3 veranschaulicht diese Grenzen im Fuß.



**Abb. 5.3.:** Ansicht des linken Fußes von unten mit Position der Fußsensoren und den unteren und oberen Fehlergrenzen.

Im grünen Bereich wird der Fehler gleich Null gesetzt. An der Grenze zum roten Be-

reich erfolgt der maximale Ausschlag des Arms. Als Stellgröße wird also folglich der Abstand zu den Grenzen, an denen der Fehler gleich Null gesetzt wird, verwendet. Die betragsmäßigen Unterschiede der Grenzen kommen dadurch zustande, dass sich das Fußgelenk und das Koordinatensystem des Fußes nicht in der Mitte des Fußes befinden. Demnach kann der ZMP im negativen Bereich nicht so groß werden wie im positiven Bereich. Dies erklärt unter anderem, warum der NAO dazu neigt, eher nach hinten als nach vorne zu fallen. Die eigentliche Regelung erfolgt über einen diskreten PID-Regler mit *Anti-Windup* Konfiguration. In Abbildung 5.3 ist die Konfiguration als Blockschaltbild dargestellt.



**Abb. 5.4.:** Blockschaltbild eines PID-Reglers mit *Anti-Windup* (nach [Wer11]).

Die *Anti-Windup* Konfiguration ist nötig, da es eine Stellgrößenbegrenzung gibt. Im Allgemeinen erhält der Regler als Führungsgröße den ZMP und berechnet daraus den entsprechenden Fehler. Die Winkelstellungen des Arms sind jedoch anschaulicher für die Erklärung und berechnen sich aus dem ZMP wie folgt. Tritt der ZMP aus dem Bereich, in dem der Fehler zu Null gesetzt wird, berechnet der Regler den Fehler im Bezug zu den Grenzen dieses Bereichs. Für den vorderen Fußbereich wird also der maximale Ausschlag des Arms nach hinten bei einem Fehler von 3 cm ausgeführt. Im hinteren Fußbereich wird der maximale Ausschlag des Arms nach vorne bei einem Fehler von 1,5 cm erreicht. In der *Nullposition* sind die Arme des NAOs gerade nach vorne gestreckt. Ausgehend davon ist die Gelenkwinkelbegrenzung für eine positive Drehung 119,5 °. Die Gelenkwinkelbegrenzung für eine negative Drehung ist analog dazu -119,5 °. In der Regelung wird jedoch 0 ° als Stellgrößenbegrenzung vorgesehen, da bei ausgestreckten Armen das durch die Armbewegung verursachte, statische Moment sein Maximum erreicht. Ohne die *Anti-Windup* Konfiguration, würde der I-Anteil des Reglers bei sehr großen Fehlern dafür sorgen, dass entweder Winkel angefahren werden sollen, die in diesem Zusammenhang keinen Nutzen haben oder durch die Gelenkwinkelbegrenzungen nicht anfahrbar sind. Durch die Integration der Fehler sorgt der I-Anteil im Folgeschritt dafür, dass dieser große Fehler in den Berechnungsprozess der neuen Regelgröße mit einfließt. Dies hat wiederum zur Folge, dass der große Fehler, der durch die Stellgrößenbegrenzung nicht direkt ausgeglichen werden konnte, erst über mehrere Schritte ausgeglichen wird. Der Regler wird in Folge dessen reaktionsträger.

## 5.2. Prozessarchitektur

Im Folgenden wird in einem Schaubild dargestellt, wie die verwendeten Module interagieren, um das dynamische Modell des NAOs aufzustellen und es anschließend zur Regelung während einer Schussbewegung zu nutzen.

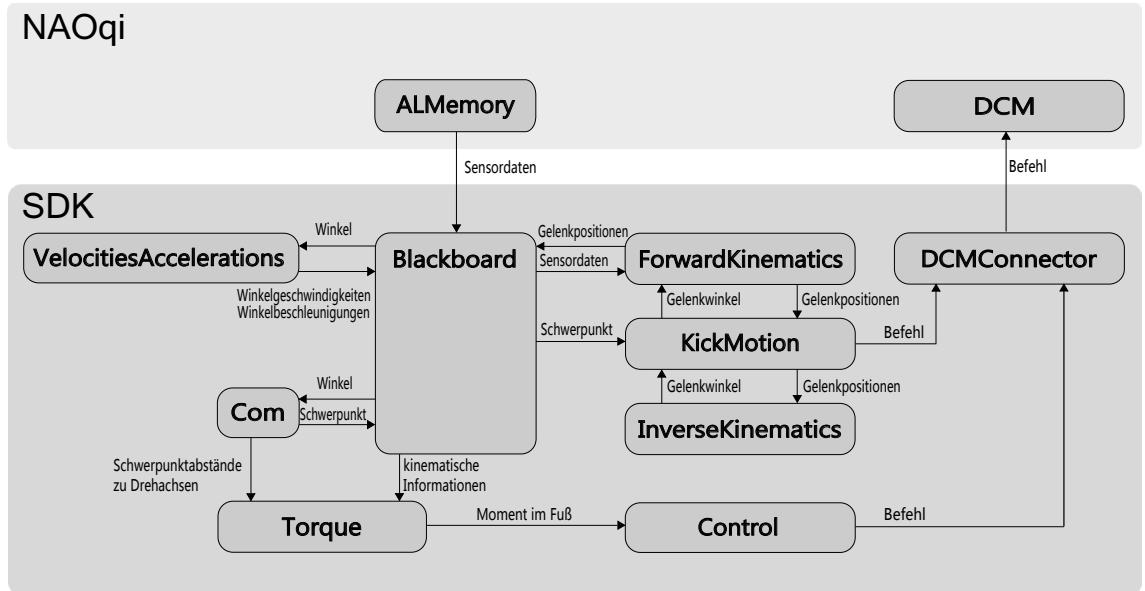


Abb. 5.5.: Vereinfacht dargestellte Prozessarchitektur.

Die Module *KinematicMatrix* und *BodyPartCom* sind auf Grund der Übersichtlichkeit nicht mit in Abbildung 5.5 einbezogen worden. *KinematicMatrix* stellt eine bestimmte Form von kinematischen Informationen dar, die im Allgemeinen auch im Blackboard abgespeichert sind. *BodyPartCom* ist nur eine andere Ausgabeform von Daten, die durch *Com* bereit gestellt werden.

Die Funktion zur Regelung aus dem Modul *Control* muss zyklisch aufgerufen werden, um den aktuellen Fehler zu berechnen. Dazu ist in der Simulation ein Aufruf der Funktion in jedem Zyklus des *Controllers* nötig. Auf dem NAO hingegen muss diese Funktion an den DCM Zyklus angebunden werden. Der Block aus *ForwardKinematics*, *KickMotion* und *InverseKinematic* stellt das Einholen der aktuellen Gelenkpositionen, die Änderung der Position des Schwerpunktes und des Schussbeins im Translationsvektor und das anschließende Umrechnen in die benötigten Gelenkwinkel dar.

# 6. Evaluation

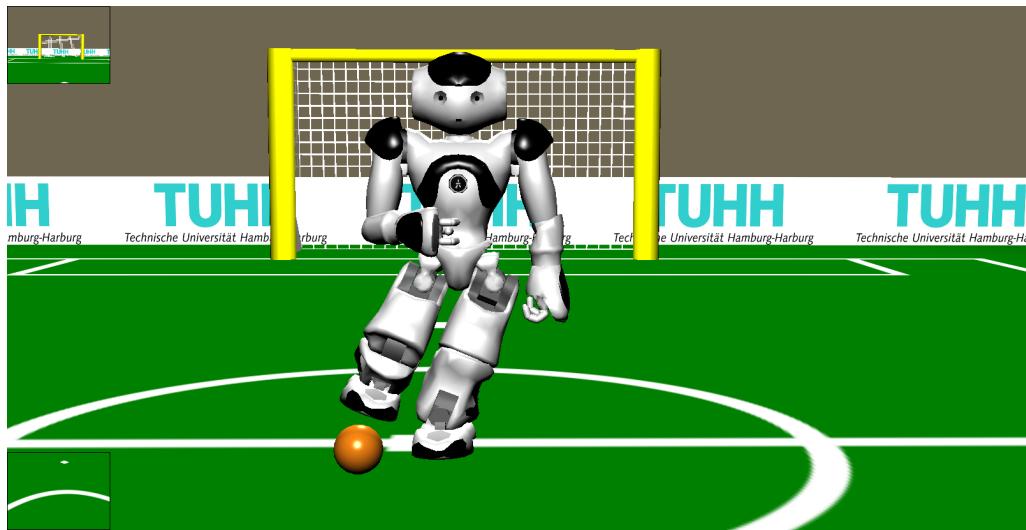
In diesem Kapitel werden die Messergebnisse des ZMP und der wirkenden Momente aus der Simulation und vom Realversuch gegenübergestellt und ausgewertet. Zudem wird auch überprüft, ob die in Abschnitt 5.1.9 angesprochene Regelung zur Stabilisierung der Schussbewegung beiträgt. Der ZMP hängt direkt mit dem Momentenverlauf im Fußgelenk zusammen. Es werden deshalb hauptsächlich die Verläufe des ZMP verglichen, da der Regler den ZMP als Stellgröße verwendet. Für den ZMP wird jeweils ein Plot mit und ohne Regelung aufgestellt, wobei stets der über die Fußsensoren berechnete ZMP als Referenz aufgetragen wird. Um eine Aussage darüber treffen zu können, ob das gewählte roboterspezifische Modell dem in Abschnitt 3.2.2 vorgestellten *Cart-table Modell* überlegen ist, wird zusätzlich der über das *Cart-table Modell* berechnete ZMP aufgetragen. Des Weiteren wird zur besseren Auswertung des Reglers auch der Verlauf der Stellgrößen im Regler betrachtet. Eine kurze Auflistung der Reglerparameter befindet sich im Anhang (s. A.4).

Der NAO weist innerhalb der Simulation ein deutlich abweichendes Verhalten zur Realität auf. So müssen viele Bewegungsabläufe auf dem echten NAO langsamer ablaufen, als in der Simulation, um ein Kippen zu vermeiden. Es werden im Simulationsabschnitt 6.1 daher Parameter gewählt, die nur in der Simulation zu einem stabilen Verhalten des NAOs führen und danach Parameter verwendet, die auch auf dem NAO zu einem stabilen Verhalten führen.

Da nur das Moment um die  $y$ -Achse betrachtet wird, ist mit ZMP stets der  $x$ -Wert des ZMP gemeint. Die gesamten Tests werden in dieser Arbeit mit dem linken Bein als Standbein und dem rechten Bein als Schussbein durchgeführt. Die Messaufnahmen beginnen, wenn das rechte Bein angehoben wird und dauern in der Simulation 2000 ms und im Realversuch 2500 ms. Im Realversuch wird eine leicht erhöhte Messdauer gewählt, da sich im Verlauf mehrerer Messungen gezeigt hat, dass der NAO in der Realität länger braucht, um sich auszuschwingen. Die Regelung beginnt mit dem Schuss und läuft 300 ms, wobei der eigentliche Schuss eine Laufzeit von 20 ms hat.

## 6.1. Simulation

Wie bereits erwähnt, erfolgt die Simulation in Webots. Es ist bei Bewegungen immer sinnvoll eine Simulation durchzuführen, da bei komplexeren Programmabläufen leicht unvorhergesehene Bewegungsabläufe entstehen, die den NAO beschädigen könnten. Zu Beginn der Simulation wird der NAO in eine standardisierte Ausgangsposition für den Schuss gebracht (s. Abb. 6.1). Darauf folgt die Schusssequenz.



