

---

# **Einsatz propriozeptiver Sensorik in der Bewegungssteuerung humanoider Roboter**

---

Der Fakultät für Elektrotechnik und Informationstechnik  
der Technischen Universität Dortmund  
vorgelegte

## **Dissertation**

zur Erlangung des akademischen Grades  
Doktor der Ingenieurwissenschaften (Dr.-Ing.)

von

**Dipl.-Inf. Sören Kerner**

Dortmund, 2013

Hauptreferent:  
Korreferent:

Prof. Dr. Uwe Schwiegelshohn  
Prof. Dr. rer. nat. Thomas Christaller



Für die wichtigsten Menschen in meinem Leben

Tabea und Sora





---

# Inhaltsverzeichnis

---

<b>Abbildungsverzeichnis</b>	<b>iv</b>
<b>Tabellenverzeichnis</b>	<b>viii</b>
<b>Abkürzungsverzeichnis</b>	<b>x</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Robotik . . . . .	2
1.2 Roboterfußball . . . . .	5
1.3 Zielsetzung und Übersicht der Arbeit . . . . .	6
<b>2 Humanoide Roboter</b>	<b>7</b>
2.1 Begriffsdefinition . . . . .	9
2.2 Koordinatensysteme . . . . .	15
2.2.1 Weltkoordinatensystem . . . . .	18
2.2.2 Roboterzentrisches Koordinatensystem . . . . .	18
2.2.3 After-Preview Koordinatensystem . . . . .	19
2.2.4 Körpereigenes Koordinatensystem . . . . .	19
2.2.5 Robotereigenes Koordinatensystem . . . . .	20
2.2.6 Fußkoordinatensystem . . . . .	20
2.2.7 Gait-Koordinatensystem . . . . .	20
2.3 State-of-the-Art Roboterplattformen . . . . .	21
2.3.1 ASIMO . . . . .	22
2.3.2 HRP . . . . .	23
2.3.3 QRIO . . . . .	24
2.3.4 NAO . . . . .	24
2.3.5 DARwIn-OP . . . . .	25
<b>3 Die Roboterplattform NAO</b>	<b>27</b>
3.1 Kinematische Struktur . . . . .	29
3.1.1 Direkte Kinematik . . . . .	31
3.1.2 Inverse Kinematik . . . . .	33
3.2 Motoren . . . . .	42
3.3 Sensoren . . . . .	45
3.3.1 Inertialsensoreinheit . . . . .	46
3.3.2 FSR-Sensoren . . . . .	52

3.3.3	Gelenkwinkelencoder . . . . .	57
3.3.4	Motorstrom . . . . .	59
3.3.5	Exterozeptive Sensorik . . . . .	60
3.4	Zentrale Recheneinheit . . . . .	61
3.5	Steuerungscomputer . . . . .	61
3.6	Software . . . . .	62
3.7	Stromversorgung . . . . .	63
<b>4</b>	<b>Propriozeption mittels Sensorik</b>	<b>65</b>
4.1	Lage im Raum . . . . .	66
4.1.1	Gelenkpositionen . . . . .	66
4.1.2	Inertialsensoren . . . . .	68
4.1.3	Optische Kamera . . . . .	69
4.2	Bestimmung des Schwerpunktes . . . . .	70
4.3	Theorie des Gleichgewichts . . . . .	71
4.3.1	Zero Moment Point . . . . .	74
4.4	Messung des Zero Moment Point . . . . .	77
4.4.1	Fußdruck-Sensoren . . . . .	78
4.4.2	Inertialsensoren . . . . .	88
4.4.3	Kraft-Drehmoment-Sensoren . . . . .	94
4.4.4	Experimenteller Vergleich der ZMP-Messverfahren . . . . .	96
4.4.5	Beurteilung der Messverfahren . . . . .	106
<b>5</b>	<b>Fortbewegung zweibeiniger Roboter</b>	<b>109</b>
5.1	Zyklische Bewegungssteuerung . . . . .	110
5.2	NDD-Walking-Engine . . . . .	113
5.2.1	Erzeugung von Laufmustern . . . . .	116
5.2.2	Regelung der Bewegung . . . . .	121
5.2.2.1	Preview-Regler . . . . .	122
5.2.2.2	Integration des Beobachter-Modells . . . . .	126
5.3	Parametrisierung . . . . .	131
5.4	Evaluation . . . . .	134
5.4.1	Simulation . . . . .	135
5.4.2	Realer Roboter . . . . .	137
5.4.3	Energieeffizienz . . . . .	142
5.4.4	Diskussion und Ausblick . . . . .	144
<b>6</b>	<b>Propriozeption der Bewegung</b>	<b>147</b>
6.1	Odometrie . . . . .	148
6.1.1	Messung der Odometrie zweibeiniger Roboter . . . . .	149
6.1.2	Experimentelle Odometriemessung . . . . .	155
6.2	Messung odometrischer Fehler mittels optischer Sensoren . . . . .	163
6.2.1	Einführung in die Grundlagen optischer Sensoren . . . . .	169
6.2.2	Messung rotatorischer und translatorischer Bewegung mittels optischer Sensoren . . . . .	173
6.2.3	Bewegungsmessung am NAO . . . . .	176
6.2.3.1	Kalibrierung der Sensoren . . . . .	178
6.2.3.2	Evaluation der Sensor-Schuhe . . . . .	179

6.2.4	Laufbewegung . . . . .	183
6.2.5	Vergleich optischer Sensoren . . . . .	185
6.2.6	Autonome Schussbewegung . . . . .	190
6.2.7	Diskussion und Ausblick . . . . .	193
6.3	Bewegungsschätzung im Raum . . . . .	194
6.3.1	Modellierung mit globalen Koordinaten . . . . .	195
6.3.2	Zustandsschätzung ohne Nebenbedingungen . . . . .	197
6.3.3	Projektion der Nebenbedingungen . . . . .	198
6.3.4	Evaluation und Diskussion . . . . .	201
<b>7</b>	<b>Diskussion und Ausblick</b>	<b>207</b>
	<b>Danksagungen</b>	<b>211</b>
	<b>Betreute Arbeiten im Kontext der Dissertation</b>	<b>213</b>
	<b>Bildnachweis</b>	<b>215</b>



---

## Abbildungsverzeichnis

---

1.1.1	Prozesskette der autonomen Robotik. . . . .	4
2.0.1	<i>Unscanny Valley</i> . . . . .	8
2.1.2	<i>Support</i> -Polygon während der <i>Double-Support</i> -Phase rot schraffiert. . .	14
2.2.3	Übersicht der Koordinatensysteme. . . . .	17
2.3.4	ASIMO. . . . .	22
2.3.5	HRP-4C und HRP-2. . . . .	23
2.3.6	QRIO. . . . .	24
2.3.7	NAO. . . . .	24
2.3.8	DARwIn-OP. . . . .	25
3.1.1	Abmessungen des Roboters NAO entnommen aus [1]. . . . .	29
3.1.2	Gelenke des NAO entnommen aus [1]. . . . .	31
3.1.3	Inverse Kinematik des NAO. . . . .	37
3.1.4	Arbeitsraum der Gelenke des rechten Fußes entnommen aus [1]. . . . .	41
3.3.5	Rauschverhalten des V2 Beschleunigungssensors über $t \approx 1$ h. . . . .	49
3.3.6	Rauschverhalten des V2 Gyrometers über $t \approx 1$ h. . . . .	49
3.3.7	Rauschverhalten des V3.3 Beschleunigungssensors über $t = 120$ s. . . .	50
3.3.8	FlexiForce A201 Kalibrierung. . . . .	54
3.3.9	Rauschverhalten des FlexiForce A201A FSR-Sensors über $t = 110$ s. . .	54
3.3.10	Wiederholte Belastung des FlexiForce A201A FSR-Sensors mit 1 kg. .	56
3.3.11	Rotationsfehler des AS5055 abhängig von der Positionierung des Sensors entnommen aus [4]. Die Legende gibt den Fehler in Grad an. . . .	59
3.3.12	Messung des Stromsensors des <i>KneePitch</i> -Gelenks mit <i>Stiffness</i> =1 während eines Laufs. . . . .	60
4.1.1	Fehlerfortpflanzung der Lagebestimmung aus Encodern. . . . .	67
4.4.2	Position der FSR-Sensoren basierend auf [1]. . . . .	79
4.4.3	Simulation 4.4.1.A - FSR vorne rechts - konstanter Bias von 100g. . .	81
4.4.4	Simulation 4.4.1.B - FSR-Gainfehler. . . . .	83
4.4.5	Simulation 4.4.1.B - Kombiniertes FSR-Gainfehler, vorne rechts - Faktor 0,8 und vorne links - Faktor 1,2. . . . .	84
4.4.6	Simulation 4.4.1.C - Fehler durch Übersprechen von vorne rechts mit 0,2 auf vorne links und hinten rechts. . . . .	85
4.4.7	Simulation 4.4.1.D - Fehler durch Rauschen mit Standardabweichung 75g. . . . .	87

4.4.8	Simulation 4.4.1.E - Ausfall FSR vorne rechts. . . . .	88
4.4.9	Schematische Darstellung des <i>Cart-Table</i> -Modells. . . . .	89
4.4.10	Simulation 4.4.2.A - Fehler in $\ddot{x}$ von +10 Prozent. . . . .	91
4.4.11	Simulation 4.4.2.B - Fehler in der $x/y$ -Position des CoM. . . . .	93
4.4.12	Simulation 4.4.2.C - Fehler in $z_c$ von -30 mm. . . . .	94
4.4.13	ZMP-Berechnung aus 6-Achs-Kraftsensor. . . . .	95
4.4.14	Experiment 4.4.4.A - Stand auf linkem Bein. . . . .	98
4.4.15	Experiment 4.4.4.B - Sturz in $x$ -Richtung. . . . .	100
4.4.16	Experiment 4.4.4.C - Sturz in $y$ -Richtung. . . . .	102
4.4.17	Experiment 4.4.4.D - Kollision mit einer Masse. . . . .	104
4.4.18	Experiment 4.4.4.E - Schussbewegung. . . . .	105
5.2.1	Übersicht der NDD-WE Struktur entnommen aus [25]. . . . .	118
5.2.2	CoM-Verlauf um eine gegebene ZMP-Trajektorie zu erreichen. . . . .	120
5.2.3	Referenz-ZMP, berechneter ZMP und CoM für unterschiedliche $N$ der <i>Preview</i> -Verstärkung. . . . .	126
5.2.4	<i>Observer</i> -Integration in die NDD-WE. . . . .	128
5.2.5	Konfiguration des Regelungssystems. . . . .	130
5.2.6	Reaktion des Reglers auf einen konstanten ZMP-Fehler, beginnend bei $t = 1,5$ s und endend bei $t = 3,0$ s . . . . .	131
5.4.7	Experiment 5.4.1.A - Unebener Boden. . . . .	136
5.4.8	Experiment 5.4.1.B - Externe Kräfte. . . . .	136
5.4.9	Experiment 5.4.1.C - Modellfehler. . . . .	137
5.4.10	CoM-Trajektorie während eines omnidirektionalen Laufs. . . . .	138
5.4.11	Experiment 5.4.2.A - <i>Close-Loop</i> -Lauf. . . . .	139
5.4.12	Experiment 5.4.2.B - Externe Kräfte. . . . .	140
5.4.13	Experiment 5.4.2.C - Unebener Boden. . . . .	141
5.4.14	Experiment 5.4.2.C - Ergebnisse. . . . .	141
5.4.15	Vergleich der Leistungsaufnahme eines <i>KneePitch</i> -Gelenks bei unter- schiedlicher Schwerpunkthöhe. . . . .	144
6.1.1	Diskrete Positionsänderung bei kontinuierlichem CoM-Verlauf. . . . .	150
6.1.2	Resultierende Abweichung $\Delta_x$ aus dem Winkelfehler des Schwung- und Standbeins. . . . .	153
6.1.3	Modellierung des Fußes mit Federn. . . . .	154
6.1.4	Experiment 6.1.2.A - Lauf in $x$ -Richtung mit $v = 200$ mm/s. . . . .	156
6.1.5	Experiment 6.1.2.A - Lauf in $x$ -Richtung mit $v = 100$ mm/s. . . . .	157
6.1.6	Experiment 6.1.2.B - Lauf in $y$ -Richtung mit $v = 50$ mm/s. . . . .	161
6.1.7	Experiment 6.1.2.C - Drehung um $z$ -Achse mit $v = 0.4$ rad/s. . . . .	162
6.2.8	Odometriefehler über die Zeit. . . . .	165
6.2.9	Drehmoment erzeugt durch Schwungfuß. . . . .	166
6.2.10	Vergleich einer LED- und Laser-Maus, entnommen aus [71]. . . . .	170
6.2.11	BlueTrack Schematik entnommen aus [77]. . . . .	171
6.2.12	Darkfield Laser Tracking <sup>TM</sup> , entnommen aus [71]. . . . .	172
6.2.13	Überlagerung des Bildfeldes bei Bewegung. . . . .	173
6.2.14	Bewegungsmessung mittels optischer Sensoren. . . . .	174
6.2.15	Unterschiedliche Gleitwege. . . . .	175
6.2.16	Sensorschuhe des NAO. . . . .	177

6.2.17	Sensor- und Empfängerplatine. . . . .	178
6.2.18	Sensorkalibrierung auf unterschiedlichen Untergründen. . . . .	179
6.2.19	Einfluss der Sensorposition auf die Messung. . . . .	180
6.2.20	Experiment 6.2.3.2.A - Manuelle Translation in $y$ -Richtung. . . . .	180
6.2.21	Experiment 6.2.3.2.B - Manuelle Rotation. . . . .	181
6.2.22	Experiment 6.2.3.2.C - Manuelle kombinierte Bewegung. . . . .	182
6.2.23	Exemplarische Schrittauswertung eines Laufs. . . . .	184
6.2.24	Experiment 6.2.4.A - Lauf in $x$ -Richtung mit $v_x = 100$ mm/s. . . . .	185
6.2.25	Experiment 6.2.5.A - Grüner Teppich. . . . .	186
6.2.26	Experiment 6.2.5.B - Zuverlässigkeit Wireless Mobile Mouse 4000. . . . .	189
6.2.27	Schemazeichnung Sensorschuhe MS-4000. . . . .	190
6.2.28	Experiment 6.2.6.A - Rotation Sensorschuh MS-4000. . . . .	191
6.2.29	Experiment 6.2.6.B - Schuss aus Stand. . . . .	192
6.3.30	Prozess der Zustandsschätzung ohne Nebenbedingungen. . . . .	198
6.3.31	Experiment 6.3.4.A - Lauf in $x$ -Richtung. . . . .	203
6.3.32	Experiment 6.3.4.B - Rotation um eigene Achse. . . . .	204
6.3.33	Experiment 6.3.4.C - Unvorhergesehene Kollision. . . . .	205





---

## Tabellenverzeichnis

---

3.1.1 Gelenkzuordnung der kinematischen Ketten des NAO. . . . .	30
3.1.2 DH-Parameter der kinematischen Kette des linken Beins. . . . .	33
3.1.3 DH-Parameter der kinematischen Kette des rechten Beins. . . . .	33
3.2.4 Leistungsdaten der Motoren des Roboters NAO. . . . .	43
3.2.5 Gelenkzuordnung der Motoren und Getriebe. . . . .	44
3.3.6 Messintervalle des AS5055 [4]. . . . .	58
4.4.1 Positionen der FSR-Sensoren im Fußkoordinatensystem. . . . .	80
4.4.2 Simulation 4.4.1.A - Fehler für konstanten Bias. . . . .	82
4.4.3 Simulation 4.4.1.B - Fehler im FSR-Gain. . . . .	84
4.4.4 Simulation 4.4.1.C - Fehler durch Übersprechen. . . . .	86
4.4.5 Simulation 4.4.1.D - Fehler durch Rauschen mit Standardabweichung 75g. . . . .	86
4.4.6 Simulation 4.4.2.A - Fehler in $\ddot{x}$ von 10 Prozent. . . . .	91
4.4.7 Simulation 4.4.2.B - Fehler in der $x/y$ -Position des CoM. . . . .	92
4.4.8 Simulation 4.4.2.C - Fehler in $z_c$ von 20 mm. . . . .	93
5.4.1 Experiment 5.4.1.A - Unebener Boden. . . . .	135
5.4.2 Experiment 5.4.1.B - Externe Kräfte. . . . .	136
5.4.3 Experiment 5.4.1.C - Modellfehler. . . . .	137
5.4.4 Durchschnittliche Energieaufnahme eines <i>KneePitch</i> -Gelenks über einen Zeitraum von 5 Sekunden beim Lauf in $x$ -Richtung mit $v_x = 200$ mm/s. . . . .	143
6.2.1 Experiment 6.2.5.A(a) - Grüner Teppich. . . . .	187
6.2.2 Experiment 6.2.5.A(b) - Grüner Teppich. . . . .	187



---

## Abkürzungsverzeichnis

---

3D-LIPM	.....	3D-Linear-Inverted-Pendulum-Modell
ARW	.....	Angular Random Walk
BRW	.....	Bias Random Walk
CoM	.....	Center of Mass
CoP	.....	Centure of Pressure
CPG	.....	Central Pattern Generator
DH	.....	Denavit-Hartenberg
DoF	.....	Degree of Freedom
DPI	.....	dots-per-inch
EKF	.....	Extended Kalman-Filter
FRI	.....	Foot Rotation Indicator
FZMP	.....	Fictitious ZMP
GKS	.....	Gait-Koordinatensystem
KeKS	.....	körper eigenes Koordinatensystem
KI	.....	Künstliche Intelligenz
MSL	.....	Middle-Size-League
NDD	.....	NAO Devils Dortmund
ODE	.....	Open Dynamics Engine
pCoM	.....	projected Center of Mass
QMB-UKF	.....	Quaternion MultiBody Unscented-Kalman-Filter
ReKS	.....	robotereigenes Koordinatensystem
RzKS	.....	roboterzentrisches Koordinatensystem
SLL	.....	Small-Size-League

SPL .....	Standard-Platform-League
TCP .....	Tool Center Point
UKF .....	Unscented Kalman-Filter
VRW .....	Velocity Random Walk
WE .....	Walking-Engine
WKS .....	Weltkoordinatensystem
ZMP .....	Zero Moment Point

## Einleitung

---

Von jeher ist der Mensch fasziniert davon, Maschinen zu entwerfen, die ihm die Arbeit erleichtern sollen. Bereits steinzeitliche Erfindungen wie der Hammer oder das Messer belegen dies. Mit fortschreitender Entdeckung von physikalischen und chemischen Zusammenhängen wurden diese Arbeitsmaschinen im Lauf der Jahre immer komplexer. Aus einfachen mechanischen Mühlen wurden Dampfmaschinen. Im Laufe der Zeit entwickelte sich immer mehr der Wunsch nach Werkzeugen, die dem Menschen nicht nur seine Arbeit erleichtern, sondern Arbeiten ganz für ihn übernehmen können. Der Schriftsteller Karel Čapek<sup>1</sup> prägte 1920 für diese Maschinen in seinem Bühnenstück „Rossum’s Universal Robots“ den Begriff *Roboter*. Diesem liegt das tschechische Wort *robotá* zugrunde, welches soviel bedeutet wie „Fronarbeit“ oder „erzwungene Arbeit“. Verbreitung erlangte der Begriff dann vor allem durch Science-Fiction-Autoren. Einer der bedeutendsten war mit Sicherheit Isaac Asimov, bekannt für seine *The Three Laws of Robotics*. Aus diesem Grund prägte sich ein sehr futuristisches Bild vom Aussehen dieser Maschinen, das bis in die heutige Zeit Bestand hat. Der Erforschung dieser Maschinen hat sich die Wissenschaft der *Robotik* verschrieben. Bereits sehr früh zeigt sich hierbei die Faszination des Menschen, Maschinen nach seinem Ebenbild zu bauen. Al-Jazari entwarf schon im Jahre 1205 etliche Gerätschaften unter denen sich auch programmierbare Automaten in Menschengestalt befanden. Auch der Tee-Automat in Form einer Frau, den der arabische Ingenieur al-Dschazari zur selben Zeit entwarf, zeugt von diesem Streben. Als erste Dokumentation eines Roboters in vollständig humanoider Gestalt gelten die Entwürfe von Leonardo Da Vinci für einen mechanischen Ritter aus dem Jahre 1495. Um dem Menschen im Alltag sinnvoll zur Hand gehen zu können, muss ein Roboter mobil sein. Die ersten solchen bedienten sich eines fahrbaren Körpers und gehören somit zu den *rollenden* Robotern. Abseits der Faszination des Menschen eine Maschine nach seinem Ebenbild zu erschaffen birgt das Konzept eines laufenden humanoiden Roboters einige Vorteile gegenüber einem rollenden Roboter. Der Mensch hat seinen Lebensraum über die Jahre immer mehr seinen Vorlieben angepasst. Dadurch

---

<sup>1</sup>Es existieren widersprüchliche Aussagen, ob der Begriff nicht ursprünglich auf Karels Bruder Josef zurückzuführen ist.

hat er eine Umgebung geschaffen, die für seinen Bewegungsapparat geeignet ist. Erst in den letzten Jahren wird verstärkt Wert darauf gelegt, dass auch Menschen mit einer Gehbehinderung sich frei in Städten bewegen können, aber dennoch sind Rollstuhlfahrer im Alltag stark eingeschränkt. Vor eben diesem Problem stehen rollende Roboter, für die eine Treppe, oder bereits ein Bordstein, ein Bewegungshindernis darstellen, das sie nicht ohne weiteres überwinden können. Dennoch ist der mit Abstand größte Teil der entwickelten und sich im Einsatz befindlichen Roboter nicht humanoider Gestalt. Der Grund hierfür ist sehr einfach - der menschliche Bewegungsapparat ist höchst komplex. So stellen bereits die Planung und Ausführung von einfachen Bewegungen, wie beispielsweise dem Laufen, algorithmische Probleme dar, die trotz Jahren der Forschung bislang nicht als gelöst gelten können. Besonders die Einhaltung der Nebenbedingung der Stabilität erweist sich noch immer als Herausforderung. Trotz enormer Fortschritte der vergangenen Jahre sind zweibeinige Roboter daher vergleichsweise langsam und instabil in ihrer Fortbewegung. Von dem erträumten Einsatz als Helfer des Menschen im Alltag sind sie noch weit entfernt und es wird noch Jahre der Forschung benötigen diesen Wirklichkeit werden zu lassen. Dennoch zeigen herausragende Roboter, wie der ASIMO, dass dieses Ziel mittlerweile in greifbare Nähe gerückt ist.

### 1.1 Robotik

Versucht man den Begriff *Roboter* genauer zu definieren, steht man vor einem Problem - es gibt keine Definition. Oder besser, es existiert keine *einheitliche* Definition. Ohne Anspruch auf Vollständigkeit seien daher hier exemplarisch einige aufgeführt:

„It’s a machine that can sense and act and react in the world and possibly involves some reasoning for performing these actions, and it does so autonomously. By that definition a thermostat would be a robot. Though it’s not ‘aware’ it has a goal, that awareness isn’t required.“ [Alan Mackworth, President der *American Association for Artificial Intelligence*]

„To me a robot is something that has some physical effect on the world, but it does it based on how it senses the world and how the world changes around it.“ [Rodney Brooks, Leiter des *Massachusetts Institute of Technology computer science and artificial intelligence laboratory*]

„They have to have a way of making measurements of the world, they have to have a way of making decisions — in other words, something like a computer, you could call that thinking informally — and they have to have a way taking actions.“ [Gregory Dudek, Leiter des *Centre for Intelligent Machines at McGill University*]

Am treffendsten beschreibt es wohl der „*Father of Robotic*“ Joseph Engelberger, Erfinder des ersten Industrieroboters:

„I can’t define a robot, but I know one when I see one.“ [Joseph Engelberger]

Ursache für diese Uneinigkeit ist vor allem die Vielfalt, der unterschiedlichen Maschinen, die im Volksmund als Roboter bezeichnet werden. Aus diesem Grund ist die Robotik dazu übergegangen keine einheitliche Definition zu suchen, sondern vielmehr Roboter in unterschiedliche Klassen zu unterteilen. Heutzutage gehört noch immer die Vielzahl der Roboter der Gruppe der *Industrieroboter*, englisch *Robotic Manipulator*, an:

**DIN EN ISO 8373:** „Ein Industrieroboter ist ein automatisch gesteuerter, frei programmierbarer [...] Mehrzweck- [...] Manipulator [...], der in drei oder mehr Achsen [...] programmierbar ist und zur Verwendung in der Automatisierungstechnik entweder an einem festen Ort oder beweglich angeordnet sein kann“

**VDI-Richtlinie 2860:** „Industrieroboter sind universell einsetzbare Bewegungsautomaten mit mehreren Achsen, deren Bewegungen hinsichtlich Bewegungsfolge und Wegen bzw. Winkeln frei (d. h. ohne mechanischen Eingriff) programmierbar und gegebenenfalls sensorgeführt sind. Sie sind mit Greifern, Werkzeugen oder anderen Fertigungsmitteln ausrüstbar und können Handhabungs- und/oder Fertigungsaufgaben ausführen.“

Auch wenn die Deutsche Industrienorm (DIN) bewegliche Industrieroboter zulässt, so sind diese doch in der Regel fest montiert und somit auf einen bestimmten Arbeitsbereich beschränkt. Daher stellt die Robotik diesen die Klasse der *mobilen Roboter* gegenüber. Eine für diese Arbeit wichtige Untergruppe hiervon bilden mobile Roboter, die *autonom* sind. Der Begriff Autonomie wird in diesem Kontext wie folgt definiert:

„Autonomy refers to systems capable of operating in the real-world environment without any form of external control for extended periods of time.“  
[101]

Roboter, die diese Anforderung erfüllen, werden als *autonome* Roboter bezeichnet:

„A mobile robot is an autonomous system capable of traversing a terrain with natural or artificial obstacles. Its chassis is equipped with wheels/tacks or legs and possibly a manipulator setup mounted on the chassis for handling of work pieces, tools or special devices. Various preplanned operations are executed based on a pre-programmed navigation strategy taking into account the current status of the environment.“ [28]

Doch auch diese Einschränkung des Begriffs eines Roboters lässt noch Maschinen unterschiedlichster Art und Ausprägung zu. Allerdings ergibt sich trotzdem eine Gemeinsamkeit. Alle autonomen Roboter müssen in ihrer Umwelt agieren können, hierzu selbstständig Entscheidungen treffen und benötigen dazu Sensoren, um ihre Umgebung wahrzunehmen. Daraus ergibt sich für alle autonomen Roboter die gleiche Prozesskette als Grundlage ihrer Handlung - Wahrnehmen, Schlussfolgern, Agieren. Legt man diesen Ablauf zugrunde, unterteilt sich die autonome Robotik in drei Hauptforschungsgebiete, wie Abbildung 1.1.1 verdeutlicht. Die genaue Bezeichnung unterscheidet sich je nach Autor, doch die grundlegende Unterteilung ist allen gemein.

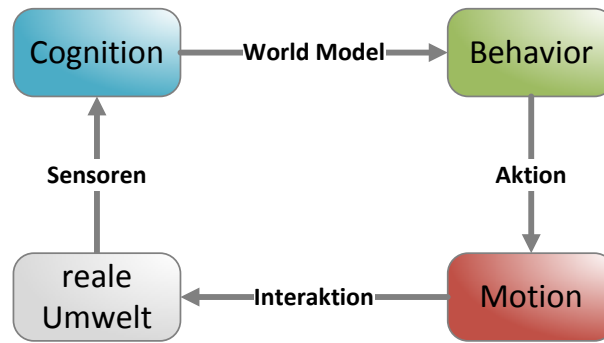


Abbildung 1.1.1: Prozesskette der autonomen Robotik.

**Cognition:** Mit dem Begriff *Cognition*, zu deutsch *Wahrnehmung*, wird die Fähigkeit des Roboters bezeichnet, seine Umwelt zu erfassen. Dies umfasst primär die Auswertung von Sensoren, mit denen der Roboter Informationen über seine Umgebung aufzeichnen kann. Ziel ist es aus den Daten Merkmale, sogenannte *Features*, zu extrahieren, die der Roboter logisch zuordnen kann. Auf deren Basis erstellt die *Cognition* ein Modell der Umwelt, auch als *World Model* bezeichnet. Der Roboter selbst ist Teil dieses Modells, so dass sowohl die *Lokalisierung*, also die Bestimmung seiner Position relativ zur Umgebung, als auch die *Propriozeption*, also die Eigenwahrnehmung, in den Aufgabenbereich der *Cognition* fällt.

**Behavior:** Der Bereich *Behavior*, oder zu deutsch *Verhaltensplanung*, übernimmt die Entscheidungsfindung des autonomen Roboters. Basierend auf der Modellierung der *Cognition* wird eine Entscheidung über die Aktion getroffen, die den Roboter seinem Ziel näher bringt. Als Grundlage des *Behavior* wird also in jedem Fall eine roboterverständliche Beschreibung einer Aufgabe vorausgesetzt. Hier finden sich viele Aufgabengebiete der *Künstlichen Intelligenz (KI)* wieder. Durch die Anwendung auf die Robotik und die damit verbundene Verknüpfung mit Unsicherheiten ist die Verhaltensforschung umfassender als die klassische KI-Forschung und erfordert die Entwicklung neuer Methoden.

**Motion:** Mit dem Begriff *Motion*, oder zu deutsch *Bewegungssteuerung*, wird die Fähigkeit eines Roboters bezeichnet, physikalische Aktionen auszuführen. Diese haben in der Regel eine Veränderung der Umwelt des Roboters zur Folge, sei es durch die Bewegung des Roboters oder durch direkte Manipulation von Objekten. Die Aufgabe der *Motion* umfasst demnach die Fähigkeit des Roboters mit seiner Umwelt physikalisch zu interagieren. Je nach physikalischer Struktur des Roboters können sich die Problemstellungen dieses Bereiches stark voneinander unterscheiden.

Eingehende Betrachtung zeigt eine enge Verknüpfung der Forschungsbereiche. Die Modellierung der *Cognition* bildet die Grundlage für das *Behavior*. Dessen Entscheidungen bestimmen die Aktionen der *Motion*. Die resultierende Bewegung beeinflusst wiederum maßgeblich die Qualität der Modellierung der *Cognition*. Eine besondere Rolle kommt hierbei der *Propriozeption* der eigenen Bewegung zugute. Betrachtet man die Rekursion dieser Prozesskette, so erklärt sich, warum es nicht sinnvoll ist, diese Bereiche der autonomen Robotik vollständig losgelöst voneinander zu erforschen.



## 1.2 Roboterfußball

Durch die Wahl der Roboterplattform, mit der die Untersuchungen dieser Arbeit ausgeführt werden, ist die Mehrzahl der Experimente dem Kontext des *Roboterfußballs* entnommen. Um die Forschung in der autonomen Robotik voranzubringen und den Austausch der Wissenschaftler zu fördern hat die 1993 gegründete *RoboCup Federation* den *RoboCup*<sup>2</sup> ins Leben gerufen. Entstanden ist die Idee aus dem Versuch Schach als *Standardproblem* der KI-Forschung abzulösen und zugleich deren Anbindung an die Robotik mehr in den Fokus zu rücken. Diese Herausforderung dann in Form eines Wettkampfes anzugehen hat schon beim Schach die Forschung vorangebracht und fördert zudem die Vergleichbarkeit von Lösungen. Keines der drei in Kapitel 1.1 vorgestellten Aufgabengebiete der autonomen Robotik ist vollständig gelöst. Sowohl die Bewegungssteuerung, Wahrnehmung also auch Verhaltensplanung sind aktuelle Forschungsbereiche der Robotik. Deren zirkulare Verknüpfung erschwert die Forschung, da keiner der Bereiche unabhängig vom Rest betrachtet werden kann. Fußball eignet sich daher als Standardproblem, da es für jeden der Bereiche Herausforderungen als *Benchmark* bereit hält. Durch die klare Abgrenzung der Welt und die eindeutige Definition der Regularien des Spiels sind diese Aufgaben im Vergleich zur Problematik einer menschlichen Umgebung stark vereinfacht. Sie sind aber so gehalten, dass durch Skalierung größere Herausforderungen zu kreiert werden können. Zugleich kann durch die Aufgabenstellung der Fokus auf unterschiedliche Bereiche des Spiels gelegt und die Problematiken so zumindest teilweise voneinander entkoppelt werden. Im RoboCup geschieht dies in Form von unterschiedlichen *Ligen*. So findet beispielsweise in der *Simulation-League* das Spiel mit autonomen Agenten rein in Software statt. Die Informationen der Umgebung werden unmittelbar bereitgestellt und von der Bewegungssteuerung des Roboters wird abstrahiert. Die Teams der Liga können sich also gänzlich auf die *Behavior*-Forschung konzentrieren. Doch werden hierdurch wichtige Bereiche der Robotik außen vor gelassen. Daher wird in der verwandten *Small-Size-League* (SSL) mit realen Robotern gespielt. Um von der Problematik der *Cognition* weiterhin zu abstrahieren wird in der Liga ein einheitliches externe Deckenkamerasystem verwendet, das alle Roboter mit sämtlichen Informationen der Umgebung versorgt. Von der Bewegungssteuerung wird weitestgehend abstrahiert, indem die am weitesten erforschte Fortbewegungsart gewählt wird - die rollende. Somit kann weiterhin der Schwerpunkt auf die Entwicklung des Verhaltens gelegt, zugleich aber auch der Einfluss von unscharfer Information und ungenauer Bewegungsausführung auf den Roboter untersucht werden. Diese Verschiebung der Fokussierung setzt sich durch die weiteren Ligen fort und endet bei zweibeinigen Robotern, deren entscheidendes Problem in der *Motion* begründet liegt. Als Folge kommt nur eine vergleichsweise einfache Verhaltensplanung zur Anwendung. Von diesen Ligen gibt es zwei. Zum einen die *Humanoid-League*, in der die Teams die Hardware der Roboter selber bauen und somit primär deren Entwicklung vorangetrieben wird. Zum anderen die *Standard-Platform-League*, kurz SPL, in der alle Teams mit dergleichen Roboterplattform antreten und daher die Entwicklung von Algorithmen im Vordergrund steht.

---

<sup>2</sup>Es darf nicht unerwähnt bleiben, dass auch die *Federation of International Robot-Soccer Association* mit dem *RoboWorld Cup* einen vergleichbaren Wettbewerb organisiert. Im Bereich der universitären Forschung spielt dieser aber nur eine untergeordnete Rolle und wird daher außen vor gelassen.

### 1.3 Zielsetzung und Übersicht der Arbeit

Diese Arbeit verschreibt sich thematisch der Forschung der Bewegungsteuerung zweibeiniger Roboter. Die in diesem Kontext relevanten Ligen des RoboCups sind die beiden zuletzt vorgestellten - *Humanoid-League* und SPL. Beide verfolgen eine andere Philosophie, in deren Unterscheidung sich die primären Fokussierungen der aktuellen Roboterforschung widerspiegelt. Bereits die kurze Einführung im vorangehenden Kapitel hat gezeigt, dass die Robotik ein interdisziplinärer Forschungszeitweig ist. Jedoch wurde bislang nicht betrachtet, dass dies nicht bei der Verknüpfung der Bereiche Sensorik, KI und Bewegungsplanung endet, sondern auch ein enger Bezug zwischen Hardware und Algorithmen in Form von Software besteht. Als Folge lässt sich eine Vielzahl der Herausforderungen der Robotik sowohl durch eine Hard- als auch eine Softwarelösung angehen. Es kann selbstverständlich nie vollkommen auf eine Anpassung der Algorithmik verzichtet werden, doch zeigen die Veröffentlichungen der vergangenen Jahrzehnte die Tendenz neue Hardwarekomponenten zur Lösung zu entwerfen. Bedenkt man, dass die Robotikhardware zu dieser Zeit noch in den Kinderschuhen steckte, ist dieses Vorgehen nachvollziehbar. Eine Fortentwicklung der Hardware führte demnach oft zu einer Verbesserung, die durch eine bessere algorithmische Ausnutzung der vorhandenen Hardware nicht zu erreichen gewesen wäre. Die Kehrseite der Medaille ist eine große Diversität der für die Forschung verwandten Roboterplattformen. Als Folge litt die Vergleichbarkeit entwickelter Algorithmen. Optimale Ergebnisse können in der Robotik jedoch nur aus dem Zusammenspiel von Hard- und Software entstehen. Der aktuelle Trend, leistungsfähige Roboterplattformen käuflich erwerben zu können, ermöglicht es erstmals entwickelte Algorithmen hardwareunabhängig zu vergleichen. Daher kommt zur Evaluation der im Rahmen der Arbeit vorgestellten Algorithmen bewusst eine kommerziellen Plattform in Form des Roboters *NAO* zum Einsatz. Folglich wird diese eingehend untersucht, um die Implikation dieser Wahl für die experimentelle Evaluation zu verstehen. Die Forschungsarbeiten fanden im Rahmen des SPL-Teams *NAO Devils Dortmund* (NDD) des Instituts für Roboterforschung (IRF) statt.

Um einige Begriffe als Grundlage der weiteren Arbeit zu erörtern, kann es nach dieser kurzen Einleitung nicht ausbleiben, in Kapitel 2 auf Besonderheiten der humanoiden Robotik einzugehen. Abgeschlossen wird diese Einführung von einer Übersicht ausgewählter *State-of-the-Art* Roboter. Das anschließende Kapitel 3 befasst sich konkret mit der Hardware des Roboters *NAO*, der als Plattform für die Experimente dieser Arbeit dient. Das Hauptaugenmerk wird hierbei auf die Untersuchung seiner Sensorik gelegt, die einen repräsentativen Querschnitt der modernen Robotik darstellt. Kapitel 4 befasst sich mit dem Einsatz dieser Sensoren zur Eigenwahrnehmung des Roboters. Dieses Wissen nutzt Kapitel 5 als Grundlage zur Untersuchung geeigneter Algorithmen zur Generierung stabiler Fortbewegungsmuster. Besonders im Hinblick auf die Stabilität ist die Prämisse, diese mittels der Sensorik des *NAO* zu überwachen und regulierend eingreifen zu können. Kapitel 6 führt dann die Gedanken aus Kapitel 4 fort und stellt Methoden vor, die Bewegung des Roboters über die Zeit zu verfolgen. Hierbei werden bewusst die Limitierungen des Roboters *NAO* außer Acht gelassen, so dass dieses Kapitel auch als Ausblick in die Entwicklung zukünftiger Roboter zu verstehen ist. Dieser wird nach einem Fazit der Arbeit in Kapitel 7 abgeschlossen.

### Humanoide Roboter

---

Zur Einordnung der Arbeit muss zunächst geklärt werden, wie im Kontext der folgenden Kapitel der Begriff „*humanoider Roboter*“ zu verstehen ist. Die allgemeine Definition eines Roboters wurde bereits einleitend in Kapitel 1.1 diskutiert und es wurde auch auf einige Unterklassen eingegangen, der Bereich der zweibeinigen Roboter allerdings noch außen vor gelassen. Durch die Artenvielfalt dieser Maschinen erweist sich eine Definition als ähnlich problematisch, wie die des allgemeinen Roboters. Aus diesem Grund existiert auch in diesem Fall kein einheitlicher Konsens unter Wissenschaftlern. Daher kann und soll auch an dieser Stelle keine letztendliche Definition postuliert werden, doch ist es für den Leser wichtig, diese im Kontext der Arbeit eindeutig verstehen zu können. Da der Fokus dieser Arbeit auf der Bewegungssteuerung liegt, soll für diese Arbeit gelten:

**Humanoider Roboter:** Ein (vollständiger) humanoider Roboter, ist ein autonomer Roboter, welcher sich ausschließlich mittels eines aufrechten Gangs auf zwei Beinen fortbewegt. Dieser hat zudem einen Torso, der als weitere Extremitäten zwei Arme und einen Kopf besitzt.

Diese Definition ist bewusst allgemein gehalten, um eine möglichst große Anzahl von Robotern einzuschließen. Klar abgrenzen soll sie aber Laufmaschinen von Robotern, die einen dem Menschen nachempfundenen Torso haben, aber zur Fortbewegung Räder verwenden. Deren Art der Fortbewegung stellt eine grundlegend andere Herausforderung dar. Des Weiteren gibt es etliche Forschungen an Laufmaschinen mit zwei Beinen, die aber den Torso eines Menschen vermissen. Auch wenn sich viele so erforschte Konzepte auf einen vollständigen humanoiden Roboter übertragen lassen, so unterscheidet sich die Problematik dennoch in wesentlichen Teilen. Die zukünftige Akzeptanz der Präsenz von Robotern im alltäglichen Leben des Menschen ist weiterhin ungeklärt. Bereits frühe Studien von Mori [79] zeigen ein Phänomen auf, das er als *unscanny valley* bezeichnet, vergleiche Abbildung 2.0.1<sup>1</sup>. Seine Untersuchungen belegen, dass ab einem gewissen

---

<sup>1</sup>Copyright ©2005 Karl F. MacDorman and Takashi Minato. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, V1.2.

Grad der Ähnlichkeit zwischen Roboter und Menschen die Vertrautheit im Umgang mit dieser Maschine rapide sinkt, um erst wieder anzusteigen, wenn der Roboter das Aussehen und die Bewegung des Menschen nahezu perfekt nachahmt. Moris Arbeit zeigt aber auch, dass vor diesem *uncanny valley* ein humanoides Äußeres der Akzeptanz der Maschine durchaus zuträglich ist. Einen analogen Schluss lassen auch moderne Studien, wie beispielsweise [12] oder [61], zu und zeigen zudem weitere Vorteile für die Mensch-Maschine-Kommunikation auf. Dies bekräftigt die Entscheidung, die Untersuchungen dieser Arbeit lediglich auf humanoide Roboter, entsprechend der gegebenen Definition, zu fokussieren, auch wenn die Arme des Roboters keinen entscheidenden Einfluss auf die Algorithmen haben. Die Beschränkung auf humanoide Körperproportionen gewährleistet zudem den Ergebnissen der Arbeit zukünftig als Basis für Forschungen dienen zu können, die Gliedmaßen explizit miteinbeziehen.

Die Definition fordert weiterhin, dass der Roboter in der Lage ist, autonom, also selbstständig, Entscheidungen zu treffen. Auch wenn Anwendungen für ferngesteuerte humanoide Roboter ebenfalls existieren - Telepräsenz bei Operationen [92] oder *Rescue*-Robotik sind hier sicherlich legitime Beispiele<sup>2</sup> - so ist Autonomie für die Mehrzahl der Applikationen dennoch von Vorteil. Diese Beschränkung erscheint also sinnvoll und es sollte beim Lesen der Arbeit bedacht werden, dass die entwickelten Algorithmen primär auf den praktischen Einsatz auf mobiler Computerhardware zielen und somit Rechenleistung einen relevanten Faktor darstellt. Mit Hinblick auf die fortschreitende Entwicklung mobiler CPUs, sollen jedoch auch Algorithmen für zukünftige Applikationen nicht außer Acht gelassen werden.

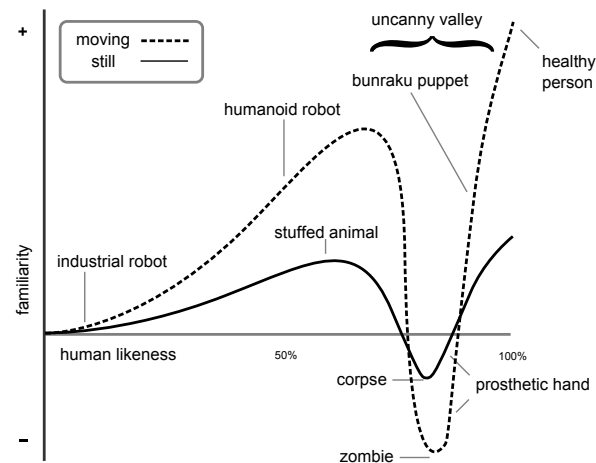


Abbildung 2.0.1: *Unscanny Valley*.

Zur Einführung in die Herausforderungen der humanoiden Robotik folgt nun zunächst ein Überblick über deren Grundlagen. Um das eindeutige Verständnis der Arbeit zu fördern, wird damit begonnen einige Begrifflichkeiten der Robotik klar zu umreißen. Ein besonderer Fokus wird in Kapitel 2.2 auf Koordinatensysteme und deren Definition gelegt. Abschließend folgt zur Einordnung in das Gesamtbild ein Überblick moderner Roboterplattformen, um den Stand der Technik aufzuzeigen.

Zur Einführung in die Herausforderungen der humanoiden Robotik folgt nun zunächst ein Überblick über deren Grundlagen. Um das eindeutige Verständnis der Arbeit zu fördern, wird damit begonnen einige Begrifflichkeiten der Robotik klar zu umreißen. Ein besonderer Fokus wird in Kapitel 2.2 auf Koordinatensysteme und deren Definition gelegt. Abschließend folgt zur Einordnung in das Gesamtbild ein Überblick moderner Roboterplattformen, um den Stand der Technik aufzuzeigen.

<sup>2</sup>Es sei darauf hingewiesen, dass in diesen Bereichen zwar Szenarien denkbar sind, in denen humanoide Roboter einen Vorteil gegenüber herkömmlichen Modellen hätten, jedoch eine praktische Anwendung bislang unbekannt ist. Die starke Fokussierung der Ausschreibung der DARPA *Robotic Challenge*, siehe (<http://www.darpa.mil>), belegt aber den Bedarf zumindest für den Bereich der *Rescue*-Robotik.

## 2.1 Begriffsdefinition

Die Forschung an autonomen Robotern hat sich über die Jahre zu einem eigenen Zweig der Robotik entwickelt. Grund hierfür ist primär die Unterschiedlichkeit der Problemstellungen zu denen der klassischen Industrierobotik, so dass grundlegend andere Lösungsansätze verfolgt werden müssen. Als Folge haben sich für dieses Gebiet spezielle Fachbegriffe eingebürgert oder es sind vorhandene im Kontext anders zu verstehen. Dieses Phänomen setzt sich weiter fort, wenn der Unterzweig der humanoiden Robotik betrachtet wird, der nochmals anders geartete Herausforderungen mit sich bringt. Hierbei sind viele in der humanoiden Robotik verwendeten Begriffe dem Menschen intuitiv verständlich. Dennoch - oder möglicherweise deswegen - sind auch in Fachkreisen einige Begriffe nicht eindeutig definiert und je nach Autor unterschiedlich besetzt. Da zudem einige verwendete Begriffe zwar für rollende autonome Roboter eindeutig sind, aber für Roboter mit Beinen ergänzt oder undefiniert werden müssen, ist eine Klärung dem präzisen Verständnis förderlich. Dieser Versuch wird zwar in diversen grundlegenden Lehrbüchern verfolgt, allerdings ist dem Autor keines bekannt, das alle hier benötigten Begriffe klar ausführt. Um einer Verwirrung durch eine Vielzahl unterschiedlicher Literaturquellen vorzubeugen folgt eine Definition in den eigenen Worten des Autors, der sich somit auch die Freiheit nimmt einige Begriffe speziell für den Kontext dieser Arbeit zu definieren oder zu ergänzen. Diese Festlegung eines einheitlichen Vokabulars erfordert zwangsläufig stellenweise einen Vorgriff auf Themengebiete und Begriffe, die erst im späteren Verlauf der Arbeit dargelegt sind. Zur Förderung des Verständnisses wird jeweils auf das entsprechende Kapitel verwiesen.

**Endeffektor** Es gibt in der Robotik keine allgemeingültige Definition des Begriffs *Endeffektor*, auch als *Effektor* bezeichnet. Im Kontext der industriellen Robotik sind zwei Definitionen gebräuchlich:

**Definition 1:** Der *Endeffektor* ist der Teil des Roboters, der zur Manipulation seiner Umgebung verwendet wird.

**Definition 2:** Unter dem *Endeffektor* eines Roboters wird das letzte Glied einer kinematischen Kette verstanden.

Für einen Industrieroboter fallen in der Regel beide Definitionen zusammen, da der klassische Endeffektor in Form eines Greifers oder eines Schweißgerätes auch meist das letzte Glied der kinematischen Kette bildet. Ein humanoider Roboter hingegen besteht aus mehreren kinematischen Ketten, im Folgenden auch als *Extremitäten* bezeichnet, die durch den Torso verbunden sind, vergleiche Kapitel 3.1. Zur direkten Manipulation der Umwelt sind in der Regel nur die Hände gedacht, die sich am Ende der jeweiligen kinematischen Armkette befinden. Nach der ersten Definition stellen also nur diese die Endeffektoren da. Ein anderes Bild zeichnet sich jedoch, wenn man ein Roboterfußballspiel beobachtet. Entgegen der Intention wird die Umwelt - in diesem Fall aufgrund der Regeln lediglich der Ball - praktisch nur mit den Füßen manipuliert. Folglich ist klar, dass im Kontext dieser Arbeit der allgemeineren zweiten Definition der Vorrang gegeben werden sollte. Daher werden auch Füße und, der Vollständigkeit halber, auch

der Kopf als Endeffektor bezeichnet werden. Der entsprechende englische Begriff lautet *end effector*. Es ist jedoch üblicher vom *Tool Center Point*, abgekürzt als *TCP*, zu sprechen, der einen festgelegten Punkt des Endeffektor beschreibt.