**Abb. 6.1.:** Ausgangsposition kurz vor dem Schuss.

Der ZMP und das wirkende Moment im Fuß werden ab Beginn der zweiten Schussphase (anheben des rechten Beins) über das erstellte dynamische Modell berechnet und über die Drucksensoren im Fuß gemessen. Die Berechnung des Moments und des ZMP über die FSRs (*Force Sensitive Resistors*) im linken Fuß erfolgt über folgende Gleichungen.

$$\begin{aligned} M_{Foot, ref} &= -0,07025 \cdot (\text{FSRsFL} + \text{FSRsFR}) \\ &\quad + 0,02965 \cdot \text{FSRsRR} + 0,03025 \cdot \text{FSRsRL} \end{aligned} \quad (6.1)$$

$$ZMP_{ref} = -\frac{M_{Foot, ref}}{m_{NAO} \cdot g} \quad (6.2)$$

FSRsFL, FSRsFR, FSRsRR und FSRsRL stehen in gleicher Reihenfolge für die Druckkräfte am Fußsensor vorne links, vorne rechts, hinten rechts und hinten links.  $m_{NAO}$  und  $g$  stehen für die Gesamtmasse des NAOs und die Erdbeschleunigung. Die Abstände in 6.1 kann man Abbildung 2.1 entnehmen.

### 6.1.1. Testbedingungen

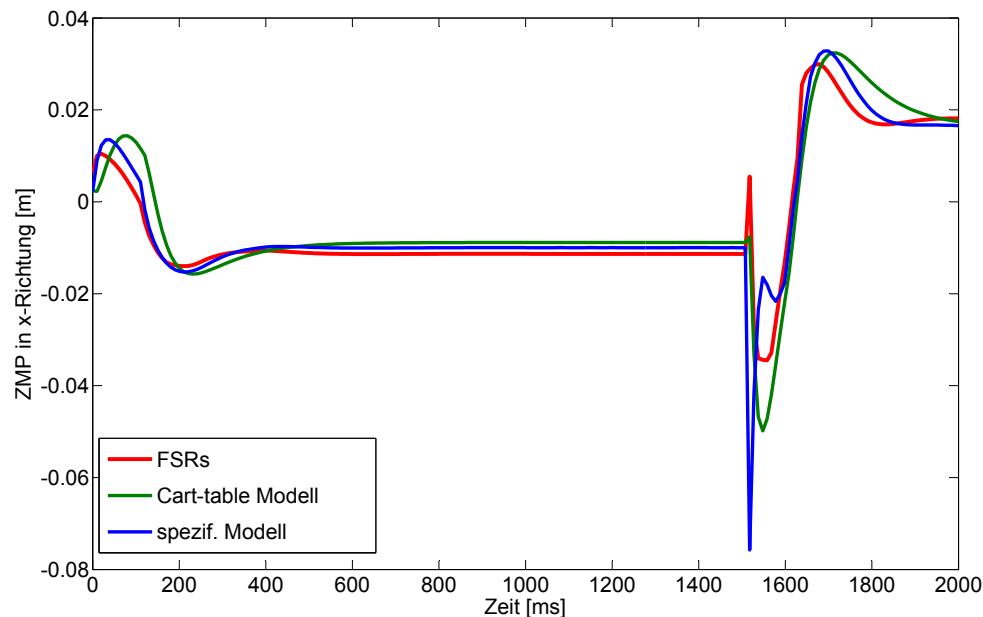
In der Simulation herrschen ideale Bedingungen. Um eine Referenz zum entwickelten dynamischen Modell zu haben, geht man davon aus, dass die Sensoren gut simulierte Werte zurückgeben. Es tritt im Allgemeinen kein Messrauschen in den Messwerten der Sensoren auf. Die Drucksensoren im Fuß werden deshalb als Referenzwert für das dynamische Modell verwendet. Ein allgemeiner Schwachpunkt der Drucksensoren ist jedoch der Messbereich, dessen Grenze bei 25 N liegt. Dieser Wert kann überschritten werden, wenn auf den NAO ein starkes Moment um die  $y$ -Achse und ein leichtes Moment um die  $x$ -Achse wirkt. Dabei wird das CoP (*Center of Pressure*) stark in die Richtung eines Sensors bewegt, worauf der Sensor nur den maximalen Wert von 25 N ausgeben kann. In der Prototyp Datei des NAOs kann das Messspektrum künstlich erhöht werden. Es ist aber nicht gewährleistet, dass diese Sensorwerte

korrekt berechnet werden. Eine optionale Erhöhung des Messspektrums, die vom Hersteller verifiziert ist, wäre eine gute Lösung, da damit keine zusätzlichen Kosten für bessere Sensoren verbunden sein würden. Die Simulation detektiert noch keine Selbstkollision. Es wird davon ausgegangen, dass der Ball auf dem virtuellen Feld, wie auf dem echten Feld rollt.

### 6.1.2. Simulationsspezifische Parameter

In der Simulation können andere Einstellparameter, als in der Realität verwendet werden. Dies betrifft sowohl Geschwindigkeit und Ausholweite einer Bewegung als auch die Regelung. Das Bein wird bei der Schussbewegung für dieses Beispiel weiter nach hinten ausgeholt, um eine größere Beschleunigung zu erreichen. Die Reglerparameter werden dementsprechend angepasst. Es wird zunächst die Schussbewegung ohne Regelung getestet und danach mit aktiver PID-Regelung.

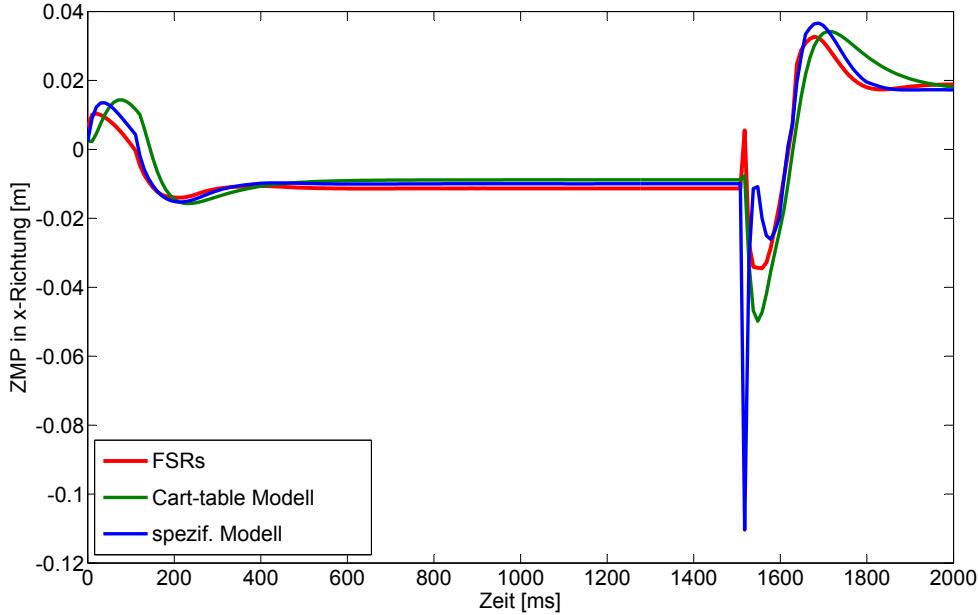
#### Schussbewegung ohne Regelung



**Abb. 6.2.:** Darstellung des berechneten ZMP mit Hilfe des roboterspezifischen Modells (blau), des *Cart-table Modells* (grün) und der FSRs als Referenz (rot).

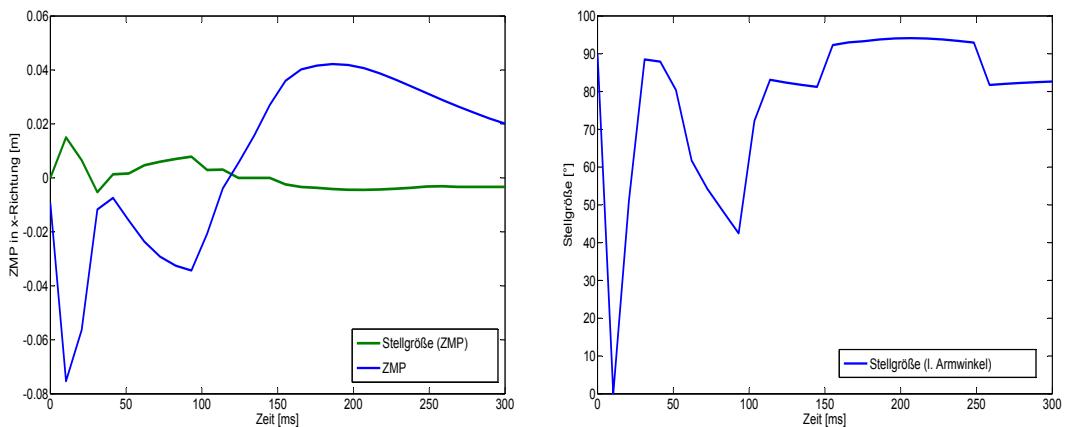
Der ZMP-Verlauf hat zu Beginn des Schusses sehr stark ausgeprägte *Peaks* und fällt danach wieder sehr rapide. Der qualitative Verlauf stimmt ungefähr mit den Referenzwerten überein. Das *Cart-table Modell* zeigt eine gute, aber etwas verzögerte Übereinstimmung mit dem Referenzgraphen. Die Maxima und Minima der Referenz werden leicht überstiegen.

### Schussbewegung mit Regelung



**Abb. 6.3.:** Darstellung des berechneten ZMP mit Hilfe des roboterspezifischen Modells (blau), des *Cart-table Modells* (grün) und der FSRs als Referenz (rot).

Das aufgestellte Modell zeigt einen sehr punktuellen und starken *Peak*. Der qualitative Verlauf des *Cart-table Modells* stimmt besser als das dynamische Modell mit dem Verlauf des Referenzwertes überein. Das Referenzmoment wird durch die Regelung nicht beeinflusst.



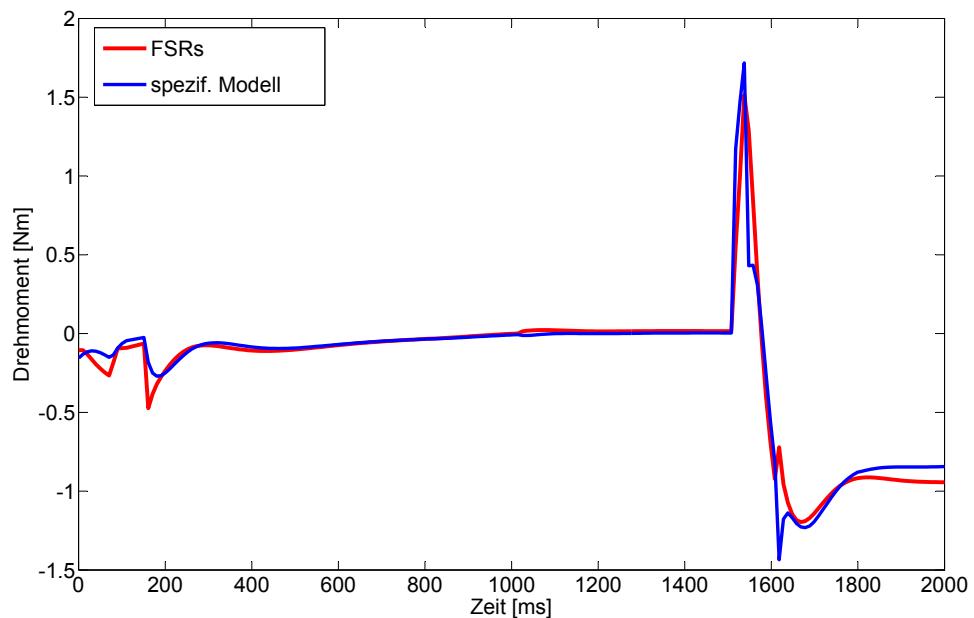
**Abb. 6.4.:** Darstellung des berechneten ZMP mit Hilfe des roboterspezifischen Modells (blau) und des ZMP als Stellgröße im Fuß (grün) auf der linken Seite. Darstellung der Stellgröße als absolute Armwinkel auf der rechten Seite.

Da der ZMP als Stellgröße nicht anschaulich ist, wird auch der Verlauf der Winkelstellungen des Arms aufgeführt. Die Berechnung der Stellgrößen wird in Abschnitt 5.1.9 erklärt. Betrachtet man die Stellgrößen ist erkennbar, dass zum Anfang der Schussbewegung ein starker Ausschlag des Arms erfolgen soll. Dieser wird allerdings nur vom dynamischen Modell wahrgenommen, da er vermutlich eine zu kurze Zeitperiode für die Ausführung hat.

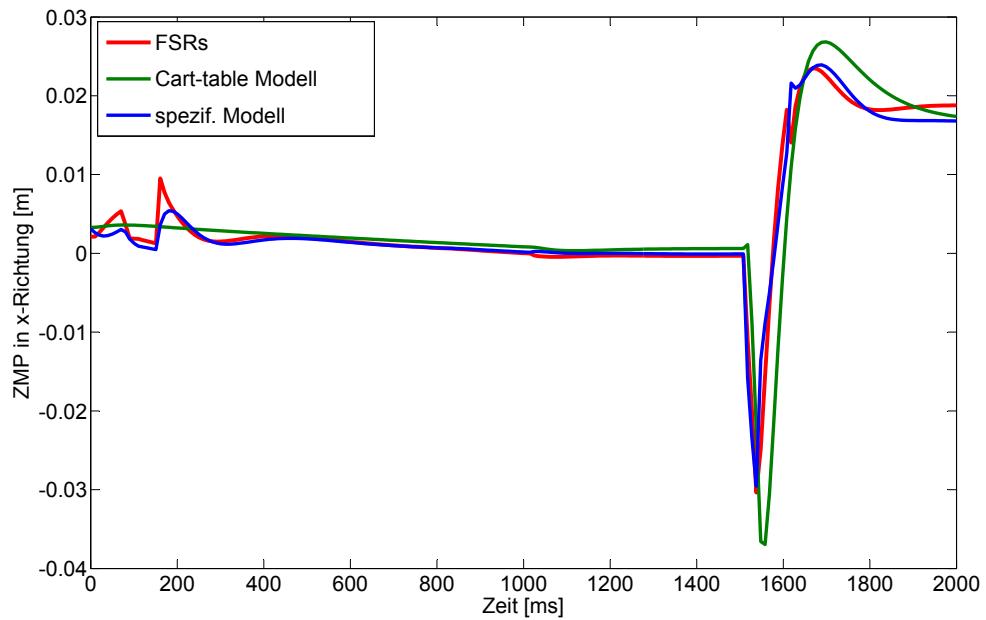
### 6.1.3. Realitätsspezifische Parameter

Zur Gegenüberstellung werden nun für die Bewegung und den Regler Parameter verwendet, die auf dem NAO in einem Realversuch kein instabiles Verhalten verursachen. Die Bewegungsabläufe sind in der Regel langsamer und schwächer. Der Momentenverlauf wird in Abbildung 6.5 zur Veranschaulichung angezeigt.

#### Schussbewegung ohne Regelung



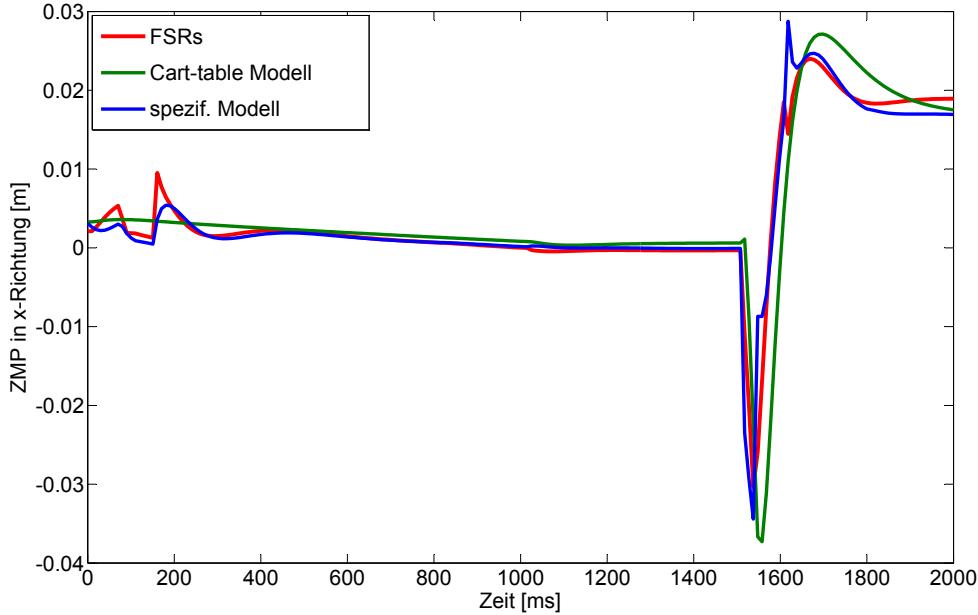
**Abb. 6.5.:** Darstellung des berechneten Moments im Fußgelenk mit Hilfe des roboterspezifischen Modells (blau) und der FSRs als Referenz (rot).



**Abb. 6.6.:** Darstellung des berechneten ZMP mit Hilfe des roboterspezifischen Modells (blau), *Cart-table Modells* (grün) und FSRs als Referenz (rot).

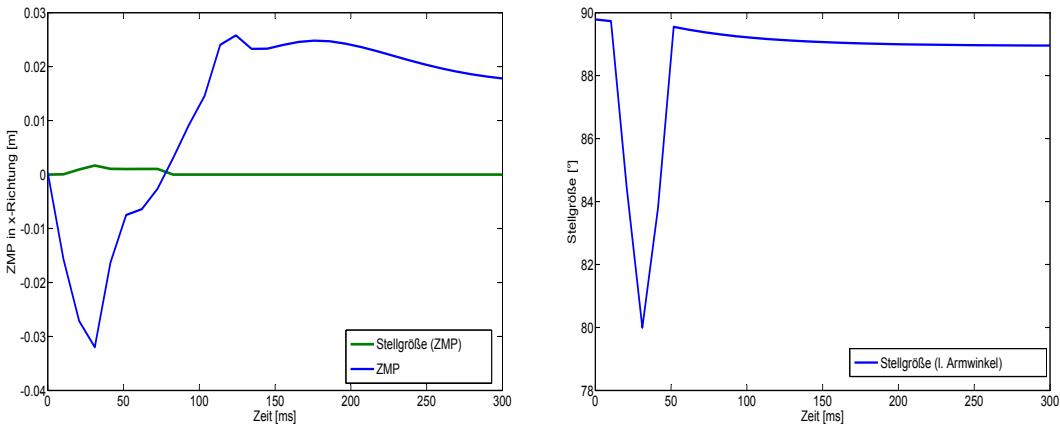
Mit realitätsspezifischen Parametern zeigt sich in der Simulation ohne Regelung eine sehr gute Annäherung durch das aufgestellte Modell. Das *Cart-table Modell* weist einen ca. 1 cm größeren ZMP auf, verläuft aber qualitativ sehr ähnlich zum dynamischen Modell und der Referenz.

### Schussbewegung mit Regelung



**Abb. 6.7.:** Darstellung des berechneten ZMP mit Hilfe des roboterspezifischen Modells (blau), des *Cart-table Modells* (grün) und der FSRs als Referenz (rot).

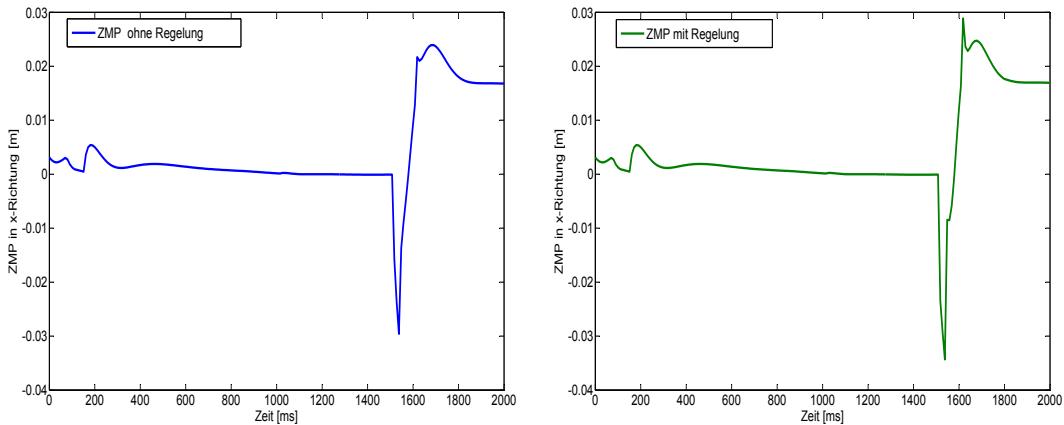
Bei aktiver Regelung zeigt sich im ZMP-Verlauf des dynamischen Modells nur eine sehr geringfügige Veränderung im Vergleich zu Abbildung 6.6. Die Regelung ist mit diesen Parametern in der Simulation also unzureichend. Die Empfindlichkeit des Modells bei einer geringen Bewegung des Arms übersteigt trotz der Tiefpassfilterung das simulierte Verhalten des NAOs. Das *Cart-table Modell* zeigt wie die Referenz keine Reaktion auf die Armbewegung.



**Abb. 6.8.:** Darstellung des berechneten ZMP mit Hilfe des roboterspezifischen Modells (blau) und des ZMP als Stellgröße im Fuß (grün) auf der linken Seite. Darstellung der Stellgröße als absolute Armwinkel auf der rechten Seite.

Anhand der Stellgrößen in Abbildung 6.8 ist zu erkennen, dass die Regelung sehr schwach und nur kurzzeitig eine Reaktion zeigt, da der ZMP während der Schussbewegung nur einmal aus dem in Abbildung 5.3 definierten Bereich austritt.

#### 6.1.4. Auswertung



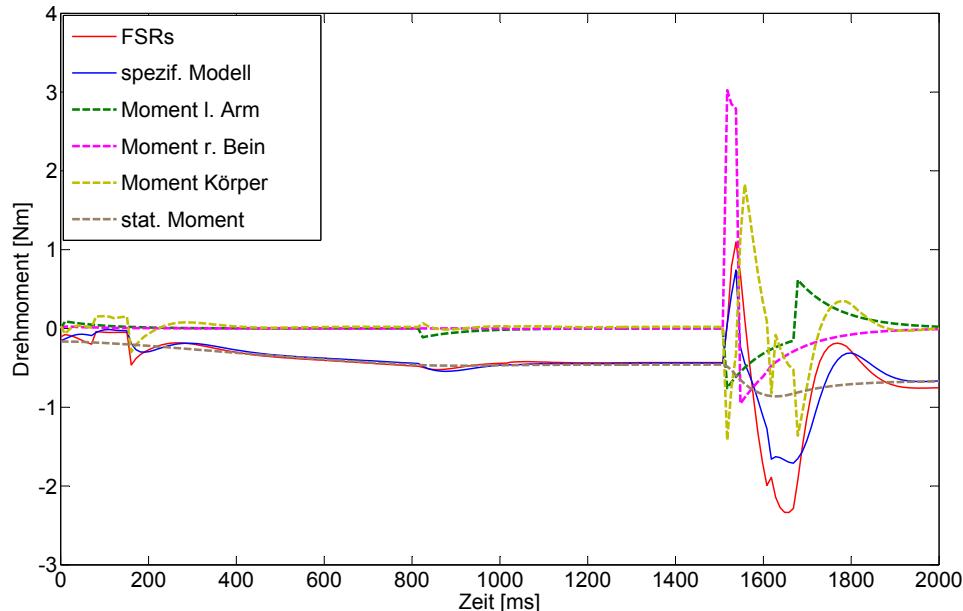
**Abb. 6.9.:** Verlauf des ZMP mit und ohne Regler mit realitätsnahen Parametern.