**Aktuator** Aus der Regelungstechnik stammt der Begriff des *Aktuators*, oft auch auch als *Aktor* bezeichnet. Dieser bezeichnet jene Komponente eines Regelungssystems, die Steuersignale in regulierende Faktoren umwandelt. Aktuatoren bilden also die Stellglieder des Systems. Übertragen auf die Robotik, beschreibt also ein Aktor die Komponente eines Roboters, die den Zustand des Systems beeinflusst. Beschränkt man sich dabei sinnvoller Weise nur auf den mechanischen Zustand des Systems, so kann diese Änderung lediglich Resultat einer Bewegung sein. Aktuatoren bezeichnen demnach die Komponenten eines Roboters, die eine Bewegung aus eigener Kraft herbeiführen können. Auf humanoide Roboter übertragen sind dies die Motoren oder elektrischen Muskeln des Systems. Da jedoch für die Bewegungsplanung dieser Arbeit von der Ansteuerung der Motoren abstrahiert wird, werden im Folgenden die Gelenke des Roboters als Aktuator bezeichnet. Um Verwirrung vorzubeugen, sei darauf hingewiesen, dass in einigen Lehrbüchern der Begriff Aktuator auch synonym mit dem Begriff des Endeffektors verwendet wird. Der englische Fachterminus lautet *actuator*.

**Festkörper** Aus der Chemie entliehen beschreibt der Begriff des *Festkörpers* Materie im festen Aggregatzustand. Ein Roboter ist aus unterschiedlichen Teilen fester Materie aufgebaut. Im Kontext der Robotik wird der Begriff jedoch zur Differenzierung enger gefasst und beschreibt lediglich die festen Verbindungselemente zwischen zwei Aktuatoren. Diese sind zwar aus unterschiedlichen Komponenten aufgebaut, doch da diese sich nicht aus eigener Kraft relativ zueinander bewegen können, werden sie abstrahierend als ein Körper betrachtet. Im Rahmen dieser Arbeit wird der Begriff Festkörper synonym mit dem Begriff des *starren* Festkörpers verwendet, also eines jenen Körpers, der sich auch durch äußere Einflüsse nicht in seiner Form verändern lässt. Dies ist eine notwendige Annahme zur grundlegenden Modellierung des Roboters, wobei klar ist, dass diese zwangsläufig von der Realität abstrahieren muss, und beispielsweise Flexibilität und Spiel in den Getrieben vernachlässigt werden. Ohne diese Vereinfachung ist eine Simulation der Systemdynamik in Echtzeit nicht möglich und folglich könnte ein Modell für die Bewegungsplanung nicht zur Laufzeit berechnet werden. Der synonym verwendete englische Begriff für Festkörper lautet *rigid body*.

**Schwerpunkt** Der *Schwerpunkt*, aufgrund des englischen Begriffs *Center of Mass* in der Regel als CoM abgekürzt, bezeichnet in der Robotik den physikalischen Massenschwerpunkt eines Roboters. Da ein humanoider Roboter per Definition aus mehreren gegeneinander beweglichen Festkörpern besteht, ist die Position des CoM im System nicht fix, sondern hängt von der aktuellen Stellung der Gelenke ab. Daher muss der CoM zu jedem Zeitpunkt  $t$  berechnet werden, indem die gewichteten Mittel der Positionen der einzelnen Massenschwerpunkte der Festkörper aufaddiert werden. Für eine genauere Betrachtung sei auf Kapitel 4.2 verwiesen. Für die folgenden Stabilitätsbetrachtungen ist nicht nur der Schwerpunkt, sondern vor allem dessen Projektion auf die Ebene des

Bodens von Interesse. In diesem Fall ist es üblich vom *projizierten* Schwerpunkt, oder englisch *(ground-)projected Center of Mass* zu sprechen, der im weiteren Verlauf mit pCoM angekürzt wird.

**Konfiguration** Die *Konfiguration* eines Roboters bezeichnet eine eindeutige Stellung des Roboters. Da diese sich lediglich durch die Bewegung der Aktuatoren verändern lässt, liegt es nahe, diese für humanoide Roboter durch die Stellung der Gelenke zu beschreiben. Somit wird die Konfiguration  $K$  des Roboters zum Zeitpunkt  $t$  eindeutig gegeben durch den Vektor der aktuellen Positionen  $J_i$  der  $n$  Gelenke,

$$\mathbf{K}(t) = (J_0(t), J_1(t), \dots, J_{n-2}(t), J_{n-1}(t))^T. \quad (2.1.1)$$

**Position** Die *Position* eines Körpers im Raum ist relativ zu seinem Bezugskoordinatensystem definiert. Durch den Freiheitsgrad in  $x$ -,  $y$ - und  $z$ -Richtung hat die Position im allgemeinen drei Freiheitsgrade und ist gegeben durch

$$\mathbf{P} = (p_x, p_y, p_z)^T. \quad (2.1.2)$$

Zur Bestimmung der Position wird ein fester Punkt benötigt, für den in der Regel der geometrische Mittelpunkt herangezogen wird. Da aber ein humanoider Roboter aus mehreren Körpern aufgebaut ist, ergibt sich analog zur Definition des Schwerpunkts die Problematik, die Position des Gesamtsystems eindeutig zu bestimmen. Hierzu sind verschiedene Ansätze zur Definition denkbar:

**Definition 1:** Geometrisches Mittel der Positionen aller Festkörper.

**Definition 2:** Position eines ausgezeichneten Festkörpers, beispielsweise des Torsos.

**Definition 3:** Position des Festkörpers, dessen relative Position zum Bezugskoordinatensystem bei der Bewegung unverändert bleibt.

Definition 1 würde die Position analog zum Schwerpunkt festlegen und erscheint daher als intuitive Lösung. Allerdings fällt bei näherer Betrachtung ein unerwünschter Effekt auf. Die Position eines Körpers im Raum soll sich nur ändern, wenn sich der Körper fortbewegt. Verharre der Roboter nun aber an einer Stelle und bewege lediglich die Arme, so würde diese Bewegung zu einer Veränderung der Position des Roboters führen, obwohl sich dieser nicht fortbewegt hat.

Definition 2 erscheint sinnvoll, wenn die Position des Torsos als Position des Roboters gewählt wird, da dieser das Zentrum der kinematischen Kette bildet. Somit könnten zwar die Arme bewegt werden ohne die Position des Roboters zu verändern, doch führt auch diese Definition zu einer nicht geeigneten Positionsdefinition während die Fortbewegung. Wie in den späteren Kapiteln erläutert wird, erfordert die Stabilisierung des Gangs eine seitliche Schwingung des Oberkörpers des Roboters. Somit führt die Fortbewegung zu einem Oszillieren der Position um den eigentlichen Pfad und wäre somit ständig in Veränderung. Dies ist aber für die *Cognition* des Roboters ein erschwerender

Zustand, da diese die Wahrnehmungen ständig mit der sich ändernden Position abgleichen müsste. Folglich wären unnötige Fehler durch die Transformation unvermeidbar.

Es verbleibt Definition 3, bei der dieses Problem per Definition ausgeschlossen ist. Durch die Art der Fortbewegung bleibt die Position mindestens eines Fußes fest im Bezugskoordinaatensystem, solange der Roboter nicht in der Lage ist, sich rennend fortzubewegen. Folglich liegt es nahe, die Position des Roboters als geometrischen Mittelpunkt aller Bodenkontaktpunkte zu definieren. In der Praxis wird der Einfachheit halber immer der Ursprung des Fußkoordinatensystems, siehe Kapitel 2.2.6, gewählt, um überflüssige Rechnungen zu vermeiden. Dies hat allerdings zur Folge, dass sich die Position des Roboters nicht mehr kontinuierlich, sondern nur noch im wahrsten Sinnes des Wortes schrittweise bewegt. Durch die Natur der menschlichen Fortbewegung ändern sich die Aufsatzpunkte der Füße immer in einem Phasenübergang, was eine diskrete Änderung der Position zur Folge hat. Allerdings erweist sich dies in der Praxis als die sinnvollste Definition und wird daher im weiteren Verlauf verwendet.

Es ist anzumerken, dass durch diese Definition der Freiheitsgrad eingeschränkt wird, da die  $z$ -Komponente der Position identisch der des Bodens ist. Im Falle des RoboCups führt diese Nebenbedingung dazu, dass die Position des Roboters auf 2 Dimensionen beschränkt ist, da der Boden eine ebene Fläche mit der Koordinate  $z = 0$  darstellt.

**Orientierung** Die *Orientierung* eines Körpers im Raum ist definiert durch die Rotation des Koordinatensystems des Körpers relativ zur Basis des Bezugskoordinaatensystems. Somit ergeben sich für die Orientierung 3 Freiheitsgrade, wenn diese frei veränderbar sein soll. Für die Überführung der Orientierung sind unterschiedliche Notationen gebräuchlich, von denen in der Robotik *Eulersche Winkel*, *Roll-Pitch-Yaw-Winkel*, *Drehvektor/Drehwinkel*, *Orientierungsmatrix* und *Quaternionen* die gebräuchlichsten sind. Eine detaillierte Erklärung kann an dieser Stelle nicht erfolgen. Vielmehr sei bei Interesse für eine vertiefende Betrachtung u. a. auf [52] und [64] verwiesen.

Durch diese Definition ergibt sich für einen humanoiden Roboter, analog zur Position, das Problem des Bezugspunktes. Daher erfolgt die Definition der Orientierung ebenfalls, indem die Orientierung des Standfußes herangezogen wird. Die Richtung der  $x$ -Achse des jeweiligen Fußkoordinatensystem bestimmt die Orientierung und somit zeigt diese in Laufrichtung des Roboters. Auch durch diese Definition reduziert sich die Anzahl der Freiheitsgrade der Orientierung im Falle des Roboterfußballs auf zwei.

**Pose** Die *Pose* ist in der Robotik definiert als Kombination der Position und Orientierung eines Körpers. Folglich hat sie im Allgemeinen 6 Freiheitsgrade. Aus der vorhergehenden Definition der Position und Orientierung ergeben sich aber analoge Nebenbedingungen für den Roboterfußball, so dass die Pose im Kontext der SPL lediglich 4 Freiheitsgrade besitzt.



**Lage im Raum** Besonders für die Selbstwahrnehmung des Roboters kommt der Pose des Oberkörpers im Bezug auf den Boden eine gesonderte Rolle zu. Um diese von der Pose des Roboters abzugrenzen und dadurch das Verständnis der Diskussionen zu fördern, wird für diese im Folgenden ein separater Begriff eingeführt. Sie wird als *Lage* des Roboters im Raum bezeichnet.

**Bewegung** Jede Veränderung der Stellung eines Aktuators des Roboters wird folgend als *Bewegung* bezeichnet. Eine Bewegung hat somit eine Veränderung der Konfiguration  $K(t)$  zu  $K(t + 1)$  zur Folge. Der im Englischen verwendete Begriff lautet *motion*. Unter diesem Begriff wird jegliche Art von Bewegungsplanung und -ausführung zusammengefasst, um diese von der *Cognition* und dem *Behavior* abzugrenzen, siehe auch Kapitel 1.1.

**Fortbewegung** Nicht jede Bewegung des Roboters resultiert in einer Veränderung der Position oder Orientierung desselbigen. Um dies trennen zu können, ist eine begriffliche Differenzierung nötig. Mit *Fortbewegung*, im Englischen als *locomotion* bezeichnet, wird im Folgenden jede Bewegung bezeichnet, die in einer Veränderung der Pose des Roboters resultiert. Es ist anzumerken, dass nach dieser Definition eine Fortbewegung logischerweise auch gleichzeitig eine Bewegung sein muss, solange sie vom Roboter selber initiiert wird. Führen jedoch externe Kräfte zur Veränderung der Pose, so kann es sein, dass die Konfiguration des Roboters unverändert bleibt und demnach eine Fortbewegung ohne Bewegung im obigen Sinne stattfindet. Wenn diese Unterscheidung betont werden soll, wird von Fortbewegung *aus eigener Kraft* gesprochen.

**Gang** Als *Gang*, oder englisch *gait*, wird in der Robotik eine Fortbewegung bezeichnet, die aus einer Abfolge von abwechselnden periodischen Fußbewegungen resultiert. Der Gang kann also durch eine Abfolge von zyklischen Fußbewegungen definiert werden:

1. Beide Füße des Roboters haben Kontakt mit dem Boden. Diese Phase wird *Double-Support-Phase* genannt.
2. Der Roboter hebt wahlweise den rechten oder linken Fuß an und bewegt diesen auf einer Trajektorie in Richtung des neuen Fußaufsatzpunktes. Das frei bewegte Bein wird als *Schwungbein* bezeichnet, das auf dem Boden verbleibende als *Standbein*. Der Oberkörper wird wahlweise relativ zum Standbein bewegt. Diese Phase wird *Single-Support-Phase* genannt.
3. Der Roboter setzt den Fuß des Schwungbeins auf den neuen Aufsatzpunkt und geht damit wieder in eine *Double-Support-Phase* über. Auch während dieser Phase ist weiterhin eine Bewegung des Oberkörpers relativ zum *Support-Polygon* möglich.
4. Nun wird wieder in eine erneute *Single-Support-Phase* übergeleitet, indem der Roboter das Standbein aus Phase 2 anhebt und nun als Schwungbein auf einer Trajektorie Richtung des nächsten Fußaufsatzpunktes bewegt. Der Oberkörper kann auch in dieser Phase relativ zum aktuellen Standbein bewegt werden.

5. Das Schwungbein wird aufgesetzt und der Roboter geht erneut in eine *Double-Support-Phase* über. Diese Phase ist identisch mit der ersten und schließt dadurch den Zyklus.

Es fällt auf, dass der Roboter zu jedem Zeitpunkt der Fortbewegung Kontakt mit dem Boden hat. Diese Art der Fortbewegung wird als *laufend*, englisch *walking*, bezeichnet. Ihr entgegen steht das *Rennen*, englisch *running*, bei dem der Roboter zwischenzeitlich den Boden nicht mehr berührt. Hierbei stößt sich der Roboter zum Ende jeder *Single-Support-Phase* mit dem Standbein vom Boden ab und ersetzt die folgende *Double-Support-Phase* durch eine sogenannte *Flugphase*, englisch *air phase*.

**Support-Polygon** Das *Support-Polygon* eines humanoiden Roboters ist definiert als die konvexe Hülle aller Kontaktpunkte des Roboters mit dem Boden. Da dies ein zentraler Begriff dieser Arbeit ist, wird zum Verständnis in Abbildung 2.1.2 das Beispiel einer Konfiguration demonstriert, in der der Roboter mit beiden Füßen den Boden berührt. Das *Support-Polygon* ist in diesem Fall die schraffierte Fläche. Für diesen englischen Begriff gibt es kein allgemein verwendetes deutsches Synonym.

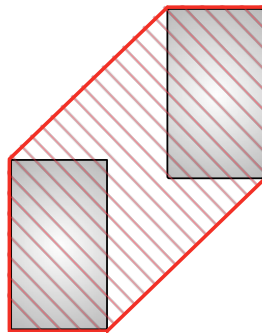


Abbildung 2.1.2: *Support-Polygon* während der *Double-Support-Phase* rot schraffiert.

**Degree of Freedom** Der Begriff *Degree of Freedom*, auch abgekürzt mit *DoF*, bezeichnet im Allgemeinen den *Freiheitsgrad* eines Roboters. Dieser ist aber abhängig vom Kontext unterschiedlich definiert, da zwischen dem Freiheitsgrad der Bewegung und der Fortbewegung unterschieden werden muss. So beschreibt der DoF für rollende Roboter die Freiheitsgrade, durch die die Pose des Roboters verändert werden kann. Für einen humanoiden Roboter ist es jedoch üblich, mit dem DoF die Anzahl der Aktuatoren zu bezeichnen, also die Möglichkeiten die Konfiguration des Roboters zu beeinflussen und diesen somit zu bewegen. Eine bei fahrenden Robotern übliche Unterscheidung zwischen DoF und *Degree of Mobility*, bzw. *Degree of Maneuverability*, vergleiche [101], findet bei laufenden Robotern üblicherweise nicht statt.

**omnidirektional** Die Fortbewegung eines Roboters wird als *omnidirektional* bezeichnet, wenn der Roboter in der Lage ist, sich in jede Richtung zu bewegen, ohne zuvor seine Orientierung ändern zu müssen. Humanoide Roboter, deren Kinematik dem Menschen nachempfunden ist, sind hierzu in der Lage. Um diese für rollende Roboter zu ermöglichen, müssen diese mit speziellen Räder, wie zum Beispiel sogenannten *Swedish Wheels*, ausgestattet sein. Ein ausführlicherer Überblick über omnidirektionale Konfigurationen ist [101] zu entnehmen.

**quasi-omnidirektional** Ein humanoider Roboter ist in der Lage, sich wie der Mensch omnidirektional fortzubewegen, wenn er eine ausreichende Anzahl Freiheitsgrade in den Beinen besitzt. Jedoch ist er aufgrund seiner Kinematik in der Regel nicht in der Lage, sich in jede Richtung gleich schnell zu bewegen. Ein Roboter, der sich omnidirektional bewegen kann, aber eine Richtung der Fortbewegung aus Gründen der Geschwindigkeit oder Stabilität bevorzugt, wird im Folgenden als *quasi-omnidirektional* bezeichnet. Auch die Fortbewegung des Menschen fällt durch die Natur seines Fortbewegungsapparates unter diese Klasse der Bewegung. Das Kniegelenk präferiert eine Bewegung in *geradeaus*<sup>3</sup> Richtung. Der Mensch vermeidet aus diesem Grund wenn möglich seitliche Schritte oder den Rückwärtsgang.

## 2.2 Koordinatensysteme

*Koordinatensysteme* spielen in der Robotik eine entscheidende Rolle. Daher widmet sich dieses Kapitel der Definition der benutzten Systeme und deren Umrechnung ineinander, auch als *Transformation* bezeichnet. In der Robotik ist es üblich, je nach Anwendung ein anderes geeignetes Koordinatensysteme zu verwenden. Messdaten der Sensoren liefern beispielsweise Daten relativ zum Roboter und werden daher auch in einem relativen Koordinatensystem verarbeitet. Die Lokalisierung des Roboters kann hingegen nur sinnvoll in einem globalen System geschehen, da seine Position im relativen immer gleich ist. Die Koordinatensysteme werden nach Möglichkeit so gewählt, dass die Anzahl der nötigen Transformationen minimiert wird, um Rechenzeit zu sparen. Eine solche Transformation kann immer durchgeführt werden. Bei Verwendung gleicher Koordinaten ist hierzu lediglich eine Überführung des Ursprung des Ausgangskordinatensystems  $A$ , mittels einer Translation, in die Position des Ursprungs des Zielkoordinatensystems  $B$ , gefolgt von einer Rotation zur Übereinbringung der Orientierungen, nötig. Seien  ${}^A\mathbf{x}$  die Koordinaten eines Vektors im System  $A$ ,  ${}^B\mathbf{x}$  die Koordinaten des gleichen Vektors im System  $B$ ,  ${}^A\mathbf{R}_B$  die Orientierungsmatrix des Systems  $B$  bezüglich des Systems  $A$  und  ${}^A\mathbf{p}_B$  der auf das System  $A$  bezogene Vektor vom Ursprung des System  $B$ , so lässt sich eine allgemeine Transformation  $\mathbf{T}$ , in der Robotik auch als *Frame* bezeichnet, von System  $A$  nach System  $B$ , gezeichnet durch  $A \rightarrow B$ , mathematisch beschreiben durch:

$$A \rightarrow B : \quad {}^A\mathbf{x} = {}^A\mathbf{p}_B + {}^A\mathbf{R}_B \cdot {}^B\mathbf{x}. \quad (2.2.1)$$

<sup>3</sup>*geradeaus* ist mathematisch nicht definiert, wird aber im Folgenden synonym mit „positive  $x$ -Richtung aus Sicht des Roboters“ verwendet.

Diese Notation führt jedoch dazu, dass nacheinander folgende Transformationen in unhandlichen Termen resultiert - hier am Beispiel  $A \rightarrow B \rightarrow C$  verdeutlicht:

$$B \rightarrow C : {}^B \mathbf{x} = {}^B \mathbf{p}_C + {}^B \mathbf{R}_C \cdot {}^C \mathbf{x} \quad (2.2.2)$$

$$A \rightarrow B \rightarrow C : {}^A \mathbf{x} = {}^A \mathbf{p}_B + {}^A \mathbf{R}_B \cdot {}^B \mathbf{p}_C + {}^A \mathbf{R}_B \cdot {}^B \mathbf{R}_C \cdot {}^C \mathbf{x}. \quad (2.2.3)$$

Da dies besonders die Handhabung kinematischer Ketten erschwert, wird im Folgenden ein in der Robotik üblicher mathematischer Kniff verwendet - *homogene Koordinaten*. Diese sind definiert als

$$\mathbf{x} = \begin{pmatrix} a \\ b \\ c \\ s \end{pmatrix} \quad \text{mit } a = s \cdot x_1, b = s \cdot x_2, c = s \cdot x_3, \quad (2.2.4)$$

wobei  $s$  den *Skalierungsfaktor* bezeichnet, der in der Robotik in der Regel gleich 1 gesetzt wird:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix}. \quad (2.2.5)$$

Unter Verwendung von homogenen Koordinaten lässt sich nun die *homogene Transformation* definieren, mit der Transformationsmatrix

$${}^A \mathbf{T}_B := \begin{bmatrix} {}^A \mathbf{R}_B & {}^A \mathbf{p}_B \\ \mathbf{0} & 1 \end{bmatrix} \quad \text{mit } \mathbf{0} = [0 \ 0 \ 0]. \quad (2.2.6)$$

Mittels dieser Definition kann nun  $A \rightarrow B$  geschrieben werden als:

$${}^A \mathbf{x} = {}^A \mathbf{T}_B \cdot {}^B \mathbf{x} = \begin{bmatrix} {}^A \mathbf{R}_B & {}^A \mathbf{p}_B \\ \mathbf{0} & 1 \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} = \begin{bmatrix} {}^A \mathbf{R}_B \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} + {}^A \mathbf{p}_B \\ 1 \end{bmatrix}. \quad (2.2.7)$$

Der Vorteil dieser Notation wird deutlich, wenn man erneut  $A \rightarrow B \rightarrow C$  betrachtet, deren Gesamttransformation sich nun aus der Multiplikation der einzelnen homogenen Transformationen ergibt:

$${}^A \mathbf{x} = {}^A \mathbf{T}_B \cdot {}^B \mathbf{T}_C \cdot {}^C \mathbf{x}. \quad (2.2.8)$$

Auch bei der Beschreibung beliebiger kinematischer Ketten können aufeinanderfolgende Translationen und Rotationen in homogener Koordinatendarstellung durch Multiplikation der einzelnen homogenen Transformationsmatrizen berechnet werden, so lange beachtet wird, dass die nachfolgende Transformation von links an die vorherigen heran multipliziert wird. Die Umkehrung der Transformation ergibt sich durch Matrixinversion:

$${}^A \mathbf{T}_B^{-1} = \begin{bmatrix} {}^A \mathbf{R}_B^T & -{}^A \mathbf{R}_B^T \cdot {}^A \mathbf{p}_B \\ \mathbf{0} & 1 \end{bmatrix}. \quad (2.2.9)$$

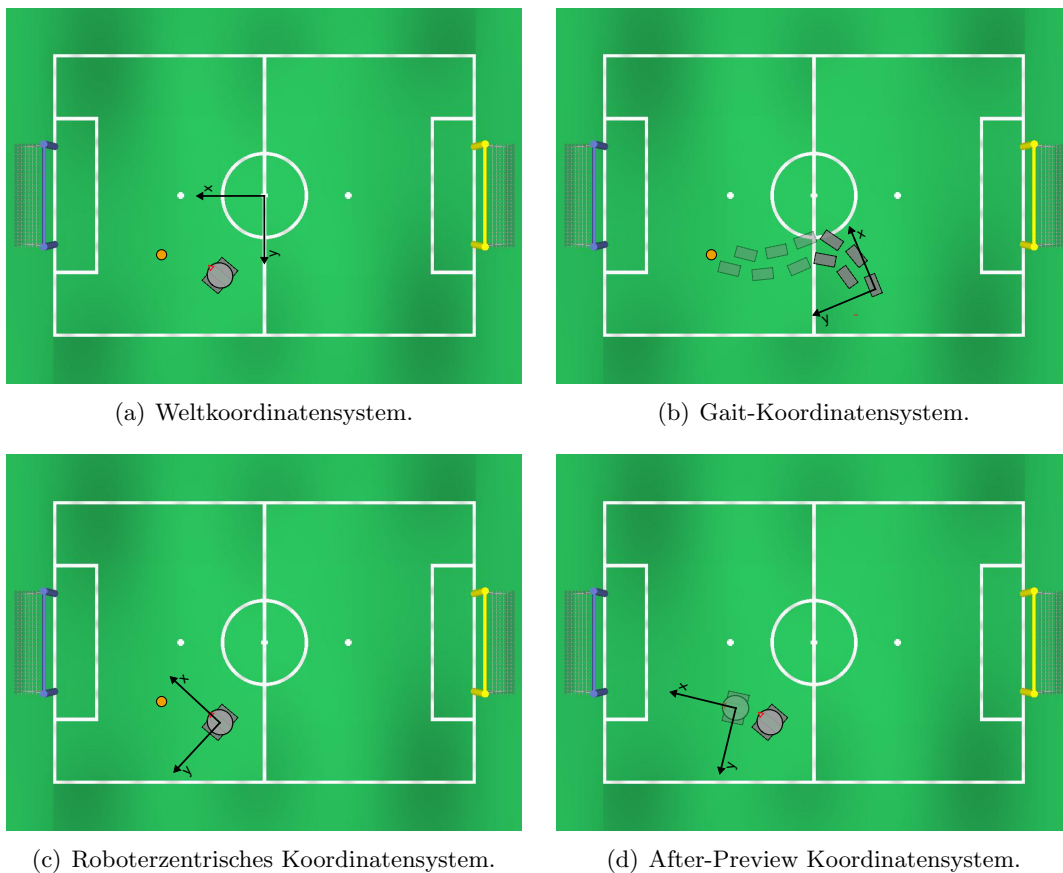


Abbildung 2.2.3: Übersicht der Koordinatensysteme.

Dieser kurze Exkurs dient lediglich dazu, ein grundsätzliches Verständnis für die im Laufe der Arbeit durchgeführten Transformationen zu liefern. Für eine weiterführende Beschreibung der Transformationen mit homogenen Koordinaten sei daher u. a. auf [64] verwiesen.

Koordinatensysteme sind besonders für die Bewegungsplanung eines Roboters entscheidend, da mit ihrer Hilfe ein Bezug zwischen Wahrnehmung und Aktion hergestellt wird. Aus diesem Grund ist ein Verständnis der in dieser Arbeit verwendeten Koordinatensysteme wichtig und es folgt eine Einführung mit dem Ziel, diese eindeutig zu definieren. Zudem wird das später vorgestellte Konzept zur Planung dynamischer Laufbewegungen die Einführung eines neuen Koordinatensystems zur Folge haben. Dies soll vorgreifend vorgestellt werden, um in den Kontext der anderen Koordinatensysteme eingeordnet zu werden, auch wenn der Grund für die Einführung hier noch nicht ausreichend motiviert werden kann.

### 2.2.1 Weltkoordinatensystem

Das *Weltkoordinatensystem*, im Folgenden auch *WKS*, bezeichnet das globale Koordinatensystem in der Robotik. Seine Wahl ist abhängig vom Kontext, in dem der Roboter operiert, so dass diese zunächst willkürlich erscheinen mag. Jedoch umfasst das WKS mindestens den gesamten Arbeitsraum des Roboters, also den Raum in dem dieser agieren und sich sinnvoll bewegen kann. So ist beispielsweise das WKS eines KUKA KR6 Industrieroboters zentrisch um seinen Sockel aufgebaut und umfasst lediglich den halbkugelförmigen Raum, den er mit ausgestrecktem Arm erreichen kann. Eine Definition seiner Position im Geo-Koordinatensystem wäre unsinnig, da er seinen Arbeitsraum nicht verlassen kann. Doch auch wenn ein mobiler Roboter sich theoretisch beliebig weit bewegen könnte, so ist ein lokal auf seinen Arbeitsraum beschränktes WKS sinnvoll, um seine Aktionen effektiver planen zu können. So ist beispielsweise der Roboter *Toomas*<sup>4</sup> in der Lage, seine Arbeitsbereich - den ihm zugewiesenen Baumarkt - zu verlassen. Dennoch kann er außerhalb nicht mehr zielgerichtet agieren, da er dort keine Information über die Umgebung besitzt. Eine Beschränkung des WKS auf den Baumarkt ist somit ausreichend. Das Auffinden der Pose des Roboters im WKS wird als *globale Lokalisierung* bezeichnet.

Im Kontext des RoboCups wird das WKS durch das Spielfeld definiert. Überschreitet ein Spieler die Spielfeldgrenzen, erhält er einen *Timeout* als Strafe und somit beschreibt dieser Bereich per Definition den maximalen Arbeitsraum. Bei den NAO Devils Dortmund kommt ein kartesisches Koordinatensystem zur Anwendung, dessen Ursprung durch den Anstoßpunkt festgelegt ist. Die positive  $x$ -Achse läuft immer senkrecht auf das gegnerische Tor zu und die positive  $y$ -Achse wird nach der Rechte-Hand-Regel gewählt. Da weder der Roboter noch der Ball in der Lage sind, kontrolliert den Boden zu verlassen, spielen Höhenunterschiede für die Lokalisierung oder Bewegungsplanung keine Rolle. In Folge dessen wird auf eine  $z$ -Komponente verzichtet und das WKS ist somit zweidimensional. Zur Veranschaulichung ist das WKS der SPL grafisch in Abbildung 2.2.3(a) dargestellt.

### 2.2.2 Roboterzentrisches Koordinatensystem

Das *roboterzentrische Koordinatensystem*, kurz RzKS, ist ein relatives Koordinatensystem und hat die Position des Roboters zum Ursprung. Seine Orientierung wird durch die Ausrichtung des Roboters bestimmt, wie Abbildung 2.2.3(c) zu entnehmen ist. Somit ist als Folgerung aus der Definition der Pose des Roboters im vorangehenden Kapitel 2.1 klar, dass es sich auch beim RzKS im Roboterfußball um ein zweidimensionales Koordinatensystem handelt. Seine zentrale Rolle kommt dem RzKS zu, da im NDD-Framework die Positionsbestimmung von Objekten in diesem erfolgt - also relativ zum Roboter. Damit bildet das RzKS die Grundlage für jede Verhaltensplanung. Durch seine relative Ausrichtung hat eine Fortbewegung des Roboters keine Verschiebung der Roboterpose im RzKS, sondern eine Verschiebung der Wahrnehmungen und somit des

---

<sup>4</sup>Von der Firma MetraLabs (<http://metralabs.com/>) für die Baumarktkette Toom entwickelter rollender Roboter.

Weltmodells in diesem zur Folge. Im NDD-Framework werden aus Gründen der Konsistenz zum WKS ebenfalls kartesische Koordinaten verwendet, doch diese Entscheidung ist willkürlich. Da die meisten Wahrnehmungen mittels der Kamera geschehen, und diesen eine Entfernung und ein Kopfwinkel zugeordnet werden, lassen sich ebenfalls gute Argumente für eine Darstellung in Zylinderkoordinaten oder Kugelkoordinaten finden, da dies der nativen Repräsentation der Messwerte entspricht. Erste Untersuchungen von Tasse [107] zeigen hier signifikante Vorteile der Darstellung in Kugelkoordinaten. Eine Transformation ins WKS erfolgt mit Hilfe der aus der Lokalisierung ermittelten Pose des Roboters. Somit sind Wahrnehmungen, wie z.B. die Position des Balls, im globalen Kontext einer zweifachen Unsicherheit ausgesetzt - zum einen der fehlerbehafteten Bestimmung durch den Sensor im RzKS und zum anderen der Unsicherheit in der Transformation durch die Positionsbestimmung des Roboters.

### 2.2.3 After-Preview Koordinatensystem

Schnelle Fortbewegung erfordert für den Roboter dynamische Stabilität, wie in Kapitel 4.3 erläutert wird. Daher kann der Roboter nicht zu jedem beliebigen Zeitpunkt anhalten und auch nicht abrupt seine Bewegungsrichtung ändern. In Kombination mit dem in Kapitel 5.2 eingeführten Konzept des *Preview*-Reglers führt dies zu einer Latenz im System. Diese hat zur Folge, dass Entscheidungen, die das Verhalten aufgrund der Wahrnehmungen im RzKS trifft, erst zu einem Zeitpunkt  $t_{\Delta}$  in der Zukunft ausgeführt werden können. Zu dem Zeitpunkt kann sich aber die Pose des Roboters bereits durch seine Eigenbewegung verschoben haben. Anschaulich lässt sich dies am Beispiel der Ballannäherung verdeutlichen. Ab einer gewissen Entfernung zum Ball entscheidet der Roboter stehen zu bleiben, um einen Schuss durchzuführen. Durch die *Preview*-Phase kann er aber erst zeitlich verzögert seine Schritte anpassen und läuft somit weiter als gedacht und stößt gegen den Ball. Um mit dieser Problematik umzugehen wird für Verhaltensentscheidungen an dieser Stelle ein korrespondierendes *After-Preview* Koordinatensystem eingeführt. Dies entspricht zu jedem Zeitpunkt dem roboterzentrischen Koordinatensystem zu dem Zeitpunkt in der Zukunft, an dem der Roboter die Bewegungen der *Preview*-Phase durchgeführt hat, also dem Zeitpunkt, an dem es frühestens möglich ist, neu entschiedene Aktionen auszuführen. Somit werden die Wahrnehmungen entsprechend transformiert, um zumindest die bekannte Eigenbewegung des Roboters implizit in die Entscheidungen des Verhaltens einzubeziehen. Exemplarisch ist dies in Abbildung 2.2.3(d) verdeutlicht.

### 2.2.4 Körpereigenes Koordinatensystem

Zusätzlich zu den Koordinatensystemen zur Verhaltensplanung des Roboters ist es sinnvoll, jedem Festkörper ein *körpereigenes Koordinatensystem*, im Folgenden auch durch *KeKS* abgekürzt, zuzuweisen. Dies ist notwendig, um mittels der Kinematik die Positionen der Endeffektoren der jeweiligen Ketten bestimmen zu können, vergleiche Kapitel 3.1.1 und 3.1.2. Um die Transformation zu vereinfachen, wird für jedes Element der kinematischen Ketten dessen Ursprung ohne Beschränkung der Allgemeinheit auf die

Gelenkachse zum vorhergehenden Festkörper gelegt. Somit kann die Transformation des Ursprungs zu jedem Zeitpunkt durch eine feste Translation mit anschließender konfigurationsabhängiger Rotation durchgeführt werden. Dies ermöglicht die Anwendung der Denavit-Hartenberg-Parameter zur Beschreibung der Ketten, vergleiche auch [49]. Bei den Aktuatoren des hier betrachteten Roboters handelt es sich ausschließlich um Drehgelenke. Den DH-Konventionen folgend wird daher die Orientierung immer so gewählt werden, dass einen Koordinatenachse mit der Drehachse übereinkommt. Eine Ausnahme bildet lediglich das KeKS des Oberkörpers, da es keinen Vorgänger in der kinematischen Kette hat, aber mehrere Nachfolger. Daher wird hier der Ursprung des KeKS in den Schwerpunkt gelegt und die Orientierung gleich der Orientierung des Körpers gewählt.

### 2.2.5 Robotereigenes Koordinatensystem

Da das RzKS für den Fall des RoboCups auf 2 Dimensionen reduziert wurde, ist es nicht geeignet die Informationen über den Roboterzustand aus den internen Sensoren modellieren zu können. Somit wird analog zum Konzept der Lage im Raum ein weiteres Koordinatensystem eingeführt, das je 3 Freiheitsgrade in der Position und Orientierung hat. Dies sogenannte *robotereigene Koordinatensystem*, abgekürzt als *ReKS*, ist identisch mit dem KeKS des Oberkörpers, damit die Sensordaten der Inertialsensoreinheit ohne weitere Transformation nutzbar sind. Der Ursprung  $ReKS_0$  des robotereigenen Koordinatensystems kann willkürlich als fester Punkt im Torso gewählt werden. In dieser Arbeit wird sich daher nach der Bemaßung von Aldebaran gerichtet und  $ReKS_0$  durch *HipOffsetZ* und *HipOffsetY* bestimmt, vergleiche Abbildung 3.1.1.

### 2.2.6 Fußkoordinatensystem

Ein weiteres spezielles KeKS stellt das *Fußkoordinatensystem*, auch als *FKS* abgekürzt, dar. Dieses entspricht jeweils dem KeKS des aktuellen Standfußes und erhält eine eigene Bezeichnung, da es in den folgenden Kapiteln der Bewegungssteuerung eine zentrale Rolle spielt. Während einer *Double-Support*-Phase wird das Fußkoordinatensystem des Standfußes der vorherigen *Single-Support*-Phase beibehalten. Sollte dies nicht eindeutig sein, wird per Definition das KeKS des rechten Fußes gewählt.

### 2.2.7 Gait-Koordinatensystem

Der Lauf des Roboters könnte direkt im RzKS geplant werden. Da dessen Position aber mit jedem Schritt springt, hätte dies eine ständige Transformation der geplante Fußaufsatzpunkte zur Folge. Um dies zu umgehen und somit eine kontinuierliche Pfadplanung zu erhalten, wird ein Hilfskoordinatensystem der NDD-Walking-Engine eingeführt, das sogenannte *Gait-Koordinatensystem*, abgekürzt als *GKS*. Dieses entspricht per Definition dem aktuellen FKS zur Startzeit des *Walking-Engine*-Moduls, siehe Kapitel 5.1. Abbildung 2.2.3(b) veranschaulicht ein Beispiel.



## 2.3 State-of-the-Art Roboterplattformen

Nachdem nun eine einheitliche Sprachgrundlage für die Arbeit gelegt wurde, steht es noch aus, zum Abschluss der kurzen Einführung in die humanoide Robotik den Fokus auf die Hardwareentwicklung der letzten Jahre zu legen. Dies ist wichtig, da die Forschung an humanoiden Robotern zwangsläufig immer im engem Wechselspiel zwischen Hardware und Algorithmen in Form von Software geschieht. So können zum einen erforschte Algorithmen nur in die Praxis umgesetzt werden, wenn die Hardware des Roboters dies auch ermöglicht. Zum anderen ist aber auch der leistungsstärkste Sensor für den Roboter keine Unterstützung, wenn seine Informationen nicht von seinem Programm interpretiert und genutzt werden können. Um die Herausforderungen an die Forschung der Bewegungssteuerung humanoider Roboter besser einordnen zu können, muss also zunächst geklärt werden, wozu die Hardware mobiler Roboter heutzutage in der Lage ist und welche Art von Sensorik zur Anwendung kommt. Daher wird an dieser Stelle abschließend zur Einführung über moderne humanoide Robotik ein Querschnitt über aktuelle zweibeinige Roboter gegeben, um den Entwicklungsstand der Hardware aufzuzeigen. Hierbei wird sowohl auf kommerziell erhältliche Roboter, als auch auf aktuelle Forschungsprojekte eingegangen. Doch um dem bereits eingehend diskutierten Gedanken - Experimente aus Gründen der Vergleichbarkeit auf einer einheitlichen Plattform auszuführen - Rechnung zu tragen, wird die Auswahl bereits bewusst in diese Richtung gelenkt.

Wie fasziniert der Mensch davon ist, humanoide Roboter zu erforschen, zeigt sich nicht zuletzt an der Vielfalt der entwickelten Modelle die jährlich wächst. Diese Aufzählung soll und kann daher nicht einen kompletten Überblick über alle verfügbaren Plattformen bieten. Vielmehr werden nur exemplarisch einzelne Modelle herausgestellt, die interessante Entwicklungen und Konzepte demonstrieren. Diese Auswahl ist natürlich mit Fokus auf die Laufforschung zugeschnitten, um sich in den Rahmen dieser Arbeit einzugliedern. Somit müssen einige interessante Projekte außen vor gelassen werden, da diese sich auf andere Bereiche der Robotik, wie beispielsweise die *Mensch-Maschine-Interaktion*, beschränken. Zur Vermeidung von Wiederholungen wird nicht im Detail bei jedem Roboter auf die verwendete Sensorik eingegangen. Es sei aber bereits vorweggenommen, dass praktisch alle modernen humanoiden Roboter mit einem sehr ähnlichem Arsenal von Sensoren ausgestattet sind. Diese unterscheiden sich aufgrund der Größe und des Preises des Roboters zwar in der Qualität, jedoch nicht grundsätzlich voneinander. So hat jeder Roboter, an die Sinnesorgane des Menschen angelehnt, mindestens eine Kamera zur Wahrnehmung der Umgebung. Zudem einen Inertialsensor, um damit die Funktion des Innenohrs nachzuahmen, und zumindest an den Füßen eine Form von taktilem Sensor, der eine Rückmeldung über Kräfte zulässt. Des Weiteren ist jeder Roboter zur Kommunikation mit Mikrofon und Lautsprecher ausgestattet.

### 2.3.1 ASIMO

Der Roboter ASIMO, zu finden in Abbildung 2.3.4<sup>5</sup>, gehört ohne Frage zu den fortschrittlichsten zweibeinigen Robotern der Welt. Entstanden als Forschungsprojekt der Entwicklungsabteilung von Honda wurde die erste Version im Jahre 2000 der Öffentlichkeit vorgestellt. Die Roboter der Honda E- und P-Serie können als Vorgänger des ASIMO angesehen werden. Die Forschung reicht also bereits weitere 15 Jahre in die Vergangenheit zurück. Ziel der Entwicklung war es, einen Roboter zu bauen, der in der Lage ist, sich in für Menschen entworfenen Umgebungen zu bewegen und dem Menschen in seinem Haushalt helfen zu können. Daher wurde als Name ein Akronym für *Advanced Step in Innovative Mobility* gewählt. Er hat eine Größe von 130 cm und wiegt, je nach Version, rund 50 kg. Der neueste 2011 vorgestellte ASIMO besitzt 57 Freiheitsgrade und ist dadurch sehr flexibel in der Bewegungsplanung. Sein Vorteil in



Abbildung 2.3.4: ASIMO.

der Beweglichkeit gegenüber anderen Robotermodellen zog ASIMO jedoch bereits zu Beginn aus dem von Honda entwickelten Algorithmus zur Regelung, der als *predicted movement control* bezeichnet wird. Diese erlaubt den Einsatz von flexibleren Gelenken und ermöglicht so weichere und gedämpfte Bewegungsabläufe. Dies ist ein gutes Beispiel für das einfühend erwähnte Zusammenspiel von Hard- und Software in der Robotik. Aufgrund dieser Gelenke war ASIMO als erster<sup>6</sup> Roboter seiner Größe in der Lage war zu rennen, also während der Fortbewegung zeitweilig mit beiden Füßen vom Boden abzuheben. Die letzte Version erreicht dadurch eine beeindruckende Geschwindigkeit von rund 9 km/h. Über die Details der Forschungen von Honda ist wenig bekannt, da nur wenige Veröffentlichungen über ASIMO existieren. Ein Überblick ohne tiefergehende Beschreibungen ist in [50] und [93] zu finden. ASIMO ist nicht kommerziell erhältlich und nur sehr ausgewählten Forschungseinrichtungen, wie dem CoR-Lab der Universität Bielefeld oder dem *Robotic Institut* der Carnegie Mellon University, wurde bislang eingeschränkter Zugriff für Forschungen gewährt, vergleiche u. a. [15].

<sup>5</sup>Copyright ©: Honda, [Quelle:[world.honda.com/ASIMO/](http://world.honda.com/ASIMO/)].

<sup>6</sup>Je nach Quelle unterscheiden sich die Angaben über den ersten „rennenden“ Roboter. QRIO von Sony ist nachweislich der erste Roboter, der während des Laufes gänzlich vom Boden abhob, doch war diese Flugphase so kurz, dass fraglich ist, ob diese Fortbewegung als *Renner* klassifiziert werden kann.

## 2.3.2 HRP



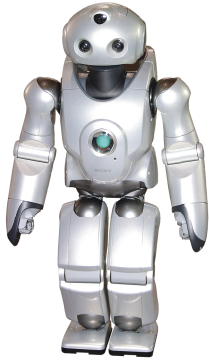
Abbildung 2.3.5: HRP-4C und HRP-2.

für die Laufsteuerung humanoider Roboter entscheidenden und wegweisenden Experimente der Forschungsgruppe um Kajita mit der HRP Reihe durchgeführt, wie u. a. die Veröffentlichungen [56], [57] oder [81] belegen. Eine sehr interessante Entwicklung stellt der HRP-4C dar, von dem ein Foto in Abbildung 2.3.5<sup>7</sup> zu finden ist. Forschungen im Bereich der Mensch-Maschine-Interaktion finden bisher primär mit Roboterköpfen oder -büsten statt. Laufroboter hingegen zeichnen sich meist nicht durch ihr realistisches Aussehen aus, wodurch der Effekt des *unscanny Valley* die Erforschung der Interaktion mit Menschen stark beeinträchtigt. Mit dem HRP-4C geht AIST nun einen entscheidenden Schritt in die Richtung integrativer Forschung beider Bereiche. Die Kombination eines HRP-4-Skeletts mit einem realistischen Roboterkopf mit nachgebildeter Haut erlaubt es Experimente aus dem Bereich der Bewegungssteuerung und der Mensch-Maschine-Interaktion zu verbinden und somit neue Forschungsbereiche zu erschließen.

Das *Humanoid Robotics Project* entstand im Jahre 1997 aus einem Honda P3 Prototyp, der von einem Zusammenschluss unterschiedlicher Firmen unter der Leitung von *Kawada Industries* weiterentwickelt wurde. Unterstützt wird das Projekt durch das *National Institute of Advanced Industrial Science and Technology* (AIST) und *Kawasaki Heavy Industries*. Über die Jahre entstanden mehrere HRP-Modelle, die in der aktuellen Version HRP-4, gipfelten. Diese hat eine Größe von 1,5 m und ein Gewicht von 40 kg. Eine detaillierte Spezifikation des HRP-4 kann beispielsweise in [62] gefunden werden. Interessant ist der HRP vor allem, da er einer der ersten menschengroßen humanoiden Roboter ist, der käuflich erworben werden kann. Durch seinen Preis von rund 250.000 € für den HRP-4 gibt es zwar nicht viele Forschungseinrichtungen die einen HRP ihr Eigen nennen, doch ist dies die einzige Plattform, die es Forschern ermöglicht Algorithmen an Hardware zu erproben, die sich auf einem mit ASIMO vergleichbaren Niveau befindet. So wurde beispielsweise die

<sup>7</sup>Copyright ©2009: AIST.

### 2.3.3 QRIO



Einer der ersten Versuche einen humanoiden Roboter kommerziell zu vermarkten wurde mit dem QRIO, Akronym für *Quest for cuRIOsity*, von der Firma Sony in Angriff genommen. Ein Foto ist zu finden in Abbildung 2.3.6<sup>8</sup>. Durch die positive Erfahrung mit dem Roboterhund AIBO zielte QRIO ebenfalls auf den Entertainment-Markt. Doch nach der ersten Vorstellung des Prototyps SDR-4X im Jahre 2002 wurde klar, dass besonders Forschungseinrichtungen Interesse an einer kommerziell erschwinglichen zweibeinigen Roboterplattform zeigten. Da aber aufgrund ihrer finanziellen Krise die Firma Sony die Roboterforschung im Jahre 2006 einstellte, hatten nur sehr wenige Forschergruppen die Gelegenheit mit einem Prototyp zu arbeiten. Die genaue Anzahl produzierter

QRIOs ist unbekannt. Der 60 cm große QRIO ist laut [35] offiziell der erste Roboter, der während des Laufens kurzzeitig den Boden mit beiden Füßen nicht mehr berührte. Diese Flugphase war jedoch so kurz, dass strittig ist, ob diese Fortbewegung bereits als Rennen klassifiziert werden kann. Dennoch ist diese Tatsache besonders durch sein verhältnismäßig hohes Gewicht von 7,3 kg bemerkenswert und zeigt, zu welcher Leistung moderne Miniaturmotoren fähig sind. Der Lauf der Firma Sony erreichte eine Geschwindigkeit von 0,8 km/h was rund 22 cm/s entspricht.

### 2.3.4 NAO

Die neu gegründete Firma *Aldebaran Robotics* rief 2004 das Projekt *NAO* ins Leben. Erklärtes Ziel ist die Entwicklung eines kommerziell erfolgreichen humanoiden Roboters. Im Jahre 2007 löste der NAO, siehe Abbildung 2.3.7<sup>9</sup>, den AIBO offiziell als Hardware der RoboCup-SPL ab. Der Roboter hat eine Größe von 58 cm und wiegt 4,5 kg. Je nach Version besitzt NAO 21, bzw. 25 Freiheitsgrade in der *Academic Version*. Kontinuierliche Weiterentwicklung und Feedback der SPL-Teams führen dazu, dass jedes Jahr für den RoboCup eine neue Version vorgestellt wird. Die aktuelle Version V3.3 ist also über die Jahre Schritt für Schritt verbessert worden. In Kombination mit dem Preis von 10.000 € hat sich daher der Roboter NAO auch außerhalb des RoboCups als eine beliebte Plattform für die Forschung etabliert. Viele Veröffentlichungen nutzen bereits diese Hardware für Experimente. Auch in Schulen hat NAO als Lernplattform Einzug gehalten und der Einsatz von Firmen als Werbe-Roboter belegt die Akzeptanz des Roboterdesigns. Der Roboter NAO ist mit einem typischen Arsenal an Sensoren ausgestattet, wobei interessant zu erwähnen ist, dass Aldebaran verschiedene Austauschköpfe mit weiteren Sensoren, beispielsweise einem Laserscanner, anbietet.



Abbildung 2.3.7: NAO.

<sup>8</sup>Copyright ©: Dschen Reinecke and Jorgebarrios. Published and reused unter CC BY-SA 3.0 license.

<sup>9</sup>Copyright ©: Aldebaran Robotics.