Vergleicht man die ZMP-Verläufe bei aktiver und ausgeschalteter Regelung, ist nur eine sehr geringe Abweichung zu erkennen. Die Ergebnisse der Simulationen zeigen, dass bei simulationsspezifischen Bewegungsparametern eine stärkere Tiefpassfilterung der d'Alembertschen Momente nötig wäre. Bei Verwendung von realitätsnahen Bewegungsparametern liefert die Filtereinstellung jedoch gute Ergebnisse. Zu beachten ist wiederum, dass die Bewegungen im zweiten Fall langsamer und mit weniger großen Ausschlägen ablaufen. Demnach fallen Arm- und Beinbewegungen nicht so

stark ins Gewicht wie im ersten Fall. Es werden daher im nächsten Abschnitt die einzelnen Momentenkomponenten gegeneinander aufgetragen, um zu sehen welche Momente ausschlaggebend sind.

### 6.1.5. Bewertung der Modellkomponenten

Es werden in einem Plot die in 4.5 erwähnten Momentenkomponenten einzeln aufgetragen, um zu erfahren, welche Komponenten einen großen Einfluss auf das Verhalten des Modells haben. Dazu wird erneut die in dieser Arbeit verwendete Schussbewegung ausgeführt, wobei der Arm gezielt eine Gegenbewegung ohne Regelung ausführt. Dies soll zu einer besseren und verlässlicheren Aussage verhelfen, da die Regelung nicht immer gleich reagiert und somit das Ergebnis verfälschen könnte. Es werden Parameter verwendet, die sowohl in der Simulation als auch auf dem NAO einen stabilen Schuss ergeben.



**Abb. 6.10.:** Plot des Gesamtmoments (blau), der Momentenkomponenten und des Referenzmoments (rot).

Abbildung 6.10 zeigt den Plot der ungefilterten Einzelmomente und der ungefilterten Addition. Die Abweichung des Gesamtmomenten vom Referenzmoment im Minimum wird durch einen plötzlichen Anstieg des Körpermoments erzeugt, welches über das Accelerometer berechnet wird. Den geringstens Einfluss hat in dieser Bewegung der Arm, der mit der geringsten Masse auch das kleinste Massenträgheitsmoment im Vergleich zu den anderen Komponenten aufweist. Es fällt allerdings kein extrem dominantes Moment auf.

## 6.2. Realversuch

In diesem Abschnitt wird das dynamische Modell und die Regelung im Realversuch besprochen. Wie auch in der Simulation, wird der NAO vor Beginn der Schusssequenz in eine standardisierte Position gebracht.



**Abb. 6.11.:** Der NAO in der Ausgangsposition kurz vor dem Schuss.

Der ZMP und das wirkende Moment im Fuß werden ab Beginn der zweiten Schussphase (anheben des rechten Beins) über das erstellte dynamische Modell berechnet und über die Drucksensoren im Fuß gemessen. Die Messungen werden über eine Messfunktion mit Ausgabe in *Choregraphe* durchgeführt. Die zu messenden Werte werden analog zu den Werten aus der Simulation aufgetragen. Es wird jedoch als zusätzliche Referenz für den ZMP die Messfunktion für das *Center of Pressure* im linken Fuß verwendet.

### 6.2.1. Testbedingungen

Für die Tests steht zum jetzigen Zeitpunkt an der TUHH noch kein Fußballfeld zur Verfügung. Dies spielt allerdings im Bezug auf das dynamische Modell nur eine geringfügige Rolle, da sich ein harter Teppichfußboden ähnlich zum Fußballfeld verhalten sollte. Insgesamt verhält der NAO sich in der Realität instabiler als in der Simulation. Hervorzuheben ist das Fußgelenk, welches um die *x*-Achse dreht. Es hat ein relativ großes Spiel. Dies ist jedoch konstruktionsbedingt, da dieses Spiel den NAO beim Gehen federt und somit die permanente Belastung der Sensoren im Fußgelenk dämpft. So entsteht jedoch beim Anheben eines Beins, ein leichtes Einknicken zur Seite.

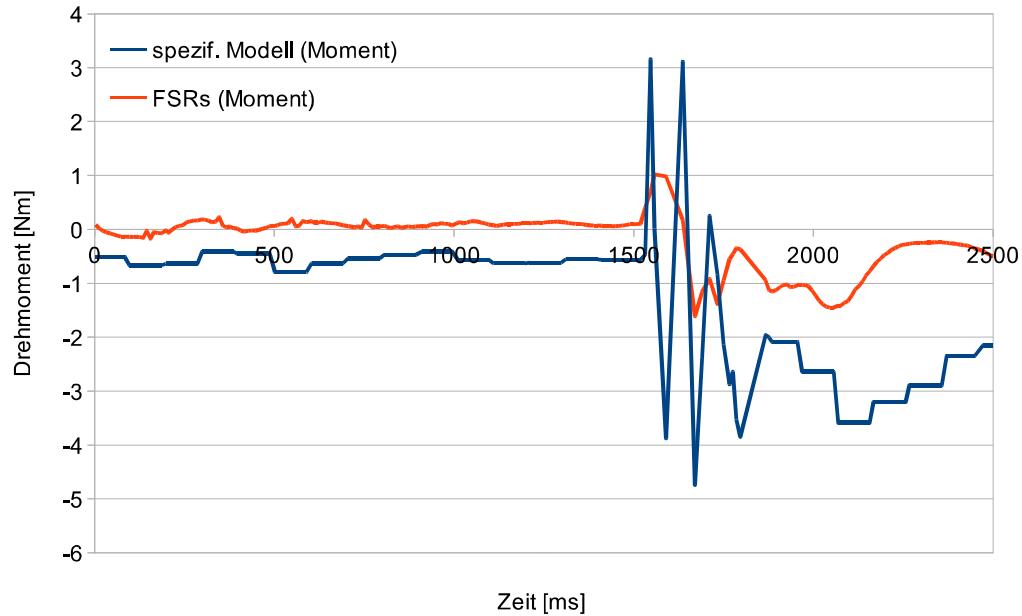
Die druckempfindlichen Sensoren in den Füßen des NAOs sind im Gegensatz zur Simulation nicht immer perfekt kalibriert und können somit schon zu Beginn der

Messung einen *Offset* haben. Für eine korrekte Kalibrierung stehen die Mittel an der TUHH noch nicht zur Verfügung. In der Dokumentation des NAOs wird erwähnt, dass die FSRs selbst bei perfekter Kalibrierung eine Abweichung von 20% pro Sensor aufweisen können. Nach längerer Benutzung kann diese Abweichung also noch weitaus höher sein. Ein zusätzliches Problem, welches schon häufiger aufgetreten ist, ist der Ausfall der gesamten Fußsensorik über eine gewisse Zeitspanne. Das Messspektrum ist auf 25 N pro Sensor begrenzt. Diese aufgeführten Punkte machen die FSRs bei der Benutzung des NAOs zu einem sehr unzuverlässigen und schlechten Messinstrument. Es wird wieder nur ein Plot mit den FSRs als Referenz aufgezeigt, da kein wirklicher Verlass auf eine realitätsnahe Wiedergabe der Kräfte besteht. Um dennoch einen, wenn auch ungenauen, Referenzwert zum ZMP zu erhalten, wird das CoP gemessen. Dieses wird zwar auch über die FSRs gemessen, ist aber einheitlich unabhängig, da es aus dem Verhältnis der vier angreifenden Druckkräfte im Fuß gemessen wird.

In dem entwickelten Modell wird auch das Accelerometer verwendet. Zu beachten ist, dass zu Beginn der Messungen ein *Offset* gemessen wurde. Dieser führt bei der Messung des ZMP zu einem um ca. 2 cm erhöhtem Wert.

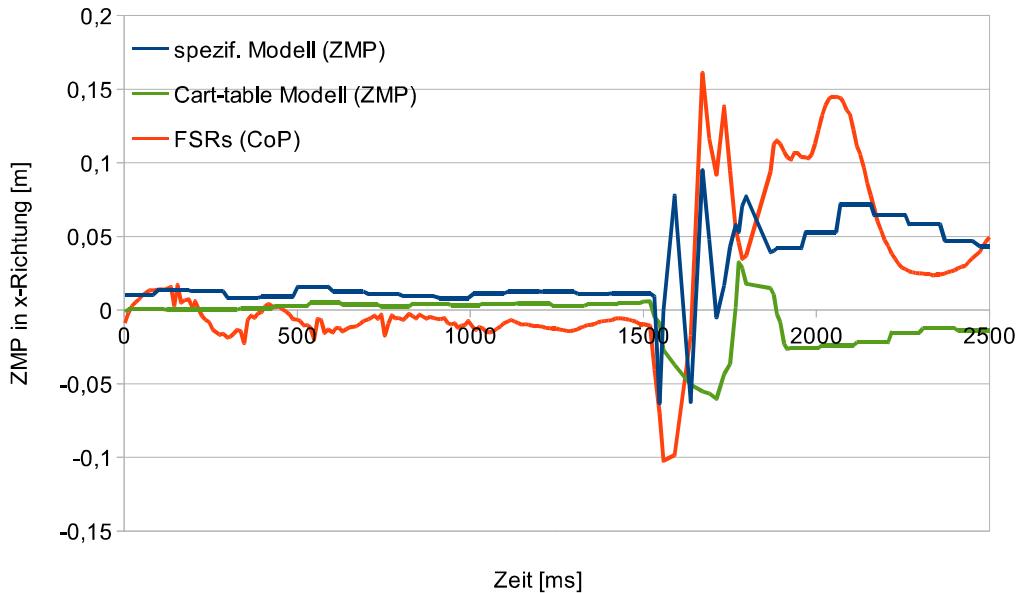
Die Aufnahmefunktion von Sensorwerten in *Choregraphe* bietet zur Zeit nur einen manuellen Startknopf wodurch Ungenauigkeiten bzw. zeitliche Verschiebungen entstehen können. Die Messwerte werden nach Beendigung der Aufnahmefunktion in einer .csv Datei abgespeichert. Diese Daten enthalten gelegentlich Fehler, die der Art sind, dass ein Komma nicht beachtet wird und z. B. aus einer 1,462 eine 1462 wird. Da diese Fehler beim Plotten jedoch sehr auffällig sind, sind sie leicht zu finden und zu eliminieren. Es sind Messaufnahmen mit einem Takt von 10 ms möglich. Ist der Prozessor des NAOs jedoch stark durch Berechnungen ausgelastet, wird dieser Takt nicht mehr eingehalten und ggf. übersprungen. Bei der Aufnahme vieler Daten und gleichzeitiger intensiver Benutzung des Prozessors hat sich außerdem herausgestellt, dass der NAO in manchen Fällen einen "ratternden" Bewegungsablauf ausführt, da nicht mehr im 10 ms Takt Befehle an die Aktoren gesendet werden können.

### Schussbewegung ohne Regelung



**Abb. 6.12.:** Darstellung des berechneten Moments im Fußgelenk mit Hilfe des roboter-spezifischen Modells (blau) und FSRs als Referenz (rot).

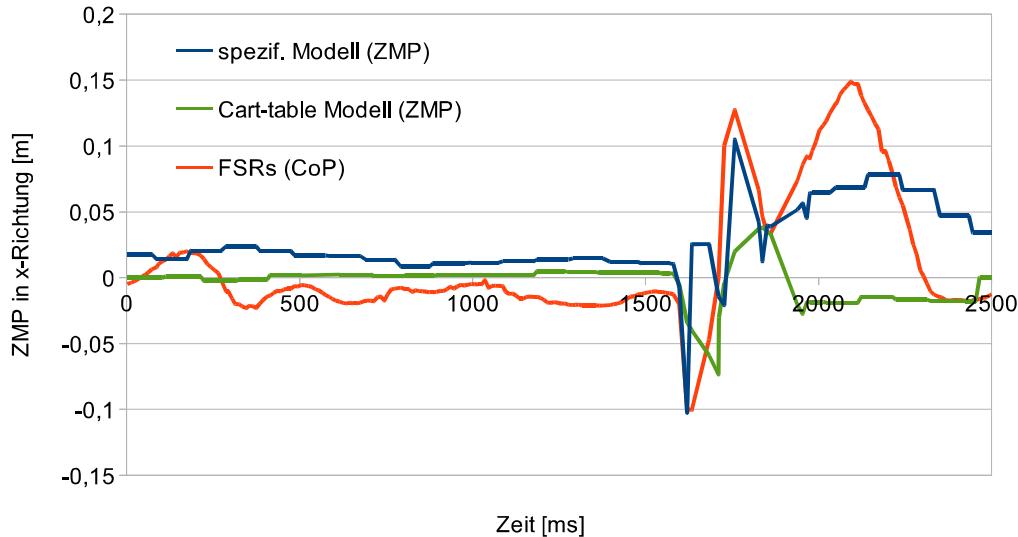
Zur Veranschaulichung wird einmal der Momentenverlauf des dynamischen Modells und der FSRs als Referenz abgebildet. Zu beachten ist, dass ZMP- und Momentenverlauf voneinander abhängen. Demnach fließt der in Abschnitt 6.2.1 erwähnte *Offset* auch hier mit umgekehrtem Vorzeichen in den Momentenverlauf ein.



**Abb. 6.13.:** Darstellung des berechneten ZMP mit Hilfe des roboterspezifischen Modells (blau), *Cart-table Modell* (grün) und FSRs als Referenz (rot).

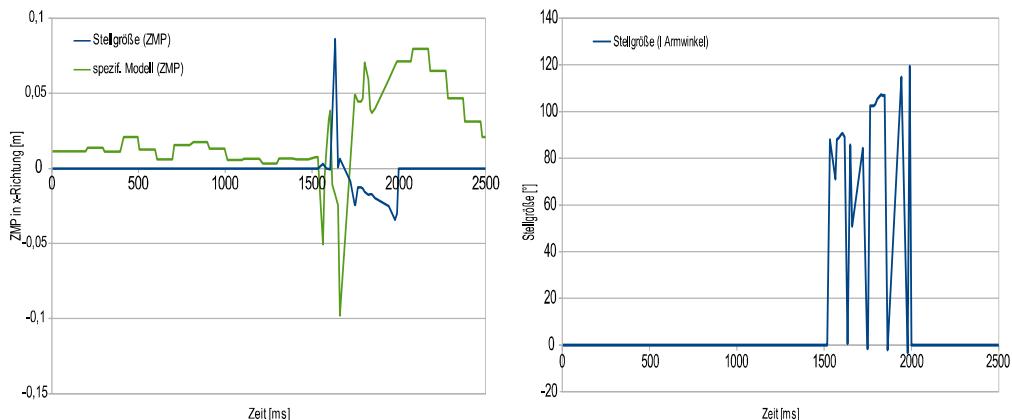
Der ZMP-Verlauf des roboterspezifischen Modells weist in der Schussphase viele schnelle Schwankungen auf. Die ZMP-Werte übersteigen die in der Simulation gemessenen Werte, welche zwischen 3 cm und -4 cm liegen. Das *Cart-table Modell* verläuft hingegen träger mit weniger ausgeprägten Peaks. Die Referenzwerte zeigen ZMP Positionen im Bereich von 15 cm bis -10 cm. Dieser maximale und minimale Wert liegt außerhalb des Fußes. Da der NAO im Test nicht gestürzt ist und die Werte über mehrere Millisekunden anhalten, müssen diese Werte falsch sein. Der qualitative Verlauf der gemessenen Werte für die FSRs stimmt jedoch am besten mit den Simulationen während des Schusses überein.

## Schussbewegung mit Regelung



**Abb. 6.14.:** Darstellung des berechneten ZMP mit Hilfe des roboterspezifischen Modells (blau), *Cart-table Modell* (grün) und FSRs als Referenz (rot).

Bei aktiver Regelung weist das dynamische Modell einen sehr simulationsnahen Verlauf auf. Die Spitzenwerte sind jedoch erneut viel zu hoch und liegen teilweise außerhalb des Fußes. Das *Cart-table Modell* hat erneut einen verzögerten Verlauf mit simulationsnäheren Werten. Die FSRs übersteigen wieder die Grenze des Vorderfußes.

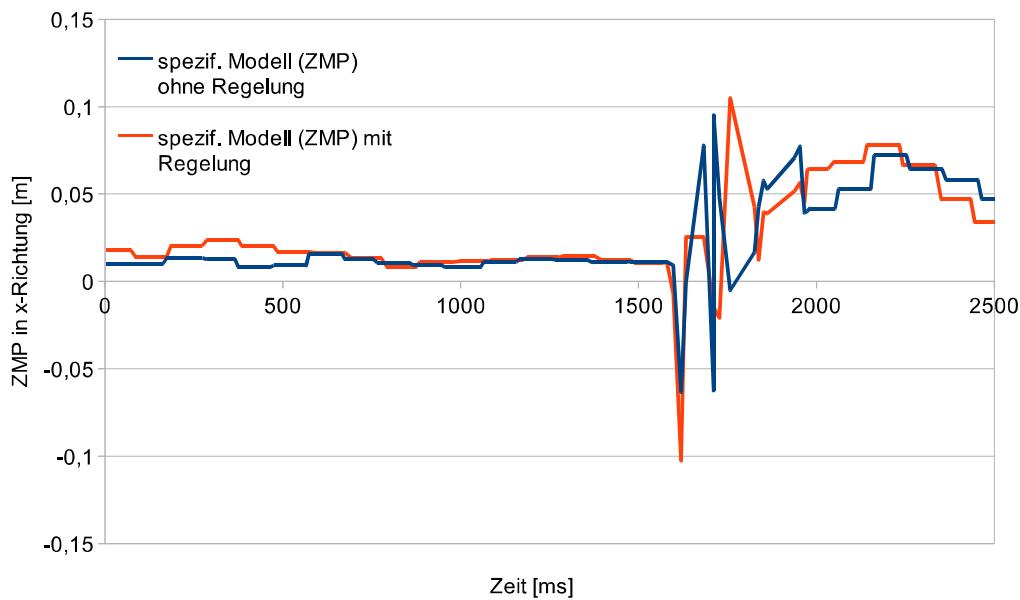


**Abb. 6.15.:** Darstellung des berechneten ZMP mit Hilfe des roboterspezifischen Modells (grün) und des ZMP als Stellgröße im Fuß (blau) auf der linken Seite. Darstellung der Stellgröße als absolute Armwinkel auf der rechten Seite.

Der ZMP ist in diesem Fall nicht gut als anschauliche Stellgröße geeignet, wird aber der Vollständigkeit halber mit aufgeführt. Die Berechnung der Stellgrößen wird in Abschnitt 5.1.9 erklärt. Der Regler setzt erst ab ca. 1500 ms ein. Betrachtet man den Verlauf des ZMP, welcher über das dynamische Modell berechnet wird, erfolgt die Änderung der Stellgröße zu spät. Zu dieser Verzögerung addiert sich die Totzeit des Systems. Auf diese Weise entsteht bei Schwankungen mit großer Wahrscheinlichkeit eine Verstärkung.

### 6.2.2. Auswertung

Die Messwerte für das roboterspezifische und das *Cart-table Modell* haben gezeigt, dass beide Modelle Stärken und Schwächen haben. Das roboterspezifische Modell reagiert zum Zeitpunkt der Tests zu empfindlich und mit zu hohen Ausschlägen. Abhilfe würde eine stärkere Tiefpassfilterung schaffen, die jedoch die Reaktionsgeschwindigkeit des Modells herabsetzen würde. Das *Cart-table Modell* zeigt im Bezug auf die ZMP-Position realistischere Werte, die auch den simulierten Werten näher kommen. Es reagiert jedoch oft zu langsam. Die langsame Reaktion kommt durch die Bestimmung der Schwerpunktbeschleunigung zustande. Durch zweifaches Ableiten der Schwerpunktposition entsteht ein starkes Rauschen. Dies kann nur durch eine Filterung unterdrückt werden, führt jedoch zur genannten Situation.



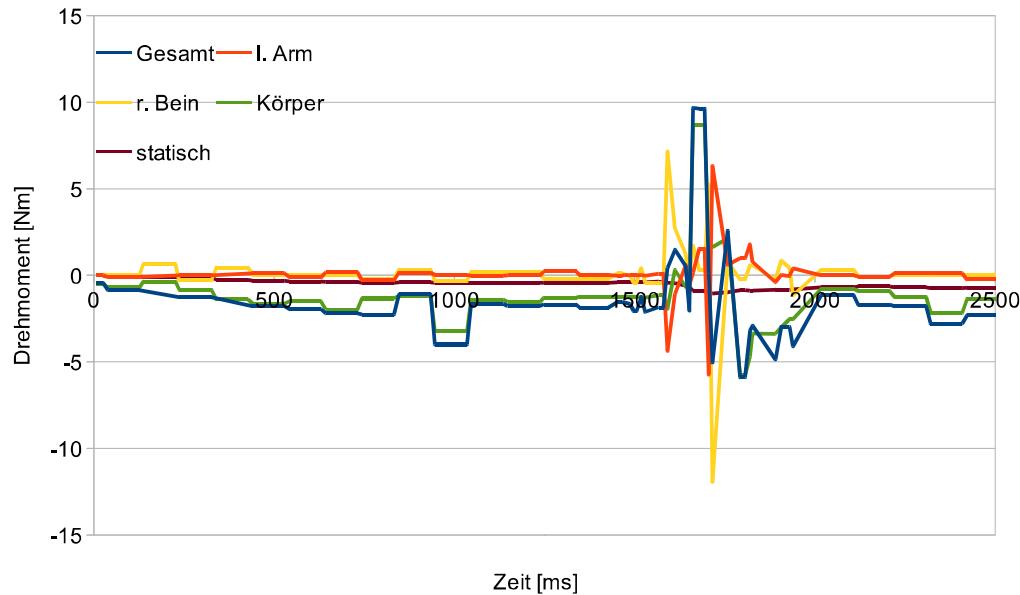
**Abb. 6.16.:** Verlauf des berechneten ZMP mit und ohne Regler.

Vergleicht man die ZMP-Positionen bei eingeschaltetem Regler und ausgeschaltetem Regler, wird deutlich, dass der Regler größere Ausschläge des ZMP verursacht. Die verzögerte Reaktion führt vermutlich zu einer Verstärkung des wirkenden Mo-

ments. Eine prädiktive Regelung oder präventiv gesteuerte Gegenbewegung könnte zu größeren Erfolgen verhelfen.

### 6.2.3. Bewertung der Modellkomponenten

Wie in Abschnitt 6.1.5 erfolgt auch eine Bewertung der Modellkomponenten im Realversuch. Die Momente werden ungefiltert aufgetragen.



**Abb. 6.17.:** Plot des Gesamtmoments (blau) und der Momentenkomponenten.

Das statische Moment spielt eine untergeordnete Rolle. Der extrem starke Ausschlag des Beins im negativen Momentenbereich ist vermutlich ein Messfehler der dadurch entsteht, dass der NAO bei hoher Rechenbelastung, wie der Aufnahme vieler Messdaten, nicht mehr in der Lage ist, den Aktualisierungszyklus von 10 ms einzuhalten. Die Geschwindigkeiten und Beschleunigungen in den Gelenken werden dennoch mit einer erwarteten Zykluszeit von 10 ms berechnet. Der extrem hohe Ausschlag des Gesamtmoments wird fast ausschließlich durch die Körperkomponente ausgelöst. Dies kann zum einen mit dem Fehlen eines Filters zu tun haben und zum anderen an Messfehlern liegen. Vergleicht man das Gesamtmoment mit dem berechneten Moment aus Abbildung 6.12, bestätigt das diese Annahme. Es wird die gleiche Schussbewegung verwendet, wobei noch eine zusätzliche Gegenbewegung des Arms ausgeführt wird.

### 6.3. Zusammenfassung und Bewertung

Hinsichtlich der Simulation liefert das dynamische Modell bei Bewegungen, die dem realistischen Verhalten des NAOs entsprechen gute Annäherungen. Werden in der Simulation hingegen Bewegungen ausgeführt, die sehr schnell sind, neigt das dynamische Modell dazu, große Abweichungen zu den Referenzwerten aufzuzeigen. Die Regelung zeigt in der Simulation mit realitätsspezifischen Parametern kaum Auswirkung auf das Verhalten. Im Realversuch ist anhand des berechneten ZMP in beiden Modellen sogar eine Verschlechterung des Verhaltens bemerkbar. Dies lässt darauf schließen, dass die Regelung zu langsam ist und in einer Schussbewegung nicht brauchbar ist.