### 2.3.5 DARwIn-OP

Der Roboter DARwIn-Op, Synonym für *Dynamic Anthropomorphic Robot with Intelligence - Open Platform* stellt ein vielversprechendes *Open-Source-Projekt*<sup>10</sup> in der Robotik dar. Er hat eine Höhe von 46 cm und ein Gewicht von rund 3 kg. Beteiligt an der Entwicklung ist eine Kollaboration unterschiedlicher Universitäten und der koreanische Hersteller ROBOTIS. Der Roboter ist zwar auch als Bausatz erhältlich, doch unterscheidet er sich von den anderen Bausätzen des Herstellers - wie beispielsweise dem *Bioid* - dahingehend, dass er als Komplettpaket aus kinematischer Struktur, Sensoren, CPU und Software entworfen wurde. Abbildung 2.3.8<sup>11</sup> zeigt zudem, dass der Roboter sich merklich vom Design anderer Bausätze unterscheidet. Als Recheneinheit kommt ein Intel Atom Z530 Prozessor mit 1.6 GHz zum Einsatz. DARwIn ist primär für den Einsatz in der Forschung der humanoiden Robotik entwickelt worden und besitzt daher eine breite Anzahl an Schnittstellen und Sensoren. Mit einem Preis von rund 10.000 € erfreut er sich bereits großer Beliebtheit unter Forschern, obwohl er erst im Jahre 2010 der Öffentlichkeit vorgestellt wurde. Durch seine Quellenoffenheit und Baupläne bietet er interessante Optionen für die Forschung, da er zum einen eine günstige aber leistungsfähige standardisierte Plattform darstellt, die zudem auch eigene Modifikation zulässt. Somit können gleichzeitig Entwicklungen in den Bereichen Hard- und Software durchgeführt werden, ohne den Aspekt der Vergleichbarkeit zu verlieren. Die Leistungsfähigkeit der Hardware konnte der DARwIn unter Beweis stellen, indem er bereits im ersten Jahr nach seiner Veröffentlichung in der Lage war, den Weltmeistertitel in der RoboCup *Humanoid-Kid-Size-League* für sich zu entscheiden.



Abbildung 2.3.8: DARwIn-OP.

<sup>10</sup><http://sourceforge.net/projects/darwinop/>

<sup>11</sup>Copyright ©: ASIMOV Robotics Pvt Ltd.



---

### Die Roboterplattform NAO

---

Das enge Zusammenspiel von Algorithmen und Hardware in der Robotik führt dazu, dass Probleme oft sowohl durch den Einsatz neuer Hardware, als auch durch verbesserte Algorithmen gelöst werden können. Dies hat eine großen Diversität an humanoiden Forschungsrobotern zur Folge, wie bereits der selektive Überblick in Kapitel 2.3 aufzeigen konnte. Da zudem aufgrund der fehlenden Praxisreife der Markt für den Einsatz humanoider Roboter noch sehr klein ist, gab es in der Vergangenheit keine weitverbreitete kommerzielle Roboterplattform, deren Hardware für die Forschung qualitativ hochwertig genug und zugleich vom finanziellen Gesichtspunkt her interessant war. Daher wurden in der Vergangenheit Forschungen an unterschiedlichster Hardware durchgeführt, wodurch die Ergebnisse und entwickelten Algorithmen oft nicht direkt übertragbar sind. Folglich fällt es schwer von den Forschungen andere Wissenschaftler zu profitieren und zu evaluieren, ob sich eigene Algorithmen durch deren Forschung verbessern lassen. Es bleibt festzuhalten, dass dennoch in den letzten Jahren große Fortschritte im Bereich der humanoiden Roboter erlangt wurden, doch bleibt die Frage offen, ob nicht eine einheitliche Forschungshardware förderlich gewesen wäre. Aufgrund mangelnder Verfügbarkeit entsprechender Hardware ist diese Frage müßig, doch eröffnet das steigende kommerzielle Angebot humanoider Roboter dahingehend neue Wege für die Forschung.

Zur Förderung der Vergleichbarkeit wurde sich daher für diese Arbeit gegen die Entwicklung eines neuen Laufroboters entschieden. Die Arbeit beschränkt sich vielmehr in weiten Teilen auf die Erforschung geeigneter Algorithmen. Für die Experimente soll eine Plattform gewählt werden, welche ohne Probleme kommerziell erhältlich und zudem mit allen gebräuchlichen Sensoren ausgestattet ist. Dies ist eine wichtige Grundlage für jegliche Art von *Motion*-Experimenten. Hightech-Roboter, wie der vorgestellte HRP, sind hierzu eine gute Wahl, da aufgrund des Kaufpreises ihrer Hardware in der Regel sehr präzise Sensoren verbaut werden können. Dennoch fällt die Wahl bewusst auf einen Roboter aus dem *low-cost* Segment. Dies wird aller Erwartung nach Einfluss auf die Sensorik haben. Besonders die starke Nachfrage im Bereich der mobilen Unterhaltungselektronik hat dazu geführt, dass moderne Sensoren immer bessere Leistung zu immer



günstigerem Preis liefern. Dennoch sind die in *low-cost* Robotern verbauten Sensoren aufgrund ihrer Größe und des Preises in der Regel nicht derart leistungsstark und präzise, wie beispielsweise die des ASIMO. Trotzdem erscheint die Entscheidung sinnvoll, da Forschung an humanoiden Robotern auf lange Sicht nur Relevanz besitzt, wenn sie auch mit kommerziellem Erfolg eingesetzt werden kann. Folglich wird ein besonderer Fokus der Arbeit darauf gelegt, wie Algorithmen mit fehlerbehafteten Sensorwerten umgehen.

Nach anfänglichen Experimenten mit Roboterbausätzen der Firmen *Kondo Kagaku* und *Lynxmotion*, wurde von Bausätzen Abstand genommen. Zwar ist die Möglichkeit diese individuell mit Hardware und Sensorik auszustatten reizvoll, doch führt dies zu einer nicht gewünschten Diversität der Plattformen. Da zudem der Steuercomputer und somit auch die Software nicht standardisiert sind, lassen sich Programme und implementierte Algorithmen nicht austauschen. Es wurde daher gezielt nach einer Alternative gesucht, die vom Hersteller als Komplettpaket angeboten wird. Nach eingehender Betrachtung und einem Vergleich der in Kapitel 2.3 vorgestellten Roboter fällt zunächst auf, dass ASIMO von Honda die am weitesten fortgeschrittene Hardwareplattform ist, aber nicht kommerziell erworben werden kann. Durch den Einsatz von speziell entwickelter Hardware sind die Herstellungskosten vermutlich so hoch<sup>1</sup>, dass der Roboter für einen kommerziellen Einsatz im Massenmarkt selbst dann zu teuer wäre, wenn Honda ihn zum Kauf anbieten würde. Ähnlich verhält es sich mit dem HRP-4 von *Kawada Industries*. Dieser ist zwar kommerziell erhältlich, aber aufgrund der hohen Anschaffungskosten von rund 250.000 € nur für wenigen Forschungseinrichtungen und noch weniger Endverbraucher erschwinglich. Aufgrund des Kostenfaktors kommen also nur kleinere Roboter in Betracht. Da Sony die Entwicklung des QRIO aus finanziellen Gründen zusammen mit der Produktion des AIBO im Jahre 2006 einstellte, ist dieser für die Experimente keine Option. Der DARwIn-OP ist ein robuster und gut ausgestatteter Roboter zu einem günstigen Preis, der sich zukünftig als Experimentalplattform eignet. Zum Zeitpunkt an dem mit der Forschung dieser Arbeit begonnen wurde, war er allerdings noch nicht verfügbar und konnte daher nicht verwendet werden. Somit verbleibt nur der NAO der Firma *Aldebaran Robotics* als geeignete Plattform. Er ist, je nach Version, für Preise zwischen 4.000 € und 10.000 € erhältlich und ist aufgrund seiner Hardwareausstattung bestens für Laufexperimente geeignet. Der Vollständigkeit halber sei erwähnt, dass auch wenn die Wahl zwischen NAO und DARwIn-OP bestanden hätte dem NAO der Vorzug gegeben worden wäre. Dies liegt aber lediglich in seiner weiteren Verbreitung begründet. Allerdings bietet für die Zukunft eine verbessertes Angebot unterschiedlicher kostengünstiger humanoider Roboter die Chance vergleichende Experimente direkt auf zwei unterschiedlichen Robotern durchzuführen und somit zu untersuchen ob sich algorithmische Vorteile gleichermaßen auf unterschiedliche Hardware übertragen lassen.

Da der Roboter NAO die Plattform für fast alle experimentellen Versuche dieser Arbeit darstellt, folgt an dieser Stelle eine detailliertere Vorstellung seiner Hardware und der verwendeten Software, um ein besseres Verständnis der Rahmenbedingungen zu bieten. Der Roboter wird von Aldebaran als komplett montierter Roboter geliefert, der sich ohne Fachwissen über die Konstruktion nicht in seine Einzelteile zerlegen lässt, ohne ihn zu zerstören. Da zudem der Zugriff auf die Sensoren des Roboters von Aldebaran

---

<sup>1</sup>Honda veröffentlicht hierzu keine Daten.



gekapselt wurde - für Details siehe Kapitel 3.5 - muss sich bei der Vorstellung meist auf Herstellerangaben über die verbaute Hardware verlassen werden. Es ist weiterhin zu beachten, dass der Roboter von seinem Hersteller kontinuierlich weiterentwickelt wird. Daher gab es im Laufe seines Entwicklungszyklus unterschiedliche Versionen der Hardware, die Forschungseinrichtungen teilweise bereits vor der Markteinführung zugänglich waren. Somit wurden für die Experimente dieser Arbeit über die Zeit unterschiedliche Versionen der Hardware, beginnend mit der Version *V2* bis hin zur derzeit aktuellen Version *V3.3* verwendet. Da sich der grundlegende Aufbau des Roboters über diese Produktevolution nicht maßgeblich verändert hat, wird an dieser Stelle nur die Version *V3.3* vorgestellt und nur gegebenenfalls auf markante Änderungen eingegangen.

### 3.1 Kinematische Struktur

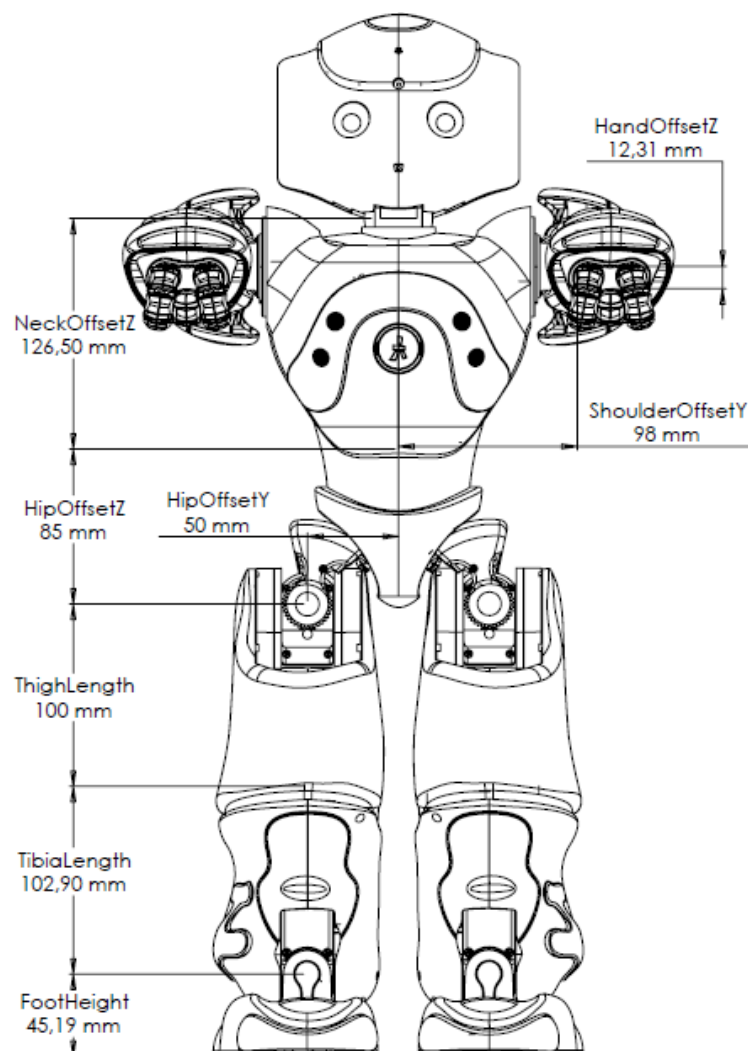


Abbildung 3.1.1: Abmessungen des Roboters NAO entnommen aus [1].

Die Lehre der *Kinematik* beschäftigt sich in der Robotik mit der mathematischen Beschreibung des strukturellen Aufbau der Mechanik des Roboters. Hierbei finden lediglich die Geometrie und die Beschreibung des Bewegungsverhaltens Berücksichtigung. Von Ursachen für die Bewegung in Form von Kräften, die auf Roboterkomponenten wirken, wird abstrahiert. Der Kinematik steht die Lehre der *Dynamik* des Roboters gegenüber, die diese physikalischen Zusammenhänge betrachtet, vergleiche auch [49].

Der Roboter NAO hat im aufrechten Stand eine Höhe von 58 cm und wiegt 4,3 kg. Er besteht aus einzelnen Körperteilen, die aus Metall und Plastik aufgebaut und jeweils mit einer Verschalung aus Plastik ummantelt sind. Eine schematische Darstellung mit detaillierteren Abmessungen ist in Abbildung 3.1.1<sup>2</sup> zu finden. Die Steifigkeit der Körperteile ist hoch genug, um diese im Weiteren als starrer Festkörper zu betrachten<sup>3</sup>. Wie bei modernen Robotern üblich, sind diese Festkörper durch Gelenke, engl. *joints*, rotatorisch untereinander verbunden, die von Motoren angetrieben werden<sup>4</sup>. Das Resultat sind sogenannte *kinematische Ketten* aus gekoppelten Körpern[49]. Ein humanoider Roboter ahmt den menschlichen Körperbau nicht nur in den Beinen, sondern auch im Aufbau der Arme und des Kopfes nach. Folglich ist klar, dass sich seine Struktur nicht durch eine einzige kinematische Kette beschreiben lässt sondern in einer Baumstruktur resultiert. In dieser stellt jede Gliedmaße eine eigene Kette dar, deren Ende der jeweilige Endeffektor bildet. Die gemeinsame Wurzel ist der Torso. Für die Planung der Bewegung eines Roboters spielen die Freiheitsgrade, aus dem Englischen abgekürzt mit DoF, eine entscheidende Rolle, da diese die variablen Größen zur Steuerung und Regelung des Systems darstellen.

Aldebaran hat die Gelenke deren DoF-Achse entsprechend in *Roll-Pitch-Yaw*-Notation bezeichnet. Um eine einheitliche Notation zwischen Veröffentlichungen zu gewährleisten, werden diese im Folgenden mit der englischen Originalbezeichnung übernommen. Die Zuordnung der einzelnen Gelenke zu den kinematischen Ketten ist Tabelle 3.1.1 zu entnehmen. Die Aufzählung von links nach rechts ist beginnend mit dem Gelenk sortiert, das sich am weitesten vom Torso entfernt befindet.

Gliedmaße	Gelenke
Bein	AnkleRoll, AnklePitch, KneePitch, HipPitch, HipRoll, HipYawPitch
Arm	ElbowRoll, ElbowYaw, ShoulderRoll, ShoulderPitch
Kopf	HeadPitch, HeadYaw

Tabelle 3.1.1: Gelenkzuordnung der kinematischen Ketten des NAO.

Jedes Bein hat demnach 6 Freiheitsgrade, jeder Arm 4 und der Kopf 2. Da der Torso unflexibel ist, würde sich damit die Anzahl der Freiheitsgrade zu 22 aufsummieren. Durch eine Besonderheit im kinematischen Design des NAO sind aber die beiden schräg stehenden Hüftgelenke gekoppelt, so dass *LeftHipYawPitch* immer den gleichen Wert wie

<sup>2</sup>Copyright ©2012: Aldebaran-Robotics.

<sup>3</sup>Es ist klar, dass diese Annahme von der Realität abstrahiert, da durch die Materialien eine gewisse, wenn auch sehr geringe, Flexibilität unvermeidbar ist.

<sup>4</sup>Eine Ausnahme hierzu stellen Roboter mit künstlichen Muskeln dar. Diese sind jedoch durch ihren fragilen Aufbau noch weit entfernt von einem praktisch relevanten Einsatz. Daher sei an diese Stelle nur auf eine Übersicht in [65] verwiesen.

*RightHipYawPitch* annimmt. Dadurch wird ein Freiheitsgrad eingebüßt und beide Beine haben in Summe nur 11 und der Roboter nur 21 Freiheitsgrade. Abbildung 3.1.2<sup>5</sup>(a) zeigt die Positionierung der jeweiligen Gelenke und Abbildung 3.1.2(b) verdeutlicht die resultierende kinematische Struktur. Der Vollständigkeit halber ist zu erwähnen, dass von der Version V3.3 des NAO auch eine sogenannte *Academics Version* existiert. Diese unterscheidet sich von der hier vorgestellten u. a. darin, dass die Hände des NAO motorisiert sind. Die Arme werden jeweils um die beiden Freiheitsgrade *Hand* und *WristYaw* ergänzt. Ihre Freiheitsgrade erhöhen sich also um je 2 und die Gesamtanzahl steigt auf 25. Da für keine untersuchte Anwendung diese DoF von Relevanz sind, wird sich weiterhin auf die Beschreibung der RoboCup-Version beschränkt.

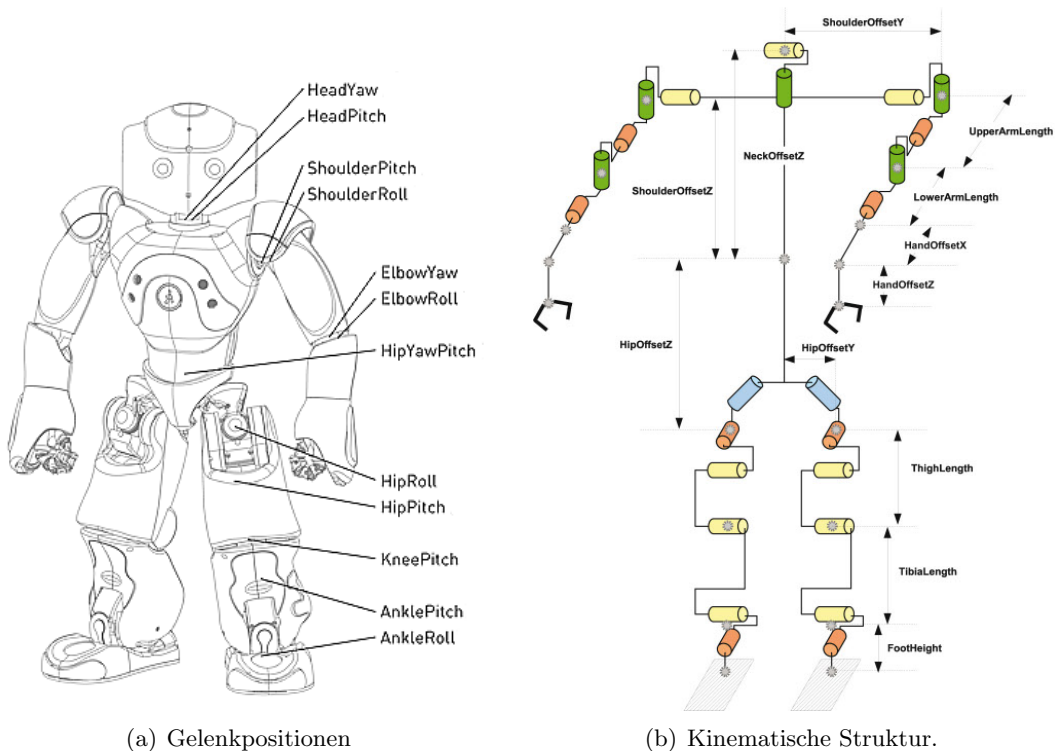


Abbildung 3.1.2: Gelenke des NAO entnommen aus [1].

### 3.1.1 Direkte Kinematik

Die *direkte Kinematik*, auch als *Vorwärtskinematik* bezeichnet, berechnet aus einer gegebenen Konfiguration der Gelenkpositionen  $K_n = J_0, \dots, J_{n-1}$  im Arbeitsraum des Roboters die Positionen seiner Festkörper im Weltkoordinatensystem (WKS). Wie in Kapitel 3.1 erläutert hat der Arbeitsraum des NAO 21 Dimensionen. Die Stellung der 21 Gelenke kann über einen Sensor an der jeweiligen Achse direkt abgelesen werden. Die Konfiguration  $K_{21}$  des Roboters im Arbeitsraum ist dadurch zu jedem Zeitpunkt eindeutig bestimmbar. Aus  $K$  kann dann wiederum über die kinematische Struktur des Roboters die Position der Körperteile im roboterzentrischen Koordinatensystem

<sup>5</sup>Copyright ©2012: Aldebaran-Robotics.

(RzKS) eindeutig berechnet werden. Da der Ursprung des RzKS im Torso liegt, und somit im Zentrum der kinematischen Baumstruktur, lässt sich diese wieder in die einzelnen kinematischen Ketten der Extremitäten aufteilen. Folglich können diese entkoppelt betrachtet werden.

Zur Berechnung sei, beginnend mit dem Torso  $F_0$ , für jeden Festkörper  $F_i$  des NAO ein körpereigenes Koordinatensystem (KeKS) festgelegt. Der Ursprung und die Orientierung von  $F_0$  sei übereinstimmend mit dem Ursprung und der Orientierung des RzKS. Weiterhin sei der Ursprung jedes weiteren Festkörperkoordinatensystems, ohne Beschränkung der Allgemeinheit, so definiert, dass er sich auf der Gelenkachse zwischen dem Festkörper  $F_{i-1}$  und  $F_i$  befindet. Ausgehend von dem Ursprung des jeweiligen Koordinatensystems des Körpers  $F_{i-1}$  kann dann der Übergang in das Koordinatensystem des angrenzenden Körpers  $F_i$  in jedem Fall durch eine Translation, gefolgt von einer Rotation, beschrieben werden

$$\mathbf{T}_{i-1,i} = \mathbf{Rot}_\alpha \cdot \mathbf{Rot}_\beta \cdot \mathbf{Rot}_\gamma \cdot \mathbf{Trans}_l. \quad (3.1.1)$$

Die Translation ist durch die Dimension des jeweiligen Bauteils festgelegt und somit kann die Translation aus der von Aldebaran angegebene Dokumentation<sup>6</sup> übernommen werden. Der rotatorische Freiheitsgrad der Gelenke verläuft immer entlang einer Achse, so dass zwei der drei Rotationen ebenfalls bauartbedingt festgelegt sind. Nur eine Rotation ist variabel. Diese kann durch die Sensoren der Gelenkachsen bestimmt werden.

Zur Berechnung der Vorwärtskinematik kinematischer Ketten ist in der Robotik die Verwendung der *Denavit-Hartenberg-Transformation*, abgekürzt als *DH-Transformation* [49] gebräuchlich. Dies ist ein mathematisches Verfahren, das auf Basis von homogenen Matrizen eine eindeutige Überführungsvorschrift von einem Ortskoordinatensystem in ein anderes beschreibt, unter der Bedingung, dass deren Koordinatensysteme so definiert sind, dass diese die DH-Konventionen erfüllen:

1. Die  $z_i$ -Achse des Festkörpers liegt entlang der Bewegungsachse des  $i + 1$ -ten Gelenks.
2. Die  $x_i$ -Achse des Festkörpers ist das Kreuzprodukt der Achsen  $z_i$  und  $z_{i-1}$ .
3. Die  $y_i$ -Achse des Festkörpers ergänzt  $x_i$  und  $z_i$  zu einem rechtshändigen System.

Weiterhin gelten folgende Definitionen:

$\theta_i$ : Winkel der Rotation um  $z_{i-1}$ , die die  $x_{i-1}$ -Achse und die  $x_i$ -Achse zur Deckung bringt.

$d_i$ : Abstand des Ursprungs des  $(i - 1)$ -ten KeKS zum Schnittpunkt der Normalen zwischen der  $i$ -ten und der  $(i + 1)$ -ten Gelenkachse mit der  $i$ -ten Gelenkachse.

---

<sup>6</sup>Es sei darauf hingewiesen, dass die Herstellerangaben nicht weiter verifiziert werden können, da die Position der Achsen durch die äußere Verkleidung des Roboters verdeckt wird und somit nur sehr grob abschätzbar ist. Da die Angaben laut Aldebaran direkt aus dem CAD-Modell der Fertigung entnommen sind, ist anzunehmen, dass diese genauer sind, als Daten, die durch Ausmessen der Körperteile mit vorhanden Mitteln hätten erreicht werden können.

$a_i$ : Abstand der Gelenke, die durch die  $z$ -Einheitsvektoren des  $i$ -ten und des  $(i-1)$ -ten KeKS verlaufen.

$\alpha_n$ : Winkel zwischen der  $z_{i-1}$ -Achse und der  $z_i$ -Achse bei Drehung um die  $x_i$ -Achse.

Dann ist die Transformationen des Koordinatensystems  $i-1$  in das Koordinatensystem  $i$  in homogenen Koordinaten eindeutig beschreiben durch:

$$\begin{aligned} \mathbf{T}_{i-1,i} &= \mathbf{Rot}_{z_{i-1}}(\theta_i) \cdot \mathbf{Trans}_{z_{i-1}}(d_i) \cdot \mathbf{Trans}_{x_i}(a_i) \cdot \mathbf{Rot}_{x_i}(\alpha_i) \\ &= \begin{pmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned} \quad (3.1.2)$$

Die DH-Parameter des linken und rechten Beines des Roboters NAO sind in den Tabellen 3.1.2 und 3.1.3 zu finden. Da die direkte Kinematik der Arme nicht für die weiteren Experimente benötigt wird, wird an dieser Stelle auf eine Berechnung verzichtet.

Gelenk	$\alpha$	a	$\theta$	d	Offset
HipYawPitch	0	0	$\theta_0^l$	85	0
HipRoll	$-\pi/2$	0	$\theta_1^l$	0	$-\pi/4$
HipPitch	$\pi/2$	0	$\theta_2^l$	0	0
KneePitch	0	100	$\theta_3^l$	0	0
AnklePitch	0	102,9	$\theta_4^l$	0	0
AnkleRoll	$\pi/2$	0	$\theta_5^l$	0	0

Tabelle 3.1.2: DH-Parameter der kinematischen Kette des linken Beins.

Gelenk	$\alpha$	a	$\theta$	d	Offset
HipYawPitch	0	0	$\theta_0^r$	85	0
HipRoll	$-\pi/2$	0	$\theta_1^r$	0	$-\pi/4$
HipPitch	$-\pi/2$	0	$\theta_2^r$	0	0
KneePitch	0	100	$\theta_3^r$	0	0
AnklePitch	0	102,9	$\theta_4^r$	0	0
AnkleRoll	$\pi/2$	0	$\theta_5^r$	0	0

Tabelle 3.1.3: DH-Parameter der kinematischen Kette des rechten Beins.

### 3.1.2 Inverse Kinematik

Die *inverse Kinematik*, auch als *Rückwärtstransformation* bezeichnet, berechnet zu einer gegebenen Position eines TCP in Weltkoordinaten eine zugehörige Konfiguration  $K_n = J_0, \dots, J_{n-1}$  im Gelenkwinkelraum des Roboters. Mathematisch ist dies gleichbedeutend mit der Bestimmung der Gelenkstellungen des Roboters aus einer vorgegebenen Gesamt-Transformationsmatrix  $\mathbf{T}_{0,n}$ . Die Lösung muss nicht eindeutig sein, da es mehrere Konfigurationen geben kann, die zu dergleichen Stellung des Endeffektors führen.

Zur Lösung dieses Problems sind unterschiedliche Verfahren gebräuchlich, die sich in *iterative* und *analytische* Methoden unterteilen lassen.

**Iterative Methode** Iterative Kinematiken versuchen eine Näherung der Lösung zu finden, indem die Bewegung in differenzielle Abschnitte unterteilt wird. Dies führt dazu, dass Teile der Bewegungsgleichungen linearisiert betrachtet werden können und für die Transformation die sogenannte *Jacobi-Matrix*, vergleiche beispielsweise [19], angewandt werden kann. Dieser, auch als *Universaltransformation* bekannte, Ansatz bietet den Vorteil unabhängig von der eigentlichen Kinematik zu sein. Er ist somit auch anwendbar, wenn diese nicht bekannt ist oder nicht mit einem analytischen Verfahren gelöst werden kann. Da es sich um eine numerische Lösung handelt, entsteht aber in der Regel eine Abweichung von der korrekten Lösung. Wenn das Verfahren konvergiert, kann dieser Fehler geringer gehalten werden, indem die Anzahl der Iterationsschritte erhöht wird, was sich aber in einem höheren Rechenaufwand nieder schlägt. Da aber eine Konvergenz des Verfahrens nicht garantiert werden kann, ist es bei dieser Methode nicht möglich eine feste obere Zeitschranke anzugeben, in der ein minimaler Fehler erreicht wird. Somit muss für eine echtzeitfähige Anwendung die maximale Anzahl Iterationen begrenzt werden, um deterministisch zu einer Lösung zu gelangen. Dies kann begünstigt werden, indem die differentiellen Bewegungsupdates klein gewählt werden, um durch den Startpunkt eine Konvergenz des Verfahren zu bevorzugen. Die logische Konsequenz ist dann aber, dass der Algorithmus mit einer höheren Frequenz aufgerufen wird, wodurch sich der Rechenaufwand erhöht.

**Analytische Methode** Dieser Klasse von Ansätzen ist gemein, dass sie die Lösung der Kinematik über eine Berechnungsvorschrift direkt ermöglichen. Dies wird auch als *geschlossene* Lösung bezeichnet. Es ist nicht in jedem Fall möglich eine analytische Methode zur Lösung der Kinematik zu verwenden. Wenn doch, so ist diese abhängig von der Art der Kinematik mit unterschiedlichen Lösungsverfahren zu erreichen:

1. **Algebraisch.** In diesem Fall kann aus der durch die Lage des TCP vorgegebenen Gleichung  $\mathbf{T}_{0,n} = \mathbf{T}_{0,1} \cdot \mathbf{T}_{1,2} \cdot \dots \cdot \mathbf{T}_{n-1,n}$  mit Hilfe der kinematischen Nebenbedingungen unmittelbar einer der Gelenkwinkel berechnet werden. Sei dieser o.B.d.A.  $\Theta_1$ , so ist das Frame  $\mathbf{T}_{0,1}$  bekannt und die Transformationsgleichung kann zu  $\mathbf{T}_{0,1}^{-1} \cdot \mathbf{T}_{0,n} = \mathbf{T}_{1,2} \cdot \dots \cdot \mathbf{T}_{n-1,n}$  vereinfacht werden. Wiederholung des Verfahrens liefert nach und nach alle Gelenkwinkel.
2. **Geometrisch.** Bei dieser Lösung wird explizites Wissen über die Struktur des Roboters ausgenutzt, um die Lösung des kinematischen Problems durch Gesetze der Geometrie zu lösen.
3. **Kombination von algebraisch und geometrisch.** In einigen Fällen kann auf einen Teil der Gelenkwinkel mittels des geometrischen Verfahrens geschlossen werden, andere müssen aber analytisch gelöst werden.

Nach diesem kurzen theoretischen Exkurs kehren wir zurück zur Rückwärtstransformation des Roboters NAO. Wie auch bei der Vorwärtstransformation besteht grundsätzlich

keine Notwendigkeit, eine Lösung für die gesamte komplexe kinematische Baumstruktur des Roboters zu finden. Daher wird analog zu dieser vorgegangen und eine Kinematik für jede Kette der Extremitäten einzeln berechnet. Da die Arm- und Kopfketten für die Untersuchungen dieser Arbeit keine Relevanz besitzen, wird im Folgenden nur die Berechnung der Bein kinematik durchexerziert.

Durch das gekoppelte *HipYawPitch*-Gelenk ist dies aber für die Beine nicht ohne weiteres möglich. Dennoch wird zunächst das Problem so betrachtet, als wenn diese unabhängig wären. Die Lösung der Bedingung wird später mit Hilfe einer *Constraint*-Berechnung gesucht. Im Allgemeinen gilt, dass bei Existenz einer analytischen Lösung, dieser grundsätzlich der Vorzug gegenüber einer numerischen Berechnung zu geben ist. Es kann gezeigt werden, dass, um das Problem der inversen Kinematik analytisch lösen zu können, eine der folgenden Bedingungen hinreichend ist [49]:

1. Drei aufeinanderfolgende Achsen des Roboters schneiden sich in einem Punkt.
2. Drei aufeinander folgende Achsen sind parallel zueinander.

Die kinematische Struktur des NAO ist in Abbildung 3.1.2(b) zu finden. Sie lässt erkennen, dass sich die drei Hüftgelenke in einem Punkt schneiden. Demnach ist die erste Bedingung erfüllt und die Existenz einer analytischen Lösung garantiert. Dennoch war das RoboCup Team des Instituts für Roboterforschung das einzige, welches bereits im Jahre 2008, dem ersten Jahr der neuen NAO SPL Liga, mit einer geschlossenen kinematischen Lösung antreten konnte. Selbst der Hersteller Aldebaran veröffentlichte erst im Jahre 2010 eine analytische Lösung der inversen Kinematik. Grund hierfür ist vermutlich nicht nur die Koppelung der *HipYawPitch*-Gelenke, sondern auch deren Schrägstellung um einen Winkel von  $45^\circ$ , vergleiche Abbildung 3.1.2(b).

**Berechnung der inversen NDD-Kinematik des NAO** Da die inverse Kinematik die Grundlage aller folgenden Experimente bildet, und diese für die Beine durch die Schrägstellung des *HipYawPitch*-Gelenks nicht unmittelbar gelöst werden kann, folgt die Herleitung der NDD-Kinematik für je eines der Beine. Hierzu werden geometrisches und analytisches Vorgehen kombiniert. Das Bein des NAO besteht aus 6 Gelenken, die beginnend mit dem obersten Hüftgelenk von 0 bis 5 durchnummeriert werden. Die Transformation  $\mathbf{T}_{0,1}$  beschreibt also die Transformation vom Ursprung des robotereigenen Koordinatensystem (ReKS) in das KeKS des ersten Hüftgelenkes *HipYawPitch* und wird bestimmt durch die Translation  $T_{0,1}$  und den Winkel  $\Theta_0$  usw. Es ist anzumerken, dass nach dieser Definition  $\mathbf{T}_{5,6}$  lediglich durch einen feste Translation  $T_{5,6}$  in den Ursprung des Fußkoordinatensystems (FKS) gegeben ist.

In homogenen Koordinaten ausgedrückt lässt sich die Stellung des Endeffektors eines Beins des NAO dann beschreiben durch:

$$\mathbf{T}_{0,6} = \mathbf{T}_{0,1} \cdot \mathbf{T}_{1,2} \cdot \dots \cdot \mathbf{T}_{5,6}. \quad (3.1.3)$$

Die Translationen  $T_{i,i+1}$  der homogenen Matrizen sind durch den kinematischen Aufbau vorgegeben und folglich bekannt, vgl. Abbildung 3.1.2. Die Winkel  $\Theta_i$  sind die gesuchten Größen der Rücktransformation. Daher muss nun zu  $\mathbf{T}_{0,6}$  die Stellung der

Gelenke  $\Theta_0, \dots, \Theta_5$  berechnet werden, die diese erfüllt. Die Translation  $T_{0,1}$  ist durch die Abmessung des Torsos gegeben.  $T_{1,2}$  und  $T_{2,3}$  sind Null, da sich die Hüftgelenke in einem Punkt schneiden. Folglich ist  $T_{0,3}$  bekannt.  $T_{5,6}$  ist durch die Abmessung des Fußes festgelegt. Damit berechnet sich die verbleibende Translation  $T_{3,5}$  aus  $T_{0,6}$  zu:

$$T_{0,6} = T_{0,3} + T_{3,5} + T_{5,6} \quad (3.1.4)$$

$$\Leftrightarrow T_{3,5} = T_{0,6} - T_{0,3} - T_{5,6}. \quad (3.1.5)$$

Die Translationen  $T_{3,4}$  und  $T_{4,5}$  sind gegeben durch die Abmessungen *TibiaLength* respektive *ThighLength* des NAO. Wie Abbildung 3.1.3 zeigt bilden diese mit  $T_{3,5}$  ein Dreieck. Alle Seitenlängen dieses Dreiecks sind bekannt und der Winkel  $\gamma$  kann mittels des Kosinussatzes berechnet werden:

$$\gamma = \arccos \frac{T_{3,4}^2 + T_{4,5}^2 - T_{3,5}^2}{2 \cdot T_{3,4} \cdot T_{4,5}}. \quad (3.1.6)$$

Da der Winkel  $\Theta_3$  bei ausgestrecktem Bein die Nullstellung einnimmt, berechnet sich dieser aus  $\gamma$  nach

$$\Theta_3 = \pi - \gamma. \quad (3.1.7)$$

In Nullstellung des des Winkel  $\Theta_4$  steht der Nao aufrecht. Abbildung 3.1.3 verdeutlicht dies, indem gestrichelt die Position des *KneePitch*-Gelenkes für diesen Fall eingezeichnet ist. Man erkennt, dass sich  $\Theta_4$  aus dem Winkel  $\alpha$  und einem Restwinkel  $\beta$  zusammensetzt. Der Winkel  $\alpha$  kann analog zu  $\gamma$  mit Hilfe des Kosinussatzes bestimmt werden:

$$\alpha = \arccos \frac{T_{4,5}^2 + T_{3,5}^2 - T_{3,4}^2}{2 \cdot T_{4,5} \cdot T_{3,5}}. \quad (3.1.8)$$

Durch Verlängerung der Translation  $T_{3,5}$  erhält man denselben Anwinkel  $\beta$  und somit kann dieser aus den Komponenten der Translation  $T_{5,6} = (x, y, z)$ , gegeben durch die Fußabmessungen, berechnet werden:

$$\beta = \arctan \frac{x}{\sqrt{y^2 + z^2}}. \quad (3.1.9)$$

Abschließend erhält man  $\Theta_4$  durch Summation

$$\Theta_4 = \alpha + \beta. \quad (3.1.10)$$

$\Theta_5$  ergibt sich aus analoger Überlegung zu  $\beta$  durch:

$$\Theta_5 = \arctan \frac{x}{z}. \quad (3.1.11)$$

Damit sind die Winkel  $\Theta_3$ ,  $\Theta_4$  und  $\Theta_5$  gelöst und die Transformation  $\mathbf{T}_{3,5}$  eindeutig bestimmt. Es verbleibt noch zur Lösung:

$$\mathbf{T}_{0,6} \cdot \mathbf{T}_{3,4}^{-1} \cdot \mathbf{T}_{4,5}^{-1} \cdot \mathbf{T}_{5,6}^{-1} = \mathbf{T}_{0,1} \cdot \mathbf{T}_{1,2} \cdot \mathbf{T}_{2,3} = \mathbf{T}_{0,3}. \quad (3.1.12)$$



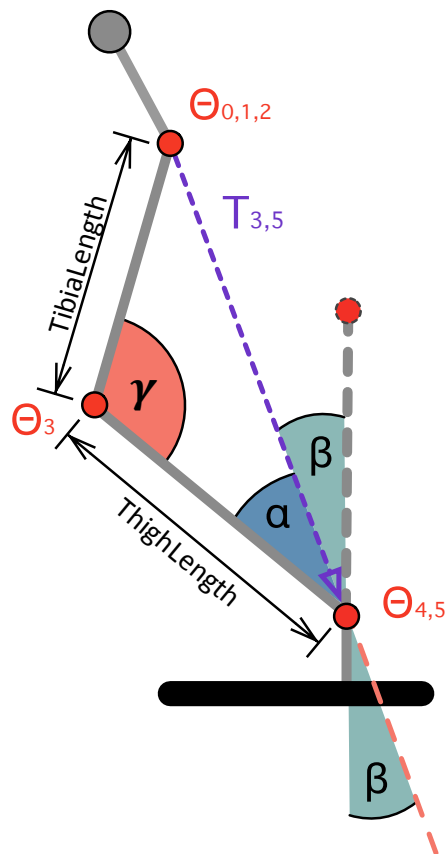


Abbildung 3.1.3: Inverse Kinematik des NAO.

Die Transformation  $\mathbf{T}_{0,3}$  kann folglich aus der Gesamt-Transformation bestimmt werden, aber die Rücktransformation ist nicht mehr auf geometrischem Wege zu lösen. Um eine analytische Lösung zu finden wird nun die Rotation  $\mathbf{R}_{0,3}$  genauer betrachtet. Bei dieser ist besonders die Querstellung des Gelenkes *HipYawPitch* zu beachten. Diese wird durch eine explizite Transformation des KeKS mittels einer festen Rotation um  $45^\circ$  und anschließende Rücktransformation berücksichtigt. Für ihre Rotation  $\mathbf{R}_{0,3}$  ergibt sich dann nach der Definition der Euler-Winkel:

$$\mathbf{R}_{0,3} = \mathbf{R}_{45} \cdot \mathbf{R}_{\Theta_0}^{Yaw} \cdot \mathbf{R}_{45}^{-1} \cdot \mathbf{R}_{\Theta_1}^{Pitch} \cdot \mathbf{R}_{\Theta_2}^{Roll} = \begin{pmatrix} R_1 & R_2 & R_3 \\ R_4 & R_5 & R_6 \\ R_7 & R_8 & R_9 \end{pmatrix}. \quad (3.1.13)$$

Ausmultiplizieren<sup>7</sup> der Rotationsmatrizen liefert für die Faktoren  $R_1$  bis  $R_9$  des rechten Beines:

$$\begin{aligned} R_1 &= \cos \Theta_0 \cos \Theta_2 - \frac{\sqrt{2}}{2} \sin \Theta_2 (\sin \Theta_0 \cos \Theta_1 + \sin \Theta_0 \sin \Theta_1), \\ R_2 &= \frac{\sqrt{2}}{2} \sin \Theta_0 \sin \Theta_1 - \frac{\sqrt{2}}{2} \sin \Theta_0 \cos \Theta_1, \\ R_3 &= \cos \Theta_0 \sin \Theta_2 + \frac{\sqrt{2}}{2} \cos \Theta_2 (\sin \Theta_0 \cos \Theta_1 + \sin \Theta_0 \sin \Theta_1), \\ R_4 &= \sin \Theta_2 \left( \cos \Theta_1 \left( \frac{\cos \Theta_0}{2} - \frac{1}{2} \right) + \sin \Theta_1 \left( \frac{\cos \Theta_0}{2} + \frac{1}{2} \right) \right) + \frac{\sqrt{2}}{2} \sin \Theta_0 \cos \Theta_2, \\ R_5 &= \cos \Theta_1 \left( \frac{\cos \Theta_0}{2} + \frac{1}{2} \right) - \sin \Theta_1 \left( \frac{\cos \Theta_0}{2} - \frac{1}{2} \right), \\ R_6 &= \cos \Theta_2 \left( \cos \Theta_1 \left( -\frac{\cos \Theta_0}{2} + \frac{1}{2} \right) + \sin \Theta_1 \left( -\frac{\cos \Theta_0}{2} - \frac{1}{2} \right) \right) + \frac{\sqrt{2}}{2} \sin \Theta_0 \sin \Theta_2, \\ R_7 &= -\sin \Theta_2 \left( \cos \Theta_1 \left( \frac{\cos \Theta_0}{2} + \frac{1}{2} \right) + \sin \Theta_1 \left( \frac{\cos \Theta_0}{2} - \frac{1}{2} \right) \right) - \frac{\sqrt{2}}{2} \sin \Theta_0 \cos \Theta_2, \\ R_8 &= \cos \Theta_1 \left( -\frac{\cos \Theta_0}{2} + \frac{1}{2} \right) + \sin \Theta_1 \left( \frac{\cos \Theta_0}{2} + \frac{1}{2} \right), \\ R_9 &= \cos \Theta_2 \left( \cos \Theta_1 \left( \frac{\cos \Theta_0}{2} + \frac{1}{2} \right) + \sin \Theta_1 \left( \frac{\cos \Theta_0}{2} - \frac{1}{2} \right) \right) - \frac{\sqrt{2}}{2} \sin \Theta_0 \sin \Theta_2. \end{aligned}$$

Aus Gründen der Symmetrie verläuft die Berechnung der Rotation des linken Beins analog, jedoch mit

$$R_2^{links} = -R_2^{rechts} \quad (3.1.14)$$

$$R_4^{links} = -R_4^{rechts} \quad (3.1.15)$$

$$R_6^{links} = -R_6^{rechts} \quad (3.1.16)$$

$$R_8^{links} = -R_8^{rechts}. \quad (3.1.17)$$

Die analytische Berechnung der verbleibenden Winkel geschieht nun, indem das Gleichungssystem durch Kombination geeigneter Terme sukzessive gelöst wird. Dies wird im Folgenden exemplarisch für das rechte Bein durchgeführt. Bei der Lösung müssen

<sup>7</sup>Auf eine Ausführung wird an dieser Stelle verzichtet, da die Berechnung unter Zuhilfenahme der Matlab *Symbolic Math Toolbox* durchgeführt wurde.

Sonderfälle betrachtet werden, die durch sogenannte *Singularitäten* in der Kinematik entstehen. Auf diese wird allerdings hier nicht weiter eingegangen, sondern lediglich der allgemein Fall betrachtet.