Das reale Verhalten des NAOs bei Bewegungen lässt sich weitaus schwieriger berechnen als in der Simulation. Die aktuellen Messungen des ZMP schwingen zu oft mit zu hohen Ausschlägen. Das *Cart-table Modell* bietet einen sehr einfach zu implementierenden Ansatz, der jedoch sehr schnell an seine Grenzen stößt. Abgesehen von den Filtereinstellungen kann keine weitere Erhöhung der Präzision erzielt werden.

## 7. Zusammenfassung und Ausblick

Ziel dieser Arbeit war es, ein vereinfachtes roboterspezifisches dynamisches Modell aufzustellen, um die im Fußgelenk wirkenden Momente anzunähern. Dieses sollte anschließend während und nach einer Schussbewegung als Basis für eine Regelung zur Stabilisierung dienen. Das dynamische Modell wurde nur für Momente, die um die  $y$ -Achse wirken, aufgestellt und getestet. Für die Tests wurde zunächst eine einfache Schussbewegung implementiert, die sowohl in der Simulation als auch in der Realität funktioniert. Es konnte gezeigt werden, dass das dynamische Modell in der Simulation dem *Cart-table Modell* überlegen ist, jedoch in der Realität noch einige Schwächen aufweist. Die Regelung zur Stabilisierung erwies sich in den meisten Fällen als zu langsam, was zur Folge hatte, dass die Regelung häufig einen gegenteiligen Effekt erzielte. Schnellere und stärkere Schussbewegungen funktionierten daher bisher nur in der Simulation und nicht in der Realität. Das Ziel einer Stabilisierung der Schussbewegung durch einen Regler konnte nicht erfüllt werden, weshalb folgende Optimierungsvorschläge gemacht werden.

Für eine stabile Schussbewegung ist eine prädiktive Regelung oder eine präventiv gesteuerte Gegenbewegung erforderlich.

Zur Zeit wird das Moment, welches durch die Körperbewegung verursacht wird über das Accelerometer berechnet. Da dies eine sensorbasierte Berechnung ist, kann keine prädiktive Regelung über diese Komponente erfolgen. Zudem geht das dynamische Modell davon aus, dass sich das Accelerometer im Schwerpunkt befindet. Der Schwerpunkt ändert jedoch den Bewegungen entsprechend seine Position. Ein Ansatz zur Lösung des Problems wäre, die Beschleunigung über das Accelerometer zu bestimmen und das Fußgelenk als Drehachse des gesamten Körpers zu betrachten. Die Beschleunigung senkrecht zur Rotationsachse ist im Fixpunkt Null. Kennt man die Beschleunigung des Accelerometers, kann linear interpoliert werden, welche Beschleunigung der Körper in diesem Zeitpunkt um die Rotationsachse erfährt. In diesem Zuge sollte auch das Gyroskop in die Berechnung der Beschleunigung einzogen werden, da die Beschleunigungen des Accelerometers immer im Koordinatensystem des Torsos gemessen werden und die Ausrichtung des Oberkörpers somit eine große Rolle spielt. Wird dieser sensorbasierte Algorithmus jedoch auf mehreren Robotern implementiert, kann dieser je nach *Offset* der Accelerometer verschiedene Ergebnisse liefern. Eine regelmäßige Kalibrierung wäre also nötig. Dieses gesamte Problem kann mit Hilfe der Vorwärtsskinematik und der inversen Kinematik umgangen werden. Die ursprüngliche Annahme des dynamischen Modells geht davon aus, dass die gesamten Momente durch die d'Alembertschen Momente beschrieben werden. Davon kann jedoch nicht ausgegangen werden, wenn das System nicht in allen Gelenken sehr starr ist. Berechnet man die Bewegungen und dementsprechenden Momente immer für den nächsten Zyklusdurchlauf voraus und vergleicht diese

mit den aktuell gemessenen Momenten, ergibt das die Momentenkomponente, die durch das Spiel in den Gelenken verloren gegangen ist. Auf diese Weise lässt sich allerdings nur der aktuelle Wert bestimmen.

Die Berechnung des statischen Moments erfolgt zur Zeit nur über den Abstand des Körperschwerpunktes in  $x$ -Richtung vom Fußgelenk. Steht der Roboter nicht mit vollem Fußkontakt auf dem Boden, ist diese Annahme nicht mehr korrekt, da dieser Algorithmus die Richtung der Schwerkraft nicht beachtet. Es wird permanent angenommen, dass die Schwerkraft in  $z$ -Richtung des Fußkoordinatensystem wirkt. Durch die Verwendung des Gyroskops der *Inertial Unit* lässt sich auch dieses Problem beheben.

Das dynamische Modell berechnet noch keine Momente bezüglich der  $x$ -Achse oder  $z$ -Achse. Die Realversuche haben gezeigt, dass durch das große Spiel im Fußgelenk ein Kippen um die  $x$ -Achse bei Störungen sehr wahrscheinlich ist. Es ist also im Bezug auf die Stabilität gegen äußere Störungen von äußerster Wichtigkeit, die Momente um die  $x$ -Achse mit einzubeziehen.

Das im Zuge dieser Arbeit aufgestellte roboterspezifische dynamische Modell stellt in der Simulation eine gute Näherung der aktuell wirkenden Momente dar. Werden die Optimierungsvorschläge implementiert, sollte vor allem im Realversuch eine genauere Berechnung zu erwarten sein. Mit Hilfe der Vorwärtsskinematik bietet das dynamische Modell die Grundlage für eine präventive Regelung und kann in zukünftigen Arbeiten erweitert werden, sodass schnellere und stärkere Schussbewegungen implementiert werden können. Diese sind im Bezug auf die zukünftige Teilnahme der TUHH am *RoboCup* von großer Bedeutung. Im Bezug auf einen Laufalgorithmus, ist dieses Modell nicht direkt anwendbar, da die Bewegung des Standbeines noch nicht miteinbezogen wird. Dieses wird beim Laufen jedoch mehr bewegt als bei einem Schuss. Nach wenigen Anpassungen sollte das dynamische Modell allerdings leicht übertragbar sein. Es ist sogar zu erwarten, dass die Annäherungen während des Laufens bessere Ergebnisse liefern als beim Schuss, da die Bewegungen der Gelenke langsamer ausgeführt werden. Demnach ist auch ein größerer Erfolg bei Verwendung einer Regelung basierend auf dem ZMP zu erwarten.

# A. Hardwareübersicht

## A.1. Abmessungen des NAO

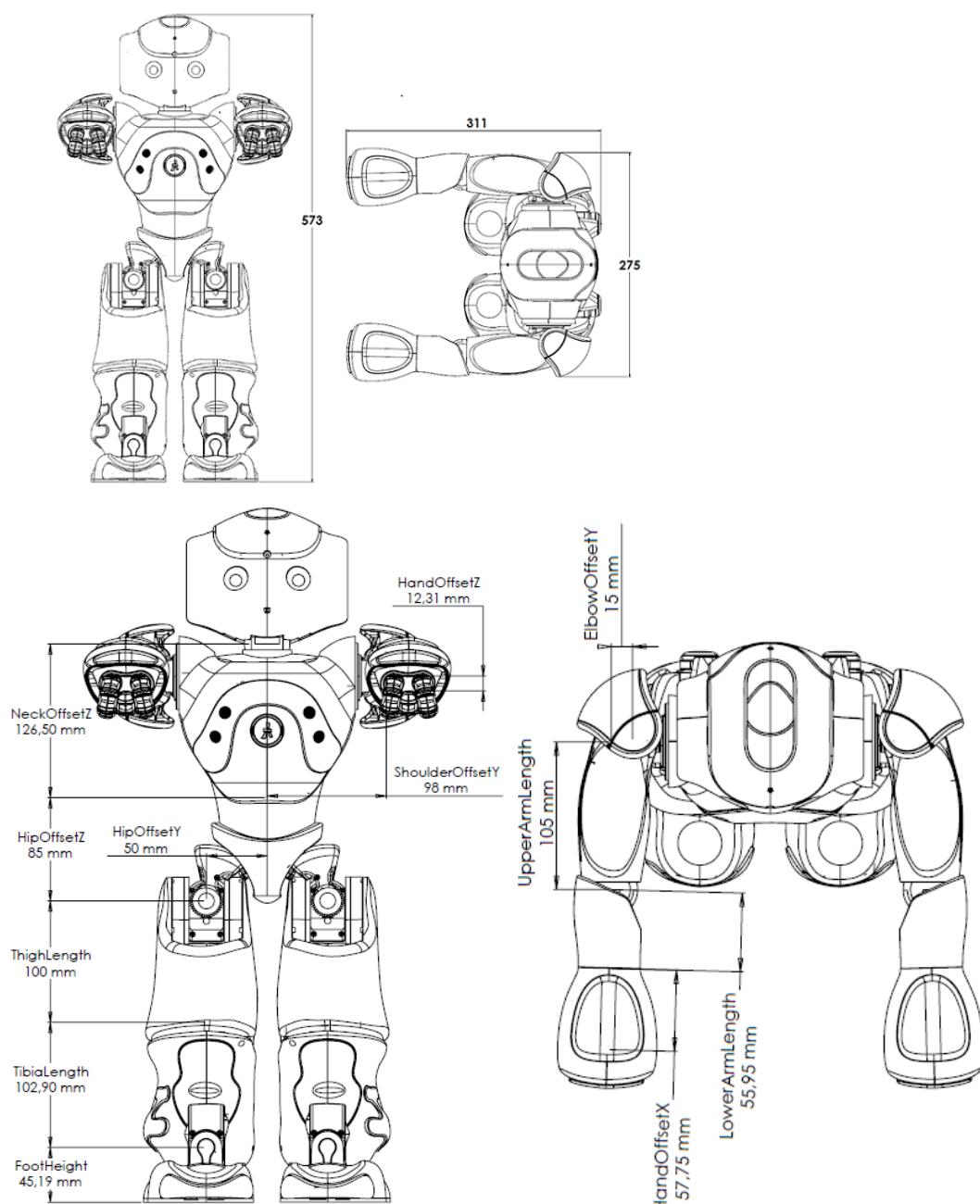
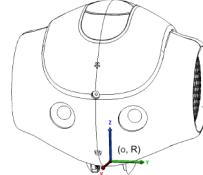


Abb. A.1.: Abmessungen des NAO Roboters. [Ald13]

## A.2. Massen, Schwerpunktpositionen und Inertialmatrizen der Bauteile

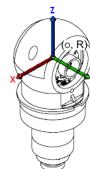
Die folgenden Angaben für Massen beziehen sich auf Kilogramm. CoM wird in Metern angegeben und die Inertialmatrix in  $\text{kg} \cdot \text{m}^2$ . Abgesehen vom Kopf sind alle Körperteil aus der Version 3.3.



**Abb. A.2.:** NAO Kopf V. 4.0. [Ald13]

$$\text{Mass} = 0,060533, \quad Com_{Head} = \begin{bmatrix} -0,00112 \\ 0,00003 \\ 0,05258 \end{bmatrix}$$

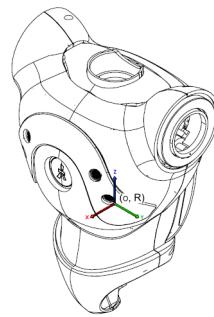
$$I_{Head} = \begin{bmatrix} 0,00263129518 & 0,00000878814 & 0,00004098466 \\ 0,00000878814 & 0,00247112488 & -0,00002995792 \\ 0,00004098466 & -0,00002995792 & 0,00098573565 \end{bmatrix}$$



**Abb. A.3.:** NAO Genick. [Ald13]

$$\text{Mass} = 0,05930, \quad Com_{Neck} = \begin{bmatrix} -0,00002 \\ 0,00017 \\ -0,02556 \end{bmatrix}$$

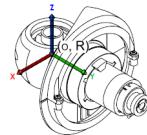
$$I_{Neck} = \begin{bmatrix} 0,00006180308 & -0,00000000205 & -0,00000004948 \\ -0,00000000205 & 0,00006281859 & -0,00000005067 \\ -0,00000004948 & -0,00000005067 & 0,00000543977 \end{bmatrix}$$



**Abb. A.4.:** NAO Torso. [Ald13]

$$\text{Mass} = 0,060533, \quad Com_{Torso} = \begin{bmatrix} -0,00415 \\ 0,00007 \\ 0,04258 \end{bmatrix}$$

$$I_{Torso} = \begin{bmatrix} 0,00496741976 & 0,00001247398 & 0,00016446814 \\ 0,00001247398 & 0,00479135246 & -0,00002477669 \\ 0,00016446814 & -0,00002477669 & 0,00159784829 \end{bmatrix}$$

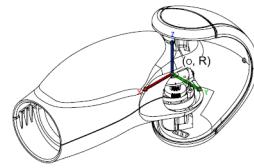


**Abb. A.5.:** NAO Schulter. [Ald13]

$$\text{Mass} = 0,06996, \quad Com_{RShoulder} = \begin{bmatrix} -0,00178 \\ 0,02496 \\ 0,00018 \end{bmatrix}, \quad Com_{LShoulder} = \begin{bmatrix} -0,00178 \\ -0,02496 \\ 0,00018 \end{bmatrix}$$

$$I_{RShoulder} = \begin{bmatrix} 0,00007102499 & 0,00000202467 & 0,00000001722 \\ 0,00000202467 & 0,0001405799 & -0,0000000841 \\ 0,00000001722 & -0,0000000841 & 0,00007316600 \end{bmatrix}$$

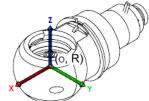
$$I_{LShoulder} = \begin{bmatrix} 0,00007102499 & -0,00000202467 & 0,00000001722 \\ -0,00000202467 & 0,0001405799 & 0,00000000841 \\ 0,00000001722 & 0,0000000841 & 0,00007316600 \end{bmatrix}$$

**Abb. A.6.:** NAO Oberarm. [Ald13]

$$\text{Mass} = 0,12309, \quad Com_{RBiceps} = \begin{bmatrix} 0,01885 \\ -0,00577 \\ 0,00065 \end{bmatrix}, \quad Com_{LBiceps} = \begin{bmatrix} 0,01885 \\ 0,00577 \\ 0,00065 \end{bmatrix}$$

$$I_{RBiceps} = \begin{bmatrix} 0,00008228413 & 0,00003978057 & -0,00000752601 \\ 0,00003978057 & 0,00029001445 & 0,00000152945 \\ -0,00000752601 & 0,00000152945 & 0,00026842331 \end{bmatrix}$$

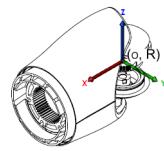
$$I_{LBiceps} = \begin{bmatrix} 0,00008228413 & -0,00003978057 & -0,00000752601 \\ -0,00003978057 & 0,00029001445 & -0,00000152945 \\ -0,00000752601 & -0,00000152945 & 0,00026842331 \end{bmatrix}$$

**Abb. A.7.:** NAO Ellenbogen. [Ald13]

$$\text{Mass} = 0,05971, \quad Com_{RElbow} = \begin{bmatrix} -0,02560 \\ 0,00001 \\ -0,00019 \end{bmatrix}, \quad Com_{LElbow} = \begin{bmatrix} -0,02560 \\ -0,00001 \\ -0,00019 \end{bmatrix}$$

$$I_{RElbow} = \begin{bmatrix} 0,00000550319 & 0,00000002243 & 0,00000001534 \\ 0,00000002243 & 0,00006225405 & -0,00000000559 \\ 0,00000001534 & -0,00000000559 & 0,00006325124 \end{bmatrix}$$

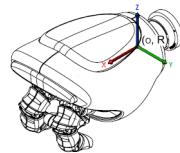
$$I_{LElbow} = \begin{bmatrix} 0,00000550319 & -0,00000002243 & 0,00000001534 \\ -0,00000002243 & 0,00006225405 & 0,00000000559 \\ 0,00000001534 & 0,00000000559 & 0,00006325124 \end{bmatrix}$$

**Abb. A.8.:** NAO Unterarm. [Ald13]

$$\text{Mass} = 0,07724, \quad Com_{R\text{Forearm}} = \begin{bmatrix} 0,02556 \\ -0,00273 \\ 0,00096 \end{bmatrix}, \quad Com_{L\text{Forearm}} = \begin{bmatrix} 0,02556 \\ 0,00273 \\ 0,00096 \end{bmatrix}$$

$$I_{R\text{Forearm}} = \begin{bmatrix} 0,00002519483 & 0,00000216293 & -0,00000071846 \\ 0,00000216293 & 0,00008890315 & 0,00000010809 \\ -0,00000071846 & 0,00000010809 & 0,00008686867 \end{bmatrix}$$

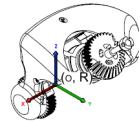
$$I_{L\text{Forearm}} = \begin{bmatrix} 0,00002519483 & -0,00000216293 & -0,00000071846 \\ -0,00000216293 & 0,00008890315 & -0,00000010809 \\ -0,00000071846 & -0,00000010809 & 0,00008686867 \end{bmatrix}$$

**Abb. A.9.:** NAO Hand. [Ald13]

$$\text{Mass} = 0,16653, \quad Com_{R\text{Hand}} = \begin{bmatrix} 0,03180 \\ -0,00083 \\ 0,00477 \end{bmatrix}, \quad Com_{L\text{Hand}} = \begin{bmatrix} 0,03180 \\ 0,00083 \\ 0,00477 \end{bmatrix}$$

$$I_{R\text{Hand}} = \begin{bmatrix} 0,00006105675 & 0,00000503640 & -0,00003895369 \\ 0,00000503640 & 0,00027947351 & 0,0000293076 \\ -0,00003895369 & 0,00000293076 & 0,00027998803 \end{bmatrix}$$

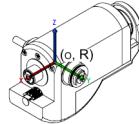
$$I_{L\text{Hand}} = \begin{bmatrix} 0,00006105675 & -0,00000503640 & -0,00003895369 \\ -0,00000503640 & 0,00027947351 & -0,00000293076 \\ -0,00003895369 & -0,00000293076 & 0,00027998803 \end{bmatrix}$$

**Abb. A.10.:** NAO Becken. [Ald13]

$$\text{Mass} = 0,07117, \quad Com_{RPelvis} = \begin{bmatrix} -0,00766 \\ 0,01200 \\ 0,02717 \end{bmatrix}, \quad Com_{LPelvis} = \begin{bmatrix} -0,00766 \\ -0,01200 \\ 0,02717 \end{bmatrix}$$

$$I_{RPelvis} = \begin{bmatrix} 0,00008997164 & 0,0000500214 & 0,00001273548 \\ 0,0000500214 & 0,00010552250 & -0,00002770075 \\ 0,00001273548 & -0,00002770075 & 0,00006688382 \end{bmatrix}$$

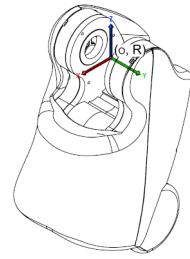
$$I_{LPelvis} = \begin{bmatrix} 0,00008997164 & -0,0000500214 & 0,00001273548 \\ -0,0000500214 & 0,00010552250 & 0,00002770075 \\ 0,00001273548 & 0,00002770075 & 0,00006688382 \end{bmatrix}$$

**Abb. A.11.:** NAO Hüfte. [Ald13]

$$\text{Mass} = 0,13530, \quad Com_{RHip} = \begin{bmatrix} -0,01649 \\ -0,00029 \\ -0,00475 \end{bmatrix}, \quad Com_{LHip} = \begin{bmatrix} -0,01649 \\ 0,00029 \\ -0,00475 \end{bmatrix}$$

$$I_{RHip} = \begin{bmatrix} 0,00002791129 & -0,0000004765 & -0,00000530958 \\ -0,0000004765 & 0,00010146847 & -0,00000000007 \\ -0,00000530958 & -0,00000000007 & 0,00009209165 \end{bmatrix}$$

$$I_{LHip} = \begin{bmatrix} 0,00002791129 & 0,0000004765 & -0,00000530958 \\ 0,0000004765 & 0,00010146847 & 0,00000000007 \\ -0,00000530958 & 0,00000000007 & 0,00009209165 \end{bmatrix}$$

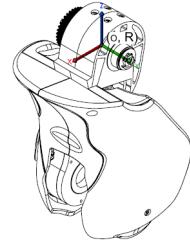


**Abb. A.12.:** NAO Oberschenkel. [Ald13]

$$\text{Mass} = 0,039421, \quad Com_{RThigh} = \begin{bmatrix} 0,00132 \\ -0,00235 \\ -0,05352 \end{bmatrix}, \quad Com_{LThigh} = \begin{bmatrix} 0,00132 \\ 0,00235 \\ -0,05352 \end{bmatrix}$$

$$I_{RThigh} = \begin{bmatrix} 0,00163011959 & -0,00000031939 & 0,00008287048 \\ -0,00000031939 & 0,0015269948 & -0,00004333919 \\ 0,00008287048 & -0,00004333919 & 0,00030120888 \end{bmatrix}$$

$$I_{LThigh} = \begin{bmatrix} 0,00163011959 & 0,00000031939 & 0,00008287048 \\ 0,00000031939 & 0,0015269948 & 0,00004333919 \\ 0,00008287048 & 0,00004333919 & 0,00030120888 \end{bmatrix}$$

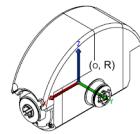


**Abb. A.13.:** NAO Schienbein. [Ald13]

$$\text{Mass} = 0,29159, \quad Com_{RTibia} = \begin{bmatrix} 0,00422 \\ -0,00252 \\ -0,04868 \end{bmatrix}, \quad Com_{LTibia} = \begin{bmatrix} 0,00422 \\ 0,00252 \\ -0,04868 \end{bmatrix}$$

$$I_{RTibia} = \begin{bmatrix} 0,00115382550 & -0,00000077529 & 0,00003231279 \\ -0,00000077529 & 0,00110377729 & -0,00004396728 \\ 0,00003231279 & -0,00004396728 & 0,00018861038 \end{bmatrix}$$

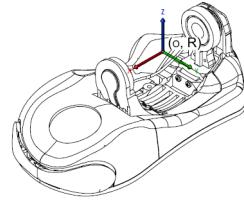
$$I_{LTibia} = \begin{bmatrix} 0,00115382550 & 0,00000077529 & 0,00003231279 \\ 0,00000077529 & 0,00110377729 & 0,00004396728 \\ 0,00003231279 & 0,00004396728 & 0,00018861038 \end{bmatrix}$$

**Abb. A.14.:** NAO Fußgelenk. [Ald13]

$$\text{Mass} = 0,13892, \quad Com_{RAnkle} = \begin{bmatrix} 0,00142 \\ -0,00028 \\ 0,00638 \end{bmatrix}, \quad Com_{LAnkle} = \begin{bmatrix} 0,00142 \\ 0,00028 \\ 0,00638 \end{bmatrix}$$

$$I_{RAnkle} = \begin{bmatrix} 0,00003882614 & 0,00000004711 & 0,00000221327 \\ 0,00000004711 & 0,0007211571 & -0,00000000686 \\ 0,0000221327 & -0,0000000686 & 0,0005352041 \end{bmatrix}$$

$$I_{LAnkle} = \begin{bmatrix} 0,00003882614 & -0,00000004711 & 0,00000221327 \\ -0,00000004711 & 0,0007211571 & 0,00000000686 \\ 0,0000221327 & 0,0000000686 & 0,0005352041 \end{bmatrix}$$