Man erkennt, dass sich durch Kombination von  $R_4$  und  $R_7$ , bzw.  $R_6$  und  $R_9$  ein Gleichungssystem ohne  $\Theta_0$  ergibt:

$$\begin{aligned}
 R_4 + R_7 &= \sin \Theta_2 \left( \cos \Theta_1 \left( \frac{\cos \Theta_0}{2} - \frac{1}{2} \right) + \sin \Theta_1 \left( \frac{\cos \Theta_0}{2} + \frac{1}{2} \right) \right) + \frac{\sqrt{2}}{2} \sin \Theta_0 \cos \Theta_2 \\
 &+ \sin \Theta_2 \left( \cos \Theta_1 \left( -\frac{\cos \Theta_0}{2} - \frac{1}{2} \right) + \sin \Theta_1 \left( -\frac{\cos \Theta_0}{2} + \frac{1}{2} \right) \right) - \frac{\sqrt{2}}{2} \sin \Theta_0 \cos \Theta_2, \\
 &= -\sin \Theta_2 (\cos \Theta_1 - \sin \Theta_1). \\
 R_6 + R_9 &= \cos \Theta_2 \left( \cos \Theta_1 \left( -\frac{\cos \Theta_0}{2} + \frac{1}{2} \right) + \sin \Theta_1 \left( -\frac{\cos \Theta_0}{2} - \frac{1}{2} \right) \right) + \frac{\sqrt{2}}{2} \sin \Theta_0 \sin \Theta_2 \\
 &+ \cos \Theta_2 \left( \cos \Theta_1 \left( \frac{\cos \Theta_0}{2} + \frac{1}{2} \right) + \sin \Theta_1 \left( \frac{\cos \Theta_0}{2} - \frac{1}{2} \right) \right) - \frac{\sqrt{2}}{2} \sin \Theta_0 \sin \Theta_2, \\
 &= \cos \Theta_2 (\cos \Theta_1 - \sin \Theta_1).
 \end{aligned}$$

Durch weitere Kombination der beiden Faktoren kann aus diesem  $\Theta_1$  eliminiert und so  $\Theta_2$  berechnet werden:

$$\frac{-(R_4 + R_7)}{R_6 + R_9} = \frac{\sin \Theta_2 (\cos \Theta_1 - \sin \Theta_1)}{\cos \Theta_2 (\cos \Theta_1 - \sin \Theta_1)} = \frac{\sin \Theta_2}{\cos \Theta_2} = \tan \Theta_2.$$

Damit erhält man eine Lösung für  $\Theta_2$ :

$$\Theta_2 = \arctan \frac{-(R_4 + R_7)}{R_6 + R_9}. \quad (3.1.18)$$

Mit dem Wissen über  $\Theta_2$  kann nun auf den Winkel  $\Theta_0$  geschlossen werden:

$$\begin{aligned}
 &R_1 + R_3 \frac{R_4 + R_7}{R_6 + R_9} \\
 &= \cos \Theta_0 \cos \Theta_2 - \frac{\sqrt{2}}{2} \sin \Theta_2 (\sin \Theta_0 \cos \Theta_1 + \sin \Theta_0 \sin \Theta_1) \\
 &+ \left[ \cos \Theta_0 \sin \Theta_2 + \frac{\sqrt{2}}{2} \cos \Theta_2 (\sin \Theta_0 \cos \Theta_1 + \sin \Theta_0 \sin \Theta_1) \right] \frac{\sin \Theta_2}{\cos \Theta_2} \\
 &= \cos \Theta_0 \cos \Theta_2 + \cos \Theta_0 \frac{\sin^2 \Theta_2}{\cos \Theta_2}.
 \end{aligned}$$

Dies kann umgeformt werden in:

$$\begin{aligned}
 \cos \Theta_0 &= \frac{R_1 + R_3 \frac{\sin \Theta_2}{\cos \Theta_2}}{\cos \Theta_2 + \frac{\sin^2 \Theta_2}{\cos \Theta_2}} \\
 &= \frac{\cos \Theta_2 \cdot R_1 + R_3 \cdot \sin \Theta_2}{\cos^2 \Theta_2 + \sin^2 \Theta_2} \\
 &= \cos \Theta_2 \cdot R_1 + \sin \Theta_2 \cdot R_3.
 \end{aligned}$$

$\Theta_0$  ist demnach bestimmt durch:

$$\Theta_0 = \arccos(\cos \Theta_2 \cdot R_1 + \sin \Theta_2 \cdot R_3). \quad (3.1.19)$$

Nachdem nun  $\Theta_0$  und  $\Theta_2$  bekannt sind, kann  $\Theta_1$  berechnet werden.

$$\begin{aligned} R_5 - R_8 &= \cos \Theta_1 \left( \frac{\cos \Theta_0}{2} + \frac{1}{2} \right) + \sin \Theta_1 \left( \frac{-\cos \Theta_0}{2} + \frac{1}{2} \right) \\ &+ \cos \Theta_1 \left( \frac{\cos \Theta_0}{2} - \frac{1}{2} \right) + \sin \Theta_1 \left( \frac{-\cos \Theta_0}{2} - \frac{1}{2} \right), \\ &= \cos \Theta_1 \cos \Theta_0 - \sin \Theta_1 \cos \Theta_0. \end{aligned}$$

Umformen liefert die Gleichung:

$$\cos \Theta_1 = \frac{R_5 - R_8}{\cos \Theta_0} + \sin \Theta_1. \quad (3.1.20)$$

Durch weitere Kombination erhält man:

$$\begin{aligned} &R_6 \frac{-(R_4 + R_7)}{R_6 + R_9} + R_7 \\ &= \left[ \cos \Theta_2 \left( \cos \Theta_1 \left( -\frac{\cos \Theta_0}{2} + \frac{1}{2} \right) + \sin \Theta_1 \left( -\frac{\cos \Theta_0}{2} - \frac{1}{2} \right) \right) + \frac{\sqrt{2}}{2} \sin \Theta_0 \sin \Theta_2 \right] \\ &\cdot \frac{\sin \Theta_2}{\cos \Theta_2} \\ &+ \left( -\sin \Theta_2 \left( \cos \Theta_1 \left( \frac{\cos \Theta_0}{2} + \frac{1}{2} \right) + \sin \Theta_1 \left( \frac{\cos \Theta_0}{2} - \frac{1}{2} \right) \right) - \frac{\sqrt{2}}{2} \sin \Theta_0 \cos \Theta_2 \right), \\ &= \sin \Theta_2 \left( \cos \Theta_1 \left( -\frac{\cos \Theta_0}{2} + \frac{1}{2} \right) + \sin \Theta_1 \left( -\frac{\cos \Theta_0}{2} - \frac{1}{2} \right) \right) + \frac{\sqrt{2}}{2} \sin \Theta_0 \frac{\sin^2 \Theta_2}{\cos \Theta_2} \\ &+ \sin \Theta_2 \left( \cos \Theta_1 \left( \frac{-\cos \Theta_0}{2} - \frac{1}{2} \right) + \sin \Theta_1 \left( -\frac{\cos \Theta_0}{2} + \frac{1}{2} \right) \right) - \frac{\sqrt{2}}{2} \sin \Theta_0 \cos \Theta_2, \\ &= \sin \Theta_2 \left( -\cos \Theta_1 \cos \Theta_0 - \sin \Theta_1 \cos \Theta_0 \right) + \frac{\sqrt{2}}{2} \sin \Theta_0 \left( \frac{\sin^2 \Theta_2}{\cos \Theta_2} - \cos \Theta_2 \right). \end{aligned}$$

Dies kann umgeformt werden in:

$$\begin{aligned} \cos \Theta_1 \cos \Theta_0 + \sin \Theta_1 \cos \Theta_0 &= \frac{R_6 \frac{\sin \Theta_2}{\cos \Theta_2} + R_7 - \frac{\sqrt{2}}{2} \sin \Theta_0 \left( \frac{\sin^2 \Theta_2}{\cos \Theta_2} + \cos \Theta_2 \right)}{-\sin \Theta_2}, \\ &= -\frac{R_6}{\cos \Theta_2} - \frac{R_7}{\sin \Theta_2} + \frac{\sqrt{2}}{2} \sin \Theta_0 \left( \frac{\sin \Theta_2}{\cos \Theta_2} - \frac{\cos \Theta_2}{\sin \Theta_2} \right). \end{aligned}$$

Aufgelöst nach  $\sin \Theta_1$  ergibt sich:

$$\sin \Theta_1 = \frac{-\frac{R_6}{\cos \Theta_2} - \frac{R_7}{\sin \Theta_2} + \frac{\sqrt{2}}{2} \sin \Theta_0 \left( \frac{\sin \Theta_2}{\cos \Theta_2} - \frac{\cos \Theta_2}{\sin \Theta_2} \right)}{\cos \Theta_0} - \cos \Theta_1. \quad (3.1.21)$$

Einsetzen von Gleichung 3.1.21 in 3.1.20 ergibt dann die gesuchte Lösung für  $\Theta_1$  die nur noch von bekannten Faktoren abhängt:

$$\Theta_1 = \arccos \frac{R_5 - R_8 - \frac{R_6}{\cos \Theta_2} - \frac{R_7}{\sin \Theta_2} + \frac{\sqrt{2}}{2} \sin \Theta_0 \left( \frac{\sin \Theta_2}{\cos \Theta_2} - \frac{\cos \Theta_2}{\sin \Theta_2} \right)}{2 \cdot \cos \Theta_0} \quad (3.1.22)$$

Somit ist eine Lösung für jedes  $\Theta_0, \dots, \Theta_5$  gefunden und das Problem der inversen Kinematik gelöst. Analog erfolgt die Lösung für das linke Bein unter den Annahmen (3.1.15) bis (3.1.17).

Es sei angemerkt, dass Team *B-Human* im Jahre 2009 einen Lösungsansatz veröffentlichte [89], der zwar zum gleichen Ergebnis führt, den Weg dorthin aber durch implizite Betrachtung der Auswirkung der Schrägstellung von *HipYawPitch* vereinfacht. Dies ist aber nur bei dieser konkreten Gelenkanordnung in der Hüfte möglich. Der hier vorgestellte Lösungsweg wurde bereits für die erste von Aldebaran veröffentlichte Gelenkkonfiguration des NAO durchgeführt, in der fehlerhafter Weise die Anordnung von  $\Theta_1$  und  $\Theta_2$  in der Reihenfolge vertauscht war. Die in [89] genutzte Vereinfachung wäre in dem Fall nicht anwendbar und so musste eine allgemeinere Lösung gefunden werden. Diese wurde nach der Auslieferung des ersten NAOs in die hier vorgestellte Form korrigiert. Deshalb wurde sich entschieden, an dieser Stelle zum Vergleich zu [89] die allgemeinere Lösungsstrategie vorzustellen, auch wenn es nicht die eleganteste Lösung des konkreten Problems darstellt.

Nicht aufgrund der Gelenkinematik, aber durch die Form der Plastikverschalung an den Füßen sind bestimmte Gelenkkombinationen von *AnkleRoll* und *AnklePitch* für den NAO nicht erreichbar. Diese Einschränkung des Arbeitsraums ist bei der Ansteuerung zu beachten und im *User Guide* des NAO [1] dokumentiert. Der Vollständigkeit halber ist der Arbeitsraum der rechten Fußgelenke zum Verständnis in Abbildung 3.1.4<sup>8</sup> zu finden.

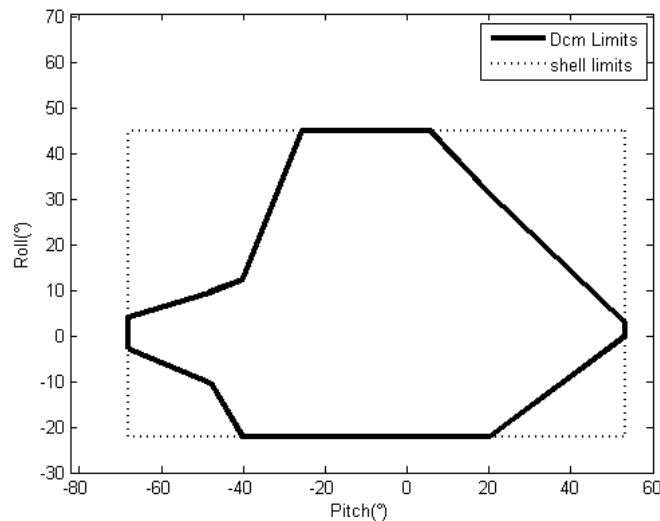


Abbildung 3.1.4: Arbeitsraum der Gelenke des rechten Fußes entnommen aus [1].

Die bisherigen Betrachtungen bezogen sich nur auf ein Bein des NAO. Durch das gemeinsame *HipYawPitch*-Gelenk sind aber beide Beine miteinander gekoppelt und somit  $\Theta_0^{links}$  nicht unabhängig von  $\Theta_0^{rechts}$ . Dies führt dazu, dass bei der getrennten Lösung der beiden Ketten in der Regel ein Fehler entstehen wird. In [89] wird dieses Problem durch die Einführung eines virtuellen *AnkleYaw*-Gelenkes umgangen und damit der Fehler auf beide Beine verteilt. Dies wird darüber begründet, dass dieser Winkel in

<sup>8</sup>Copyright ©2012: Aldebaran-Robotics.

der Praxis den geringsten Einfluss auf die Stabilität besitzt. Allerdings vernachlässigt dies die Tatsache, dass der Fehler des Standbeines in die Berechnung der Lage des Roboters durch die Gelenkwinkel eingeht, vergleiche Kapitel 4.1.1. Da sich im Laufe der Arbeit zeigen wird, dass das Wissen über die Pose des Oberkörpers für die Bewegungssteuerung von entscheidender Bedeutung ist, ist der Einfluss dieses Fehlers auf das System um einiges größer, als der Schwungbeinfehler<sup>9</sup>. Aus diesem Grund wird für die NDD-Kinematik ein anderes Vorgehen gewählt. Zunächst wird über die vorgestellte Kinematik für das Standbein die Rücktransformation gelöst. Anschließend berechnet man die Kinematik für das Schwungbein unter der Bedingung, dass  $\Theta_0^{Schwung}$  vorgegeben ist. Dies hat im Allgemeinen einen Fehler in der Schwungfußposition zur Folge. Dessen Einfluss kann jedoch durch geeignete Wahl von  $\Theta_1$  und  $\Theta_2$  minimiert werden. Analog zum Vorgehen des Standbeines kann nach Gleichung (3.1.7) der Winkel  $\Theta_3$  aus der Kinematik der Beine bestimmt werden. Damit lassen sich die Überlegungen für die Winkel  $\Theta_4$  und  $\Theta_5$  wiederholen und folglich sich diese ebenfalls nach den Gleichungen (3.1.10) und (3.1.11) zu berechnen. Die gewünschte Position des Fußgelenkes im Raum ist durch die Transformation

$$\mathbf{T}_{0,6}\mathbf{T}_{5,6}^{-1} = \mathbf{T}_{0,5}. \quad (3.1.23)$$

bekannt. Da  $\Theta_0$  vorgegeben ist, kann die Position des Fußgelenkes  $\mathbf{T}_{0,5}$  für den Fall bestimmt werden, dass  $\Theta_1$  und  $\Theta_2$  den Wert Null haben. Ausgedrückt in den Koordinatensystemen der *HipRoll*, bzw. *HipPitch*-Gelenke können nun jeweils die Rotationen bestimmt werden, die beide Punkte in der Ebene zur Deckung bringen. Diese Winkel sind die gesuchten Winkel  $\Theta_1$  und  $\Theta_2$ . Der Fehler führt zu einer Rotation des Fußes um die, in das Fußkoordinatensystem transformierte, *HipYawPitch*-Achse und somit zu einer Schräglage des Fußes, da diese nicht vollständig durch  $\Theta_1$  und  $\Theta_2$  ausgeglichen werden kann. Der Fehler wird aber geringer, je näher sich beide Beine am Boden befinden und beträgt aufgrund der Symmetrie theoretisch Null, sobald der NAO den Schwungfuß auf den Boden aufsetzt. Da aber nur zu diesem Zeitpunkt die exakte Orientierung des Schwungfußes zum Boden wichtig ist, hat dieser Fehler in der Praxis lediglich einen Einfluss, wenn der Fuß frühzeitig den Boden berührt.

## 3.2 Motoren

Eng mit der kinematischen Struktur des Roboters verbunden und für die Fortbewegung von entscheidender Bedeutung sind die Aktuatoren. Der NAO kann jeweils die Stellung eines Gelenkes durch einen Aktuator beeinflussen und damit einen Freiheitsgrad verändern. Dies geschieht mittels eines elektrischen Motors, der die Position des Gelenkes über ein Getriebe ansteuert. Ausnahme bildet das Hüftgelenk, das über ein Schneckengetriebe die Stellung des rechten und linken Hüftgelenks zugleich beeinflusst.

Über den Hersteller und den genauen Typ des Motors macht Aldebaran keine Angaben, allerdings finden sich in der NAO Dokumentation technische Angaben, die in Tabelle 3.2.4 zu finden sind. Diese beiden Motoren werden bei der RoboCup-Version des

---

<sup>9</sup>Es sei angemerkt, dass B-Human zur Berechnung der Lage lediglich die Inertialsensoreinheit verwendet. Daher ist in ihrem Fall der Einfluss nicht so signifikant.

NAO mit zwei unterschiedlichen Getrieben kombiniert<sup>10</sup>. Abhängig vom Motor führt dies durch das auf der Achse fest montierte Zahnrad zu einer unterschiedlichen Übersetzung. Folglich resultieren verschiedene nominale Drehmomente und eine Geschwindigkeiten, wie in Tabelle 3.2.5 nachzulesen ist. Die Geschwindigkeit wurde hierzu auf Grad pro Taktzyklus des Motors normiert, um ein besseres Verständnis zu ermöglichen. Die Verwendung unterschiedlicher Getriebe für die korrespondierenden Kopf-, Schulter- und Armgelenke ist vermutlich bauartbedingt. Jedenfalls bewegen sich diese Gelenke meist ohne direkten Kontakt mit einem externen Körper und müssen daher nur geringe Kraft aufbringen. Dies spiegelt sich demnach im Einsatz des leistungsschwächeren Motors vom Typ 2 wieder und in einem geringeren nominalen Drehmoment dieser Gelenke. Bei den Gelenken der Beine zeigt sich, dass die unterschiedlichen Getriebe gezielt verwendet wurden, um verschiedene maximale Drehmomente auf den Gelenken zu erzeugen. *HipRoll* und *AnkleRoll* sind die Gelenke, die beim Laufen für die Seitwärtsschwingung des Roboters sorgen. Folglich müssen sie entsprechend große Kräfte zur Beschleunigung und Verzögerung aufbringen, um den Roboter in einem stabilen Zustand zu halten, wenn dieser auf nur einem Bein balanciert. Spätere Experimente werden sogar noch aufzeigen, dass die Kraft des *HipRoll*-Gelenkes nicht ausreicht, um den Roboter ohne dynamische Einflüsse aus eigener Kraft kippen zu lassen, vergleiche Kapitel 4.4.4. Der *HipYawPitch*-Motor muss über ein Getriebe zwei Gelenke in einer Scherbewegung gegeneinander bewegen, also erscheint auch hier die Wahl des erhöhten Drehmoments sinnvoll. Verwunderlich ist hingegen die Entscheidung, im *KneePitch*-Gelenk das Getriebe mit dem geringeren Übersetzungsverhältnis einzusetzen. Dies ist ein Gelenk, das beim Laufen sehr starken Belastungen ausgesetzt ist, da in der Phase mit nur einem Standbein das gesamte Gewicht von diesem Gelenk gehalten wird. In der Praxis äußert sich dies, indem dieses Gelenk bei einem Gang mit niedrigem Schwerpunkt zuerst überhitzt. Details hierzu finden sich in den Untersuchungen zur Energieeffizienz in Kapitel 5.4.3. Hier wäre eine bessere Übersetzung hilfreich, um diesen Motor zu entlasten.

Motor	Type 1	Type 2
Leerlauf	8000 rpm	11900 rpm
Kippmoment	59,5 nMN	15,1 nMN
Nenngeschwindigkeit	6330 rpm	8810 rpm
Nennmoment	12,3 mNm	3,84 mNm

Tabelle 3.2.4: Leistungsdaten der Motoren des Roboters NAO.

Zur Ansteuerung der Position verwendet Aldebaran einen Proportional-Differential-Integral-Regler, kurz PID, der direkt in die Steuerungshardware des jeweiligen Motors integriert ist. Da dieser sich in einer für den Benutzer nicht zugänglichen Schicht befindet, können die Parameter nicht beeinflusst werden und müssen im Falle des NAO als Charakteristik des Motors angesehen werden. Laut Aldebaran erfolgt die Regelung der Motoren drehmomentgesteuert, die Kommandos hingegen positionsgesteuert. Der Benutzer kann also eine Position und einen Zeitpunkt vorgegeben, an dem dieser erreicht werden soll. Aus diesen Angaben interpoliert Aldebarans Middleware *NAOqi*, siehe Kapitel 3.6, Zwischenpositionen und sendet diese der Motorsteuerung. Werden hierdurch

<sup>10</sup>In der Academic-Version kommen noch zwei weitere Getriebe für die Gelenke *WristsYaw* und *Hand* zum Einsatz.

Gelenk	Motor	Getriebe	Drehmoment	Geschwindigkeit
HeadYaw	Typ 2	150:1	0,57 Nm	3,51°/10 ms
HeadPitch	Typ 2	173:1	0,66 Nm	3,05°/10 ms
ShoulderPitch	Typ 2	150:1	0,57 Nm	3,51°/10 ms
ShoulderRoll	Typ 2	173:1	0,66 Nm	3,05°/10 ms
ElbowYaw	Typ 2	150:1	0,57 Nm	3,51°/10 ms
ElbowRoll	Typ 2	173:1	0,66 Nm	3,05°/10 ms
HipYawPitch	Typ 1	201:1	2,47 Nm	1,89°/10 ms
HipRoll	Typ 1	201:1	2,47 Nm	1,89°/10 ms
HipPitch	Typ 1	130:1	1,61 Nm	2,90°/10 ms
KneePitch	Typ 1	130:1	1,61 Nm	2,90°/10 ms
AnklePitch	Typ 1	130:1	1,61 Nm	2,90°/10 ms
AnkleRoll	Typ 1	201:1	2,47 Nm	1,89°/10 ms

Tabelle 3.2.5: Gelenkzuordnung der Motoren und Getriebe.

theoretische Grenzen in der Position oder Geschwindigkeit überschritten, wird bereits vorher in Software das Steuerkommando begrenzt. Diese Grenzen sind vom Hersteller vorgegeben und können nicht variiert werden. Veränderbar ist aber das maximale Drehmoment der Motoren, das über den Parameter *Stiffness* vom Benutzer festgelegt werden kann. Dieser kann im Bereich von 0 bis 1 mit einer Schrittweite von 0,1 gesetzt werden. Der genaue Einfluss dieses Parameters ist undokumentiert und ein mathematischer Zusammenhang nicht bekannt. Eine experimentelle Untersuchung ist nicht durchgeführt, um keinen der Roboter durch den Ausbau eines Motors zu beschädigen. Somit ist lediglich klar, dass eine *Stiffness* von 1 zum maximal möglichen Drehmoment führt und eine *Stiffness* von 0 den entsprechenden Motor abschaltet. Da die Motoren in den Beinen aber auch bei einer sehr geringen *Stiffness* von 0,1 noch in der Lage sind, die Masse des Roboters zu bewegen, kann angenommen werden, dass der Zusammenhang nichtlinear ist. Aufgrund fehlender Modellierbarkeit wurde daher davon abgesehen, diesen Parameter in die theoretischen Überlegungen folgender Kapitel mit einzubeziehen und er wurde nur als manueller Korrekturparameter in den entsprechenden Konfigurationsdateien eingeführt, wie Kapitel 5.3 entnommen werden kann. Es ist zu erwähnen, dass der *Stiffness* Wert von Aldebaran bei steigender Temperatur der Motoren dynamisch gegen Null geregelt wird, um einer Überhitzung der Motoren vorzubeugen. Dies führt dazu, dass das Bewegungsverhalten des Roboters unvorhersehbar wird, sobald einer der Motoren die Temperatur von 70° Celsius überschreitet.



### 3.3 Sensoren

Sensoren bilden für den Roboter die Schnittstelle, mit der er Informationen über seine Umgebung sammeln kann. Folglich sind die die entscheidende Grundlage zur Modellierung seiner Umwelt. Sensoren die Informationen über die Umgebung liefern, werden in der Literatur als *exterozeptive* Sensoren bezeichnet. Für den Roboter ist es aber ebenso entscheidend Informationen über seinen internen Zustand zu sammeln. Diese Sensoren werden in der Klasse der *propriozeptiven* Sensoren zusammengefasst. Sie ermöglichen dem Roboter die Selbstwahrnehmung und bilden somit die Grundlage zur Modellbildung seines internen robotereigenen Zustandes, im Folgenden auch als *Ego-Modell* bezeichnet. Daher ist klar, dass diese Klasse von Sensoren das grundlegende Feedback zur Bewegungsteuerung bildet und somit den Schwerpunkt des folgenden Kapitels darstellen wird. Eine detailliertere Erörterung der Ego-Modellierung ist im weiteren Verlauf in Kapitel 4 zu finden, nachdem darauf eingegangen wurde, wie die propriozeptiven Informationen konkret zur Selbstwahrnehmung eingesetzt werden können.

Der Roboter NAO ist mit einer Vielzahl unterschiedlicher Sensoren beider Klassen ausgestattet. Da es sich bei dem Roboter um eine geschlossene Plattform handelt, können diese nicht ausgetauscht werden. Die verbauten Sensoren stellen einen repräsentativen Querschnitt möglicher Sensorklassen dar, die gebräuchlicher Weise in humanoiden Robotern verwendet werden. Eine Ausnahme bilden lediglich deutlich teurere und größere Roboter, wie beispielsweise ASIMO, der weitere Kraftsensoren in den Gelenken besitzt. In den folgenden Kapiteln werden daher nicht nur konkret die im NAO verwendeten Sensoren vorgestellt, sondern es wird auch jeweils auf den generellen Sensortyp eingegangen. Die Auswahl konkreter Sensoren folgt verschiedenen Kriterien, weswegen beim NAO nicht nur die Präzision und Leistungsfähigkeit der Sensoren, sondern auch die Größe und der Preis entscheidend sind. Als Folge ist diese Roboterplattform nicht mit den präzisesten oder robustesten Sensoren ausgestattet, die erhältlich sind. Da der Fokus der Arbeit bewusst auf finanziell erschwingliche Hardware gelegt ist, bilden diese Sensoren dennoch eine gute Grundlage für eine Evaluation der Leistungsfähigkeit dieser Klasse von humanoiden Robotern. Aus diesem Grund werden die für diese Arbeit relevanten Sensoren in den entsprechenden Kapitel auch einer eingehenderen experimentellen Untersuchung unterzogen. Da der NAO aber nur durch professionelle Unterstützung geöffnet werden kann, ohne ihn zu zerstören, können diese Experimente nicht mit dem Sensor selbst, sondern nur mit dem Roboter durchgeführt werden. Letztendlich können aber nur so diese Ergebnisse als realistische Grundlage späterer Simulation des Roboters dienen. Folglich ist dieser Effekt durchaus erwünscht. Da der Roboter aber nicht demontiert werden wird, kann sich nur auf Angaben des Herstellers bezüglich des konkreten Sensormodells verlassen werden. Eine Verifikation bleibt aus. Darüber hinaus ist durch Aldebaran kein direkter Zugriff auf die Sensordaten zugelassen, sondern diese sind nur über die DCM-Einheit, siehe Kapitel 3.5, ansprechbar. Daher können teilweise nur aufbereitete Messwerte und nicht die Rohdaten selber ausgewertet werden. Dennoch wird versucht, die Ergebnisse soweit möglich gegen die Datenblätter der Hersteller abzugleichen.

#### 3.3.1 Inertialsensoreinheit

Aldebaran stattet den NAO mit einem *Inertialsensor*, englisch auch als *Inertial Measurement Unit* bezeichnet, aus, der im Torso des Roboters verbaut ist. Diese Art von Sensoren dient der Messung von translatorischen und rotatorischen Bewegungen und kann unterschiedliche Typen von Sensoren umfassen, die in der Regel der Bauklasse der *Micro-Electro-Mechanical Systems*, auch als MEMS bezeichnet, angehören. Die von Aldebaran entwickelte Inertialsensoreinheit besteht aus einem MEMS-Bewegungssensor vom Typ *LIS302DL* der Firma *STMicroelectronics* mit 3 Achsen und einem 2-Achsen-MEMS-Gyrometer vom Typ *IDG-300* der Firma *Invensense*. Diese werden durch einen Mikrocontroller kombiniert, um die Messwerte durch eine Filterung vorzuverarbeiten. Der Beschleunigungssensor, Englisch *accelerometer*, misst die Beschleunigungen des Oberkörpers entlang der  $x$ -,  $y$ - und  $z$ -Achse. Ein Gyrometer, englisch *gyroscope*, überwacht die Rotation um die entsprechenden Achsen, indem er deren Winkelgeschwindigkeit mißt. Für einen tieferen Einstieg in die Funktionsweise und verwendeten Technologien moderner Inertialsensoren sei [6] oder [66] empfohlen. Da das verbaute Gyrometer des NAO nur die Rotationsgeschwindigkeit um die  $x$ - und  $y$ -Achse, nicht aber um die Hochachse  $z$ , misst, ist offensichtlich, dass von den 6 Freiheitsgraden eines Körpers von diesem Inertialsensor lediglich 5 direkt gemessen werden können.

Über das Gyrometer IDG-300 ist aus dem Datenblatt des Herstellers [104] bekannt, dass es bis zu einer Geschwindigkeit von rund  $500^\circ/\text{s}$  und Beschleunigungen von bis zu  $5000\text{ g}$  verlässlich arbeitet. Der Hersteller gibt die initiale Kalibrierungstoleranz mit  $\pm 10$  Prozent an, in der Dokumentation des NAO verbessert Aldebaran diese Angabe auf  $\pm 5$  Prozent. Es finden sich keine Angaben über die Temperaturabhängigkeit des Sensors. Aus dem Datenblatt des Beschleunigungssensors [104] ist zu entnehmen, dass dieser in zwei unterschiedlichen Modi betrieben werden kann, die sich in ihrem Arbeitsbereich - maximal  $\pm 2\text{ g}$  im Modus A, bzw.  $\pm 8\text{ g}$  im Modus B - unterscheiden. Aldebaran verwendet im NAO den erstgenannten Modus A. Laut Datenblatt erreicht der Sensor somit eine Auflösung von  $16,2 - 19,8\text{ mg}$ . Angaben zur Abweichung der Messung durch Rauschen finden sich im Datenblatt des Herstellers nicht, doch Aldebaran gibt als Messgenauigkeit eine Toleranz von 1 Prozent an. Durch die geringe Temperaturabhängigkeit des Sensors - laut Datenblatt  $0,01\%$  pro  $^\circ\text{C}$  - kann dieser Fehler in guter Näherung als konstant angenommen werden.

Die Inertialsensoreinheit wird im weiteren Verlauf der Arbeit eine wichtige Grundlage für die Messung des Gleichgewichts des Roboters, siehe Kapitel 4.4.2, und der Bewegungsschätzung, siehe Kapitel 6.3, bilden. Um eine bessere Vorstellung des Einflusses von falschen Messwerten auf diese folgende Berechnung der Gleichgewichtskriterien zu bekommen, ist ein Verständnis der möglichen Ursachen solcher Abweichungen nötig. Daher werden diese im Folgenden erörtert. Auf deren Einfluss auf die Gleichgewichtsmessung kann allerdings nicht eingegangen werden, ohne die zugrunde liegende Theorie vorher zu erläutern. Daher werden die Auswirkungen der Fehler in Kapitel 4.4.2 nochmals eingehender diskutiert. Die Signale am Ausgang der Inertialsensoreinheit unterliegen Störungen unterschiedlicher Herkunft, die allgemein als Sensorrauschen bezeichnet werden. Die möglichen Quellen für diese Störung sind mannigfaltig, doch lassen sich diese in den folgenden Phänomenen vereinen. Eine detaillierte mathematische Betrachtung

deren Charakteristik wurde von Woodman als *Technical Report* [117] zusammengefasst. Da die Sensorsimulation dieser Arbeit auf der Theorie von Woodman beruht, werden im Folgenden dessen Ideen zumindest grundlegend aufgegriffen und skizziert.

**Konstanter Bias** Die gemittelte Ausgabe eines Sensors in Ruhe wird als *konstanter Bias* bezeichnet. Der konstante Bias kann durch eine Langzeitmessung abgeschätzt werden. Ist das Ergebnis dieser ungleich Null spricht man auch von einem Biasfehler  $e_b$ , der sich demnach zu jeder Messung addiert. Es handelt sich um einen *systematischen* Fehler, der sich nicht bei wiederholter Messung im Mittel ausgleicht. Es ist darauf hinzuweisen, dass es nicht möglich ist, den Beschleunigungssensor in Ruhe zu messen, da er ständig der Gravitation ausgesetzt ist. Somit muss darauf geachtet werden die Erdbeschleunigung  $g$  von der Messung zu subtrahieren und den Bias als durchschnittliche Differenz zwischen der gemessenen und der korrekten Beschleunigung zu erhalten.

Der Einfluss des Biasfehler ist konstant auf die direkte Messung des entsprechenden Sensors:

$$\Delta a = e_b^a, \quad (3.3.1)$$

$$\Delta \omega = e_b^\omega. \quad (3.3.2)$$

Dies führt durch die Integration über die Zeit zu einem linearen Winkelfehler  $\Delta \Theta_b$  durch das Gyrometer

$$\Delta \Theta_b(t) = \int_0^t e_b^\omega d\tau = e_b^\omega t. \quad (3.3.3)$$

und im Falle des Beschleunigungssensors zu einem quadratischen Fehler  $\Delta p_b$  in der Position

$$\Delta p_b(t) = \int_0^t \int_0^t e_b^a d\tau = e_b^a \frac{t^2}{2}. \quad (3.3.4)$$

Der Biasfehler kann theoretisch durch geeignete Kalibrierung beseitigt werden. Es ist allerdings davon auszugehen, dass dies in der Praxis nicht gänzlich gelingen wird, da die Inertialsensoreinheit nur innerhalb des Roboters NAO kalibriert werden kann. Die Folge ist ein Restbiasfehler, der jedoch klein gegenüber anderen Fehlereinflüssen ist.

**Thermomechanisches Weißes Rauschen** Die Genauigkeit eines Inertialsensors wird durch die Überlagerung der Messung mit einem thermomechanischen Rauschen beeinflusst, welches sich durch eine Sequenz weißen Rauschens beschreiben lässt. Hierbei handelt es sich um einen stochastischen Fehler, welcher nach Woodman [117] durch unkorrelierte, normalverteilte Zufallsvariablen  $\mathcal{X}_i$  mit einem Erwartungswert von  $E[\mathcal{X}_i] = 0$ , einer Varianz  $\text{Var}[\mathcal{X}_i] = \sigma^2$  und einer Kovarianz  $\text{Cov}[\mathcal{X}_i, \mathcal{X}_j] = 0$  modelliert werden kann. Der Erwartungswert von  $E[\mathcal{X}_i] = 0$  zeigt, dass im Mittel der Fehler keinen Einfluss auf die direkte Messung hat, dennoch pflanzt sich der Fehler natürlich bei der Bestimmung der Orientierung und der Position mit Hilfe dieser Sensoren entsprechend fort.

Wird im folgenden für ein Gyrometer angenommen, dass die Messwerte zwischen den Abtastzeitpunkten  $T_s$  konstant sind, dann ergibt sich dadurch nach der Zeitspanne  $t = nT_s$  ein Fehler in der Orientierung von

$$\Delta\Theta_w = T_s \sum_{i=1}^n \mathcal{X}_i. \quad (3.3.5)$$

Damit gilt für den Erwartungswert  $E[\Delta\Theta_w]$  und die Varianz  $\text{Var}[\Delta\Theta_w]$

$$E[\Delta\Theta_w] = T_s n E[\mathcal{X}] = 0, \quad (3.3.6)$$

$$\text{Var}[\Delta\Theta_w] = T_s^2 n \text{Var}[\mathcal{X}] = T_s t \sigma^2. \quad (3.3.7)$$

Somit ist das Resultat des weißen Rauschens eine Zufallsbewegung in der Orientierung, deren Standardabweichung

$$\sigma_{\Delta\Theta}(t) = \sigma \sqrt{T_s t}, \quad (3.3.8)$$

proportional zur Quadratwurzel der Zeit beschrieben wird. Woodman klassifiziert dies als *Angular Random Walk*, im Folgenden abgekürzt durch ARW, der in der Einheit  $^\circ/\sqrt{s}$  beschrieben ist durch:

$$\text{ARW} = \sigma_{\Delta\Theta}(1) = \sigma \sqrt{T_s}. \quad (3.3.9)$$

Entsprechend der Modellierung des Gyrometers durch den ARW verwendet Woodman für das weiße Rauschen des Beschleunigungssensors den *Velocity Random Walk*, der im Folgenden mit VRW bezeichnet und in der Einheit  $m/s/\sqrt{h}$  angegeben wird. Für die Abweichung in der Positionsbestimmung durch den Beschleunigungssensor ergibt sich damit analog zum Gyrometer:

$$\Delta p_w(t) = \int_0^t \int_0^\tau e_w(\tau) d\tau \quad (3.3.10)$$

$$= T_s \sum_{i=1}^n T_s \sum_{j=1}^i \mathcal{X}_j \quad (3.3.11)$$

$$= T_s^2 \sum_{i=1}^n (n - i + 1) \mathcal{X}_i. \quad (3.3.12)$$

Somit ist der erwartete Positionsfehler

$$E[\Delta p_w(t)] = T_s^2 \sum_{i=1}^n (n - i + 1) E[\mathcal{X}_i] = 0, \quad (3.3.13)$$

und die Varianz

$$\text{Var}[\Delta p_w(t)] = T_s^4 \sum_{i=1}^n (n - i + 1)^2 \text{Var}[\mathcal{X}_i] \quad (3.3.14)$$

$$= T_s^4 \frac{n(n+1)(2n+1)}{6} \text{Var}[\mathcal{X}] \quad (3.3.15)$$

$$= \frac{T_s^4 (2n^3 + 3n^2 + n)}{6} \sigma_w^2. \quad (3.3.16)$$

Unter der Annahme, die Abtastzeiten  $T_s$  seien kurz, lässt sich  $\text{Var} [\Delta p_w(t)]$  für  $t \rightarrow \infty$  zu

$$\text{Var} [\Delta p_w(t)] \approx \frac{T_s^4 n^3}{3} \sigma_w^2 \quad (3.3.17)$$

$$= \frac{T_s t^3 \sigma_w^2}{3} \quad (3.3.18)$$

vereinfachen. Dabei wurde bei der Vereinfachung der Zusammenhang  $t = nT_s$  ausgenutzt. Als Ergebnis erhält man somit eine Zufallsbewegung zweiter Ordnung mit einer Standardabweichung von

$$\sigma(t) = \sigma_w t^{3/2} \sqrt{\frac{T_s}{3}}, \quad (3.3.19)$$

welche proportional zu  $t^{3/2}$  wächst.

Da es sich beim weißen Rauschen nicht um einen systematischen Fehler handelt, kann dieser nicht durch eine geeignete Kalibrierung eliminiert werden. Wie erwähnt ist es nicht möglich auf die ungefilterten Werte der Sensoren zurückzugreifen, da Aldebaran nur die bereits durch die Inertialsensoreinheit vorverarbeiteten Werte dem Benutzer zur Verfügung stellt.

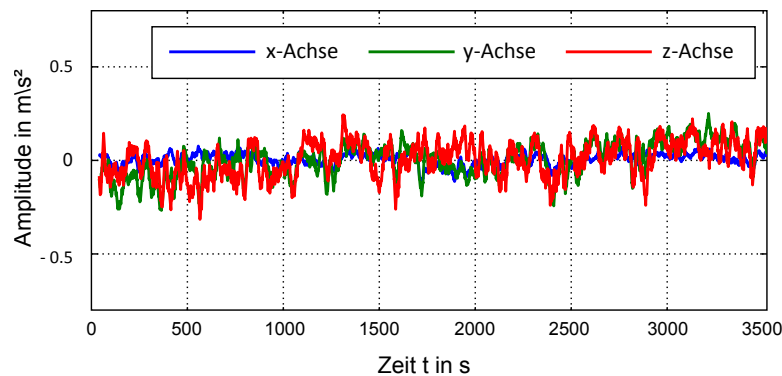


Abbildung 3.3.5: Rauschverhalten des V2 Beschleunigungssensors über  $t \approx 1$  h.

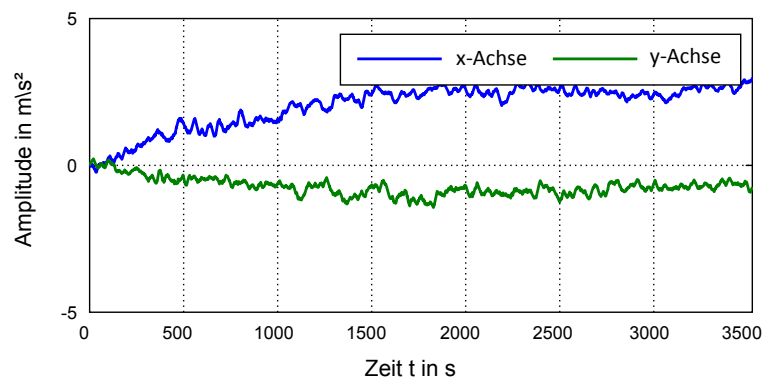


Abbildung 3.3.6: Rauschverhalten des V2 Gyrometers über  $t \approx 1$  h.

Diagramm 3.3.5 zeigt das Rauschverhalten des Beschleunigungssensors eines NAO V2 in  $x$ -,  $y$ - und  $z$ -Achse über einen Zeitraum von rund einer Stunde. Zur Normierung

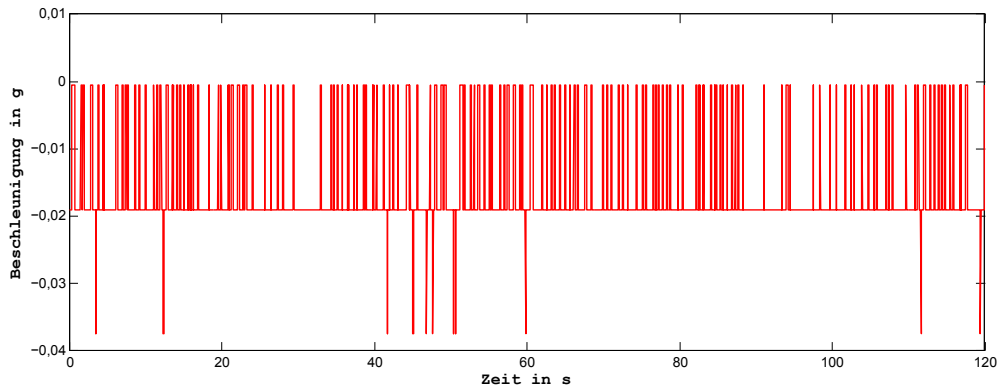


Abbildung 3.3.7: Rauschverhalten des V3.3 Beschleunigungssensors über  $t = 120$  s.

auf Null wurde die Erdbeschleunigung - angenähert<sup>11</sup> durch  $g = 9,81 \text{ m/s}^2$  - herausgerechnet. Auch wenn es sich hierbei nicht im Rohdaten des Sensors handelt - Aldebaran hat im DCM bereits eine Filterung vorschaltete, die nicht umgangen werden kann - so bietet diese Messung des NAO V2 doch die beste Option, um Rückschlüsse auf die Charakteristik des Sensors zu ziehen. Aldebaran hat die Filterung von V2 zu V3.3 derart verstärkt, dass keine Rückschlüsse auf die Eigenschaften des Sensors mehr gezogen werden können, wie der direkte Vergleich des Rauschverhaltens zwischen Abbildung 3.3.5 und 3.3.7 verdeutlicht. Allen drei Beschleunigungskanälen des NAO V2 werden in ihrem Messfehler in Ruhelage durch ein statistisches weißes Rauschen dominiert. Ein signifikanter Biasfehler über die Zeit scheint nicht aufzutreten und die Messung zeigt auch keine Anzeichen eines Einflusses der Betriebstemperatur des *Chestboards*. Zum Vergleich zeigt Abbildung 3.3.7 einen kürzeren Ausschnitt der  $z$ -Beschleunigung der gleichen Messung mit einem NAO V3.3. Es fällt auf, dass sich der Rauschanteil zur V2 signifikant verringert hat. Der Wert bleibt über den Zeitraum der Messung von zwei Minuten praktisch konstant, so dass die Vermutung naheliegt, dass in der V3.3 einen fortschrittlicheren Filteralgorithmus in der Inertialsensoreinheit zum Einsatz kommt. Bei dem resultierenden Rauschverhalten handelt es sich zwar immer noch um ein weißes Rauschen, allerdings scheint es nur noch drei Zustände anzunehmen und kommt somit vom Verhalten beinahe einem *Salt-And-Pepper*- oder *Impulsrauschen* gleich. Bei genauer Betrachtung fällt auf, dass der Messwert jeweils im Bereich von  $\pm 18,5 \text{ mg}$  schwankt, was im Vergleich mit dem Datenblatt genau der Auflösung des Sensors von  $16,2 - 19,8 \text{ mg}$  entspricht. Es handelt sich also um ein Quantisierungsrauschen. Aldebaran selber macht keine Aussagen darüber welcher Algorithmus für die Vorfilterung des V3.3 angewandt wird. Doch aufgrund der Messwerte liegt es nahe, dass der Messung ein starkes Tiefpassfilter, möglicherweise in Kombination mit einem Prädiktionsfilter, wie beispielsweise einem Kalman-Filter, zugrunde liegt. Daher ist anzunehmen, dass die Messung durch die unveränderte Position des NAO begünstigt wurde und in Bewegung ein stärkerer Rauscheinfluss zu erwarten ist.

<sup>11</sup>Der Standardwert der Erdbeschleunigung ist international auf  $g = 9,80665 \text{ m/s}^2$  festgelegt, weicht aber lokal durch die Verformung der Erdoberfläche geringfügig ab.

Zum Vergleich finden sich die Messwerte des Gyrometers eines NAOs V2 über eine Stunde in Ruhe in Abbildung 3.3.6. Man kann erkennen, dass das weiße Rauschen auch in dieser Messung einen sichtbaren Einfluss hat. Jedoch wird dieser Fehler im Vergleich zum Beschleunigungssensor bereits nach rund einer Minute von einer Biasinstabilität dominiert. Diese steigt zunächst für beide Achsen, um dann einen Sättigungswert zu erreichen. Es liegt also die Vermutung nahe, dass es sich hierbei um eine Temperaturinstabilität handelt, da auch die Temperatur im Inneren des NAO nach einer gewissen Zeit eine Sättigung erreicht. Da es aber nicht möglich ist, das Gyrometer unabhängig vom Betrieb des NAO zu testen, kann diese Vermutung keiner weiteren experimentellen Verifikation unterzogen werden. Da auch der Hersteller keine Angabe zur Temperaturabhängigkeit seines Sensor macht, kann auch hieraus kein weiterer Rückschluss gezogen werden.

**Biasinstabilität** In der Regel ist der Bias im Ausgangssignal eines Sensor nicht konstant, sondern variiert über die Zeit. Der Grund dafür ist das sogenannte Funkelrauschen, auch als  $1/f$ -Rauschen bezeichnet. Dieses kann unterschiedliche technische Ursachen haben, auf die an dieser Stelle nicht weiter eingegangen werden soll, doch tritt es bei nahezu allen elektronischen Bauteilen auf und somit auch bei den Inertialsensoren des NAO. Anzumerken ist, dass das Rauschen sich bei geringerer Frequenz bemerkbar macht und bei hohen Frequenzen durch das weiße Rauschen komplett überlagert wird. Die Biasinstabilität BS, auch als *Drift* bezeichnet, beschreibt die Veränderungsrate des Bias über die Zeit. Woodman [117] modelliert die Biasinstabilität durch den *Bias Random Walk*, im folgenden mit BRW abgekürzt, da sie näherungsweise als Zufallsbewegung beschrieben werden kann. Der Zusammenhang der beiden Größen kann nach Woodman über

$$\text{BRW} = \frac{\text{BS}}{\sqrt{T}}, \quad (3.3.20)$$

hergestellt werden, wobei  $T$  die Zeitspanne beschreibt, für die BS definiert ist. Durch die Zufallsbewegung des Bias steigt der erwartete Biasfehler  $e_b$  über die Zeit an. Analog zur Betrachtung der thermomechanischen weißen Rauschens spiegelt sich dies für das Gyrometer in einer Zufallsbewegung zweiten Grades des resultierenden Orientierungsfehlers  $\Delta\Theta_{bs}$  wider. Die Unsicherheit der Geschwindigkeit aus der Beschleunigungssensormessung resultiert zu einer Zufallsbewegung zweiter Ordnung proportional zu  $t^{3/2}$ . Und letztlich kann die Unsicherheit in der Position durch eine Zufallsbewegung dritter Ordnung proportional zu  $t^{5/2}$  abgebildet werden, vergleiche [117]. Hierbei ist zu beachten, dass dieses Modell nur für kurze Zeiten Gültigkeit besitzt. Der Einfluss in der Praxis ist zudem beschränkt, da der Biasfehler durch einen Neustart oder Reset der Hardware wieder auf den Ursprungszustand zurückgesetzt wird. Somit ist im Allgemeinen der Einfluss der Biasinstabilität als klein gegenüber anderen Faktoren anzusehen.

**Temperaturinstabilität** Viele Sensoren verändern ihr Messverhalten in Abhängigkeit von der Umgebungstemperatur und folglich variieren auch die Messwerte entsprechend. Dies entspricht im Grunde einer speziellen Biasinstabilität, doch da diese nur indirekt von der Zeit, primär aber von Umwelteinflüssen abhängt, sei diese nochmals gesondert hervorgehoben. Die resultierenden Schwankungen sind in der Regel nichtlinearer Natur

und schwer zu modellieren. Es handelt sich zwar um einen in Näherung deterministischen und folglich systematischen Fehler, allerdings ist es in der Praxis schwer, dieses Verhalten durch Kalibrierung auszugleichen. Da der NAO keine Sensoren hat, um die Temperatur zu messen, wird dies Vorhaben zusätzlich erschwert. Theoretisch wäre es dennoch möglich eine Abschätzung der Temperatur zur Korrektur über die Laufzeit vorzunehmen. Da diese aber stark von der Beanspruchung der den Sensor umgebenden anderen elektronischen Bauteile abhängt, ist auch diese Variante nicht praktikabel.

**Kalibrierungsfehler** In dieser Klasse werden unterschiedliche Fehler zusammengefasst, die aus fehlerhaften Annahmen über den Sensor entstehen. Dies sind zum einen Fehler, die direkt mit dem Sensor verknüpft sind, wie beispielsweise inkorrekte Skalierungsfaktoren zur Umrechnung des digitalen Messwertes in seine physikalische Repräsentation. Solche Fehler können nur vom Hersteller minimiert werden, da hierzu Referenzmessungen durchgeführt werden müssen, die nach dem Einbau nur noch mit großem Aufwand durchgeführt werden können. Zum anderen sind dies Kalibrierungsfehler durch einen inkorrekten Einbau des Sensors, der entweder werksseitig bereits vorhanden ist oder durch leichte Verschiebung des Sensors im Betrieb geschehen kann. Dies resultiert zwar nicht in einem direkten Fehler des Messwertes, aber einer fehlerhaften Umrechnung in das körpereigene-Koordinatensystem. Diese Fehler können entweder aus einer Drehung oder Verschiebung des Sensors relativ zu der angenommenen Position bestehen. Da keine Möglichkeit besteht die genaue Position der Sensoren innerhalb des Roboters zu bestimmen, sind letztere auch nur mit großem Aufwand zu messen und können daher vom NAO nicht ausgeglichen werden. Da die Erfahrung gezeigt hat, dass die Dekalibrierung des NAO - vermutlich bauartbedingt - verstärkt im rotatorischen Einbau geschieht, wurde hierzu regelmäßig ein halbautomatisches Kalibrierungsverfahren durchgeführt. Der Roboter wird dabei der Reihe nach jeweils in eine Ruhelage gebracht, in der die Erdbeschleunigung genau entlang seiner der  $x$ -, bzw.  $y$ - oder  $z$ -Achse verläuft. Anhand dieser Messung können nun ausgleichende Korrekturfaktoren für die Beschleunigungen berechnet werden, um die Messwerte in Deckung mit den Achsen des körpereigenen Koordinatensystems zu bringen. Die direkte Bestimmung der rotatorischen Korrekturfaktoren für das Gyrometer sind in der Praxis experimentell nicht zu erreichen, da hierzu der Roboter in eine kontrollierte rotatorische Bewegung um seine Achsen gebracht werden müssten. Da aber das Gyrometer Teil der Inertialsensoreinheit ist und daher starr mit dem Beschleunigungssensor verbunden ist, kann dessen Kalibrierung in guter Näherung übertragen werden.