**Abb. A.15.:** NAO Fuß. [Ald13]

$$\text{Mass} = 0,16175, \quad Com_{RFoot} = \begin{bmatrix} 0,02540 \\ -0,00332 \\ -0,03241 \end{bmatrix}, \quad Com_{LFoot} = \begin{bmatrix} 0,02540 \\ 0,00332 \\ -0,03241 \end{bmatrix}$$

$$I_{RFoot} = \begin{bmatrix} 0,00026961204 & 0,00000587690 & 0,00013924446 \\ 0,00000587690 & 0,00064387818 & -0,00001887096 \\ 0,00013924446 & -0,00001887096 & 0,00052519167 \end{bmatrix}$$

$$I_{LFoot} = \begin{bmatrix} 0,00026961204 & -0,00000587690 & 0,00013924446 \\ -0,00000587690 & 0,00064387818 & 0,00001887096 \\ 0,00013924446 & 0,00001887096 & 0,00052519167 \end{bmatrix}$$

### A.3. Gelenkwinkel

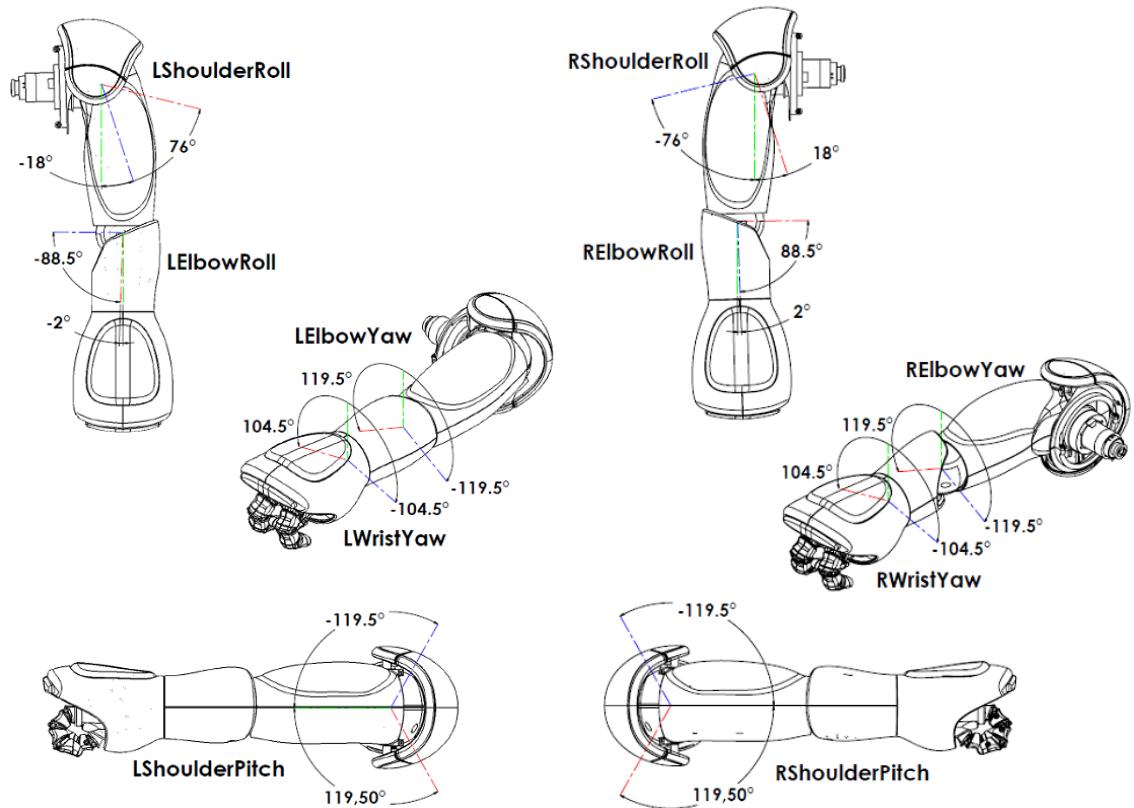


Abb. A.16.: Gelenkwickelbegrenzungen beider Arme. [Ald13]

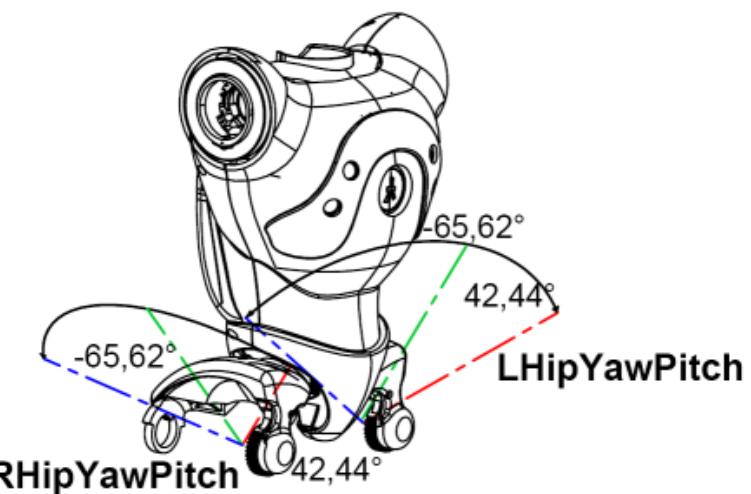


Abb. A.17.: Gelenkwickelbegrenzungen des Torsos. [Ald13]

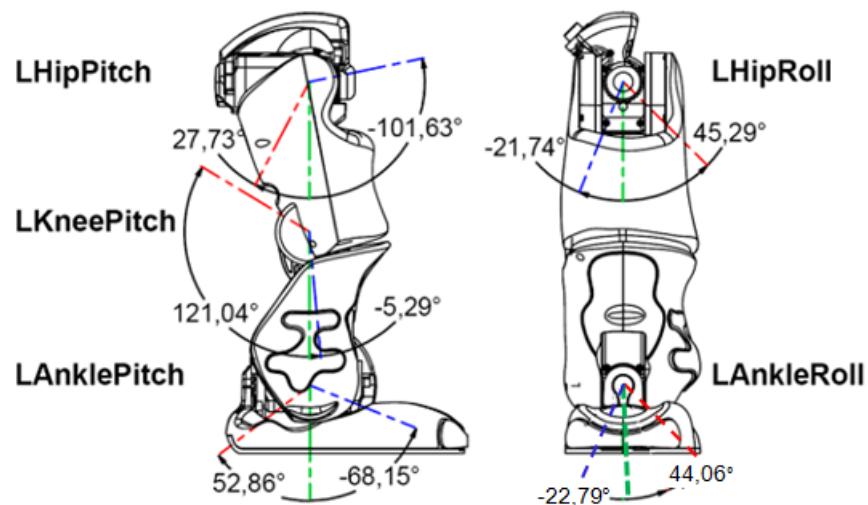


Abb. A.18.: Gelenkumfangsgrenzen des linken Beins. [Ald13]

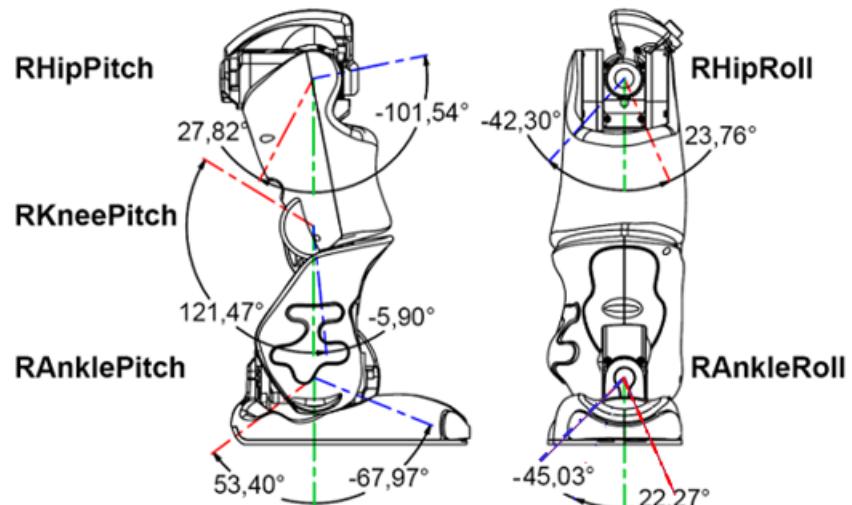


Abb. A.19.: Gelenkumfangsgrenzen des rechten Beins. [Ald13]

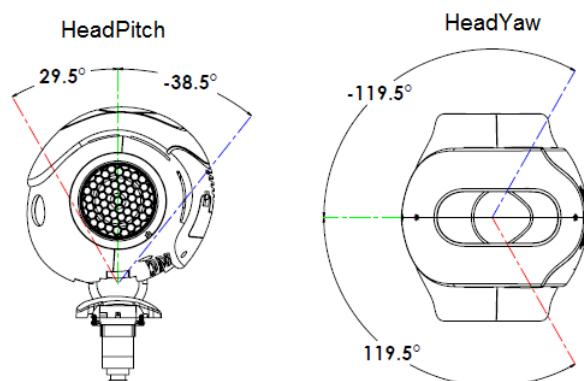


Abb. A.20.: Gelenkumfangsgrenzen des Kopfes. [Ald13]

## A.4. Reglerparameter

### Simulationsspezifische Parameter

- $K_P = 0, 2$
- $T_n = 0, 05$
- $T_v = 0, 008$
- $K_{Aw} = 1, 5$

### Realitätsspezifische Parameter

- $K_P = 0, 04$
- $T_n = 0, 012$
- $T_v = 0, 0005$
- $K_{Aw} = 1, 5$

## B. Inhalte der CD

- Bericht
- Messwerte vom Realversuch
- Quellcode

# Literaturverzeichnis

- [Ald13] ALDEBARAN ROBOTICS (Hrsg.): *NAO Software documentation.* 1.14. Aldebaran Robotics, 2013
- [DD09] DANKERT, Jürgen ; DANKERT, Helge: *Technische Mechanik Statik, Festigkeitslehre, Kinematik/Kinetik.* 5. Vieweg+Teubner, 2009
- [GR11] GRAF, Colin ; RÖFER, Thomas: A Center of Mass Observing 3D-LIPM Gait for the RoboCup Standard Platform League Humanoid. In: *RoboCup 2011: Robot Soccer World Cup XV* (2011), S. 26–36
- [HJ10] HYUN, Cheollwan ; JO, Sungho: Dynamically Stable Movement Generation of a Humanoid Robot from Demonstration: Kicking a Ball. (2010)
- [Hol12] HOLST, B. Sc. Nico D.: *Erkennung und Klassifikation von Objekten zur Selbstlokalisierung durch digitale Bildverarbeitung unter Verwendung der integrierten HD-Kameras des NAO-Robotiksystems.* 2012
- [Kau11] KAUFMANN, Dipl.-Ing S.: *Implementierung und Analyse verschiedener Regelstrategien für dynamische Laufbewegungen humanoider Roboter auf Basis des NAO-Robotiksystems,* Technische Universität Hamburg-Harburg, Diplomarbeit, 2011
- [KKK<sup>+</sup>01] KAJITA, Shuuji ; KANEHIRO, Fumio ; KANEKO, Kenji ; YOKOI, Kazuhito ; HIRUKAWA, Hirohisa: The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern. (2001), S. 239–246
- [MLR10] MÜLLER, Judith ; LAUE, Tim ; RÖFER, Thomas: Kicking a Ball - Modeling Complex Dynamic Motions for Humanoid Robots. In: *RoboCup 2010: Robot Soccer World Cup XIV* (2010), S. 110–120
- [MSF05] MANN, Prof. Dipl.-Ing H. ; SCHIFFELGEN, Prof. Dipl.-Ing H. ; FRORIEP, Prof. Dipl.-Ing R.: *Einführung in die Regelungstechnik Analoge und digitale Regelung, Fuzzy-Regler, Regler-Realisierung, Software.* 10. Hanser, 2005
- [SK08] SICILIANO, B. ; KHATIB, O.: *Handbook of Robotics.* Springer, 2008. – 361–389 S.
- [VB04] VUKOBRAZOVIĆ, Miomir ; BOROVAC, Branislav: Zero-Moment-Point - Thirty Five Years Of Its Life. In: *International Journal of Humanoid Robotics* 1 (2004), Nr. 1, S. 157–173

- [Wer11] WERNER, Prof. Dr. H.: *Introduction to Control Systems.* September 2011