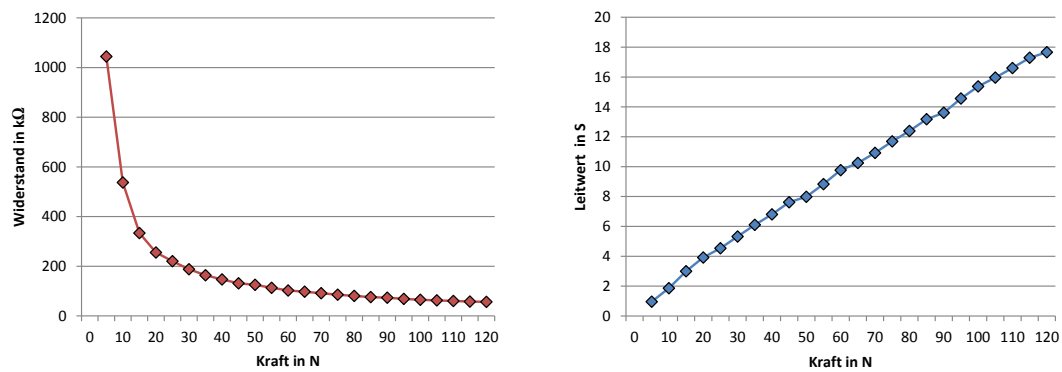
#### 3.3.2 FSR-Sensoren

Drucksensoren sind elektrische Widerstände, die ihren Widerstandswert unter Einwirkung von äußeren Kräften verändern. Gebräuchlich in der Robotik sind hierfür *FSR*-Sensoren, die in die Klasse der *taktilen* Sensoren fallen. Diese Abkürzung wird offiziell für den *Force-sensing Resistor Sensor* verwendet, der in den 80er Jahren von der Firma *Interlink Electronics* vorgestellt wurde und bis heute ein eingetragenes Markenzeichen ist. Dennoch hat die verwendete Technik in der Industrie viele Nachahmer gefunden und somit gibt es eine Reihe Sensoren, die nach demselben Prinzip arbeiten, zur Abgren-



zung aber oftmals als *Force-Sensitive Resistor* bezeichnet werden. Im Sprachgebrauch hat es sich durchgesetzt, die gesamte Klasse als FSR-Sensoren zu bezeichnen und daher wird auch hier so verfahren. FSR-Sensoren bestehen aus einem Polymere, das erst durch Krafteinwirkung leitfähig wird und diese Eigenschaft je nach Stärke der Kraft verändert. Dieses Polymere wird in Form einer dünnen Schicht oder einer Flüssigkeit zwischen zwei dünnen leitende Kontaktfolien angebracht und trennt diese elektrisch durch seinen initialen Widerstand von rund  $1\text{ M}\Omega$  nahezu vollständig. Bei Druck auf die Kontaktfolien verringert sich dieser, um irgendwann einen Sättigungswert anzunehmen, der vom verwendeten Polymere abhängt. Somit ermöglicht der Sensoraufbau auf verhältnismäßig kleiner Fläche die senkrechte Kraft auf diese zu messen. Es handelt sich also im Grunde um einen 1-Achs-Kraftsensor. Da FSR-Sensoren bauartbedingt eine Dicke von unter  $0,5\text{ mm}$  erreichen können, eignen sich diese um unter der Sohle eines humanoiden Roboter angebracht zu werden und somit die Gewichtsverteilung über die Fußfläche messen zu können.

Aldebaran verwendet für den NAO die FSR-Sensoren *FlexiForce*®A201 der Firma Tekscan [108]. Diese haben eine Dicke von  $0,208\text{ mm}$  und können durch den geringen Durchmesser der Messfläche von  $9,53\text{ mm}$  Kräfte in guter Näherung punktgenau messen. In jedem der Füße sind je 4 FSR-Sensoren integriert, wodurch insgesamt 8 dieser Sensoren am NAO zu finden sind. Die Sensoren sind nicht direkt an der Auflagefläche der Füße angebracht, sondern im Inneren. Hierzu hat Aldebaran ein Design erdacht, dass aus einer doppelten Fußplatte besteht. Die eigentliche Sohle ist nur an vier Punkten mit dem Fuß verbunden. An diesen sind die FSR-Sensoren angebracht. Die Reaktionszeit der Sensoren gibt Tekscan mit  $< 5\ \mu\text{s}$  an. Für die Messung können drei Auflösungen gewählt werden, in denen die maximal messbare Kraft  $4,4\text{ N}$ ,  $110\text{ N}$  respektive  $440\text{ N}$  beträgt. Aldebaran betreibt die FSR-Sensoren im mittleren Arbeitsbereich, da bereits die Gewichtskraft des NAO die  $4,4\text{ N}$  Grenze überschreitet. Kräfte jenseits der  $110\text{ N}$  sind hingegen nur durch starke äußere Krafteinwirkung zu erreichen. Somit sichert diese Wahl die maximal mögliche sinnvolle Auflösung der Sensoren. In Ruhe beträgt der Widerstand des Sensors  $> 5\text{ M}\Omega$  und fällt nichtlinear mit steigendem Druck ab. Um aus dem Widerstand des Sensors auf die einwirkende Kraft zu schließen, muss die Charakteristik des Sensors untersucht werden und eine Funktion zur Kalibrierung gefunden werden. Dies geschieht in der Regel durch experimentelle Aufnahme von Stützpunkten und anschließender Interpolation einer mathematischen Funktion mittels Ausgleichsrechnung. Durch den DCM sperrt Aldebaran ein direktes Auslesen der Daten dieser Sensoren. Daher können hier zur Veranschaulichung nur exemplarische Messwerte des Herstellers aus dem Datenblatt herangezogen werden. Abbildung 3.3.8(a) zeigt deutlich den nichtlinearen Zusammenhang zwischen Kraft und Widerstand. Wird aber der Widerstand in den elektrischen Leitwert umgerechnet, es gilt  $G = 1/R$ , so ist bereits mit bloßem Auge ein linearer Zusammenhang zwischen F und G zu erkennen. Tekscan gibt den Fehler dieser Linearisierung mit maximal  $\pm 3\%$  an. Der mathematische Zusammenhang lässt sich also durch Berechnung einer Ausgleichsgerade bestimmen, die die  $y$ -Achse bei dem Wert  $1/R_{max}$  schneiden muss. Dies ergibt laut Datenblatt einen elektrischen Leitwert von  $G < 0,2 \cdot 10^{-6}\text{ S}$ . Somit kann in guter Näherung davon ausgegangen werden, dass die Gerade durch den Ursprung verläuft. Zur Kalibrierung des Sensors reicht es also aus die Steigung der Geraden experimentell zu verifizieren.



(a) Widerstand.

(b) Leitwert.

Abbildung 3.3.8: FlexiForce A201 Kalibrierung.

In Kapitel 4.4.1 wird sich zeigen, dass mittels Drucksensoren auch im dynamischen Fall ein Kriterium für das Gleichgewicht eines Roboters berechnet werden kann. Somit stellen auch die FSR-Sensoren eine wichtige Grundlage der folgenden Kapitel und Experimente dar. Aus diesem Grund soll, wie bei der Inertialsensoreinheit, auf mögliche Ursachen für Messfehler eingegangen werden. Da die Fehlerquellen bei FSR-Sensoren grundlegend ähnlich denen der MEMS-Sensoren sind, wird nicht mehr detailliert auf die einzelnen Begriffe eingegangen. Eine ausführliche Betrachtung erfolgte bereits in Kapitel 3.3.1. Des Weiteren kann nur unzureichend auf den Einfluss der Fehler auf die Gleichgewichtsmessung eingegangen werden, ohne die zugrunde liegende Theorie vorher zu erläutern. Daher werden die Auswirkungen der Fehler in Kapitel 4.4.1 nochmals eingehender diskutiert. Es sei aber bereits vorweggenommen, dass für die Gleichgewichtsmessung ein Verhältnis aus den FSR-Sensorwerten gebildet werden muss. Folglich sinkt der Einfluss von Fehlern signifikant, wenn sie alle Sensoren gleichermaßen beeinflussen.

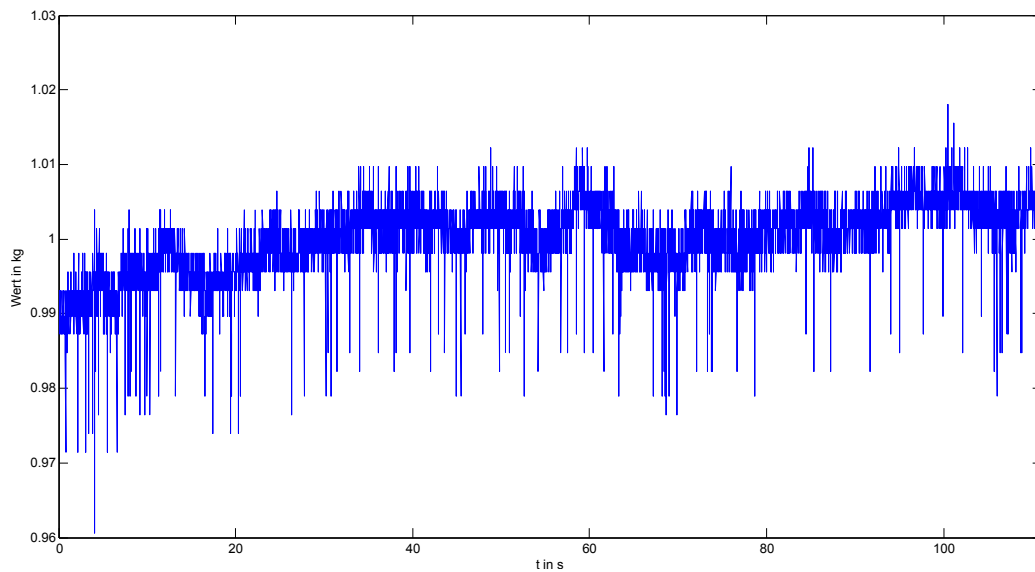


Abbildung 3.3.9: Rauschverhalten des FlexiForce A201A FSR-Sensors über  $t = 110$  s.

**Konstanter Bias** Auch bei den FSR-Sensoren tritt in der Regel ein konstanter Bias auf. Dieser ist aber gering gegenüber sonstigen Fehlerquellen. Da für die Gleichgewichtsmessung nur das Verhältnis der Sensoren zueinander benötigt wird, verringert sich der Einfluss dieses Störfaktors nochmals. Dies gilt im Besonderen, wenn dieser bei allen verwendeten Sensoren gleichermaßen auftritt.

**Biasinstabilität** Auch eine zeitliche Biasinstabilität kann bei FSR-Sensoren auftreten. Zur Evaluierung wird die Fußsohle eines NAOs über einen Zeitraum von 110 Sekunden mit einem Gewicht von 1 kg belastet. Diese hat eine Pyramidenform, so dass durch die sehr geringe Auflagefläche das Gewicht punktgenau auf die Position eines A201 FSR-Sensors wirken kann. Die Abbildung der aufgenommenen Messwerte in Grafik 3.3.9 zeigt, dass sich trotz des starken Rauschens ein Drift des Mittelwertes erkennen lässt, der vor allem in den ersten 30 Sekunden merklich ansteigt. Da der Roboter beim Laufen aber periodisch die Füße vom Boden abhebt und somit die FSR-Sensoren entlastet, kommt es nur selten vor, dass über einen längeren Zeitraum kontinuierlich gemessen wird. Somit ist der Einfluss dieses Fehlers auf die Praxis als gering einzuschätzen.

**Temperaturinstabilität** Tekscan gibt die Temperaturinstabilität mit  $\pm 0.2\%$  pro Grad F an. Diese kann demnach als sehr gering angenommen werden. Da im Fuß des Roboters, mit Ausnahme der LEDs und eines Kontaktschalters, keine weiteren elektronischen Komponenten verbaut sind, kann zudem davon ausgegangen werden, dass sich die Temperatur während des Betriebes kaum verändert. Der Einfluss auf die Praxis ist daher als gering zu klassifizieren. Selbst eine existente Temperaturinstabilität wirkt sich, analog zur Biasinstabilität, nur verringert auf die Messung aus, da alle Sensoren in gleichem Maße von ihr betroffen sind.

**Thermomechanisches weißes Rauschen** Wie bei allen elektronischen Messinstrumenten wird auch das Signal der Drucksensoren in der Praxis von einem statistischen Rauschen überlagert - deutlich erkennbar in Abbildung 3.3.9. Wie das charakteristische Band um den Mittelwert zeigt, scheint auch dieses Rauschen gut durch ein weißes Rauschen modelliert werden zu können. Die Schwankungen liegen im Bereich von rund 10 g und fallen damit mit ungefähr 1 Prozent recht gering aus. Allerdings weist das Ergebnis dieses Experiments nicht die typische Charakteristik eines reinen weißen Rauschens auf, da immer wieder deutliche Abweichungen nach unten auffallen, die bis zu 40 g, also 4 Prozent ausmachen können. Da diese nur sporadisch auftreten lassen sie sich durch eine Überlagerung des weißen mit einem *Salt-and-Pepper*-Rauschen am sinnvollsten modellieren.

Das Langzeit-Rauschverhalten der FSR-Sensoren für die Fortbewegung eines humanoiden Roboters nicht von Bedeutung, wie bereits bei der Diskussion der Biasinstabilität erörtert wurde. Die Sensoren werden in der Regel nicht ununterbrochen über einen Zeitraum von mehreren Sekunden belastet. Interessanter ist daher der Rauscheinfluss auf wiederholte kurze Messungen. Daher wird das gleiche Experiment nochmals wiederholt, der Sensor aber immer nur für einen kurzen Zeitraum von rund 10 Sekunden belastet.

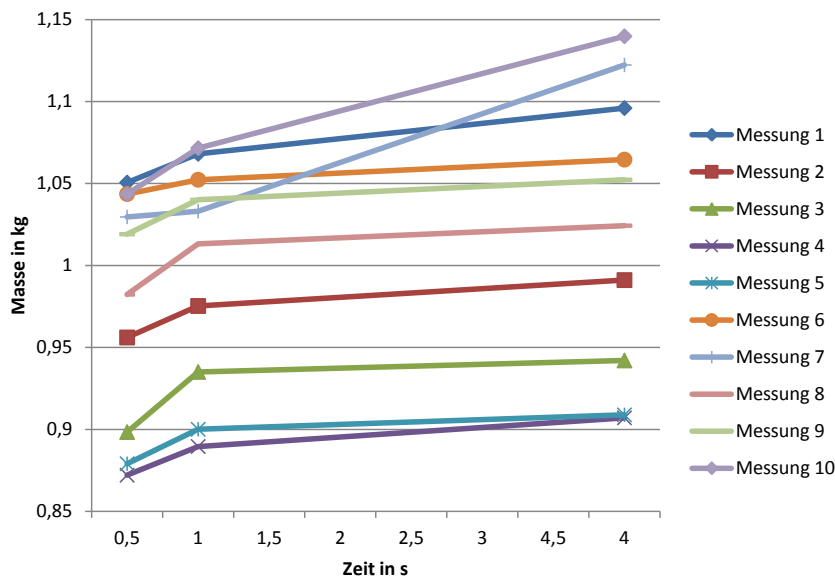


Abbildung 3.3.10: Wiederholte Belastung des FlexiForce A201A FSR-Sensors mit 1 kg.

Abbildung 3.3.10 zeigt für 10 Wiederholungen des Experiments jeweils einen Messwert des Sensors nach 0,5, 1 und 4 Sekunden. Zur Veranschaulichung des Trends wurden die Messpunkte jeweils linear verbunden. Es zeigt sich, dass die initialen Messwerte über einen Bereich von rund  $\pm 125$  Gramm verteilt sind. Allerdings kann diesem Ergebnis kein hoher Stellenwert beigemessen werden, da das Experiment manuell durchgeführt werden musste. Somit ist die verhältnismäßig große Streuung leicht damit zu erklären, dass das Gewicht nicht immer exakt an dergleichen Stelle ausgesetzt werden konnte. Dennoch kann diese Abschätzung des Rauschen durchaus als *Worse Case* für eine spätere Simulation angenommen werden, vergleiche Kapitel 4.4.1. Folglich soll die sich ergebende Standardabweichung  $\sigma$  von 75 g über alle Messungen für die Auswertung dennoch festgehalten werden. Interessant ist aber die Tatsache, dass in jeder der Wiederholungen der Messwert über die ersten Sekunden stetig steigt. Obwohl der Sensor eine sehr geringe Reaktionszeit hat, scheint es doch eine gewisse Zeit zu benötigen, bis der Messwert eine Sättigung erreicht hat. Dies muss bei der Kalibrierung berücksichtigt werden. Da für die Gleichgewichtsberechnung bei rhythmischen Fortbewegungen nur von kurzen Belastungszyklen der Sensoren ausgegangen werden kann, wird zur Kalibrierung die Messung zum Zeitpunkt  $t = 1$  s nach Initialisierung gewählt.

**Mechanische Defekte** Die FSR-Sensoren wurden von Aldebaran nicht direkt unter der Fußsohle angebracht, um diese vor mechanischer Abnutzung zu schützen. Statt dessen bestehen die Füße aus zwei beweglich gegeneinander gelagerten Teilen - dem eigentlichen Fuß und der Sohle. Zwischen diesen beiden sind die Sensoren so montiert, dass eine Kraft auf die Fußsohle sich möglichst direkt auf den entsprechenden Sensor auswirkt. Durch dieses physikalische Übertragungssystem können weitere Fehler bei der Messung auftreten. Dies kann zum einen die *Dämpfung* des Messwertes sein, da das physikalische System durch Reibung Teile der Kraft in Wärme umwandelt. Da die

Kalibrierung des Sensors aber durch äußere Krafteinwirkung auf die Sohle des Roboters geschieht, wird dieser Fehler implizit durch die Kalibrierung behoben. Anders verhält es sich, wenn sich die Aufhängung der Sohle verklemmt und es so zu einem, teilweise temporären, Totalausfall des Sensors kommen kann, da dieser entweder ständig mit einer hohen Kraft belastet wird, oder keine Kraft von außen mehr auf ihn wirkt. Zuletzt sei noch der Fehler des *Übersprechens* eines Sensors auf andere Sensoren erwähnt. Dies bezeichnet den Fall, dass durch einen mechanischen Fehler im System Kraftausübung auf einen Sensor ebenfalls zu einer Kraft auf einen oder mehrere der benachbarten Sensor führt.

**Kalibrierungsfehler** Tekscan gibt für den A201 einen Kalibrierungsfehler von 3 Prozent an, der durch die Linearisierung der Messwerte entsteht. Dieser könnte verringert werden, indem der Zusammenhang zwischen dem Leitwert und der Kraft durch mehrere Stützstellen ausgemessen und mittels einer komplexeren Funktion approximiert wird. Eine bessere Genauigkeit lässt sich hierdurch aber nur erzielen, wenn die Kalibrierung automatisch erfolgen würde, da die manuelle Wiederholbarkeit der Messung als unzureichend angesehen werden kann. In der Praxis entstehen zudem die gravierendsten Fehler durch die mechanische Aufhängung des NAO, so dass diese Art der Kalibrierung keinen großen Einfluss auf die Praxis hat.

### 3.3.3 Gelenkwinkelencoder

Zur Bestimmung der Gelenkpositionen verwendet der NAO magnetische Rotationsencoder der Firma *austriamicrosystems*<sup>12</sup>. Diese Sensoren vom Typ AS5055 sind direkt an den Gelenkachsen angebracht und messen die Bewegung des jeweiligen Gelenkes. Als Folge können Fehler des Getriebes nicht gemessen werden, aber die Sensoren umgehen diese implizit, da laut Aldebaran der Gelenkwinkel und nicht die Motorposition die Grundlage für die Regelung der Motoren bildet.

Um die Position der Gelenke auch während der Bewegung bestimmen zu können, muss die Latenz des Sensors niedrig genug sein. Eine direkte Aussage über die Latenz des Sensors trifft der Hersteller nicht. Abhängig vom Energiemodus indem der Sensor betrieben wird, gibt er aber ein *readout* Zeitintervall an, das Tabelle 3.3.6 entnommen werden kann. Da dieses den minimalen Zeitraum angibt, in dem Daten ausgelesen werden können ist klar, dass die Latenz des Sensors kleiner sein muss, da sonst kein Sensor-Update in dem entsprechenden Zeitraum erfolgen könnte. Die Motoren des NAO V3.3 werden mit einer Frequenz von 100 Hz angesteuert, aus der eine Zykluszeit von 10 ms resultiert. Folglich genügt es den Sensor im  $< 500 \mu\text{A}$  Modus zu betreiben, um entsprechend aktualisierte Werte zu liefern. Ein Modus in dem der Sensor mehr Strom verbraucht, könnte aber die Aktualität der Positionsbestimmung vermutlich verbessern. In welchem Modus der Sensor letztendlich von Aldebaran betrieben wird, kann nicht gesagt werden, da sich der Hersteller diese Information über seine Middleware versteckt und keine Informationen darüber veröffentlicht hat. Es bleibt aber festzustellen, dass

<sup>12</sup><http://www.ams.com>

der Sensor in der Lage ist, im gleichen Zeitintervall Informationen über die Position zu liefern, wie der Motor neue Positionsanforderungen entgegen nehmen kann.

Power Mode	Intervall
< 8 mA	620 $\mu$ s
< 5 mA	1 ms
< 500 $\mu$ A	10 ms
< 53 $\mu$ A	100 ms

Tabelle 3.3.6: Messintervalle des AS5055 [4].

Da die Stellung der Gelenke die aktuelle Pose des Roboters bestimmt, hat die Genauigkeit und der Fehler der Gelenkwinkelencoder an vielen Stellen Einfluss auf die in dieser Arbeit vorgestellten Algorithmen. Daher soll nun eine genauere Betrachtung dieser Einflussgrößen erfolgen. Der Sensor hat eine Auflösung von 12 Bit, ist also in der Lage 4096 Position über einen Winkelbereich von  $360^\circ$  zu bestimmen und kann somit theoretisch Winkeländerungen von  $0,09^\circ$  messen. Jedoch gibt der Hersteller für einen perfekt auf der Drehachse platzierten Sensor einen Fehlerbereich von  $0 - 0,5^\circ$  an, wie Abbildung 3.3.11<sup>13</sup> zu entnehmen ist, so dass die theoretische Auflösung des Sensors im weißen Rauschen der Messung untergeht. Es ist daher zielführender die Genauigkeit mit  $\pm 0,5^\circ$  nach oben abzuschätzen. Diese Annahme setzt aber eine perfekte Positionierung voraus, die bei einer manuellen Fertigung - wie beim NAO - im Allgemeinen nicht erreicht werden kann. Daher wird sich zu dem statistischen Messfehler noch ein systematischer Fehlereinfluss addieren. Die experimentelle Untersuchung des Herstellers in Abbildung 3.3.11 zeigt einen nichtlinearen Fehlereinfluss der Deplazierung des Sensors von der Achse. Die Positionierung des Sensors kann am NAO nicht überprüft werden, da diese zusätzlich zur Verkleidung durch eine Abschirmung verdeckt ist. Dennoch ist davon auszugehen, dass bei der Fertigung eine Positionierungshilfe in Form von Einkerbungen auf der Platine verwendet wird. Trotzdem kann eine Deplazierung im Radius von 0,5 mm selbst bei sorgfältiger händischer Fertigung nicht ausgeschlossen werden, so dass der stochastische Fehler nochmals durch einen systematischen um  $\pm 0,5^\circ$  erhöht wird. Zu dem translatorischen Fehler addiert sich außerdem ein potentieller rotatorischer Fehler durch die Sensorplatzierung. Unter Verwendung einer Arretierung sollte er sich ebenfalls in mit kleinen Bereich der Größe  $\pm 0,5^\circ$  abschätzen lassen. Somit folgt im *Worst Case* ein systematischer Fehler von  $\pm 1^\circ$ , der nochmal von dem stochastischen zu einer oberen Grenze von  $\pm 1,5^\circ$  überlagert wird.

Da weitere technische Details des Sensor für die weitere Arbeit nicht von Belang sind, wird an dieser Stelle auf eine Beschreibung des Aufbaus und der technischen Realisierung des Sensors verzichtet. Für eine eingehendere Betrachtung und weitere Spezifikationen sei daher auf das Datenblatt des Herstellers verwiesen [4].

---

<sup>13</sup>Copyright ©1997-2012, ams AG.

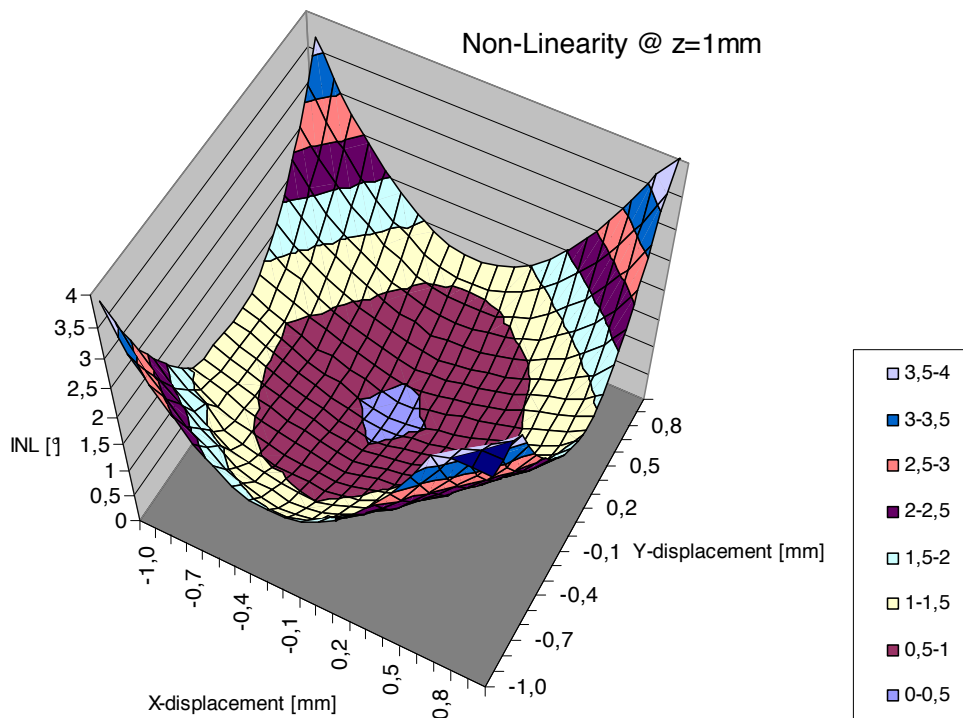


Abbildung 3.3.11: Rotationsfehler des AS5055 abhängig von der Positionierung des Sensors entnommen aus [4]. Die Legende gibt den Fehler in Grad an.

### 3.3.4 Motorstrom

Die Motoren des Roboters NAO haben die Fähigkeit den aufgenommenen Strom jedes Gelenks zu messen. Genaue Angaben über die Art der Strommessung ist in den Datenblättern des Roboters nicht zu finden. Da die Sensoren integrierter Bestandteil der Motoren sind, deren Hersteller unbekannt ist, kann auch darüber keine weitere Information eingeholt werden. Des Weiteren gibt die NAO Dokumentation keinen Einblick in die Interpretation des über die API abfragbaren Sensorwertes. Da aber in der Praxis nie Werte über 2000 zurückgelesen wurden, und der Motor laut Aldebaran Datenblatt eine maximale Stromaufnahme von 2 A besitzt, liegt die Vermutung nahe, dass dieser Wert in mA skaliert ist. Allerdings ist zu beobachten, dass der Wert durch den Motorparameter *Stiffness* beeinflusst wird. Da dessen mathematische Interpretation unbekannt ist, kann auch für *Stiffness*-Werte ungleich 1 keine qualifizierte Aussage über die Größe des Stroms in Relation zur Messung gemacht werden.

Theoretisch steht der Strom, den ein Motor aufnimmt im Verhältnis zum Drehmoment  $M$ , das dieser aufbringt. Dies lässt sich für jeden Motor  $i$  beschreiben durch:

$$M_i = k_i \cdot I_i, \quad (3.3.21)$$

wobei  $k_i$  der Motorkonstanten des Motors entspricht. Somit könnte über die Soll- und die Ist-Ansteuerung des Motors, in Kombination mit dem Stromverbrauch des Motors ein Rückschluss auf die auf ihn wirkenden Kräfte gezogen werden, wenn die Motorkonstante bekannt ist. In Kombination mit einem kompletten Dynamikmodell des Roboters

- das bislang nicht existiert und auch im Rahmen dieser Arbeit nicht erstellt wird - könnten somit Rückschlüsse auf externe Kräfte gezogen werden, die auf den Roboter wirken. Diese Diskussion ist aber zunächst ohne jegliche Praxisrelevanz, da die Motorkonstante der Motoren des Roboters NAO nicht bekannt ist und sich mit einem vollständigen Roboter auch nur schwerlich experimentell bestimmen lässt, ohne diesen zu zerstören. Der mögliche Nutzen soll aber dennoch nicht unerwähnt bleiben, da Aldebaran zum Zeitpunkt dieser Erstellung dieser Ausarbeitung das *Mechatronic Ankle Kit* angekündigt hat. Dieses für den Bereich Bildung gedachte Produkt besteht aus der vollständigen Struktur eines NAO Fußgelenks, also *AnkleRoll* und *AnklePitch*. Da diese laut Datenblatt vollständig ohne Verkleidung auskommt besteht somit die Hoffnung, in Zukunft mehr über die Motoren im Roboter NAO zu lernen und es damit zu ermöglichen diese Diskussion als Grundlage für weiterführende Arbeiten zu nutzen.

Zur Evaluation der Güte zeigt Abbildung 3.3.12 den Ausschnitt einer Messung des Stromsensors im Knie des NAO während eines Laufs in  $x$ -Richtung über mehrere Schritte. Die Charakteristik des Verlaufs der Stromaufnahme des Motors ist grundsätzlich soweit abgebildet, dass einzelne Schrittmuster optisch unterschieden werden können. Jedoch fällt auf, dass das Rausch-Signal-Verhältnis des Sensors signifikant ist und somit eine Anwendung in der Praxis ohne Filterung erschwert.

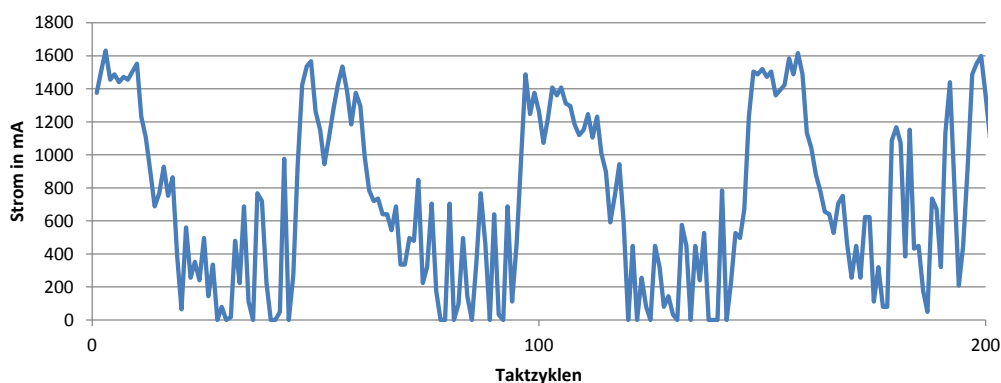


Abbildung 3.3.12: Messung des Stromsensors des *KneePitch*-Gelenks mit *Stiffness*=1 während eines Laufs.

#### 3.3.5 Exterozeptive Sensorik

Zusätzlich zu den im vorherigen Kapitel vorgestellten propriozeptiven Sensoren verfügt der Roboter NAO über weitere Sensoren, die der Klasse der exterozeptiven Sensorik zuzuordnen sind. Diese bieten ebenfalls einen repräsentativen Querschnitt über die Klasse der Sensoren, die in humanoiden Robotern zum Einsatz kommen. Doch da diese für die weiteren Kapitel dieser Arbeit keine Rolle spielen, werden sie an dieser Stelle nur der Vollständigkeit halber angeführt. Wie in der humanoiden Robotik gebräuchlich ist auch der Roboter NAO mit einer **Kamera** ausgestattet. Genauer gesagt sind es zwei, jedoch kann der Roboter zu jedem Zeitpunkt nur exklusiv auf eine Kamera zugreifen. Dieses Setup ermöglicht daher dem Roboter kein stereoskopisches Sehen, sondern soll



dem Roboter die Wahl eines für die Aufgabe geeigneten Sichtfelds ermöglichen. Die Kamera ist vom Modell Milliard OV7670 und hat eine Auflösung von  $640 \times 480$  Bildpunkten bei einer Bildwiederholrate von 30 fps. Ein weiterer Sensor dieser Klasse sind die **Ultra-Schall-Sensor** unbekanntem Herstellers, die im Torso des Roboters verbaut sind. Deren Reichweite beträgt laut Aldebaran 25 bis 255 cm. Die letzten exterozeptiven Sensoren im Roboter NAO sind **Mikrophone**, von denen 4 im Kopf verbaut sind. Da diese sowohl nach links und rechts, als auch nach vorne und hinten ausgerichtet sind, würden sich diese zur Ortung von Geräuschquellen eignen. Da aber während der Spiele im RoboCup die Geräuschkulisse eine sinnvolle Auswertung verhindert, wird auf deren Einsatz verzichtet. Der Vollständigkeit halber sei erwähnt, dass die FSR-Sensoren nicht eindeutig klassifizierbar sind, da diese durch ihre taktile Messung auch Rückschlüsse über die Umgebung ermöglichen. Da diese im Kontext der Arbeit aber lediglich der Bestimmung des internen Gleichgewichtszustands dienen, werden diese der Klasse der propriozeptiven Sensoren zugeordnet.

### 3.4 Zentrale Recheneinheit

Der Roboter NAO ist mit einem Single-Core-Prozessor vom Typ AMD *Geode LX* mit einer Taktfrequenz von 500 Mhz ausgestattet. Diese x86 Architektur ist ein Low-Power Mobilprozessor, der der Klasse *System-on-a-Chip* angehört. Folglich ist er primär für die Anwendung im Bereich des *embedded computings* gedacht und seine Rechenleistung entsprechend begrenzt. Dies muss bei der Entwicklung von Algorithmen und deren praktischen Umsetzung beachtet werden, wenn diese echtzeitfähig sein sollen. Hierbei ist anzumerken, dass dies einer der größten Nachteile der Plattform NAO V3.3 gegenüber anderen kleinen mobilen Robotern wie z.B. dem DARwIn-OP<sup>14</sup> ist. Bereits Eigenentwicklungen des Instituts für Roboterforschung in Form des *Bender DHB-1* [20] zeigen, dass es möglich ist herkömmliche Hardware für mobile Rechner, in diesem Fall ein Intel Centrino Core Duo mit 1,66 Ghz, auf humanoiden Robotern zu verwenden. Die Stromaufnahme stellt hierbei kein primäres Problem dar, da diese auch beim Bender vernachlässigbar klein gegenüber der Stromaufnahme der Motoren ist. Im Falle von Aldebaran hat die Wahl vornehmlich thermische Gründe, da CPU und Mainboard aus Designgründen im Kopf des Roboters verbaut sind und dieser nur schwer gekühlt werden kann. Für die Version V4 hat Aldebaran aber bereits ein Upgrade auf einen Intel Atom Prozessor angekündigt.

### 3.5 Steuerungscomputer

Zusätzlich zur zentralen Recheneinheit des Roboters im Kopf befindet sich im Torso des Roboters das sogenannte *Chestboard* auf dessen ARM Prozessor der *Device Communication Manager*, auch als *DCM* bezeichnet, operiert. Auch wenn es sich beim DCM um ein Softwareprogramm handelt, wird dieser Begriff von Aldebaran synonym zur Hardware verwendet. Aufgabe des Boards ist es die Sensoren auszulesen und die

---

<sup>14</sup>Intel Atom Z530 1.6 GHz

Kommandos an den Motor zu übertragen. Dies geschieht mittels eines I2C-Busses. Seit Version V3 hat sich die Frequenz des Steuercomputers von 50 auf 100 Hz erhöht. Als Folge können sowohl die Sensoren mit einer höheren Frequenz ausgelesen werden, als auch die Bewegungskommandos in kürzeren Zeitabständen an die Steuerboards der Motoren übertragen werden. Der DCM übernimmt Schutzfunktionen für die Hardware, wie beispielsweise den Übersetzungsschutz der Motoren, ebenso wie die Vorfilterung der Sensorwerte. Technische Details der Umsetzung dieser Funktionen sind nicht bekannt.

## 3.6 Software

Auf dem NAO wird ein Linux Betriebssystem mit einer *Real-time Extension* ausgeführt. Dieses führt die proprietäre Aldebaran Framework-Software *NAOqi* aus. Primär fungiert NAOqi durch das DCM-Modul als Schnittstelle für den Benutzer zur Hardware des Roboters. Darüber hinaus ergänzt Aldebaran das Programm aber um eine Vielzahl von Komfortfunktionen, die dem unerfahrenen Benutzer einen leichten Einstieg ermöglicht. Zusätzlich vertreibt Aldebaran auch externe Tools, die eine grafische Programmierung und intuitives Design von Bewegungsabläufen zulassen. Auch wenn dieses Softwarepaket durch seinen Funktionsumfang hilfreich ist, um sich bei der Entwicklung auf bestimmte Bereiche zu fokussieren und Funktionalität in anderen bereitstellen zu lassen, so ist es doch für den Einsatz in der Forschung nicht performant genug und schränkt die eh schon geringe Rechenleistung des Roboters noch weiter ein. Da zudem die einzelnen Module nur eingeschränkt konfigurierbar sind und es sich zudem um eine geschlossene Software handelt, die vom Benutzer nicht erweitert werden kann, fiel die Wahl auf ein anderes Framework. Das *German Team Framework* [91], das von der Humbolt-Universität zu Berlin, der Universität Bremen, dem DFKI Lab Bremen, der Technischen Universität Darmstadt und der Technischen Universität Dortmund für den Sony AIBO entwickelt wurde, fand über die Jahre eine weite Verbreitung in der SPL Liga des RoboCups und wurde daher kontinuierlich weiterentwickelt. Da auch der Sony AIBO nur eine sehr geringe Rechenleistung zur Verfügung hatte, kommt das Framework mit den geringen Ressourcen des Roboters NAO ebenfalls problemlos zurecht. Zudem lag der Fokus der Software auf der verteilten Entwicklung für das GermanTeam. Daher ist sie grundsätzlich modular aufgebaut und ermöglicht die Kombination und den Austausch von Lösungen unterschiedlicher Teams. Da es sich um Open Source Software handelt besteht die Option das Framework bei Bedarf zu erweitern und anzupassen und so wurde es im Jahre 2007 vom Team BreDoBrothers<sup>15</sup>, einer Kooperation, der Universität Bremen, dem DFKI Lab Bremen und der Technischen Universität Dortmund, für den Roboter NAO portiert und erstmals beim RoboCup Suzhou erfolgreich eingesetzt. Die Tatsache, dass in den folgenden Jahren sowohl das Team BHuman, der Universität Bremen, als auch die NAO Devils, der Universität Dortmund, wiederholt ihre neueste Version der Software veröffentlicht haben, hat dazu geführt, dass die Software in der NAO SPL mittlerweile eine weite Verbreitung erlangt hat. NAOqi lässt sich aber nicht gänzlich deaktivieren, da zumindest das DCM Modul benötigt wird, um das Chestboard anzusprechen. Dies ist eine Vorkehrung von Aldebaran, da dieses Modul zusätzlich zu den Treibern für die Hardware auch eine Software-Sicherheitsschicht zur Ansteuerung

---

<sup>15</sup><http://www.bredobrothers.de>

der Motoren bereitstellt. Aus diesem Grund wurden alle Module bis auf das DCM deaktiviert und NAOqi von Team BHuman um eine Schnittstellen-Bibliothek erweitert. Diese *libBHuman* kommuniziert mit dem eigentlichen *GermanTeam*-Framework und leitet Informationen und Befehle der Sensoren und Aktoren in beide Richtungen weiter.

Teil des *GermanTeam*-Frameworks ist der Simulator *SimRobot* [69]. Zur Simulation der Dynamik der Festkörper kommt die verbreitete *Open Dynamics Engine*<sup>16</sup>, abgekürzt als *ODE*, zum Einsatz. Diese Open Source Simulation wird seit 2001 kontinuierlich weiterentwickelt und findet bei unterschiedlichen anderen Forschungsprojekten Anwendung. Weiterhin setzen auch kommerzielle Simulatoralternativen für den Roboter NAO, wie beispielsweise *Webots* von Cyberbotis<sup>17</sup>, auf die ODE. Diese bieten zwar durch optimierte Programmierung einen Performance-Vorteil gegenüber *SimRobot*, doch bietet *SimRobot* durch den offenen und veränderbaren Quellcode zum Ausgleich Flexibilität. So konnte bereits bevor der Roboter NAO kommerziell verfügbar war, ein Modell des Roboters aus den von Aldebaran veröffentlichten Konstruktionsdaten erstellt werden und der Simulator um die nicht vorhandenen Inertial- und FSR-Sensoren ergänzt werden. Diese Sensormodelle bilden - soweit möglich - die in Kapitel 3.3 untersuchten Charakteristiken und Störeinflüsse der Sensoren des NAO ab. Die Güte der Simulation lässt sich über Parameter adaptieren und so konnten u. a. schon Heibel [68] und Urbann [113] zeigen, dass der Simulator in der Lage ist das physikalische Verhalten von realen Robotern korrekt abzubilden.

## 3.7 Stromversorgung

Der Roboter NAO wird mit einem Lithium-Ionen-Akkumulator mit Strom versorgt, der hinter einer Klappe im Rücken des Roboters eingebaut wird. Ein Netzteil kann mit dem Akku direkt verbunden werden, um den Roboter auch im laufenden Betrieb mit Strom zu versorgen. Über die Jahre wurden von Aldebaran Akkus mit unterschiedlicher Kapazität ausgeliefert, so dass keine allgemeine Aussage über die Laufzeit des Roboters getroffen werden kann. Der aktuell ausgelieferte Akku mit 2 Ah Kapazität reicht aus, um den NAO im Spielbetrieb über rund 15 bis 25 Minuten mit Strom zu versorgen, abhängig davon, wie viel der Roboter sich bewegt.

---

<sup>16</sup><http://www.ode.org/>

<sup>17</sup><http://www.cyberbotics.com/overview>



---

### Propriozeption mittels Sensorik

---

Der Fokus dieser Arbeit liegt auf der Bewegungsplanung und -ausführung humanoider Roboter. Da die Bereiche *Motion* und *Cognition* in der Robotik eng miteinander verbunden sind, ist es nicht möglich sich mit fortgeschrittener Bewegungssteuerung zu beschäftigen, ohne zumindest interne Sensorik als Grundlage zu betrachten. Das Ziel einer Bewegung des Roboters ist es immer, diesen von einer Ausgangspose  $P_i$ , in eine Zielpose  $P_j$  zu überführen. Um die hierzu notwendigen Schritte im Bewegungsablauf zielgerichtet planen zu können, ist es zwingend erforderlich, die aktuelle Ausgangspose  $P_i$  zu kennen. Diese Kenntnis ist Teil des sogenannten *Ego-Modells*, in dem der Roboter Wissen über seinen internen Zustand verwaltet. Informationen für diese Modellierung kann der Roboter nur durch den Einsatz geeigneter Sensorik erfassen. Es folgt daher ein einführender Exkurs in das Arbeitsgebiet der *Cognition* mit dem Fokus der *Propriozeption*, also der Eigenwahrnehmung, des Roboters. Dieser schafft notwendige Grundlage für das Verständnis des weiteren Verlaufs der Arbeit. In Kapitel 4.1 finden sich hierzu Überlegungen über den Einsatz von Sensoren zur Messung der aktuellen Lage des Roboters im Raum. Eine hervorstechende Rolle für die folgenden Kapitel wird der Schwerpunkt - im weiteren Verlauf vom Englischen Ausdruck *Center of Mass* abgekürzt mit CoM - des Roboters einnehmen. Aus diesem Grunde geht Kapitel 4.2 besonders auf dessen Wahrnehmung ein, um die Implikation auf folgende Kapitel besser einschätzen zu können. Da für humanoide Roboter aufgrund ihrer Struktur die Stabilität des Roboters eine Nebenbedingung ist, die nicht in jedem Bewegungsablauf gewährleistet ist, muss dann in Kapitel 4.3 ein Blick auf den aktuellen Stand der Forschung bezüglich Stabilitätskriterien zweibeiniger Roboter folgen. Dieser erfolgt zunächst theoretisch und wird dann in Kapitel 4.4 um eine Diskussion unterschiedlicher Messmethoden ergänzt werden. Im Vordergrund steht dabei der Einfluss von Sensorfehlern auf die Beurteilung des Gleichgewichts. Deren Zusammenhang wird zunächst simulatorisch untersucht und zum Abschluss des Kapitels um praktische Untersuchungen am realen Roboters NAO ergänzt.

## 4.1 Lage im Raum

Die Lage des Roboters im Raum wird durch sein roboterzentrisches Koordinatensystem (RzKS) im Bezug auf das Weltkoordinatensystem (WKS) bestimmt, siehe auch Kapitel 2.2. Der Roboter registriert Wahrnehmungen immer im eigenen Koordinatensystem, also relativ zur Position des entsprechenden Sensors und plant Bewegungsabläufe im eigenen Gelenkwinkelraum. Im Umkehrschluss ist Wissen um seine Lage im Raum eine entscheidende Voraussetzung, um diese Informationen wieder in das WKS und somit in das Weltmodell zu übertragen. Das Zentrum des RzKS bildet der Torso des Roboters, vergleiche Kapitel 2.2.2, da dieser auch die Basis der kinematischen Ketten der einzelnen Glieder bildet. Somit erlangt der Roboter Wissen über seine Lage im Raum, wenn er die Orientierung und Position seines Oberkörpers relativ zum WKS bestimmen kann. Hierzu stehen dem Roboter NAO eine Vielzahl unterschiedlicher Sensoren zur Verfügung. Wie diese zur Bestimmung der Pose des Oberkörpers eingesetzt werden können bildet den Schwerpunkt dieses Kapitels und wird nach einer theoretischen Betrachtung abschließend nochmals im Bezug auf den praktischen Einsatz diskutiert.

### 4.1.1 Gelenkpositionen

Zu jedem Zeitpunkt, an dem der Roboter mit mindestens einem seiner Füße planen Kontakt mit dem Boden hat, wird die Position seines Oberkörpers relativ zum Boden lediglich durch die Stellung seiner Gelenke beeinflusst. Somit kann unter dieser Bedingung mit Hilfe der in Kapitel 3.1.1 vorgestellten Methode der Vorwärtskinematik aus den Messwerten der Gelenkwinkelsensoren zu jeder Zeit seine Lage im Raum eindeutig berechnet werden. Es ist lediglich zu beachten, dass die bislang vorgestellte kinematische Kette den Oberkörper zum Ursprung hat, wohingegen nun das Fußkoordinatensystem (FKS) des Standfußes der Ausgangspunkt ist. Die kinematische Kette muss demnach umgekehrt betrachtet werden. Die Inversen der Transformationsmatrizen, vergleiche Kapitel 2.2, ist daher gegeben durch:

$$\mathbf{T}_{6,0} = \mathbf{T}_{5,6}^{-1} \cdot \mathbf{T}_{4,5}^{-1} \cdot \dots \cdot \mathbf{T}_{0,1}^{-1}. \quad (4.1.1)$$

Ist durch diese Rechnung die Pose des Oberkörpers bekannt, kann wie gewohnt mit der Vorwärtskinematik fortgefahren werden, um die Position der verbleibenden Extremitäten im Raum zu bestimmen. Kapitel 3.3.3 hat aufgezeigt, dass die Gelenkwinkelencoder nicht frei von Fehlern sind. Da aber in jede Transformation der Formel (4.1.1) der Messwert eines solchen Sensors in Form einer Rotation einfließt, wird sich dieser Fehler in die Berechnung der Position und Orientierung des Oberkörpers fortpflanzen. Dieses Phänomen wird als *Kinematikfehler* bezeichnet. Der Fehler ist rotatorischer Natur. Die angrenzenden Festkörper wirken folglich als Hebel und so fließt deren Dimension ebenfalls in den resultierenden Fehler ein. Da es sich bei der Rechnung um eine Kombination unterschiedlicher Gelenksensoren zu einer kinematischen Kette handelt, ist deren gegenseitiger Einfluss nicht unmittelbar ersichtlich. Vielmehr ist eine detaillierte Betrachtung in Form einer Fehlerfortpflanzungsanalyse erforderlich. Der rotatorische Fehler eines einzelnen Sensors wurde durch eine obere Grenze von rund  $\pm 1,5^\circ$  abgeschätzt und wird im Folgenden jeweils mit  $\delta_{Joint}$  bezeichnet.

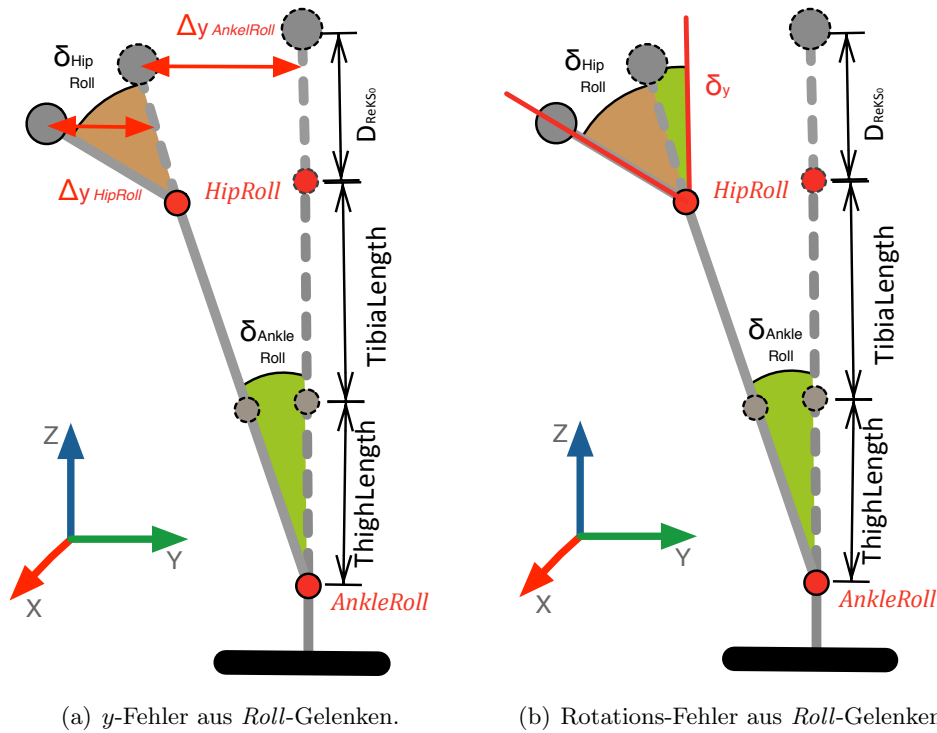
(a)  $y$ -Fehler aus *Roll*-Gelenken.(b) Rotations-Fehler aus *Roll*-Gelenken.

Abbildung 4.1.1: Fehlerfortpflanzung der Lagebestimmung aus Encodern.

Zur Fehlerfortpflanzung ist der Einfluss jedes einzelnen Gelenkes auf die Position und Orientierung des ReKS-Ursprungs zu untersuchen. Bedenkt man die kinematische Struktur des NAO, vergleiche Abbildung 3.1.2, so erkennt man, dass die Gelenke jeweils nur einen Einfluss auf  $x$ - oder  $y$ -Achse des FKS haben. Eine Ausnahme stellt das *HipYawPitch*-Gelenk dar, dessen Achse schräg gestellt ist. Da dieses aber das Ende der kinematischen Kette bildet, können die Fehler in  $x$ - und  $y$ -Richtung dennoch getrennt voneinander betrachtet werden. Das Prinzip der gegenseitigen Beeinflussung demonstriert Abbildung 4.1.1 anhand der *Roll*-Gelenke. Die reale Position und Stellung der Gelenke ist gestrichelt, die fälschlich berechnete jeweils durchgehend dargestellt. Betrachtet wird zunächst das Fußgelenk *AnkleRoll*. Dessen Fehler  $\delta_{AnkleRoll}$  hat den größten Einfluss auf den Fehler  $\Delta_y$  in  $y$ -Richtung, wenn das Bein vollkommen gestreckt ist und sich auch der Oberkörper in gleicher Linie befindet. Für den resultierenden Fehler  $\Delta_{yAnkleRoll}$  auf die  $y$ -Komponente der Lage kann Abbildung 4.1.1(a) folgende trigonometrische Gesetzmäßigkeit entnommen werden:

$$\Delta_{yAnkleRoll} = \sin \delta \cdot (L_{Tibia} + L_{Thigh} + D_{ReKS_0}). \quad (4.1.2)$$

Führt man diese Analyse weiter fort, fällt bei Betrachtung von Abbildung 4.1.1(a) auf, dass der resultierende Fehler  $\Delta_{yHipRoll}$  nicht mehr unmittelbar aus  $\delta_{HipRoll}$  berechnet werden kann. Die vermeintliche Position des *HipRoll*-Gelenkes wurde bereits durch  $\delta_{AnkleRoll}$  beeinflusst. Folglich muss der Fehler des Hüftgelenks noch entsprechend transformiert werden. In diesem Fall kann dies noch unmittelbar durchgeführt werden, doch wird spätestens bei der Betrachtung des *HipYawPitch*-Gelenkes klar, dass dies resultie-

renden Transformationen komplizierter werden. Betrachtet man nun die  $x$ -Richtung, so verkompliziert sich die Rechnung weiter durch das zusätzliche Kniegelenk. Aus diesem Grund wird auf eine Weiterführung der theoretischen Betrachtung des Kinematikfehlers an dieser Stelle verzichtet. Stattdessen wird zur Abschätzung einer ungefähren oberen Grenze die Implementierung des Kinematik-Moduls genutzt, welche nach Aufrundung folgenden Ergebnisse liefert:

$$\Delta x \lesssim 20 \text{ mm}, \quad (4.1.3)$$

$$\Delta y \lesssim 15 \text{ mm}, \quad (4.1.4)$$

$$\Delta z \lesssim 1 \text{ mm}. \quad (4.1.5)$$

Das Vorgehen der Fehleranalyse für die Orientierung erfolgt analog zum bisherigen Vorgehen. Wie Abbildung 4.1.1(b) verdeutlicht, summieren sich die rotatorischen Fehler entsprechend ihrer Ebene zu  $\delta x$ ,  $\delta y$  und  $\delta z$  unmittelbar auf. Lediglich durch die Schrägstellung des *HipYawPitch*-Gelenkes ist die Rechnung nicht gänzlich trivial. Erneute Abschätzung mit dem Kinematik-Modul ergibt als obere Grenzen:

$$\delta x \lesssim 4^\circ, \quad (4.1.6)$$

$$\delta y \lesssim 4^\circ, \quad (4.1.7)$$

$$\delta z \lesssim 1^\circ. \quad (4.1.8)$$

Es ist zu beachten, dass diese Analyse nur unter der Annahme korrekt ist, dass sich der Roboter auf ebenem Boden befindet und das FKS somit keiner weiteren Rotation relativ zum Weltkoordinatensystem unterliegt. Andernfalls müsste diese Rotation des Fußes zum Boden durch ein weitere virtuelles Gelenk in der Kette simuliert werden. In der Praxis gestaltet sich dies schwierig, da dieser Winkel in der Regel nicht gemessen werden kann. Für den NAO im Spielbetrieb des RoboCups stellt dieser Fall aber keine Einschränkung dar.

#### 4.1.2 Inertialsensoren

Eine Methode die Orientierung im Raum unmittelbar aus Messwerten zu bestimmen, ist der Beschleunigungssensor des NAO, der Teil der Inertialsensoreinheit ist, vergleiche Kapitel 3.3.1. Diese ist im Torso des Roboters NAO verbaut und weicht daher nur durch eine feste Translation vom Zentrum des Roboterkoordinatensystems ab, da ihre Orientierung analog zur Orientierung des Oberkörpers gewählt ist. Befindet sich der Roboter in Ruhe, wird also nicht durch externe oder interne Kräfte beschleunigt, so misst dieser Sensor lediglich die Erdbeschleunigung  $g$  von rund  $9,81 \text{ m/s}^2$ . Deren Richtung im RzKS kann bestimmt werden, da der Sensor Beschleunigungswerte für drei Achsen messen kann. Die Gravitation beschleunigt Masse in radialer Richtung zum Erdkern, wodurch die Erdbeschleunigung senkrecht auf ebenen Flächen steht. Misst der Roboter also eine Gravitation, die parallel zu seiner Hochachse verläuft, befindet sich der Torso des Roboters aufrecht und folglich seine  $x$ - und  $y$ -Achse parallel zum Boden. Ist dies nicht der Fall, kann aus der  $x$ - und  $y$ -Komponente der Beschleunigung die Lage dieser Achsen bestimmt werden und daher die Rotation des Torsos im Raum.



Der Fehler in der Orientierungsmessung ergibt sich daher im statischen Fall direkt aus dem Fehler des Beschleunigungssensors. Im dynamischen Fall hingegen führt die Bewegung des Roboters dazu, dass die Messung des Beschleunigungssensors von der Roboterbeschleunigung überlagert wird. Um dies zu kompensieren, können zusätzlich die Geschwindigkeitsinformationen des Gyrometers zur Bestimmung der rotatorischen Bewegung des Oberkörpers verwendet werden. Diese verwendet Aldebaran beispielsweise in Form eines - nicht weiter dokumentierten - Kalman-Filters für ihre NAOqi-Methode zur Bestimmung der Orientierung. Auch wenn diese die Messung gegenüber dem reinen Beschleunigungssensor verbessert, so ist sie durch die ständige Bewegung des Oberkörpers beim der Fortbewegung eines humanoiden Roboters dennoch soweit Störungen überlagert, dass diese Methode in der Praxis ohne fortgeschrittene Filtertechniken lediglich als Richtwert verwendet werden kann.

Bislang wurde nur über die Messung der Orientierung gesprochen. Grund hierfür ist die Tatsache, dass die Position des RzKS nicht mittels der Inertialsensoren direkt bestimmt werden kann. Wenn aber eine Ausgangsposition im Raum bekannt ist, können die Bewegungsinformationen aus Beschleunigungssensor und Gyrometer verwendet werden, um über inkrementelle Updates die Position zu verfolgen. Hierauf soll aber an dieser Stelle nicht weiter eingegangen werden, da dies in Kapitel 6.3 anhand fortgeschrittener Filtertechniken ausführlicher aufgegriffen wird.

### 4.1.3 Optische Kamera

Auch wenn der Mensch mit seinem Innenohr ein hochentwickeltes Organ zur Wahrnehmung seiner Lage im Raum hat, so ist dies doch nicht das einzige Sinnesorgan, das er hierzu verwendet. Vielmehr fusioniert das Gehirn möglichst viele Sinneseindrücke, die Informationen über die Orientierung liefern können. Das menschliche Auge ist zwar nicht direkt in der Lage Informationen über den körpereigenen Zustand zu liefern, sehr wohl aber über seine Relation zur Umgebung. Wenn der Mensch sich orientieren kann, hat er in der Regel soviel Wissen über die Struktur seiner Umwelt, dass er hierdurch implizit Rückschlüsse auf seine eigene Lage im Raum ziehen kann. Das gleiche Konzept lässt sich auch auf humanoide Roboter anwenden. Beispielsweise ein Torpfosten oder ein Querbalken befinden sich in der Regel senkrecht, bzw. waagrecht, zum Boden. Werden diese also in einem Kamerabild erkannt, kann dadurch auf Orientierungsinformationen der Kamera zurückgeschlossen werden. So konnte dies Verfahren beispielsweise für den Roboter Bender mit sehr guten Ergebnissen eingesetzt werden, dessen Omnivision-Kamera starr mit dem Oberkörper verbunden ist. Da im Jahre 2007 in der *Humanoid League* des RoboCups noch mit vier eindeutigen Landmarken an den Eckpunkten des Spielfeldes gespielt wurde, konnten aus deren Position im Kamerabild zuverlässig die Oberkörper-Orientierung mittels Triangulation berechnet werden, solange alle vier gleichzeitig im Bild zu sehen waren.

Für den Roboter NAO ergeben sich allerdings in der Praxis zwei Schwierigkeiten. Zum einen ist der Roboter nicht in der Lage Schlussfolgerungen intuitiv zu ziehen, sondern braucht explizites Wissen über seine Umwelt. Durch die fortschreitende Reduktion der eindeutigen Landmarken des RoboCups wird es damit immer schwerer, eindeutige

Schlüsse über die Orientierung der Umgebung zu ziehen. Zwar konnte das *NAO-Team HTWK Leipzig* [110] zeigen, dass die Position der Kamera auch aus den Feldlinien und deren Winkeln bestimmt werden kann, doch ist dies nur im statischen Fall möglich. Zum anderen kann aus dieser Information nur auf die Lage der Kamera im Raum geschlossen werden. Diese ist beim NAO nicht starr mit dem Oberkörper verbunden und daher muss die Oberkörper-Orientierung aus dieser erst ermittelt werden. Dies ist über die kinematische Kette des Halses nach der vorgestellten Methode der direkten Kinematik theoretisch möglich. Allerdings ist dieses Vorgehen für den NAO mit einem praktischen Problem verbunden. Die Kamerabilder und die Gelenkwinkel kommen aus unterschiedlichen Nachrichten des DCM. Für beide Sensorinformationen sind zwar Zeitstempel verfügbar, doch verwenden diese unterschiedliche, von Aldebaran undokumentierte, Formate. Weiterhin ist bekannt, dass es einen zeitlichen Verzug zwischen diesen Informationen gibt, unbekannt ist allerdings, wie groß dieser ist. Somit pflanzt sich der resultierende Synchronisationsfehler auf unbekannte Weise in die Orientierungsmessung fort und macht diese im dynamischen Fall unbrauchbar. Aus diesem Grund findet diese Methode im weiteren Verlauf der Arbeit keine Anwendung.

## 4.2 Bestimmung des Schwerpunktes

Die Physik definiert den *Massenmittelpunkt*  $\mathbf{r}_s$  der Masse  $M$  eines Festkörpers als gewichtetes Integral der Ortsvektoren  $\mathbf{r}$  über alle Massepunkte  $dm$ . Alternativ kann das Volumenintegral  $dV$  über die Dichteverteilung  $\rho(\mathbf{r})$  des Körpers gebildet werden:

$$\mathbf{r}_s = \frac{1}{M} \int \mathbf{r} \, dm = \frac{1}{M} \int \mathbf{r} \rho(\mathbf{r}) \, dV. \quad (4.2.1)$$

Im Allgemeinen ist der Massenmittelpunkt nicht gleichbedeutend mit dem *Gravizentrum*, dem Punkt also, an dem das Mittel der gewichteten Gravitationskräfte angreift. Da aber in guter Näherung davon ausgegangen werden kann, dass das Gravitationsfeld in dem der NAO sich bewegt homogen ist, sind Massenmittelpunkt und Gravizentrum in den durchgeführten Experimenten identisch und werden im Folgenden zusammenfassend als Schwerpunkt (CoM) bezeichnet.

Der Roboter NAO besteht aus einer Vielzahl starrer Festkörper, deren Schwerpunktposition folglich fest im jeweiligen körpereigenen Koordinatensystem ist. Gruppen dieser Festkörper sind wiederum starr miteinander verbunden, und bilden somit erneut Festkörper, die die Glieder der kinematischen Kette darstellen. Somit ist klar, dass es sich bei den Gliedern nicht um Festkörper mit einer homogenen Massenverteilung handelt und daher der Schwerpunkt nicht mit dem Volumenmittelpunkt, auch als geometrischer Schwerpunkt bezeichnet, zusammenfällt. Aus diesem Grund muss zur Berechnung der einzelnen Schwerpunkte die Dichteverteilung  $\rho(\mathbf{r})$  gemäß Formel 4.2.1 berücksichtigt werden. Diese ist aber nicht bekannt. Über die einzelnen Baugruppen ist der NAO Dokumentation lediglich die Geometrie und das Gesamtgewicht zu entnehmen. Somit steht keinerlei Information über die Massenverteilung zur Verfügung. Bis Aldebaran genauerer Daten veröffentlicht, kann daher für die Gliedmaßen jeweils der eigentliche Schwerpunkt durch den geometrischen angenähert werden.

Sind die Schwerpunkte der einzelnen Gliedmaßen  $i$  bekannt, kann der Gesamtschwerpunkt des Systems durch die diskretisierte Schwerpunktformel berechnet werden. Das Volumenintegral durch die Summe über alle Ortsvektoren  $\mathbf{r}_i$  substituiert:

$$\mathbf{r}_s = \frac{1}{M} \sum_i \mathbf{r}_i m_i, \quad (4.2.2)$$

wobei  $\mathbf{r}_i$  den jeweiligen Ortsvektor des Schwerpunktes des  $i$ -ten Körpers im ReKS bezeichnet. Folglich ist klar, dass die Position des resultierenden Roboterschwerpunktes zu jeder Zeit abhängig von der derzeitigen Konfiguration ist. Verändert sich diese, so verschieben sich auch die Schwerpunkte der einzelnen Festkörper relativ zum Ursprung des ReKS. Somit müssen die Ortsvektoren  $\mathbf{r}_i$  der einzelnen Körperschwerpunkte schrittweise durch die vorgestellte Vorwärtskinematik berechnet werden. Damit wird die Berechnung des CoM ebenfalls durch den bereits diskutierten Kinematikfehler beeinflusst, vergleiche Kapitel 4.1.1. Die Abschätzung des Einflusses auf die CoM-Position ist aber nicht so einfach möglich, wie bei der Diskussion zur Lage im Raum. Grund ist die Abhängigkeit von der aktuellen Konfiguration.

Die kinematische Struktur des NAO ist axialsymmetrisch zur Hochachse des Roboters aufgebaut. Folglich gibt es Konfigurationen, in denen sich aufgrund dieser Symmetrie die Fehler vollständig ausgleichen. Doch auch in den anderen Stellungen gleichen sich zumindest Teile der Fehler aus. Es ist sicher möglich auch in diesem Fall eine Abschätzung einer oberen Grenze mit einer *Worst-Case*-Konfiguration zu finden. Diese hätte aber keinen Bezug zur Praxis, da es bei den Bewegungen humanoider Roboter immer gewisse Symmetrien in der Konfiguration gibt, um dessen Gleichgewicht zu gewährleisten. Aus diesem Grund wirkt sich der Fehler verstärkt nur auf die Höhe des CoM und nicht auf dessen Projektion in die  $x$ - $y$ -Ebene des Boden aus.

### 4.3 Theorie des Gleichgewichts

Für einen humanoiden Roboter spielt das Gleichgewicht eine entscheidende Rolle für die Fortbewegung. Um diesen Aspekt besser verstehen zu können, folgt eine Einführung in den Begriff der Stabilität und den Einfluss der Bewegung auf diese. Die Physik versteht unter dem Begriff *Gleichgewicht* einen Zustand eines Systems, vergleiche u. a. [111]. Dieser kann entweder *stabil*, *labil* oder *indifferent* sein. Bei einem stabilen Gleichgewicht variieren sich die Potentiale und somit die Kräfte im System aus, so dass bei einer kleinen Auslenkung das System erneut diese Ruhelage erreicht. Die *Systemtheorie* nennt diese Ruhelage einen *Attraktor* des Systems. Ein labiles Gleichgewicht hingegen zeichnet sich dadurch aus, dass sich das System in einer Ruhelage auf einem Maximum des Potentials befindet und dies bei einer kleinen Auslenkung verlässt und einem anderen Potential zustrebt. Eine Ruhelage wird als indifferent bezeichnet, wenn das System zwar bei einer kleinen Auslenkung eine neue Ruhelage einnimmt, sich das Potential dabei aber nicht ändert. Für humanoide Roboter reicht eine einfache Übertragung der physikalischen Prinzipien nicht aus. Vielmehr soll das Gleichgewicht für diese wie folgt definiert sein:

**Gleichgewicht:** *Ein humanoider Roboter befindet sich im Gleichgewicht, wenn er aus eigener Kraft seine Lage im Raum kontrollieren kann. Kontakt zum Boden darf nur mit den Füßen bestehen.*

Gleichgewicht beschreibt also die Eigenschaft des Roboters, die Stabilität seines Systems zu erhalten. Die Robotik unterscheidet hierbei zwischen zwei Arten von Stabilität, der *statischen* und der *dynamischen* Stabilität, vergleiche beispielsweise [100] oder [101]. Eine allgemein anerkannte Definitionen dieser Unterscheidung gibt es nicht, und kann daher nicht zitiert werden. Vielmehr umschreibt jeder Autor den Zusammenhang anschaulich:

**Statische Stabilität:** *Der Zustand eines Roboters wird als statisch stabil bezeichnet, wenn dieser das Gleichgewicht ohne Bewegung erhalten kann.*

Betrachten wir nun das System Roboter unter physikalischen Gesichtspunkten, so lässt sich hierfür als hinreichende Bedingung für einen stationären Roboter eine eindeutige Definition festhalten:

*[...] when the velocities and accelerations are small, and assuming that contact points are motionless during each phase of the step, the balance condition reduces to the classical one: the projection of the robot's center of mass belongs to the convex hull of contact points. [100]*

Bedenkt man den Umkehrschluss, erkennt man, dass dies stimmt. Befindet sich die Projektion des Schwerpunktes, aus dem englischen abgekürzt als pCoM, nicht innerhalb des *Support-Polygons*, so entsteht durch die Erdbeschleunigung ein Drehmoment, das den Roboter um eine Kante seiner Auflagefläche rotieren lässt. Als Folge fällt dieser. Auch eine Bewegung kann statisch stabil sein und ist dies, wenn zu jedem Zeitpunkt die Bedingung der statischen Stabilität erfüllt ist und der Roboter in einen stationären Zustand übergehen kann, ohne instabil zu werden.

Eine Bewegung, die dieser Bedingung nicht genügt, kann dennoch stabil sein, wenn die durch die Bewegung entstehenden Kräfte den Roboter im Gleichgewicht halten. Man sagt dann, der Roboter sei *dynamisch* stabil:

**Dynamische Stabilität:** *Der Zustand eines Roboters wird als dynamisch stabil bezeichnet, wenn dieser durch seine Bewegung das Gleichgewicht erhält.*

Auch hierfür wird ein Kriterium benötigt, um zu entscheiden ob dynamische Stabilität vorliegt, oder nicht. Dieses ist aber nicht so trivial zu definieren, wie das der statischen Stabilität. Daher widmet sich Kapitel 4.3.1 der Definition eines dynamischen Stabilitätskriteriums, nachdem einige theoretische Grundlagen erörtert wurden.

Nach den hier vorgestellten Definitionen ist ein Roboter entweder stabil oder instabil. Dies deckt sich mit dem vorgestellten physikalischen Konzept des Gleichgewichts. Jedoch ist dieses Wissen nicht ausreichend, um die Güte einer geplanten Bewegung genügend beschreiben zu können. Aus diesem Grund wird ein Maß für die Stabilität

eingeführt, um beurteilen zu können, *wie* stabil der Roboter zu einem Zeitpunkt ist:

*Das Maß der Stabilität gibt für einen Roboter an, in welchem Umfang er aus seiner Ruhelage ausgelenkt werden kann, ohne in einen instabilen Zustand überzugehen. Ob es sich dabei um eine stabiles oder indifferentes Gleichgewicht handelt, ist nicht von Belang.*

Diese Definition ist zunächst bewusst unkonkret und soll lediglich die Notwendigkeit verdeutlichen, ein Maß für den Grad der Stabilität zu haben. Eine Konkretisierung dieser Definition erfolgt nach Einführung der benötigten Grundlagen in Kapitel 4.3.1.

Basierend auf diesem Wissen kann diskutiert werden, warum sich zwar jeder Roboter für die Fortbewegung im Gleichgewicht befinden muss, dies aber besonders für humanoide Roboter eine entscheidende Rolle spielt. Die Fortbewegung eines Roboters resultiert immer in einer Bewegung der Position seines CoM. Rollende Roboter bewegen sich fort, indem Aktuatoren relativ zum Boden bewegt werden. Diese sind in der Regel Räder oder Ketten, die rotierend gelagert sind, deren Position relativ zum CoM des Roboters jedoch fest ist. Somit führt die Fortbewegung zu einer Bewegung des Schwerpunktes im Raum, aber nicht zu einer Bewegung des Schwerpunktes relativ zu den Aktuatoren. Weiterhin sind die Aktuatoren rollender Roboter in der Regel in durchgehenden Kontakt mit dem Boden. Das *Support*-Polygon des rollenden Roboters verändert daher nicht die Position im RzKS. Rollende Roboter, mit mehr als zwei Rädern<sup>1</sup>, sind also per Definition statisch stabil, solange nicht die Projektion des CoM auf den Boden durch eine Schräglage des Roboters das *Support*-Polygon verlässt. Die Fortbewegung laufender Roboter basiert ebenfalls auf der Bewegung von Aktuatoren relativ zum Boden. Aufgrund der, durch die Gelenke veränderbaren, Konfiguration der Roboter führt aber diese Bewegung zu einer Verschiebung des CoM relativ zu den Aktuatoren. Zudem müssen diese aufgrund der Art der Fortbewegung wiederholt vom Boden angehoben werden, was eine ständige Veränderung des *Support*-Polygons zur Folge hat. Da die Auflagefläche der Füße eines Laufroboters klein im Vergleich zur Höhe desselben sind, ist die relative Verkleinerung des *Support*-Polygons beim Anheben eines Fußes groß. Dies gilt augenscheinlich besonders für zweibeinige Roboter, bei denen das Anheben eines Beines dazu führt, dass das *Support*-Polygon per Definition nur noch aus der Auflagefläche des Standfußes besteht. In Kombination mit dem, durch die menschliche Struktur gegebenen, relativ hohen Schwerpunkt führt diese dazu, dass ein humanoider Roboter leicht instabil werden kann und beginnt über eine Fußkante zu kippen. Die Veränderung des Potentials und der resultierende Sturz führen dazu, dass der Roboter sich einem neuen Attraktor, der Lage auf dem Boden, annähert. Diese ist zwar stabil, doch ist der Roboter in diesem Zustand per Definition nicht mehr im Gleichgewicht.

Nach dieser Diskussion des Gleichgewichts für humanoide Roboter wird klar, warum dieses eine entscheidende Rolle in der Bewegungssteuerung spielen muss. Für eine Planung der Bewegung, oder Regelung derselbigen, wird immer ein Kriterium zur Bewertung benötigt. Für statische Stabilität wurde bereits ein Kriterium beschrieben, für den praxisrelevanten Fall der dynamischen Stabilität steht dieses noch aus. In den

---

<sup>1</sup>Es sei angemerkt, dass auch ein Roboter mit zwei Rädern statisch stabil sein kann, allerdings muss hierzu der CoM unterhalb der Achse liegen.

vergangenen 40 Jahren ist dieses Thema von unterschiedlichen Wissenschaftlern erforscht worden, so dass dieses Problem, zumindest für waagerechte Untergründe, als umfassend theoretisch gelöst angesehen werden kann. Daher soll an dieser Stelle keine erneute tiefgehende Erörterung erfolgen. Auf eine einführende Vorstellung kann aber nicht verzichtet werden, da für das weitere Verständnis der Arbeit eine Diskussion und Abgrenzung der verschiedenen Konzepte unabdingbar ist. Aus diesem Grund folgt ein rudimentärer Überblick, dessen Vertiefung bei Interesse anhand der zusammenfassenden Ausarbeitung von Vukobratović [115] erfolgen kann.

### 4.3.1 Zero Moment Point

Die ersten Bemühungen, ein Kriterium für das Gleichgewicht humanoider Roboter zu definieren, wurden von Vukobratović bereits in den 70er Jahren des 20. Jahrhunderts durchgeführt. Seine Theorie beruht auf grundlegenden physikalischen Beobachtungen. Bei der Bewegung wirken auf den Roboter unterschiedliche Kräfte. Hierbei handelt es sich sowohl um systemimmanente Kräfte, als Resultat der eigenen Beschleunigung, als auch externe Kräfte, wie die Gewichtskraft oder Stoßkräfte. Nach den Gesetzen der Physik greifen diese an den Schwerpunkten der verschiedenen Festkörper an und summieren sich zu einer Gesamtkraft auf. Sie lassen sich also jeweils durch eine resultierende Kraft  $\mathbf{F}_i$  je Festkörper ersetzen.

Betrachtet wird nun ein Roboter mit  $n$  Festkörpern, die durch Gelenke miteinander verbunden sind. Wird die Flexibilität der Gelenke und Getriebe vernachlässigt und weiterhin davon ausgegangen, dass die elektrischen Haltekräfte  $F_{M_i}$  der jeweiligen Gelenkmotoren groß gegenüber dem Betrag der Kräfte  $\mathbf{F}_i$  und  $\mathbf{F}_{i+1}$  auf die jeweiligen Schwerpunkte der verbundenen Festkörper sind

$$F_{M_i} \gg \max(F_i, F_{i+1}), \quad (4.3.1)$$

so führen diese Kräfte nicht zu einer Veränderung der Gelenkkonfiguration des Roboters. Dann kann der Roboter zu jedem Zeitpunkt der Bewegung als einzelner starrer Festkörper betrachtet werden. Folglich können die Kräfte  $\mathbf{F}_i$  auf die einzelnen Festkörper-Schwerpunkte durch eine resultierenden Gesamtkraft  $\mathbf{F}_{CoM}$  auf den Schwerpunkt des Roboters ersetzt werden:

$$\sum_i \mathbf{F}_i = \mathbf{F}_{CoM} = (F_x \ F_y \ F_z)^T. \quad (4.3.2)$$

Die  $z$ -Komponente  $F_z$  wirkt direkt auf den Boden der Umgebung, da nach der Voraussetzung in Formel (4.3.1) der Roboter weiterhin als einzelner starrer Körper betrachtet werden kann. Sei weiterhin der Boden unbeweglich und unflexibel, erzeugt dieser nach den Axiomen von Newton eine Gegenkraft  $F_{R_z}$ , die  $F_z$  kompensiert. Folglich kann  $F_z$  für die weiteren Betrachtungen außer Acht gelassen werden kann.

Analog zu der Betrachtung von  $F_z$  lässt sich auch für Komponenten  $F_x$  und  $F_y$  argumentieren. Auch diese übertragen sich über die Festkörper auf den Boden, der ihnen

eine Reibungskraft entgegen bringt. Kann davon ausgegangen werden, dass die Haftreibungskraft  $F_{Haft}$  des Bodens auf den Fuß des Roboters groß genug ist,

$$F_{Haft} \gg \max(F_x, F_y), \quad (4.3.3)$$

so werden auch diese kompensiert und es resultiert keine translatorische Bewegung des Roboters aus  $\mathbf{F}_{CoM}$ .

Unter diesen Annahmen kann der Fußpunkt des Roboters als fest angesehen werden und das System verhält sich wie ein starres Pendel. Durch das Hebelgesetz führt daher die Kraft  $\mathbf{F}_{CoM}$  über die Länge  $h$  des starren Körper des Roboters zu einem Drehmoment  $\mathbf{M}_{CoM}$  auf den Kontaktpunkt:

$$\mathbf{M}_{CoM} = \mathbf{h} \times \mathbf{F}_{CoM}. \quad (4.3.4)$$

Um die Auswirkungen auf den Roboter zu diskutieren sei zunächst angenommen, dass die Kontaktfläche des Roboters punktförmig ist, aber weiterhin (4.3.3) erfülle. Für den Fall, dass

$$M_{CoM} = 0, \quad (4.3.5)$$

gilt, entspricht dies dem Fall eines starren Pendels im instabilen Gleichgewicht. Sobald aber

$$M_{CoM} \neq 0, \quad (4.3.6)$$

gilt, bewirkt das Moment eine Veränderung des Potentials und das Pendel beginnt um den Kontaktpunkt zu rotieren - der Roboter fällt. In der Praxis hat die Kontaktfläche des Roboters mit dem Boden aber eine Ausdehnung ungleich der eines Punktes. Somit bildet diese einen Hebel, über den durch die Reaktionskraft  $\mathbf{F}_R$  des Bodens ein Gegen-Drehmoment  $\mathbf{M}_R$  erzeugt wird. Der Punkt, an dem diese Reaktionskraft  $\mathbf{F}_R$  am Fuß angreift ist der Punkt, an dem das Reaktionsmoment das Moment des CoM ausgleicht

$$\mathbf{M}_{CoM} = \mathbf{M}_R, \quad (4.3.7)$$

und somit gilt in diesem Punkt

$$M_x = M_y = 0. \quad (4.3.8)$$

Aus Gleichung (4.3.7) folgt, dass der Roboter trotz der angreifenden Kraft  $\mathbf{F}_{CoM}$  im Gleichgewicht bleibt, wenn der Punkt sich innerhalb des *Support*-Polygons befindet. Gibt es keinen Punkt innerhalb der Kontaktfläche des Fußes für den Bedingung (4.3.7) erfüllt ist, so existiert kein Angriffspunkt für die Reaktionskraft  $\mathbf{F}_R$ . Folglich beginnt der Roboter um eine Fußkante zu rotieren. Dies wird als *passiver* DoF bezeichnet. Somit kann die Existenz eines solchen Punktes als Kriterium für die Stabilität des Roboters benutzt werden. Vukobratović war der Erste, der diesen Zusammenhang Ende der 70er Jahren postulierte [116] und führte einige Jahre später für diesen Punkt, aufgrund der Gleichungen (4.3.7) und (4.3.8), den Namen *Zero Moment Point* (ZMP) ein. Seit der Einführung des ZMPs wurde das Konzept von vielen Forschern untersucht und teilweise neu definiert, so dass kein einheitlicher Konsens über eine Definition besteht. Bei Betrachtung der revidierten Definition von Vukobratović aus dem Jahre 2000 fällt auf, dass er einen Fakt betont, der sonst in Definitionen gerne außen vor gelassen wird:

*(The notion of the ZMP:) The pressure under supporting foot can be replaced by the appropriate reaction force acting at a certain point of the mechanism's foot. Since the sum of all moments of active forces with respect to this point is equal to zero, it is termed the zero-moment point (ZMP). [114]*

Der ZMP existiert nur innerhalb des *Support*-Fußes, da sonst kein Angriffspunkt für die Reaktionskraft  $\mathbf{F}_R$  besteht. Existiert also ein ZMP, ist der Roboter in einem dynamisch stabilen Gleichgewicht, existiert er nicht, so ist er instabil. Da dies oft nicht beachtet wurde, führte Vukobratović später zur Unterscheidung den Begriff des *Fictitious ZMP*, ein:

*If we suppose for a moment that the Point  $P$  is outside the support polygon, let us consider what would then be the meaning of this point. In view of the fact that this position of  $P$  was obtained from the condition  $M_x = M_y = 0$  we can consider it as a **fictitious ZMP (FZMP)**. [115]*

Dies ist keine klare Definition. Betrachtet man aber die Implikation des FZMP auf das Gleichgewicht so kann dieser im weiteren Verlauf eindeutig definiert werden als:

**Fictitious ZMP:** *Der Fictitious ZMP (FZMP) ist der Punkt am Boden, an dem sich der ZMP befinden würde, wenn der Fuß des Roboters eine unendliche Ausdehnung hätte. [94]*

Auch wenn er nicht physikalisch existiert, hat dieser Punkt für die Praxis durchaus Relevanz. Er ist der Punkt, an den der Roboter einen Fuß setzen müsste, um wieder in einen dynamisch stabilen Zustand zu gelangen. Folglich ist er für die Stabilisierung des System durch Ausfallschritte interessant. Weiterhin ist der euklidische Abstand dieses Punktes vom *Support*-Polygon ein Indiz dafür, wie groß das Drehmoment  $\mathbf{M}_{CoP}$  ist, das nicht kompensiert wurde. Somit ist es auch ein Maß für die Beschleunigung, mit der der Roboter um seinen Fußpunkt kippt.

Die Tatsache, dass sich in den Jahrzehnten seit der Definition des ZMPs viele Forschergruppen mit dem Verständnis des ZMPs und seiner Bedeutung für die Praxis auseinandergesetzt haben, führte dazu, dass neben dem ZMP noch weitere Begriffe eingeführt wurden - der FZMP ist nur einer davon. Da dies zu potentiellen Missverständnissen führt, hat Vukobratović sich in seinem Rückblick [115] bemüht, diese gegeneinander abzugrenzen. Um ein eindeutiges Verständnis dieser Arbeit zu fördern, sollen daher die Ergebnisse seiner Untersuchung hier kurz vorgestellt werden. Im Kontext des ZMPs wird oft auch vom *Center of Pressure* (CoP) gesprochen. Sardain und Bessonnet definieren diesen wie folgt:

**Centre of Pressure:** *The field of pressure forces (normal to the sole) is equivalent to a single resultant force, exerted at the point where the resultant moment is zero. This point is termed CoP. [94]*

Vukobratović's Definition des ZMPs legt nahe, dass das CoP analog zum ZMP verstanden werden kann. Dies ist in der Tat korrekt und es gilt  $CoP = ZMP$ , dann wenn der Roboter sich in dynamisch stabilem Gleichgewicht befindet, der ZMP also existiert. Es



ist aber zu betonen, dass im Allgemeinen der CoP ungleich dem ZMP ist, wie Vukobratović in [115] zeigt, da der CoP im Falle der Instabilität noch immer auf der Kante definiert ist, der ZMP in dem Fall aber nicht existiert. Goswami postulierte 1999 ein neues Stabilitätskriterium für zweibeiniges Laufen, den *Foot-Rotation Indicator* (FRI):

*The foot rotation indicator (FRI) point is a point on the foot/ground contact surface, within or outside the convex hull of the foot support area, at which the resultant moment of the force/torque impress on the foot is normal to the surface. [37]*

Als Reaktion wies Vukobratović in [114] nach, dass der FRI lediglich eine andere Definition des ZMPs ist, die den FZMP mit einschließt. Somit ist der Konsens in der Literatur den FRI und den FZMP synonym zu verwenden.

Nach diesem kurze Einblick in die Historie der Stabilitätsanalyse steht noch eine klare Definition aus, wie im Kontext dieser Arbeit der Begriff ZMP verwendet wird. Aus Gründen der Lesbarkeit wird der Begriff ZMP einheitlich für das dynamische Stabilitätskriterium verwendet und nicht zwischen den unterschiedlichen Definitionen unterschieden. Somit bezeichnet ZMP den *realen* ZMP solange dieser existiert und sonst den *fictitious* ZMP. Eine Sonderstellung kommt dem CoP zu Gute. Dieser wird im Folgenden synonym für den mittels der FSR-Sensoren gemessenen ZMP verwendet, siehe Kapitel 4.4.1, um diesen besonders von der Messung der Beschleunigungssensoren, siehe Kapitel 4.4.2, abgrenzen zu können.

Nachdem mit der Existenz des ZMP ein allgemeines Kriterium für die Stabilität eingeführt wurde, kann die Diskussion über den Grad der Stabilität erneut aufgegriffen werden. Dieser muss beschreiben, wie weit der Roboter ausgelenkt werden kann, ohne das Gleichgewicht zu verlieren. Hierzu eignet sich die Position des ZMPs relativ zum Rand des *Support-Polygon*. Je näher er sich an der Kante befindet, desto näher ist der Roboter an einem instabilen Systemzustand. Somit wird als Maß der Stabilität das Minimum des euklidischen Abstands zu den Kanten des *Support-Polygon* gewählt:

*Im Falle der Existenz des ZMPs gibt der minimale euklidische Abstand zu einem Rand des Support-Polygon den Grad der Stabilität an.*

Diese Definition gilt sowohl für statische als auch dynamische Stabilität, da im statischen Fall der ZMP identisch mit dem auf den Boden projizierten CoM (pCoM) ist.

## 4.4 Messung des Zero Moment Point

Bislang wurden die Gleichgewichtskriterien aus rein theoretischer Sicht betrachtet und die zugrundeliegenden physikalischen Zusammenhänge erläutert. In Kapitel 5.2.1 wird aufgezeigt, dass diese Erkenntnisse als Grundlage zur Erzeugung stabiler Bewegungsmuster zur Fortbewegung dienen können. Der Vorteil dieser Herangehensweise liegt in erster Linie darin begründet, dass sich die Stabilität während der Ausführung der Bewegung durch eine Regelschleife beeinflussen lässt. Dazu muss jedoch die Regelgröße

überwacht werden, um eine Abweichung der Ist- von der Soll-Position festzustellen. Der Roboter muss also in der Lage sein, zu einem beliebigen Zeitpunkt seiner Bewegung nicht nur theoretisch beurteilen zu können, ob er sich in einem stabilen Zustand befindet, sondern dies auch durch eine Messung überprüfen können. Der Mensch hat hierfür mit dem Innenohr ein sehr ausgeprägtes Instrument entwickelt, um sein Gleichgewicht intuitiv zu überwachen. Gegenwärtig gibt es im Bereich der Sensorik kein entsprechendes Analogon und folglich steht dem Roboter kein direkter Gleichgewichtssensor zur Verfügung. Daher werden in den folgenden Kapiteln Ansätze und Algorithmen vorgestellt und diskutiert, mit deren Hilfe die vorhandenen Sensoren des Roboters NAO dazu genutzt werden können, die aktuelle ZMP-Position des Roboters zu messen. Wie dieses Wissen vom Roboter letztendlich genutzt werden kann, um über ein Stellglied Einfluss auf die Stabilität seiner Bewegung zu nehmen, wird in Kapitel 5.2.1 vorgestellt.

Da das Konzept des ZMPs sich über die letzten Jahren mehr und mehr als geeignete Methode zur Überwachung der Stabilität von humanoiden Robotern etabliert hat, wurde bereits von verschiedenen Forschern der Einsatz unterschiedlicher Sensortypen zur Messung der ZMP-Position an diversen Roboterplattformen untersucht. Es fehlt jedoch an Untersuchungen, die sich auf die Vergleichbarkeit der Messungen verschiedener Sensoren fokussieren. Der Roboter NAO bietet hier den Vorteil, mit einer Vielzahl von propriozeptiven Sensoren ausgestattet zu sein, mit denen der ZMP gemessen werden kann. Somit bildet er eine geeignete Grundlage, um die verschiedenen Konzepte miteinander zu vergleichen und deren Vorzüge gegeneinander zu diskutieren und evaluieren. Wie Kapitel 3 gezeigt hat, enthält der Roboter NAO alle, in seiner Preisklasse, üblichen Sensoren. Deren Auswertung hat zudem eine repräsentative Qualität für das Preissegment aufgezeigt. Somit ist zu erwarten, dass sich die Ergebnisse dieses Kapitels sinnvoll auch auf andere Roboter mit vergleichbarer Hardware übertragen lassen.

Ein entscheidender Faktor für die Güte der Regelung ist die Qualität der Messung der Regelgröße. Bereits in Kapitel 3 wurde über die Ursachen mögliche Abweichungen der Messwerte von der Realität diskutiert. Da sich gezeigt hat, dass in der Praxis beim NAO solche Fehler nicht unüblich sind, widmen sich die folgenden Kapitel der Untersuchung des Einflusses von Messfehlern auf die ZMP-Bestimmung. Die Diskussion findet zunächst anhand von Simulationsdaten statt, um alle möglichen Fehlerursachen gezielt in die Analyse der Robustheit einfließen lassen zu können. Kapitel 4.4.4 wird diese Betrachtung durch konkrete Messung am Roboter NAO abschließen.

### 4.4.1 Fußdruck-Sensoren

Eine naheliegende Methode, das Gleichgewicht eines Roboters zu messen stellen die FSR-Sensoren dar. In Kapitel 4.3.1 wurde darauf eingegangen, dass es unterschiedliche Definitionen und Interpretationen des ZMPs gibt und gezeigt, dass der ZMP identisch mit dem *Center of Pressure* (CoP) ist. Nach der Definition von Sardain und Bessonnet ist der CoP der Punkt, an dem eine resultierende Gesamtkraft angreifen würden, die alle auf die Sohle wirkenden Druckkräfte ersetzt. Der Druck auf die Sohle des Fußes wird durch die Kraftkomponente  $F_z$  in senkrechter Richtung zum Boden erzeugt. Er lässt sich somit mittels den in Kapitel 3.3.2 vorgestellten FSR-Sensoren direkt messen. Wäre

also in jedem infinitesimal kleinen Punkt der Fußsohle ein solcher Sensor angebracht, so könnte die Position des CoP, und somit des ZMPs, direkt aus der gemessenen Verteilung des Sensor-Arrays bestimmt werden. Die Messung des CoP kann durch mathematische Umrechnung aber auch mit einer geringeren Anzahl von Sensoren durchgeführt werden. Solange mindestens drei Sensoren so an der ebenen Fußsohle eines Roboters angebracht werden, dass sie eine Fläche aufspannen, kann die Position des CoP innerhalb dieses Polygons durch lineare Interpolation aus den Kräften der Eckpunkte berechnet werden, vergleiche u. a. [100].

Nach dem Superpositionsprinzip ergibt sich die Position  $\mathbf{p}_{ZMP}$  des ZMP bei N Eckpunkten der Position  $\mathbf{p}_i$ , auf die die Kräfte  $F_{i_z}$  wirken, zu:

$$\mathbf{p}_{ZMP} = \frac{\sum_{i=0}^{N-1} \mathbf{p}_i F_{i_z}}{\sum_{i=0}^{N-1} F_{i_z}}. \quad (4.4.1)$$

Der CoP kann also als gewichtete Summe aus den Messwerten der FSR-Sensoren berechnet werden. Durch diese Betrachtung ist offensichtlich, dass der gemessene CoP nicht die Fläche des FSR-Polygons verlassen kann. Befindet sich der CoP noch innerhalb des Fußes, aber außerhalb des Messbereiches, so wird er also fehlerhaft auf der Kante gemessen. Diese Rechnung zieht bislang nur die *Single-Support*-Phase des Roboters in Betracht. Im Falle der *Double-Support*-Phase kann der resultierende ZMP aber ohne Probleme mittels Superposition aus dem linken und rechten CoP ermittelt werden.

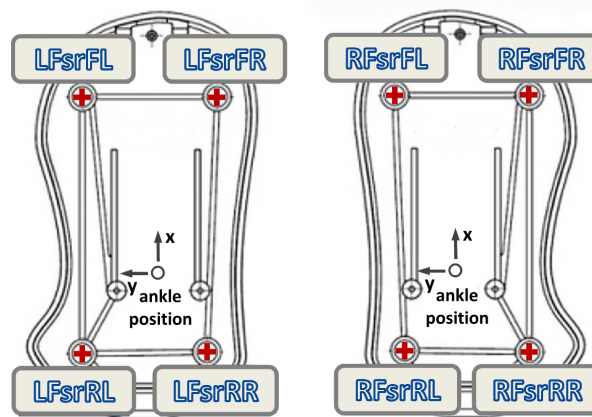


Abbildung 4.4.2: Position der FSR-Sensoren basierend auf [1].

Aldebaran hat den Roboter NAO mit jeweils 4 FSR-Sensoren pro Fuß ausgestattet. Wie Abbildung 4.4.2<sup>2</sup> zeigt, sind diese im Viereck angeordnet, so dass ein großer Teil, aber nicht die gesamte Fläche der Fußsohle des Roboters abgedeckt ist. Aldebaran gibt die Positionen der FSR-Sensoren relativ zum Ursprung des Fußkoordinatensystems (FKS) in der Dokumentation an, so dass diese nicht experimentell bestimmt werden müssen, siehe Tabelle 4.4.1.

<sup>2</sup>Basierend auf Zeichnung aus [1]. Copyright der Vorlage ©2012: Aldebaran-Robotics.

FSR Sensor	x [mm]	y [mm]
LFsrFL	70,25	29,9
LFsrFR	70,25	-23,1
LFsrBL	-30,25	29,9
LFsrBR	-29,65	-19,1
RFsrFL	70,25	23,1
RFsrFR	70,25	-29,9
RFsrBL	-30,25	19,1
RFsrBR	-29,65	-29,9

Tabelle 4.4.1: Positionen der FSR-Sensoren im Fußkoordinatensystem.

Es wurde in Kapitel 3.3.2 bereits darauf eingegangen, welche Quellen zu Abweichungen in der Kraftmessung der FSR-Sensoren führen können, jedoch beschränkte sich die Diskussion auf einen einzelnen Sensor. In die Messung des CoP fließen aber die Messwerte aller vier Sensoren ein und daher muss nun die Fehlerdiskussion fortgesetzt werden. Die Fortpflanzung des Fehlers erfolgt nach Formel 4.4.1 über die gewichtete Summe der Ortsvektoren. Aus diesem Grund ist die Auswirkung des Messfehlers abhängig von der Position des tatsächlichen CoP. Um eine bessere Interpretation der Resultate zu ermöglichen, wird im Folgenden der Einfluss auf den CoP grafisch mit Hilfe einer MATLAB-Simulation durchgeführt, die von Fiegel in Rahmen seiner Bachelorarbeit [32] erstellt wurde. Ohne Beschränkung der Allgemeinheit wird nur der Fall betrachtet, in dem der Roboter auf einem Bein steht. Um den Einfluss der Fehler besser interpretieren zu können, wird die FSR-Polygonfläche durch ein Rechteck angenähert. Weiterhin wird dieses diskretisiert, indem die Länge und Breite in 20 jeweils äquidistante Abschnitte unterteilt werden. Der CoP kann also nicht mehr jede beliebige Position annehmen, sondern wird auf  $21 \cdot 21 = 441$  mögliche beschränkt. Die Evaluierung beschränkt sich auf den statischen Fall, d.h. keine Kräfte außer der Gravitation wirken auf den Roboter. In diesem Fall lässt sich die Summe der Kräfte immer auf die Gewichtskraft des Roboters NAO normieren. Unter dieser Bedingung ergibt sich für jede der 441 festen CoP Positionen jeweils genau ein 4-Tupel aus Kräften, das diesen CoP zur Folge hat. Nun können durch Variation dieser Tupel die Auswirkungen von Messfehlern bestimmter Sensoren auf die ZMP-Berechnung simuliert werden. Dies geschieht durch Veränderung der Kräfte mittels einer 4x4 Matrix, die es erlaubt, Verstärkungen einzelner, sowie gegenseitige Beeinflussung mehrerer Sensoren abzubilden.

Das Resultat der folgenden Simulationen wird jeweils in einem Diagramm als Vektorfeld dargestellt. Blaue Kreise zeigen die korrekte CoP-Position an, die korrespondierenden roten Kreuze den jeweiligen CoP, der durch den Fehler gemessen würde. Um die Zuordnung zu erleichtern werden korrespondierende Kreise und Kreuze durch eine grüne Linie miteinander verbunden, die keinerlei physikalische Bedeutung hat. Die Länge dieser Linie, also der euklidische Abstand der Punkte, wird zur Fehlerabschätzung verwendet. Als Vergleichsmaß dafür, wie gravierend der Einfluss ist, wird zum einen die Varianz über alle Fehler herangezogen. Als Maß der Statistik hat dies bei systematischen Fehlern im Grunde wenig Aussagekraft. Dennoch liefert sie ein Grad für die Streuung der Fehler und kann durch die Art der Auswertung somit als Maß für den Einflussbereich des Fehlers herangezogen werden. Für die Robustheit ist aber in der Praxis nicht

nur die Größe des Fehlerbereichs entscheidend, sondern besonders auch die Schwere einzelner Fehler. Daher wird zum anderen auch der maximale Fehler verglichen. Da dieser aber nicht zwangsläufig zur Instabilität des Systems führen muss, wird dessen Einfluss im Einzelnen diskutiert. Hierbei ist entscheidend, ob Fälle auftreten können, in denen der Roboter instabil ist, obwohl durch den Messfehler der Zustand des Roboters als stabil beurteilt wird.<sup>3</sup> Eine Klassifikation eines stabilen Zustands als instabil ist nicht wünschenswert, jedoch ist der Einfluss auf die Stabilität nicht derart gravierend.

#### Simulation 4.4.1.A - Konstanter Bias

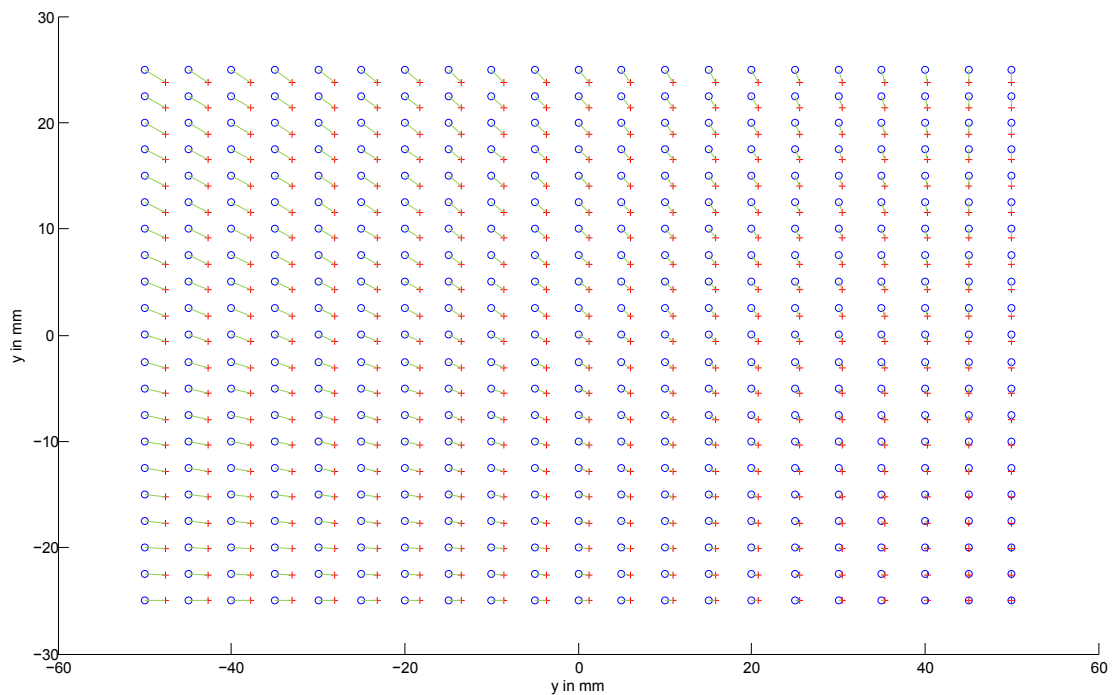


Abbildung 4.4.3: Simulation 4.4.1.A - FSR vorne rechts - konstanter Bias von 100g.

Die erste Simulation untersucht den Einfluss eines konstanten Bias eines Sensors auf den CoP. Hierzu wird die Gewichtskraft einer Masse von 100 g, respektive 200 g und 400 g auf den FSR-Sensor vorne rechts addiert. In der Regel handelt es sich beim Bias um einen Fehler, der den Sensor selbst betrifft. Für dem NAO lässt sich hiermit allerdings auch ein typischer Effekt modellieren, der mechanische Ursachen hat. Da die Fußsohle des NAOs aufgrund der FSR-Sensoren beweglich gelagert ist, kann es passieren, dass sich diese gegen den Fuß leicht verkantet. In dem Fall wird die Messung der FSR-Sensoren mit einem konstanten Druckverlauf überlagert, der den Wert des betroffenen FSR-Sensors um einen konstanten Wert erhöht.<sup>4</sup> Das resultierende Vektorfeld für die

<sup>3</sup>Zu diesem Zweck wird davon ausgegangen, dass die FSR-Sensoren das gesamte *Support-Polygon* des Roboters aufspannen, auch wenn dies in der Realität nicht der Fall ist.

<sup>4</sup>Es sei angemerkt, dass dies eine Vereinfachung darstellt. Durch Übersprechen werden auch die Nachbartsensoren betroffen. Zudem kann das Verkanten aber auch zu einer Entlastung der anderen Sensoren zur Folge haben. Durch die Verhältnisrechnung des CoP heben sich diese Effekte aber teilweise auf, so dass der Effekt sich zumindest qualitativ als Bias modellieren lässt.

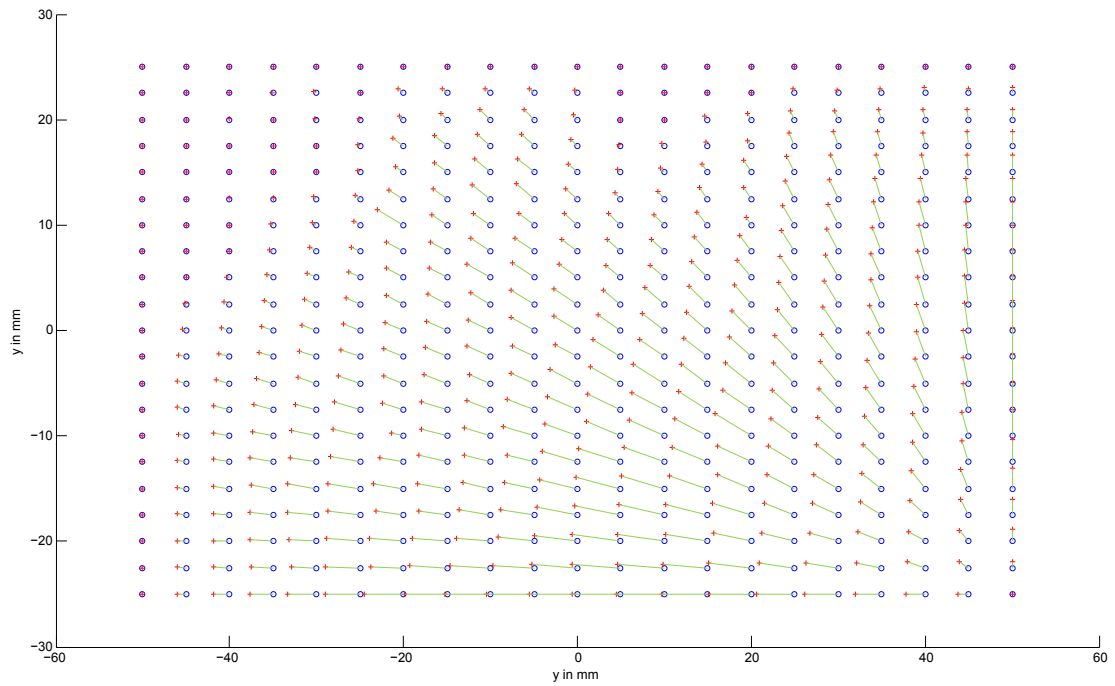
100 g Simulation ist in Abbildung 4.4.3 dargestellt. Es zeigt sich, dass der defekte Sensor als Attraktor für den gemessenen CoP wirkt. Der Einfluss des Fehlers nimmt zu, je weiter der zu messende CoP vom fehlerhaften Sensor entfernt liegt. Bedenkt man, dass durch die Verhältnisbildung nicht der absolute, sondern der relative Wert der Kräfte die CoP-Position bestimmt, ist dies nachvollziehbar. Befindet sich der korrekte CoP also genau auf dem fehlerhaften Sensor, so ist der Messwert der drei anderen Sensoren Null. Folglich hat der Fehler keinen Einfluss mehr. Liegt der reale CoP hingegen genau auf einem der anderen Sensoren, so kann dieser in keinem Fall gemessen werden, da durch den konstanten Bias der dekalibrierte Sensor nicht den Wert Null annehmen kann. Als Folge ist es möglich, dass der ZMP außerhalb des Messbereiches liegt, aber von den FSR-Sensoren fehlerhaft als innerhalb liegend wahrgenommen werden. Ein konstanter Bias kann also zu kritischem Systemverhalten führen und sollte daher nach Möglichkeit durch Kalibrierung ausgeglichen werden. Tabelle 4.4.2 zeigt jeweils die Varianz und den maximalen Fehler über alle Abweichungen für unterschiedliche Gewichte. Es lässt sich erkennen, dass der maximale Fehler nahezu linear mit der Verdopplung der Kraft ansteigt.

Bias	Varianz [mm <sup>2</sup> ]	maximaler Fehler
100 g	2,5406	2,73 mm
200 g	9,6844	5,32 mm
400 g	35,2961	10,16 mm

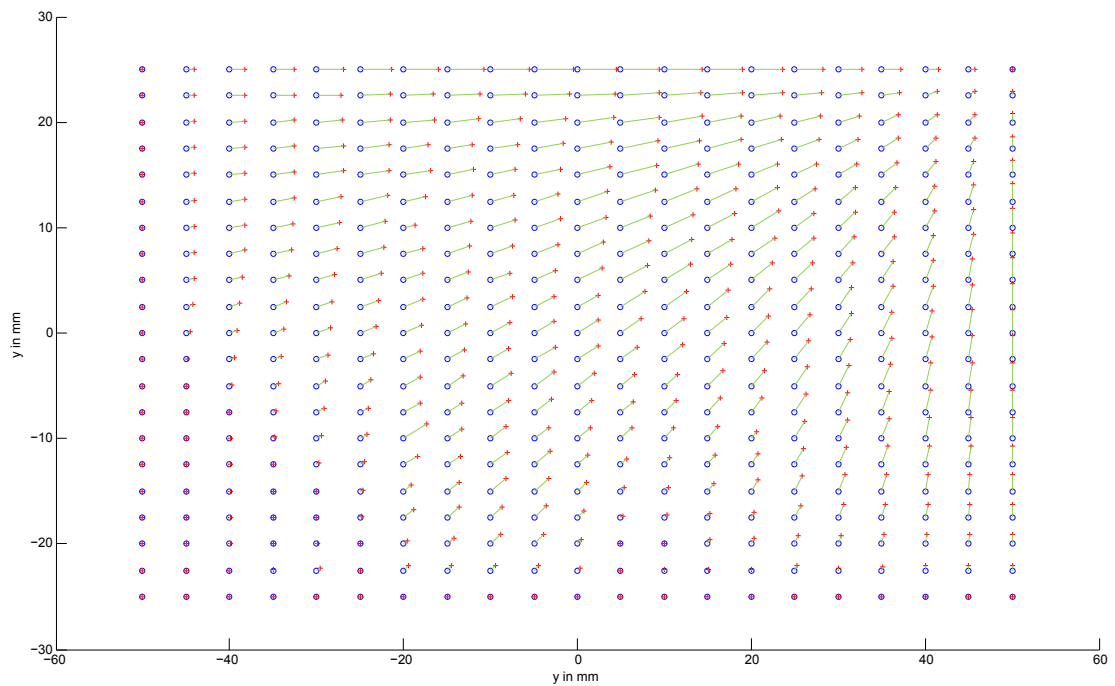
Tabelle 4.4.2: Simulation 4.4.1.A - Fehler für konstanten Bias.

#### Simulation 4.4.1.B - Gainfehler

Die zweite Simulation untersucht den Einfluss eines Gainfehlers in der Messung der FSR-Sensoren. Dieser Fehler tritt bei Drucksensoren in der Praxis häufig auf, da zur Interpolation der Kalibrierungsgeraden nur die Steigung ermittelt wird, vergleiche Kapitel 3.3.2. Auch wenn die Eichung durch wiederholte Messung verbessert werden kann, so führt ein Fehler doch immer zu einer kleineren oder größeren Geradensteigung und somit zu einem Gainfehler. Die Abbildung 4.4.4(a) und (b) zeigen die Ergebnisse der Simulation mit einem Faktor von 0,8 vorne rechts, bzw. 1,2 vorne links. Analog zu Simulation 4.4.1.A ergibt sich ein Vektorfeld, bei dem der CoP entweder bei Verstärkung von dem fehlerhaften Sensor angezogen, oder bei Abschwächung abgestoßen wird. Es ist zu erkennen, dass der Fehler in der Nähe des FSR-Sensors zunächst klein ist, dann größer wird, um bei Annäherung an die anderen Sensoren auf Null abzufallen. Ein ZMP-Messfehler kann also nur im unkritischen Bereich um den Fußmittelpunkt entsteht und ist für die Stabilität nicht signifikant. Dies lässt sich ebenfalls anhand der physikalischen Gegebenheiten erklären - je mehr der CoP sich dem dekalibrierten Sensor annähert, desto geringer werden die Kräfte auf den anderen Sensoren und desto weniger fällt der Fehler bei der Verhältnisbildung ins Gewicht. Umgekehrt gilt, wenn der CoP sich auf einem der anderen Sensoren befindet, befindet sich kein Gewicht auf dem dekalibrierten Sensor. Durch die multiplikative Art des Gainfehlers misst der Sensor dann aber auch Null und die Abweichung beeinflusst die Messung nicht mehr. Der Fehler wird also in einigen Fällen ausgeglichen, was sich auch in der Varianz widerspie-



(a) Vorne rechts - Faktor 0,8.



(b) Vorne links - Faktor 1,2.

Abbildung 4.4.4: Simulation 4.4.1.B - FSR-Gainfehler.

gelt. Befindet sich der CoP im Fußmittelpunkt des Roboters NAO, so wirkt auf jeden der Sensoren eine Gewichtskraft eines 1 Kilogramms, so dass ein Messfehler von 20 Prozent mit einer Abweichung von 200 g bei Simulation 4.4.1.A entsprechen sollte. Ein Vergleich mit Tabelle 4.4.3 zeigt aber, dass die Varianz in diesem Fall deutlich höher ist, als bei einer 20 prozentigen Gain-Abweichung. Abbildung 4.4.5 zeigt das Resultat einer Überlagerung beider Fehler, also Faktor 0,8 vorne rechts und 1,2 vorne links. Dies führt zu einer Varianz die mit der konstanten 200g Abweichung zu vergleichen ist. Tabelle 4.4.3 zeigt, dass dies den maximalen Fehler aber nicht beeinflusst. Diese Fehlerart kann durch eine Kalibrierung der Sensoren behoben werden.

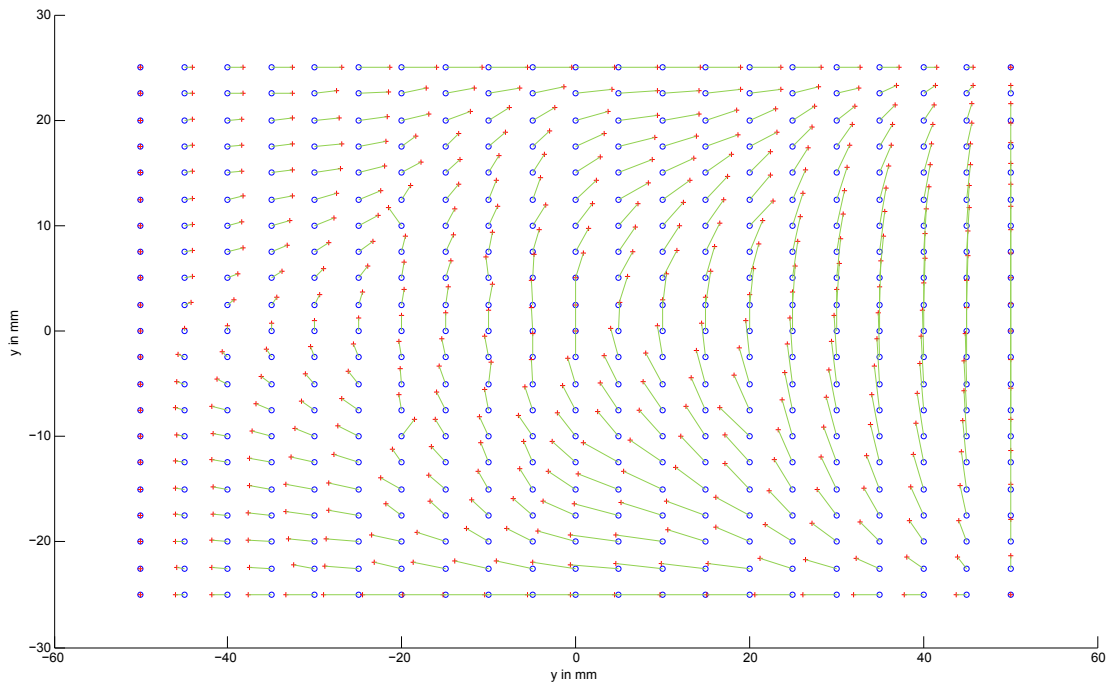


Abbildung 4.4.5: Simulation 4.4.1.B - Kombiniertes FSR-Gainfehler, vorne rechts - Faktor 0,8 und vorne links - Faktor 1,2.

Bias	Varianz[mm <sup>2</sup> ]	maximaler Fehler
0,8	4,7886	4,55 mm
1,2	6,5403	5,59 mm
0,8 vorne rechts + 1,2 vorne links	9,5497	5,59 mm

Tabelle 4.4.3: Simulation 4.4.1.B - Fehler im FSR-Gain.

### Simulation 4.4.1.C - Übersprechen

Durch die Art der Fußaufhängung bedingt, kommt es beim NAO vor, dass bei der Belastung eines Sensors eine mechanische Übertragung der Kraft auf benachbarte Sensoren stattfindet. Es handelt sich demnach um einen systematischen Fehler, der nicht aus einer ungenauen Kalibrierung des Sensors hervorgeht, sondern physikalische Ursachen hat. Dieses Phänomen wird in Simulation 4.4.1.C untersucht, deren Ergebnis für ein



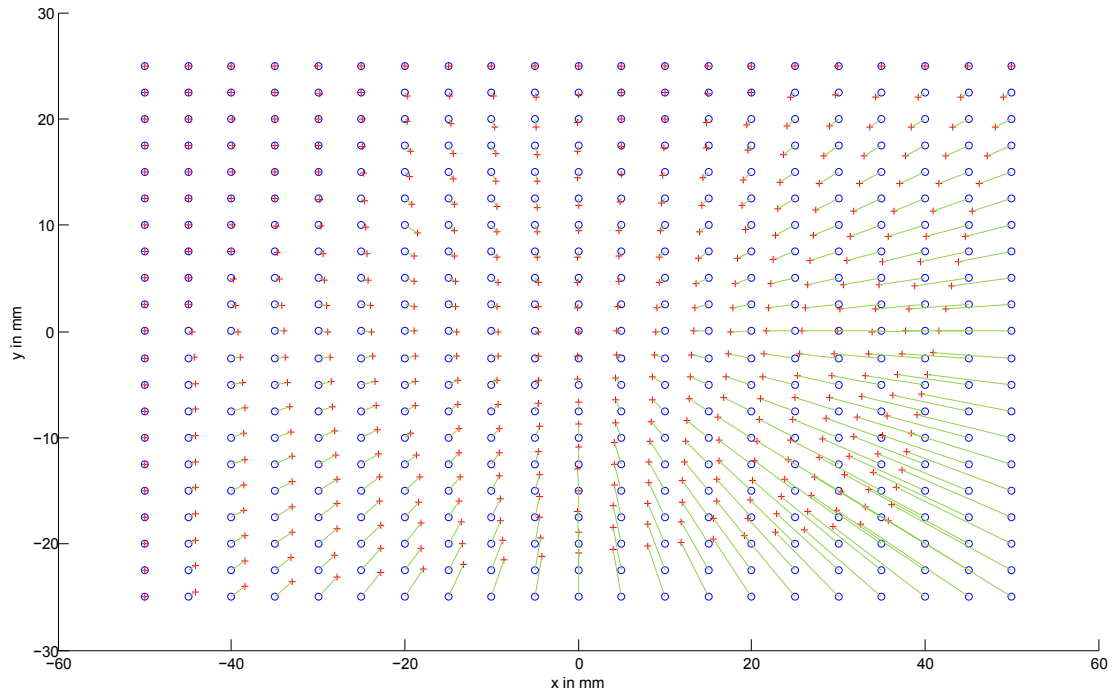


Abbildung 4.4.6: Simulation 4.4.1.C - Fehler durch Übersprechen von vorne rechts mit 0,2 auf vorne links und hinten rechts.

Übersprechen des Sensors vorne rechts mit je 20 Prozent auf die Sensoren vorne links und hinten rechts in Abbildung 4.4.6 zu finden ist. Der übersprechende Drucksensor wirkt in seinem Nahbereich als Repellor für den gemessenen CoP. Je weiter sich der tatsächliche CoP der Mitte annähert, desto kleiner wird der Fehler, um dann im Fußmittelpunkt den Wert Null anzunehmen und mit Entfernung vom Mittelpunkt wieder anzusteigen. Analog zu den vorherigen Simulationen verschwindet der Fehler bei Annäherung an die anderen Sensoren. Dieses Verhalten erscheint zunächst komplexer, lässt sich aber dennoch logisch erklären. Je näher der CoP am übersprechenden Sensor liegt, desto größer ist der absolute, und somit auch relative, Fehler der beiden anderen Sensoren. Daher wird der Fehler dieser beiden Sensoren kontinuierlich kleiner, je weiter der reale CoP sich vom Sensor entfernt. Bei Annäherung an den Mittelpunkt gleicht sich der Übersprechenfehler der beiden anderen Sensoren immer mehr aus. Dies führt logischer Weise dazu, dass der Fehler im Mittelpunkt den Wert Null erreicht. Tabelle 4.4.4 referenziert die Fehlerabweichungen der Untersuchung. Die Varianz eines Übersprechens von 20 Prozent ist merklich größer als bei den bisherigen Simulationen. Zudem liegen die großen Abweichungen des gemessenen vom realen CoP am kritischen Rand des Fußes und unterschätzen die Entfernung zu diesem. Der gemessene CoP kann also nie die Position des übersprechenden Sensors erreichen. Als Folge kann der ZMP sich außerhalb der FSR-Fläche befinden, der gemessene CoP aber noch eindeutig innerhalb liegen. Der Roboter befände sich in diesem Fall in einer instabilen Kippbewegung, obwohl er seine Lage als stabil ansieht. Der Fehler ist demnach kritisch für den Zustand des Systems. Ist das Maß des Übersprechens bekannt, kann eine Ausgleichsrechnung die Messung korrigieren, indem die Werte der betroffenen Sensoren entsprechend dem Messwert des übersprechenden Sensors verringert werden. Dieser Korrekturfaktor muss jedoch expe-

rimentell bestimmt werden und unterliegt folglich anderen Fehlerquellen, beispielsweise Rauschen. Deren Einfluss auf das System würde also weiter zunehmen. Problematisch beim NAO ist vor allem die Änderung des Übersprechverhaltens während des Betriebs, da durch Belastung der Fußaufhängung deren Übertragungseigenschaften temporär variieren können. Eine Korrektur durch Kalibrierung erweist sich somit in der Realität als nicht praktikabel.

Übersprechen	Varianz[mm <sup>2</sup> ]	maximaler Fehler
20 Prozent	16,1684	15.98 mm

Tabelle 4.4.4: Simulation 4.4.1.C - Fehler durch Übersprechen.

#### Simulation 4.4.1.D - Sensorrauschen

Zusätzlich zu systematischen Fehlern tritt durch Rauschen in der Praxis bei elektronischen Bauteilen immer auch eine statistische Komponente im Fehler auf. Um dies zu untersuchen wird ein Gauß'scher Fehler zu jedem der Sensoren addiert und so weißes Rauschen simuliert. Zur Auswertung werden 50 Simulationen durchgeführt jeweils mit einer Standardabweichung von 75 g, welche in Kapitel 3.3.2 als charakteristisch für die A201 Sensoren ermittelt wurde. Um einen Eindruck von der Art der entstehenden Fehlerfelder zu bekommen zeigt Abbildung 4.4.7 exemplarisch eine dieser Simulationen. Diese Felder folgen keiner erkennbaren Struktur, sondern sind als deterministisch chaotisch zu klassifizieren. Aber auch in diesem Fall gilt, dass die Fehler zu den einzelnen Sensoren hin kleiner werden und die größeren Fehler im unkritischen Bereich des Fußmittelpunktes auftreten. Dennoch kann es durch einen Fehler in einem unbelasteten Sensor dazu kommen, dass der CoP fehlerhaft als innerhalb des Fußes klassifiziert wird. Es ist durch die statistische Natur des Fehlers aber unwahrscheinlich, dass dies über eine Folge von Messungen geschieht. Somit braucht dieses Verhalten auch nicht als kritisch eingestuft zu werden. Eine Fehlerauswertung über alle 50 Simulationen ist in Tabelle 4.4.5 zu finden. Die Ergebnisse sind nicht direkt mit den anderen Simulationen zu vergleichen, da in diesem Fall alle 4 Sensoren gestört sind. Jedoch weist auch die Varianz aller Messungen ebenso wie das abgebildete Beispiel darauf hin, dass dieser Fehler in der Praxis nicht gravierend ist, da sich viele Einflüsse im Mittel aufheben.

$\sigma$	Varianz[mm <sup>2</sup> ]	maximaler Fehler
75 g	3,0816	9,67 mm

Tabelle 4.4.5: Simulation 4.4.1.D - Fehler durch Rauschen mit Standardabweichung 75g.

#### Simulation 4.4.1.E - Sensorausfall

Abschließend wird der Fall eines totalen Sensorausfalls betrachtet, der durch einen Kabelbruch oder Hardwaredefekt auftreten kann. Dies stellt einen Spezialfall des Bias dar, dem in der Praxis besondere Relevanz zukommt. Aus diesem Grund wird er nochmals gesondert diskutiert. Abbildung 4.4.8 illustriert das Resultat für einen defekte Sensor,

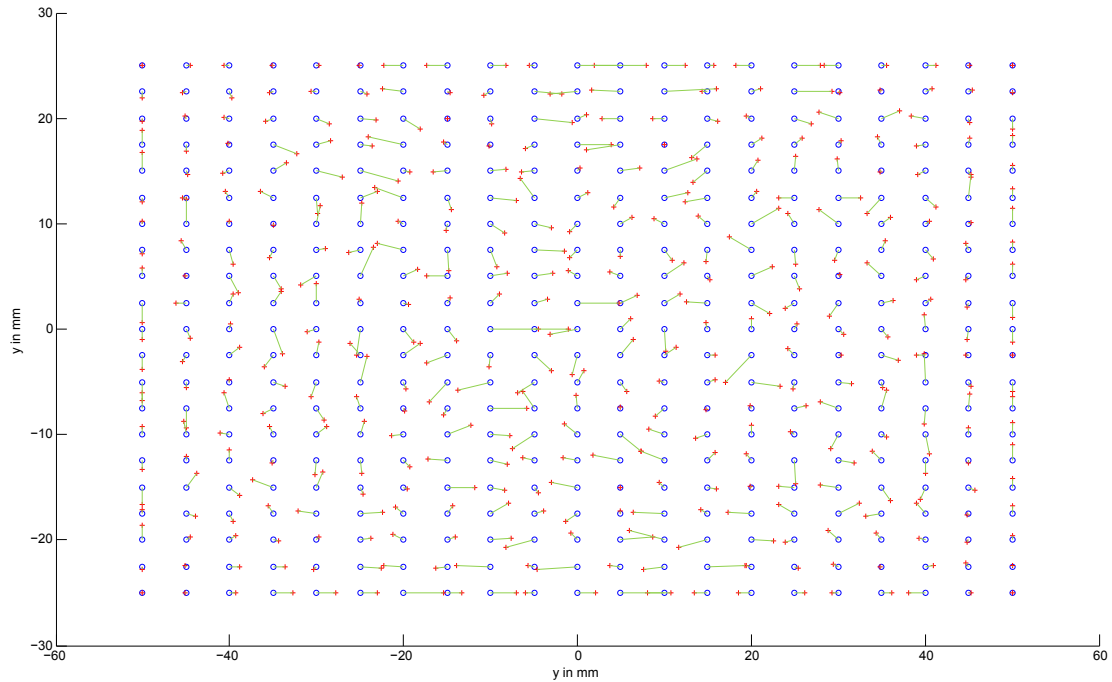


Abbildung 4.4.7: Simulation 4.4.1.D - Fehler durch Rauschen mit Standardabweichung 75g.

der nur noch den Wert Null annimmt. Es ist offensichtlich, dass die Messung nicht als Grundlage für eine Regelung eingesetzt werden kann, da nur noch in einem kleinen Bereich der CoP korrekt wahrgenommen wird. Im schlimmsten Fall wird ein realer CoP sogar in der Nähe der rechten Fußkante, nahe der linken Fußkante gemessen. In diesem Fall würde eine Korrektur des gemessenen CoP dazu führen, dass der reale CoP noch näher an die Kante verlagert wird. Die Regelung würde also das System destabilisieren. Aus diesem Grund wird auf eine Betrachtung der Varianz verzichtet, da der totale Ausfall eines Sensors in jedem Fall zu einem kritischen Systemzustand führt.

### Evaluation der FSR-Sensoren

Ziel der Simulationen ist es die Robustheit der FSR-Sensoren zur Messung des ZMPs zu evaluieren. Hierzu wurden die Auswirkung der in der Praxis am häufigsten auftretenden Fehler untersucht. Allen Simulationsergebnissen ist gemein, dass sich der Fehler immer nur auf einen gewissen Bereich des Fußes auswirkt, da zwischen allen vier FSR-Sensoren interpoliert wird. Der Fehler wird hierdurch gedämpft und beeinflusst den CoP nicht in seiner vollen Stärke. Zudem können sich Fehler einzelner Sensoren lokal ausgleichen. Diese Art der ZMP-Messung erweist sich demnach als sehr robust gegen Störungen und Dekalibrierung der Sensoren. Da zudem die Ergebnisse des Kapitels 3.3.2 zeigen, dass die Latenzzeit der Sensoren gering ausfällt, können diese als praxistauglich beurteilt werden. Lediglich das Übersprechen stellt durch seine Variabilität ein relevantes Problem für den NAO dar. Allerdings muss festgehalten werden, dass die Messung des ZMP über Drucksensoren die Gefahr eines *Single-Point-of-Failure* bietet. Sollte ein Sensor

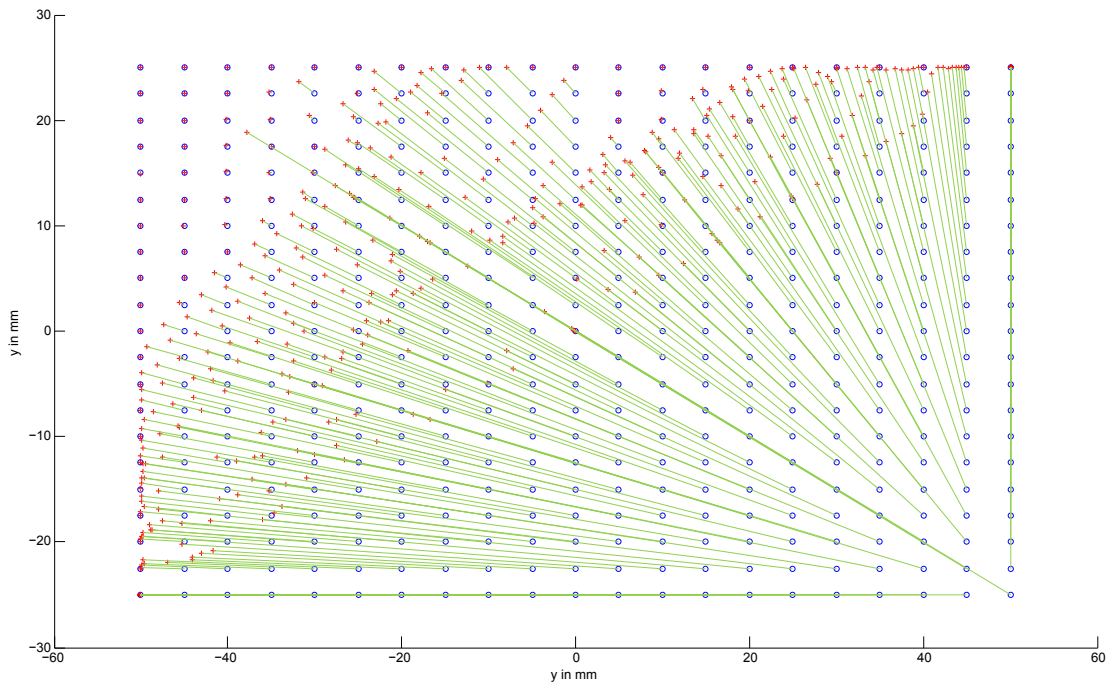


Abbildung 4.4.8: Simulation 4.4.1.E - Ausfall FSR vorne rechts.

ausfallen, so ist es nicht mehr möglich die Information der übrigen Sensoren für eine Regelung zu verwenden. Dieses Problem wird durch die physikalische Konstruktion des Fußes verstärkt. Durch die Art der Aufhängung kann es leichter passieren, dass sich die beiden Fußteile gegeneinander verschieben und einen Sensor einklemmen. Folglich kann in der Praxis ein temporärer Ausfall eines Sensors durchaus auftreten und zu einem instabilen System führen. Hierbei handelt es sich aber um ein NAO-spezifisches Problem und könnte durch eine bessere Konstruktion des Fußes behoben werden.

#### 4.4.2 Inertialsensoren

Der Mensch nimmt sein Gleichgewicht mit Hilfe des Vestibularapparates wahr. Dies sind Organe, die im Innenohr zu finden sind und auf Beschleunigungen und Drehbewegungen reagieren. Somit ist der Mensch in der Lage, implizit die Richtung der Erdbeschleunigung zu bestimmen und so ein Gefühl für sein Gleichgewicht zu bekommen. Dieses Verhalten kann von einem humanoiden Roboter mit Hilfe der Inertialsensoreinheit imitiert werden. Im statischen Fall ist dies unmittelbar verständlich. Befindet sich der Roboter nicht in Bewegung, so wirkt auf den Beschleunigungssensor lediglich die Erdanziehungskraft, deren Richtung somit direkt gemessen werden kann. Aus dieser Information ist die Orientierung des Roboters im Raum ableitbar und demnach auch die Projektion des CoM auf den Boden. Per Definition ist der Roboter statisch stabil, solange diese sich innerhalb des *Support-Polygons* befindet. Im Falle der dynamischen Stabilität lässt sich dies nicht so einfach abbilden, da nun noch weitere externe Kräfte auf den Roboter wirken. Um diese zu beachten, kann das sogenannte *Cart-Table-Modell* invertiert werden, das erstmals von Kajita et al. [55] zur Generierung von dynamischen

Laufbewegungen veröffentlicht wurde. Kajita schlug vor, zur Abbildung der Dynamik eines humanoiden Roboters das abstrakte Modell eines masselosen Tisches zu verwenden. Auf diesem befindet sich ein Wagen mit der Masse  $M$ , der den CoM des Systems repräsentiert. Eine schematische Abbildung des Systems ist Abbildung 4.4.9 zu entnehmen. Solange sich im statischen Fall die Projektion des Massenschwerpunktes des Wagens innerhalb der *Support*-Fläche des Tisches befindet, ist der Tisch stabil. Bewegt sich der Wagen hingegen soweit zur Tischkante, dass sich dessen Projektion außerhalb der Auflagefläche befindet, so wirkt auf den Punkt  $p$  ein Drehmoment  $\mathbf{M}$  mit der Komponente

$$M_x = mg(x - p_x), \quad (4.4.2)$$

in  $x$ -Richtung. Wird der Wagen nun einer gleichmäßigen Beschleunigung  $\ddot{x}$  in  $x$ -Richtung ausgesetzt, ohne sich relativ zum Tisch zu bewegen, so kompensiert die resultierende Kraft das Drehmoment,

$$M_x = mg(x - p_x) - m\ddot{x}z_c, \quad (4.4.3)$$

wobei  $z_c$  die Höhe des CoM bezeichnet. Für den ZMP muss per Definition  $M_x = 0$  gelten. Stellt man analoge Betrachtungen für ein Modell in  $y$ -Richtung auf, so erhält man für die  $x$ - und  $y$ -Koordinaten des Punktes  $p$  folgende Gleichungen:

$$p_x = x - \frac{z_c}{g}\ddot{x}, \quad (4.4.4)$$

$$p_y = y - \frac{z_c}{g}\ddot{y}. \quad (4.4.5)$$

Diese beschreiben die Position des ZMP relativ zum Koordinaten-Ursprung in Abhängigkeit von der Position  $x$  und  $y$  des CoM und dessen Beschleunigungen  $\ddot{x}$  und  $\ddot{y}$ . Somit besteht ein direkter mathematischer Zusammenhang zwischen der durch die Inertialsensoren gemessenen Beschleunigung und der Position des ZMPs auf dem Boden.

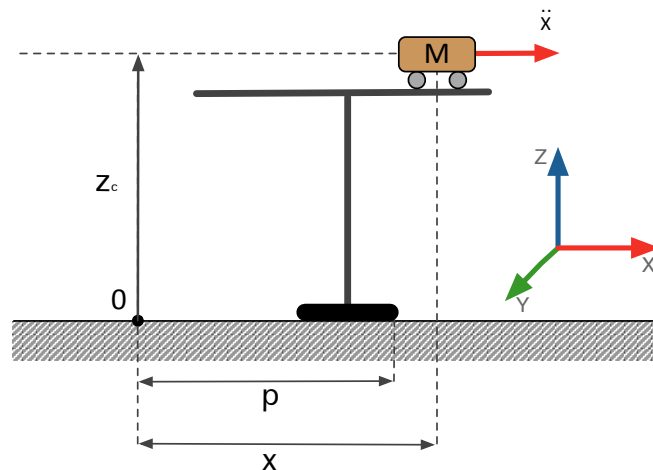


Abbildung 4.4.9: Schematische Darstellung des *Cart-Table*-Modells.

Betrachtet man dieses Modell eingehender, fällt auf, dass Kajita diese ZMP-Formeln unter der Bedingung aufstellt, dass der Tisch und somit der Roboter senkrecht zum Boden steht, der Vektor der Erdbeschleunigung also parallel zur  $z$ -Achse des RzKS

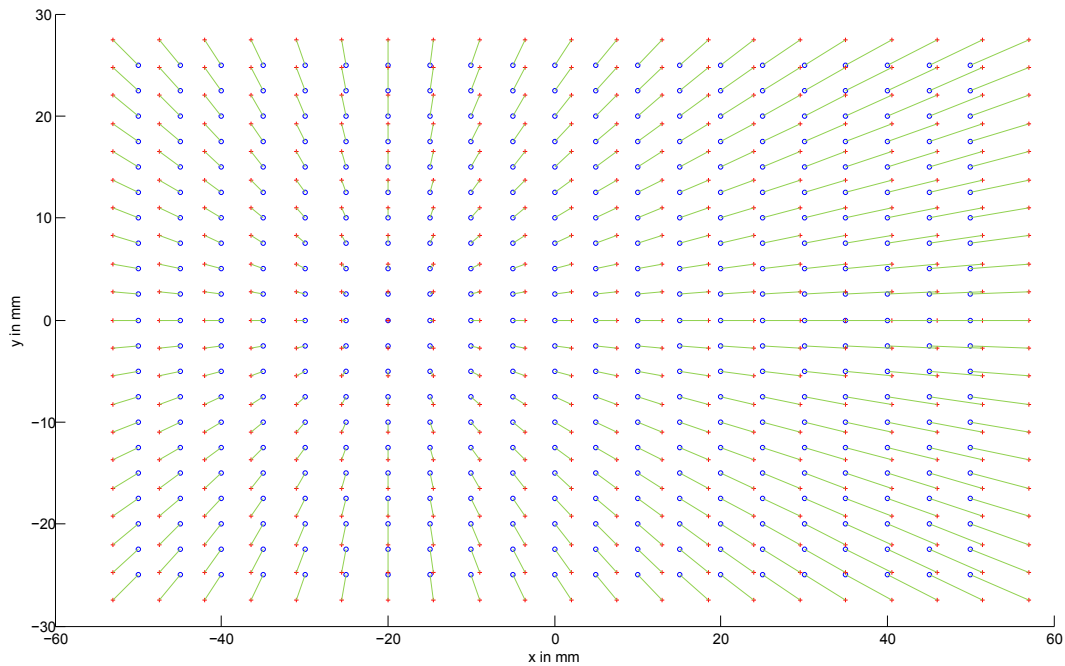
verläuft. Sollte dies nicht gelten, müssen gemessene Beschleunigungen in ein bodenparalleles Koordinatensystem transformiert werden. Diese Transformation ist eine reine rotatorische Rücktransformation und daher theoretisch unproblematisch. Allerdings erfordert die Berechnung Wissen über die aktuelle Orientierung des Oberkörpers zum Boden. Da diese aus den Gelenkwinkeln berechnet werden muss, ergibt sich eine Beeinflussung der Messung durch den Kinematikfehler, der bereits in Kapitel 4.1.1 eingehend diskutiert wurde. Eine Berechnung der Oberkörper-Orientierung aus der Messung der Beschleunigungssensoren ist in diesem Fall verständlicher Weise nicht zielführend.

Um die Praxistauglichkeit der ZMP-Messung mittels des Beschleunigungssensor beurteilen zu können wird analog zu den FSR-Sensoren eine Betrachtung der Fortpflanzung der in Kapitel 3.3.1 betrachteten Sensorfehler durchgeführt. Erneut findet die Untersuchung mittels Fiegels MATLAB-Simulation statt, die den Messbereich in  $21 \times 21$  Messpunkte diskretisiert. Da die Position des ZMPs bei fester Schwerpunkthöhe aber sowohl durch eine Veränderung der CoM-Position, als auch dessen Beschleunigung erreicht werden kann, ist eine eindeutige Zuordnung der Punkte zu einem Systemzustand nicht möglich. Daher wird für die folgenden Untersuchungen die Position des projizierten Schwerpunkts (pCoM) fix im Ursprung des FKS gewählt und für jeden der Simulationspunkte die entsprechende Beschleunigung berechnet. Auch wenn die ZMP-Bestimmung durch den Beschleunigungssensor den Vorteil gegenüber den FSR-Sensoren bietet, nicht lediglich den ZMP in der gesamten Fußfläche des Roboters, sondern auch den FZMP messen zu können, wird hiervon zu Gunsten der Vergleichbarkeit Abstand genommen und nur die FSR-Region des Fußes untersucht. Für alle Simulationen wurde vom *Worst Case* des aufrecht stehenden Roboters ausgegangen, was einer CoM-Höhe von rund  $z_c = 330 \text{ mm}$  entspricht. Aus Betrachtung der Formeln (4.4.4) und (4.4.5) geht hervor, dass die ZMP-Berechnung von den Variablen  $x$ ,  $\ddot{x}$  und  $z_c$  abhängig ist. Diese stellen also die möglichen Fehlerquellen für die Messung dar und müssen auf ihren Einfluss hin untersucht werden. Wie bei der FSR-Simulation zeigen in den folgenden Abbildungen blaue Kreise die reale ZMP-Position und rote Kreuze die durch die Störung verschobenen ZMPs. Grüne Linien verdeutlichen lediglich die Zuordnung der Punkte.

### Simulation 4.4.2.A - Fehler in der Beschleunigung

Die offensichtlichste Fehlerquelle der ZMP-Messung mittels Beschleunigungssensoren sind Fehler in der Messung der Beschleunigung selbst. Der Messfehler des Sensors wird laut Aldebaran mit rund 1 Prozent angegeben und die vorangehenden Untersuchungen im Kapitel 3.3.1 konnten eine geringe Fehleranfälligkeit bestätigen. Es gibt aber außer dem reinen Messfehler noch eine weitere Quelle für Abweichungen in der Beschleunigung. Wie beschrieben geht die von Kajita hergeleitete Formel davon aus, dass die Beschleunigung achsparallel zum Boden verläuft - sie muss also gegebenenfalls transformiert werden. Grundlage dieser Transformation ist die Orientierung des Oberkörpers, für die in Kapitel 4.1.1 bereits ein Fehler von rund 5 Grad aufgrund des Kinematikfehler hergeleitet wurde. Während der Bewegung addiert sich hierzu ein weiterer Fehler durch eine phasenverschobene Messung und den schwankenden Gang des Roboters. Die-

ser wird aus Erfahrung mit weiteren rund 20 Grad abgeschätzt<sup>5</sup>. Nach  $\cos 25^\circ = 0.906$  resultiert der Fehler in Summe zu rund 10 Prozent in der Messung. Die Auswertung der Simulation eines solchen Fehlers ist in Abbildung 4.4.10 für den Fall einer Überschätzung um 10 Prozent dargestellt. Bei genauerer Betrachtung fällt auf, dass der Fehler in der ZMP-Messung mit der Entfernung vom pCoM zunimmt. Da der Fall  $ZMP = pCoM$  den statischen Fall ohne Beschleunigung darstellt, verwundert diese Tatsache nicht. Die Beschleunigung, und folglich auch der prozentuale Fehler, nimmt erst bei Abstand vom pCoM zu. Aus diesem Grund ist der Fehler am Rande des Fußes jeweils am größten. Wenn der Messwert überschätzt wird, hat dies keinen kritischen Einfluss, es werden lediglich Positionen nahe des Randes bereits als instabil<sup>6</sup> klassifiziert, obwohl sie dies nicht sind. Bei einer Unterschätzung entsteht hingegen ein kritischer Bereich am Rand des Fußes. Die statistische Auswertung beider Fälle ergibt dieselben Werte und ist in Tabelle 4.4.6 zu finden. Sie weist sowohl eine geringe Streuung als auch einen geringen maximalen Fehler auf. Angesichts der Tatsache, dass der maximale Fehler für einen Extremfall gewählt wurde, zeigt sich, dass ein Gain des Beschleunigungssensors kaum einen Einfluss auf die ZMP-Messung hat.

Abbildung 4.4.10: Simulation 4.4.2.A - Fehler in  $\ddot{x}$  von +10 Prozent.

$\sigma$	Varianz[mm <sup>2</sup> ]	maximaler Fehler
± 10 Prozent	15,4583	7,43 mm

Tabelle 4.4.6: Simulation 4.4.2.A - Fehler in  $\ddot{x}$  von 10 Prozent.

<sup>5</sup>Diese Abschätzung stellt eine signifikante Überschätzung des normalen Orientierungsfehlers dar, kann allerdings durch Aufschwngen des Oberkörpers in Extremsituationen durchaus kurzzeitig vorkommen.

<sup>6</sup>Da die Simulation sich auf den Bereich der FSR-Sensoren beschränkt, ist auch der Randbereich noch stabil. Am Rande des Fußes würde dieses Problem jedoch genau so auftreten.

**Simulation 4.4.2.B - Fehler in der  $x/y$ -Position des CoM**

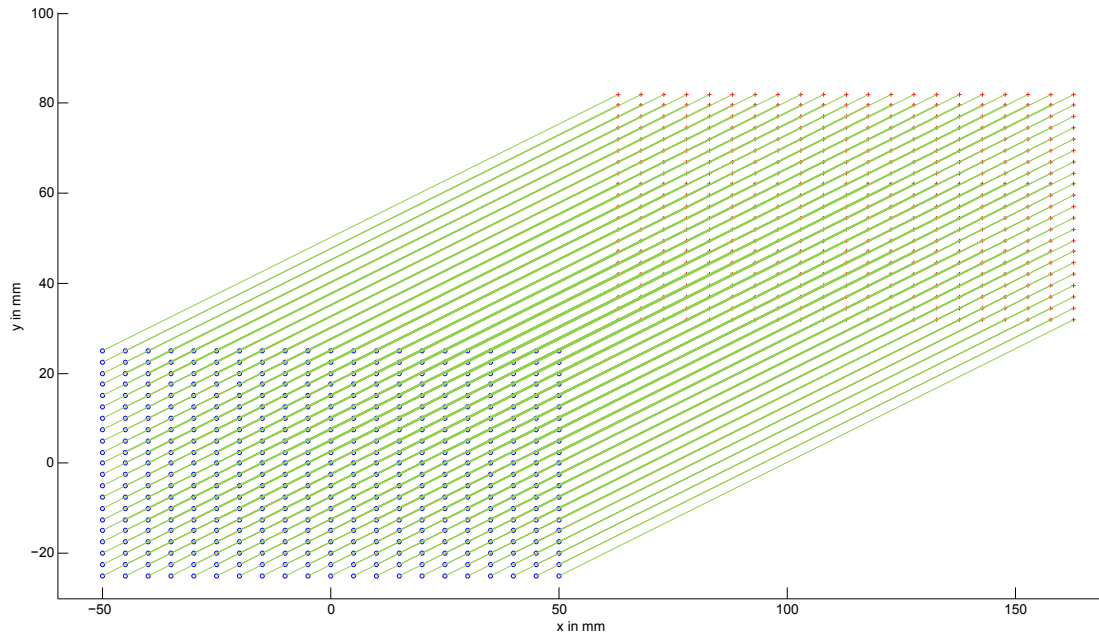
Zur Berechnung des ZMPs wird die  $x$ -, bzw.  $y$ -Position des CoM benötigt. Auf deren Bestimmung im RzKS ging bereits Kapitel 4.2 genauer ein. Dort wurde argumentiert, warum der Fehler der CoM-Berechnung klein gegenüber dem Fehler der Transformation der CoM-Position aus dem RzKS ins WKS ist. Analog lässt sich dies für das FKS des Roboters darlegen, indem  $x$ , bzw.  $y$  berechnet wird und so kann für diesen Fehler die Abschätzung aus Kapitel 4.1.1 herangezogen werden. Jedoch addiert sich erneut im dynamischen Fall ein zusätzlicher Fehler durch die Fehleinschätzung der Oberkörper-Orientierung. Im Gegensatz zu Simulation 4.4.2.A kann diese aber nicht mit einer prozentualen Abweichung angegeben werden, da die Größen  $x$ , bzw.  $y$ , von der willkürlichen Wahl des Ursprungs des Referenz-Koordinatensystems abhängen. Im schlimmsten Fall entsteht erneut ein Orientierungsfehler von  $25^\circ$ . Dieser resultiert jedoch aus Fehlern entlang der kinematischen Kette, so dass er sich nicht über die gesamte Höhe auf die CoM-Position auswirkt. Eine detaillierte Betrachtung übersteigt den Rahmen dieser Analyse, daher wird davon ausgegangen, dass ein summierter Fehler von  $20^\circ$  in  $x$ -Richtung eine gute Abschätzung für den *Worst Case* angibt. Unter dieser Annahme stellte sich ein zusätzlicher Fehler in  $x$  von  $z_c \cdot \sin 20^\circ = 112,9$  mm ein. Durch den Aufbau des Roboters und das fehlende *KneeRoll*-Gelenk wird der Fehler in  $y$ -Richtung geringer ausfallen, so dass er für die Simulation halbiert wird. Wie kritisch die Auswirkung ist, verdeutlicht Abbildung 4.4.11. Nicht ein ZMP wird als innerhalb der Fußfläche liegend gemessen. Zwar stellen diese den unkritischeren Fall dar, in welchem ein stabiler Roboter als instabil beurteilt wird, jedoch ist der Fehler dermaßen gravierend, dass eine Korrektur zwangsläufig zu einer Instabilität des Systems führt. Auch dies stellt einen selten auftretenden *Worst Case* dar. Doch die Ergebnisse demonstrieren, wie entscheidend eine exakte Positionierung des CoM für das Verfahren ist.

$\sigma$	MSE[mm <sup>2</sup> ]	maximaler Fehler
$x = 112,9$ mm, $y = 56,5$ mm	15994	126.47 mm

Tabelle 4.4.7: Simulation 4.4.2.B - Fehler in der  $x/y$ -Position des CoM.**Simulation 4.4.2.C - Fehler in der Höhe  $z_c$** 

Der letzte zu untersuchende Fehlereinfluss ist eine Fehleinschätzung der Höhe  $z_c$  des CoM. In Kapitel 4.1.1 konnte gezeigt werden, dass diese zwar ebenfalls durch den Kinematikfehler beeinflusst wird, aber in deutlich geringerem Maße von rund 0,5 mm. Dies entspricht einem Fehler in der Höhe  $z_c$  von etwas über einem Prozent. Auch in diesem Fall kann der Fehler aber noch durch eine weitere Fehleinschätzung der Orientierung des Oberkörpers vergrößert werden, wodurch bei einer Annahme von  $20^\circ$  ein weiterer Fehler von  $z_c - z_c \cdot \cos 20^\circ = 19,9$  mm entstehen würde. Somit wird der Fehler für die Simulation in Summe mit 20 mm abgeschätzt. Auswertung 4.4.12 zeigt ein ähnliches Bild, wie die vorhergehenden Untersuchungen des Cart-Table-Modells in Simulation 4.4.2.A. In diesem Fall kann allerdings eine Unterschätzung der Höhe dazu führen, dass der pCoM als Attraktor fungiert und somit kritische Stabilitätsbereiche am Rand entstehen. Es bleibt aber dennoch festzuhalten, dass sich ein Fehler in der Höhe  $z_c$  des



Abbildung 4.4.11: Simulation 4.4.2.B - Fehler in der  $x/y$ -Position des CoM.

CoM nicht so gravierend auswirkt, wie eine Abweichung in der  $x$ - und  $y$ -Position.

$\sigma$	MSE[mm <sup>2</sup> ]	maximaler Fehler
$\pm 10$ Prozent	12,1613	6,76 mm

Tabelle 4.4.8: Simulation 4.4.2.C - Fehler in  $z_c$  von 20 mm.

### Evaluation des Beschleunigungssensors

Ziel der Simulationen ist es die Robustheit der Messung des ZMPs mittels des Beschleunigungssensors zu untersuchen. Die Untersuchungen zeigen, dass auch diese Methode in der Praxis robust gegen Messfehler des Beschleunigungssensors ist. Kritisch kann sich ein Fehler in der Positionsbestimmung des CoM und somit des Beschleunigungssensors relativ zum Support-Polygon auswirken, da dieser linear in den ZMP-Fehler eingeht. Hierdurch können gravierende Fehleinschätzungen der Stabilität entstehen. Somit ist die Messung stark abhängig von der Güte der Lage- und Orientierungsbestimmung. Auch wenn Simulation 4.4.2.B nahelegt, dass dies in der Praxis ein Problem darstellen kann, werden jedoch die folgenden praktischen Laufexperimente in Kapitel 5 zeigen, dass der Einfluss nicht derart signifikant ist. Auch die Messung mittels des Beschleunigungssensors resultiert in einem *Single-Point-Of-Failure*. Jedoch kann ein Ausfall des Beschleunigungssensors leichter detektiert werden als ein FSR-Defekt. Weiterhin ließe sich der Sensor ohne größeren Aufwand redundant auslegen, so dass dies für die Zuverlässigkeit im Einsatz kein Hindernis darstellen würde.

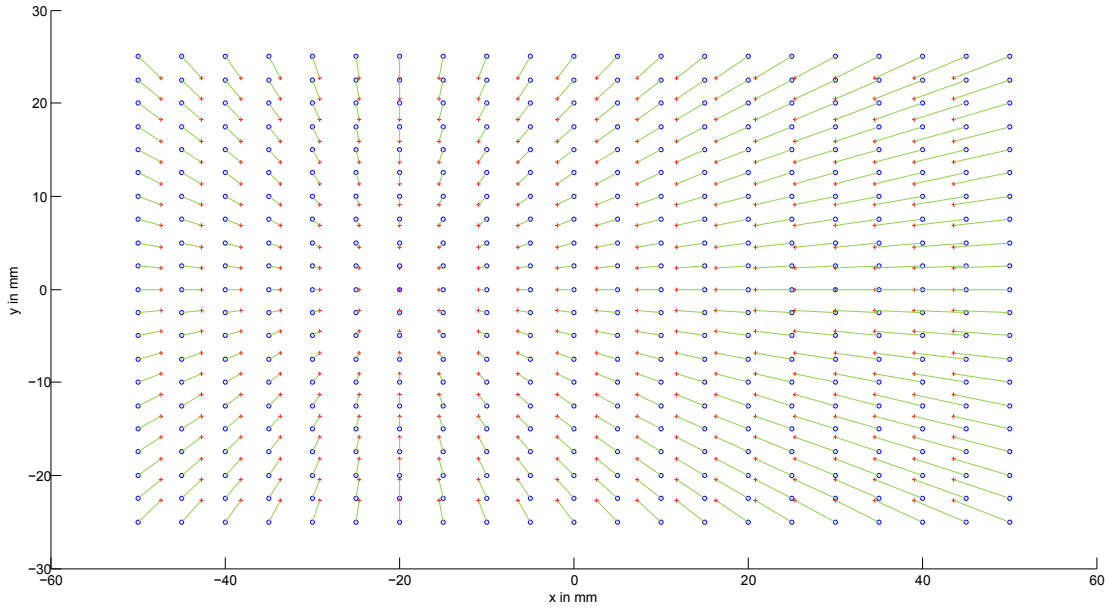


Abbildung 4.4.12: Simulation 4.4.2.C - Fehler in  $z_c$  von -30 mm.

### 4.4.3 Kraft-Drehmoment-Sensoren

Eine unmittelbare Möglichkeit die Position des ZMP zu bestimmen sind Kraft-/ Drehmomentsensoren im Fußgelenk des Roboters. Nach der Definition des CoP folgt, dass der ZMP das Zentrum der Kraftverteilung des Fußes ist. Betrachtet man nun den kontinuierlichen Fall, so ergibt sich aus der diskreten Formel 4.4.1 der FSR-Berechnung für die Koordinaten der CoP-Position  $\mathbf{p}_{CoP}$  :

$$p_x = \frac{\int x F_z(x) dx}{\int F_z(x) dx}, \quad (4.4.6)$$

$$p_y = \frac{\int y F_z(y) dy}{\int F_z(y) dy}. \quad (4.4.7)$$

Stehen Messungen der Kraft  $\mathbf{F}$  und des Drehmomentes  $\mathbf{M}$  zur Verfügung, kann somit der ZMP bestimmt werden durch:

$$p_x = \frac{M_y}{F_z}, \quad (4.4.8)$$

$$p_y = \frac{M_x}{F_z}. \quad (4.4.9)$$

Da solche Messungen aus praktischen Gründen nur im Fußgelenk des Roboters stattfinden können und dieses sich nicht auf der Kontaktfläche mit dem Boden befindet, muss die Rechnung noch um die Fußhöhe  $h$  angepasst werden, vergleiche Abbildung 4.4.13:

$$p_x = \frac{M_y}{F_z} + \frac{h \cdot F_x}{F_z}, \quad (4.4.10)$$

$$p_y = \frac{M_x}{F_z} + \frac{h \cdot F_y}{F_z}. \quad (4.4.11)$$

In der *Double-Support*-Phase kann analog zur Messung des ZMPs durch die FSR-Sensoren der resultierende ZMP mittels Superposition aus den Messungen des linken und des rechten ZMPs berechnet werden:

$$p_x^{DoubleSupport} = \frac{M_y^r \cdot p_x^r + M_y^l \cdot p_x^l}{F_z^r + F_z^l}, \quad (4.4.12)$$

$$p_y^{DoubleSupport} = \frac{M_x^r \cdot p_x^r + M_x^l \cdot p_x^l}{F_z^r + F_z^l}. \quad (4.4.13)$$

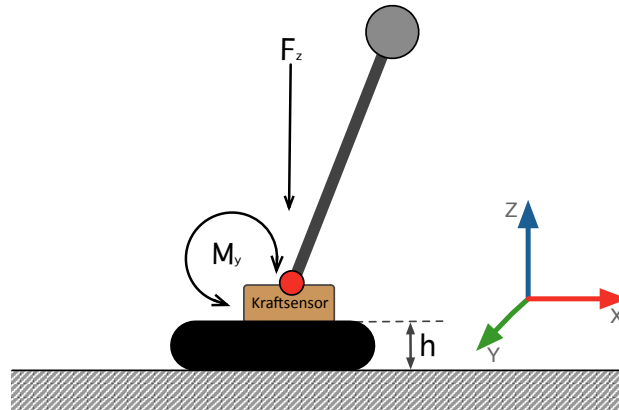


Abbildung 4.4.13: ZMP-Berechnung aus 6-Achs-Kraftsensor.

Als Sensoren zur Kraftmessung können beispielsweise 6-Achs-Kraftsensoren eingesetzt werden, wie sie in Industrierobotern üblich sind [49]. Diese waren aber in der Vergangenheit durch den Fokus ihrer Anwendung auf die Messung verhältnismäßig großer Kräfte ausgelegt und zudem meist nicht für die mobile Anwendung konzipiert. Dies führte über einen langen Zeitraum dazu, dass diese Art von Sensoren nur in Größen zu erwerben waren, die sie für den Einsatz in einem kleinen Roboter, wie dem NAO, unbrauchbar machte. Entwicklungen der letzten Jahre führen aber auch in diesem Bereich zu einer fortschreitenden Miniaturisierung. So entwarf beispielsweise das *Deutsche Zentrum für Luft- und Raumfahrt* einen zweibeinigen Roboter mit Kraftsensoren, indem zwei *KUKA-DLR-Lightweight*-Roboterarme [9] als Beine verwendet wurden, vergleiche [85]. In größeren humanoiden Robotern, wie dem ASIMO oder dem HRP-4 ist der Einsatz eines 6-Achs-Kraftsensors mittlerweile Standard. Und auch im RoboCup finden sich in der *TeenSize Humanoid League* erste erfolgreiche Roboter mit Kraftsensoren, wie der *Robo-Erectus Senior III* (RESr-III) [121]. Forschungen des ARICC zeigen auch erste Erfolge beim Einsatz am deutlich kleineren *Kid-Size* Roboters *Robo-Erectus Junior* (REJr)<sup>7</sup>. Jedoch ist derzeit kein mit dem NAO vergleichbarer, kommerziell erhältlicher Roboter bekannt, der mit solchen Sensoren ausgestattet ist.

Obwohl der Roboter NAO keine solchen Sensoren verbaut hat, kann von dieser Messart dennoch theoretisch Gebrauch gemacht werden. Durch das Getriebe besteht ein Zusammenhang zwischen dem Drehmoment, das die Motoren aufbringen und dem, welches auf die Achse resultiert. Die Motoren des NAO werden von Aldebaran drehmomentbasiert angesteuert, siehe Kapitel 3.2. Dies ist allerdings durch NAOqi gekapselt, so

<sup>7</sup>[http://www.rob-erectus.org/research/dynamic\\_walking.php](http://www.rob-erectus.org/research/dynamic_walking.php), Stand Juni 2012.

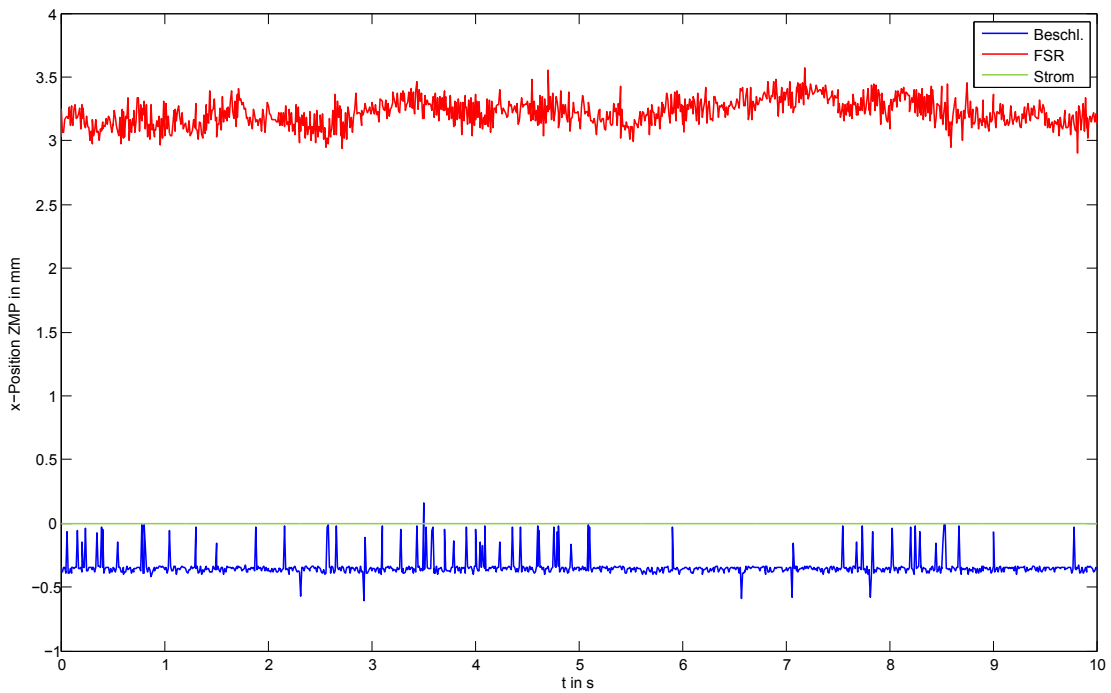
dass das Drehmoment nicht zurückgelesen werden kann. Jedoch besteht über die Motorkonstante ein Zusammenhang zwischen Drehmoment und aufgenommener Leistung, vergleiche Kapitel 3.3.4. Die Konstante ist zwar nicht bekannt, doch führt dies nur zu einem linearen quantitativen Fehler. Da die Stromaufnahme nicht vorzeichenbehaftet ist, kann auf diese Weise nur der Betrag und nicht die Richtung des Drehmoments bestimmt werden. Hierzu ist eine separate Betrachtung der Gelenkpositionen nötig, um die Richtung aus der Bewegung abzuleiten. Die Kraft  $F_z$  senkrecht zum Boden kann in Näherung als Gewichtskraft des Roboters approximiert oder über die FSR-Sensoren aufsummiert werden. Die Untersuchung der Stromaufnahme in Abbildung 3.3.12 legt allerdings bereits nahe, dass die Messwerte der Motoren des NAO durch hohes Rauschen in der Praxis unbrauchbar für die ZMP-Messung sind. Aus diesem Grund wird bewusst auf eine detailliertere Untersuchung der Fehlereinflüsse in der Simulation verzichtet.

### 4.4.4 Experimenteller Vergleich der ZMP-Messverfahren

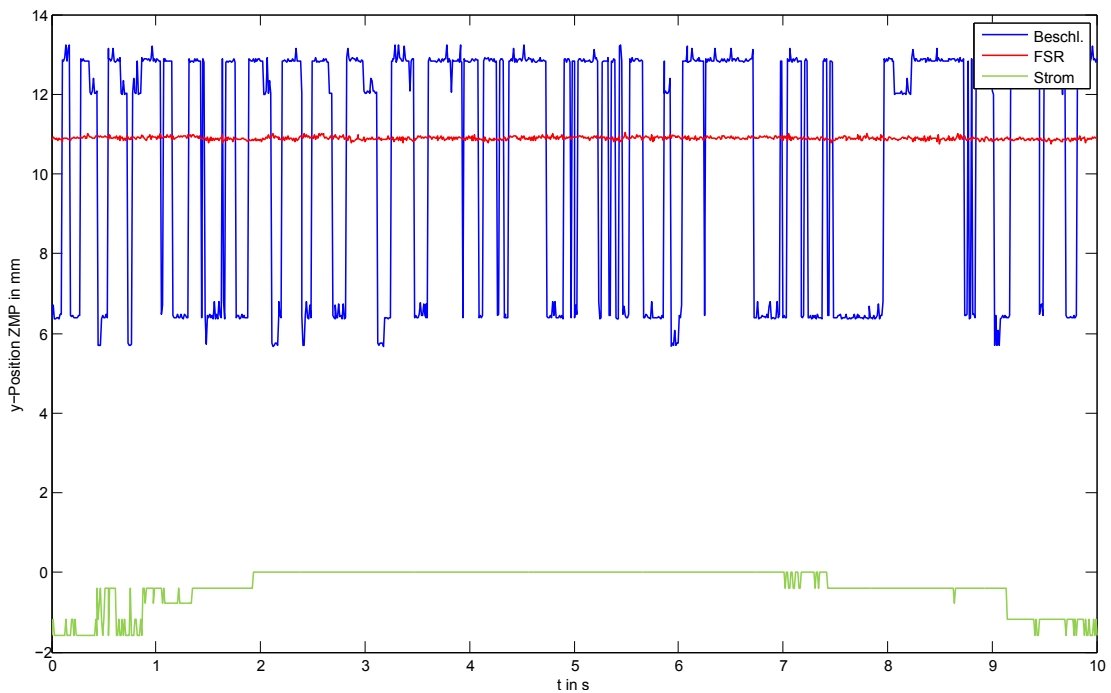
Das vorangehende Kapitel hatte das Ziel die Methoden der ZMP-Bestimmung mittels unterschiedlicher Sensoren zu untersuchen und hierbei insbesondere den Fokus auf deren Robustheit und Fehleranfälligkeit zu legen. Dies beschränkte sich bislang jedoch auf rein theoretische Betrachtungen mittels Simulation. In Kombination mit dem in vorangegangenen Experimenten erworbenen Wissen über die Charakteristik der im NAO verbauten Sensoren kann diese zwar einen guten Einblick in die Praxistauglichkeit des jeweiligen Messverfahrens bieten. Dennoch kann dessen Güte abschließend erst nach einer Untersuchung am realen Roboter beurteilt werden. Dies Unterfangen wirft aber ein Problem auf. Um die Messung der Sensoren bewerten zu können, muss der tatsächliche ZMP des Systems als Referenz bekannt sein, um die Abweichung der Messung bestimmen zu können. Dieser ist aber nicht exakt zu berechnen und soll mittels der zu untersuchenden Sensoren gemessen werden. Somit bleibt nur die Messung des Referenz-ZMPs durch eine externe Quelle. Denkbar wäre dies mit einem System von Druckplatten, wie sie zur Laufanalyse im orthopädischen Bereich verwendet werden. Hierbei handelt es sich um kapazitive Platten, die in der Lage sind über eine ganze Fläche die Druckverteilung aufzuzeichnen. Dies ermöglicht eine exakte Messung des CoP über die Druckverteilung der Füße des NAO und zeichnet so eine Referenz-ZMP-Trajektorie für einen Lauf auf. Ein solches System steht dem IRF jedoch nicht zur Verfügung. Um dennoch eine Analyse der Methoden durchführen zu können, muss daher ein anderer Weg gewählt werden. In einer Serie von Experimenten wird der Roboter NAO jeweils in eine Bewegung gebracht, in der der ungefähre Verlauf des ZMPs logisch nachvollzogen werden kann. So kann anhand von Messdaten, die von Fiegel im Rahmen seiner Bachelorarbeit [32] aufgezeichnet wurden, zumindest der qualitative, wenn auch nicht der quantitative Aspekt der Messung beurteilt werden. Es werden jeweils während eines Experiments gleichzeitig alle Verfahren zur ZMP-Messung mit einer Frequenz von 100 Hz aufgezeichnet, so dass ein relativer Vergleich möglich ist. Somit lassen sich zumindest Unterschiede und Auffälligkeiten zwischen den verschiedenen Methoden aufzeigen und diskutieren.

#### Experiment 4.4.4.A - Balancieren auf einem Bein

Mit dem ersten Experiment soll zunächst gezeigt werden, dass die bisherigen Untersuchungen zur Güte der Sensoren und Robustheit der ZMP-Messverfahren sich auf die eigentliche Messung übertragen lassen. Hierzu bringt sich der Roboter NAO so in Position, dass er nur noch auf dem linken Bein steht und der pCoM möglichst genau im Ursprung des FKS liegt. Da es sich nun um einen Fall von statischer Stabilität handelt, fällt der pCoM mit dem ZMP zusammen und dessen Koordinaten sind per Definition bekannt als  $ZMP_{x,y} = (0, 0)$ . Die Diagramme in Abbildung 4.4.14 zeigen den Verlauf des ZMPs über die Dauer des Experiments von 10 s, wobei die unterschiedliche Skalierung der  $y$ -Achse in (a) und (b) zu beachten ist. Bei Betrachtung der FSR-Kurven fällt auf, dass diese sehr stabil und rauscharm sind. Das Rauschen weist die typische Charakteristik eines weißen Rauschens auf was die Vermutung nahe legt, dass dies aus dem Sensorrauschen selbst resultiert. Beide Messungen zeigen einen konstanten Fehler von rund 3,3 mm, bzw. 11 mm, der systematischer Natur ist. Diese sind vermutlich auf eine nicht ideale Kalibrierung zurückzuführen. Abbildung 4.4.14(a) weist für den ZMP aus den Daten des Beschleunigungssensors ebenfalls eine konstante Messung mit geringem Rauschen auf, dessen Charakteristik sich jedoch von den FSR-Sensoren unterscheidet. Das Rauschen scheint aus einer Überlagerung zweier Komponenten zu bestehen. Dies ist zum einen ein weißes Rauschen mit geringer Amplitude. Diese könnte aus dem Sensor selbst stammen, jedoch zeigten die Untersuchungen in Kapitel 3.3.1 auf, dass statistisches Rauschen im statischen Fall von der Vorfilterung des NAO weitestgehend unterdrückt wird. Daher liegt die Vermutung nahe, dass dieses Rauschen aus der Positionsbestimmung des CoM durch die Gelenkwinkelencoder resultiert. Zum anderen ist eine Komponente erkennbar, die das Rauschen mit Peaks überlagert. Ursache hierfür ist vermutlich das Quantisierungsrauschen der Sensoren. Das Verhalten in  $y$ -Richtung weicht jedoch stark ab, da ein deutliches Rauschen zu erkennen ist. Es scheint zwischen zwei Zuständen zu springen, wird aber von dem Rauschen aus 4.4.14(a) überlagert, wie gut an den Plateaus zu erkennen ist. Das Verhalten ist ähnlich dem Quantisierungsrauschen aus Abbildung 3.3.7, doch unter Beachtung des geringen Fehlereinflusses der Sensordaten aus Simulation 4.4.2.B wäre der Effekt zu groß. Zudem müsste der Effekt im selbem Maße auch in  $x$ -Richtung sichtbar sein. Dies lässt den Schluss zu, dass der Effekt aus der Asymmetrie der Pose resultiert, da der Oberkörper des Roboters um die  $x$ -Achse rotiert ist. Um die Stabilität zu halten, kippt das Gelenk *HipRoll* den Oberkörper nach links und der Roboter streckt den linken Arm und das rechte Bein aus, um die Balance zu verbessern. Als Resultat ist das Drehmoment, dass das Gelenk *HipRoll* halten muss, recht hoch, was die Schlussfolgerung nahe legt, dass der Regler dieses Motors bei kleinen Störungen anfängt zu oszillieren. Als Folge würde diese Oszillation den Oberkörper in  $y$ -Richtung in Bewegung halten, wodurch erklärbar wäre, warum diese Bewegung vom dort lokalisierten Beschleunigungssensor wahrgenommen wird, von den FSR-Sensoren hingegen nicht. Da es sich nur um kleine Bewegungen handelt, werden sie durch die Flexibilität der Gelenke gedämpft. Diese Annahme wird ebenfalls von dem Stromsensor des *AnkleRoll*-Gelenkes bekräftigt, da dieser nicht auf die Bewegung des Oberkörpers reagiert. Dies erklärt allerdings noch nicht in vollem Maße die Stufigkeit der Messung, da diese in folgenden Experimenten wiederholt auch in Bewegung auftritt. Diese muss in einer Eigenschaft der Vorfilterung seitens Aldebaran begründet sein. Die ZMP-Berechnung des Stromsensors ist in diesem Fall exakt. Allerdings kann hieraus



(a) ZMP-Verlauf  $x$ -Position.



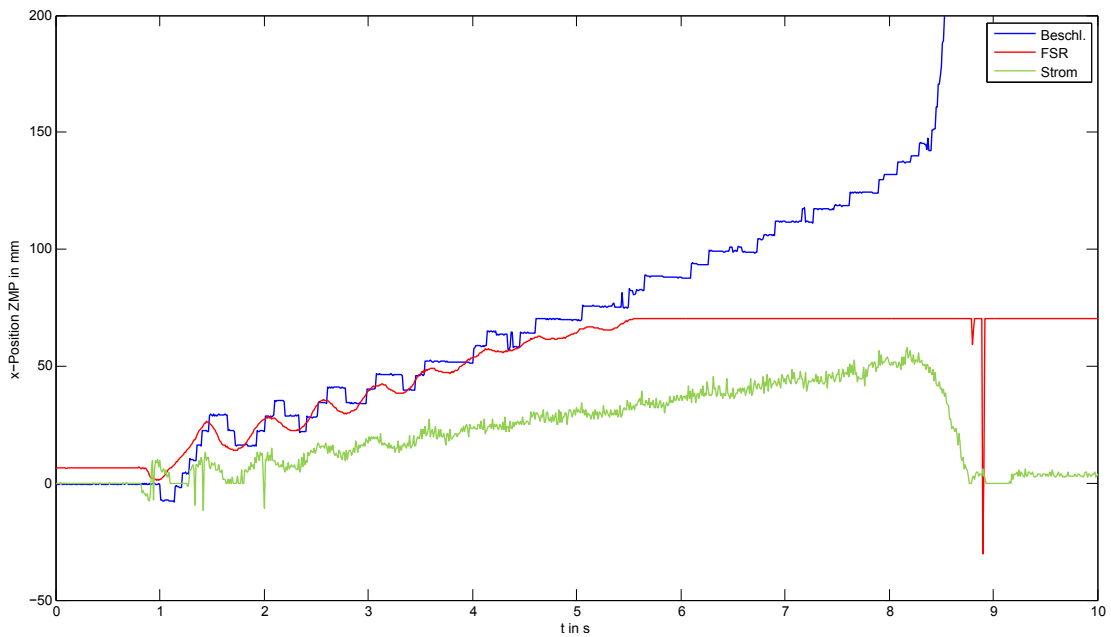
(b) ZMP-Verlauf  $y$ -Position.

Abbildung 4.4.14: Experiment 4.4.4.A - Stand auf linkem Bein.

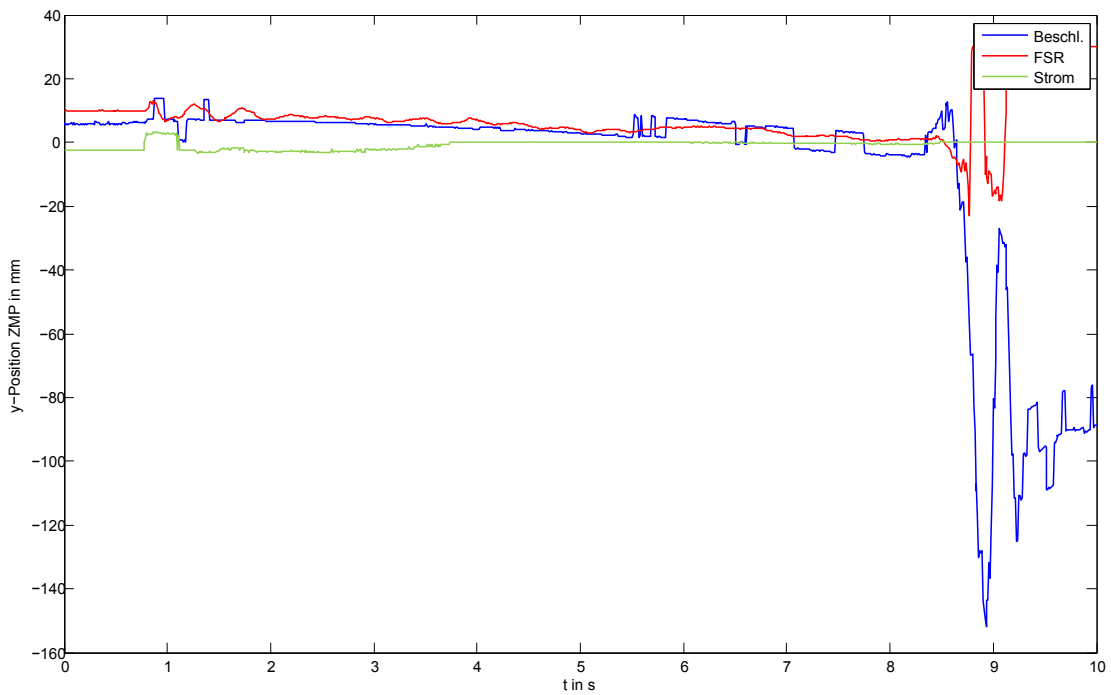
keine weitere Aussage abgeleitet werden, da es zum einen sein kann, dass der Sensor nicht akkurat genug ist, um kleinen Änderungen verlässlich wahrzunehmen, und zum anderen bewegt sich das Gelenk möglicherweise wirklich nicht, da die aufzubringende Kraft die Reibung des Gelenks nicht übersteigt.

#### **Experiment 4.4.4.B - Sturz in $x$ -Richtung**

In diesem Experiment wird ein einfacher ZMP-Verlauf untersucht, bei dem der Roboter stürzt. Hierzu nimmt der Roboter NAO die gleiche Grundstellung ein, wie in Experiment 4.4.4.A und beginnt dann das *AnklePitch*-Gelenk zu bewegen, so dass der Roboter das Gewicht nach vorne verlagert. Dies hat in der Theorie einen linearen Verlauf der ZMP-Trajektorie in  $x$ -Richtung zur Folge. Sobald sich der ZMP außerhalb des Fußes bewegt, fängt der Roboter an unkontrolliert weiter nach vorne zu beschleunigen, da er dann fällt. Der ZMP-Verlauf in  $x$ -Richtung ist in Abbildung 4.4.15(a) zu finden und zeigt deutlich, dass der Roboter sich zum Zeitpunkt  $t = 1$  s in Bewegung versetzt hat. Alle drei Sensoren nehmen diese Bewegung wahr, wobei auffällt, dass die FSR-Sensoren ein paar Zyklen vor dem Beschleunigungssensor reagieren. Dies kann mit der Filterung des Inertialsensor erklärt werden. Noch vor den anderen Sensoren reagieren die Stromsensoren, was schlüssig ist, wenn man bedenkt, dass die Stromaufnahme bereits vor der eigentliche Bewegung ansteigen muss, um das Losbrechmoment des Gelenks zu erreichen. Im Folgenden spiegeln alle drei Sensoren den linearen ZMP-Verlauf wider, der von einer Schwingung überlagert ist. Dieses Phänomen kann theoretisch erklärt werden. Da das gesamte Gewicht des Roboters NAO nur mittels des einen Fußgelenks bewegt wird, muss der Motor durch den langen Hebel eine hohe Kraft aufbringen, um den Roboter zu bewegen. Da die Trajektorie langsam angesteuert wird, kann der Regler im Motor anscheinend die Position nicht über die gesamte Zeit halten, und muss folglich in manchen Fällen gegenregeln, wodurch die Schwingung erklärbar wird. Der Verlauf des ZMPs aus FSR-Sensoren und Beschleunigungssensoren ist nahezu identisch, lediglich der Beschleunigungssensor weist erneut ein stufiges Verhalten auf. Dies geht vermutlich erneut auf die Quantisierung der Vorfilterung zurück. Der Effekt ist zwar für das reine Quantisierungsrauschen zu groß, doch da dieser Effekt bereits bei den Beschleunigungswerten auftritt, bleibt keine andere Erklärung, als dass es sich um ein Artefakt der Filterung in Kombination mit der Gelenkansteuerung handelt. Eine genaue Untersuchung steht allerdings noch aus. Die Steigung der Stromsensorgeraden fällt flacher aus, was durch eine falsche Motorkonstante erklärt werden kann. Ihr Verlauf zeigt jedoch, dass der Sensor zumindest für eine einfache Bewegung qualitativ in der Lage ist, den ZMP-Verlauf wahrzunehmen. Auffällig ist, dass bei rund  $x = 70$  mm der Verlauf der FSR-Sensoren stagniert. Die Erklärung hierfür ist offensichtlich - die Sensoren können nur innerhalb der von ihnen aufgespannten Fläche den ZMP messen. Zu diesem Zeitpunkt ist deren Grenze erreicht, auch wenn sich der ZMP im weiteren Verlauf noch innerhalb der Fußfläche befindet. Diese wird erst ungefähr bei  $x = 105$  mm verlassen. Sowohl Strom- als auch Beschleunigungssensor messen den ZMP-Verlauf auch noch bis zu diesem Punkt. Der Beschleunigungssensor tut dies sogar noch darüber hinaus und demonstrieren somit deutlich, dass er in der Lage ist, den FZMP zu messen. Kurz nach  $t = 8$  s geht der Roboter in den freien Fall über und verliert die Bodenhaftung. Dies ist daran zu erkennen, dass die Leistungsaufnahme im Fußgelenk zunächst absinkt, was



(a) ZMP-Verlauf  $x$ -Position.



(b) ZMP-Verlauf  $y$ -Position.

Abbildung 4.4.15: Experiment 4.4.4.B - Sturz in  $x$ -Richtung.

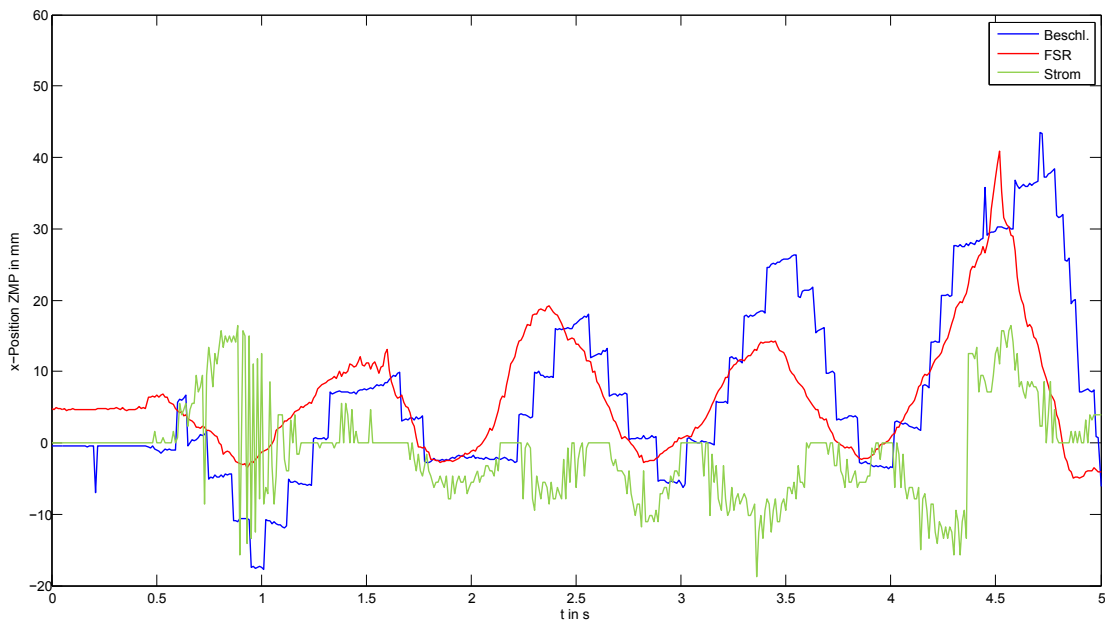


durch ein Wegrutschen des Fußes erklärt werden kann. Anschließend greift die *NDD-Fall-Protection*<sup>8</sup> ein und schaltet die Motoren des NAO ab. Die Tatsache, dass der FSR-Sensor aber noch bis kurz vor  $t = 9$  s des Experiments einen ZMP an der Kante misst, spricht dafür, dass zumindest die Fußspitze noch immer Kontakt mit dem Boden haben muss. Abbildung 4.4.15(b) zeigt grundsätzlich den erwarteten Verlauf für die  $y$ -Richtung mit der Ausnahme, dass der Roboter leicht seitlich nach rechts vorne kippt, was sich aber durch die Gewichtsverlagerung und die abgerundeten Füße des Roboters erklären lässt.

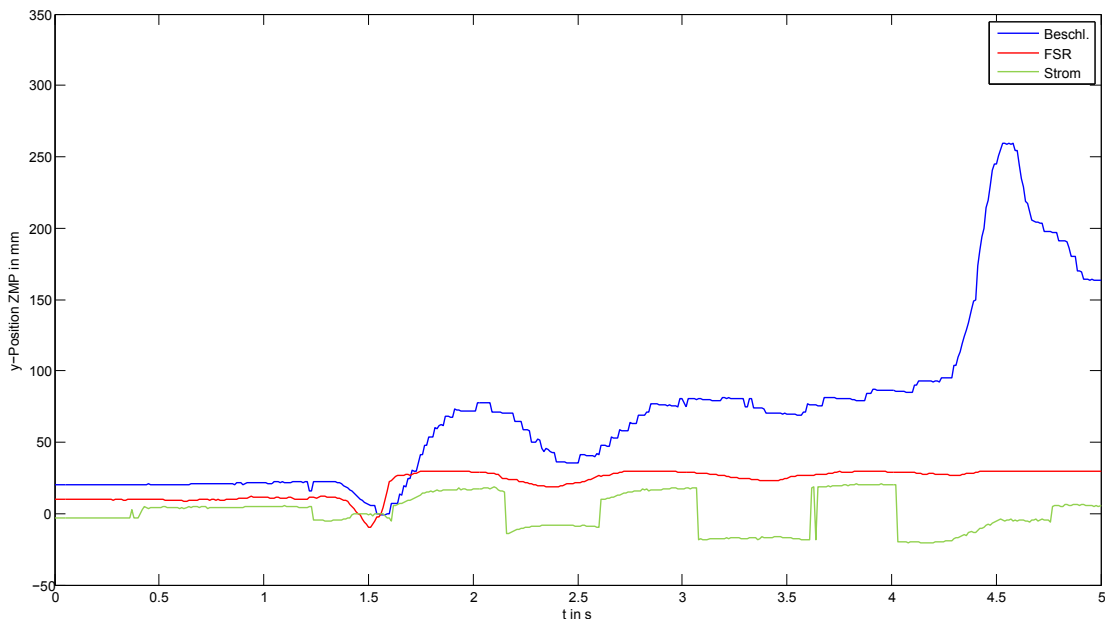
#### Experiment 4.4.4.C - Sturz in $y$ -Richtung

In diesem Experiment wird der Aufbau aus 4.4.4.B wiederholt, nur wird dieses Mal das *AnkleRoll*-Gelenk benutzt, um den Roboter nach links, also in  $y$ -Richtung kippen zu lassen. Allerdings ist dieser Bewegungsablauf mit dem Roboter NAO nicht so einfach, da sich in dieser Stellung das *AnkleRoll*-Gelenk bereits im Anschlag befindet und der Roboter nicht durch dieses Gelenk weiter nach links bewegt werden kann. Da das korrespondierende Hüftgelenk nicht genug Drehmoment aufbringen kann, um den NAO aus dem Stand über die linke Fußkante fallen zu lassen, muss dieses Experiment mittels eines dynamischen Bewegungsablaufs realisiert werden. Der Roboter wird zunächst über das Fußgelenk nach rechts ausgelenkt, um dann zu bremsen und den Oberkörper nach links zu bewegen. Durch die Beschleunigung wird dabei soviel Kraft erzeugt, dass der Roboter über die linke Fußkante kippt. Der entsprechende ZMP-Verlauf in  $y$ -Richtung, dargestellt in Abbildung 4.4.16(b), zeigt in Folge dessen auch einen merklich komplexeren Verlauf. Durch den Wechsel in der Beschleunigung wird der Roboter in Schwingung versetzt, wie an den ZMP-Verläufen zu sehen ist, um dann nach einem Rückschwinger um  $t = 2,5$  s durch die weitere Bewegung in  $y$ -Richtung nach  $t = 4$  s über die linke Fußkante zu kippen. Der Plot weist abermals das stufige Verhalten des ZMPs aus den Werten des Beschleunigungssensors auf. Ebenfalls ist zu erkennen, dass der Wert der FSR-Sensoren ein wenig vor dem ZMP aus den Beschleunigungssensoren herläuft. Erneut ist dies aller Wahrscheinlichkeit nach auf die Filterung zurückzuführen. Doch bereits nach kurzer Zeit stößt er an die Grenze der aufgespannten FSR-Fläche, wodurch die Schwingung, die der Beschleunigungssensor noch wahrnimmt, nicht mehr gemessen werden kann. Durch eine sanfte Schwankung am Grenzbereich ist diese jedoch noch zu erahnen. Auch auf den Stromsensor hat die Schwingung, erwartungsgemäß falsch skaliert, eine Auswirkung, allerdings ist diesmal der Verlauf markant anders. Dies kann mit dem PID-Regler des Gelenkes erklärt werden, der vermutlich versuchsweise stabilisierend gegen die Schwingung zu regeln und somit die Messung des ZMPs durch das Gelenk beeinflusst. Der Plot in Abbildung 4.4.16(a) belegt, dass der Roboter sich durch die Abbremsung in Kombination mit dem Spiel in den Gelenken auch in  $x$ -Richtung aufschwingt. Die Schwingung wird von den FSR- und Beschleunigungssensoren ähnlich gut verfolgt. Abermals ist hierbei die leichte Phasenverschiebung der beiden Messungen gegeneinander zu erkennen. Der Stromsensor misst zwar Schwingungen, doch fällt erneut auf, dass die Werte stark verrauscht sind. Es zeichnet sich zudem noch ein weiteres Phä-

<sup>8</sup>Ein Modul des Teams NDD, das mittels der Oberkörper-Orientierung und Beschleunigung entscheidet ob der Roboter fällt. Wird ein Sturz detektiert, werden die Motoren ausgeschaltet, um bei Kollision mit dem Boden die Gelenke nicht übermäßig zu belasten.



(a) ZMP-Verlauf  $x$ -Position.



(b) ZMP-Verlauf  $y$ -Position.

Abbildung 4.4.16: Experiment 4.4.4.C - Sturz in  $y$ -Richtung.

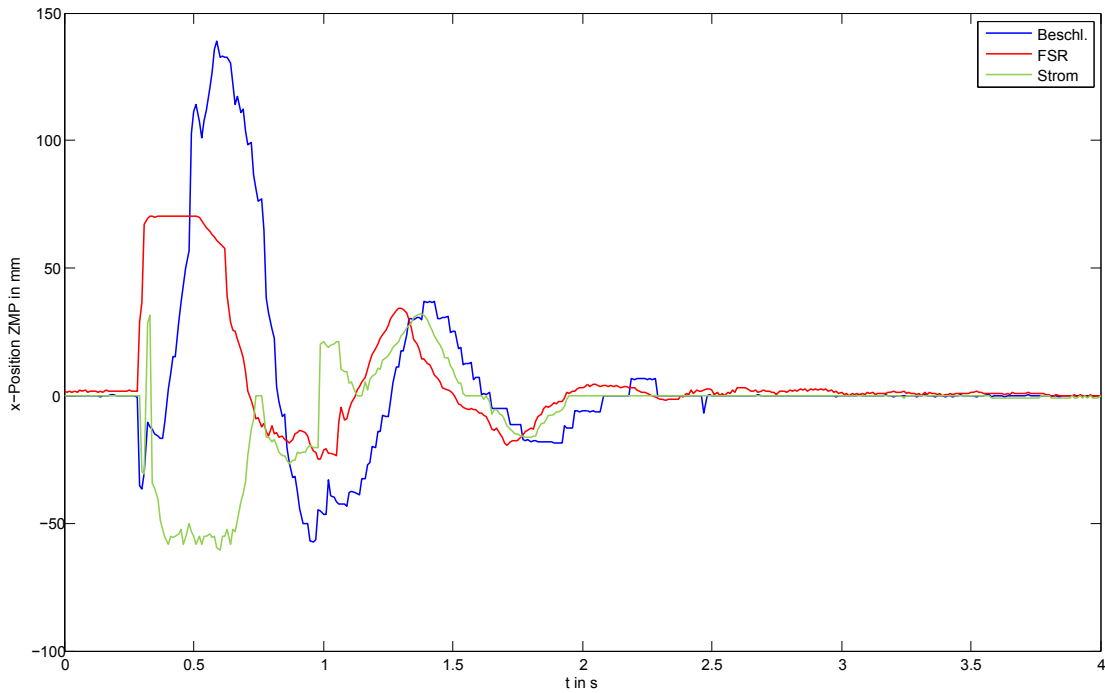
nomen ab, das bereits angesprochen wurde - die Richtung der Amplitude wird teilweise falsch aus der Bewegung bestimmt. Hier würde eine tiefere Integration der Messung in das *Motion*-Modul Abhilfe schaffen.

#### **Experiment 4.4.4.D - Kollision mit einer Masse**

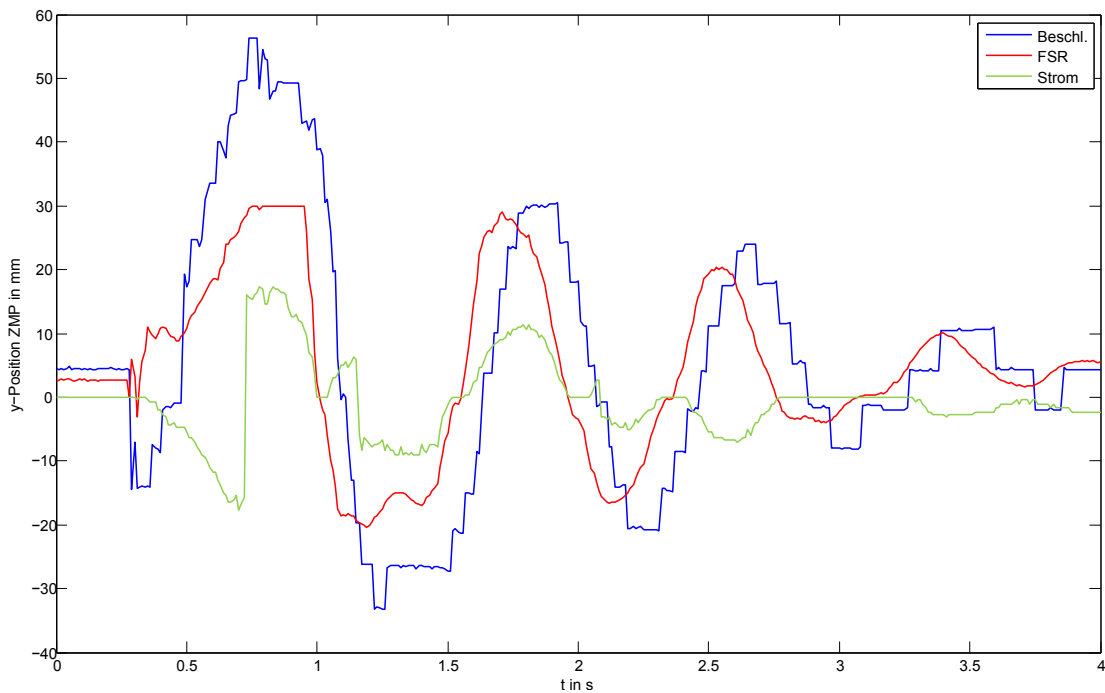
Bislang wurden Bewegungen evaluiert, die der Roboter selbstständig ausführt. Um zu zeigen, dass die Sensoren auch Änderungen des ZMP wahrnehmen können, die von Außen auf den Roboter einwirken, wird in Experiment 4.4.4.D die Kollision des Roboters mit einer Masse untersucht. Hierzu wird der Roboter erneut in die balancierende Pose auf dem linken Bein versetzt. Eine Plastikflasche, gefüllt mit 0,5 Liter Wasser, wird vor dem Roboter aufgehängt und kollidiert als Pendel von hinten mit dem Torso des Roboters. Die resultierenden ZMP-Trajektorien sind in Plot 4.4.17 zu finden. Alle Sensoren zeigen eine Schwingung des Roboters sowohl in  $x$ - als auch in  $y$ -Richtung. Dies ist zu erwarten, jedoch fällt auf, dass die Schwingung in  $x$ -Richtung deutlich schneller abklingt, als in  $y$ -Richtung. Eine logische Erklärung folgt aus der Natur des Experiments. Durch die Kollision mit dem Roboter schwingt die Flasche nach dem Aufprall in die entgegengesetzte  $x$ -Richtung, um dann nach kurzer Zeit erneut mit dem Roboter zu kollidieren. Das Verhältnis der Periode der beiden Schwinger ist offensichtlich derart, dass hierdurch die Schwingung gedämpft wird. Dieser Effekt tritt im Anschluss nochmals auf und dann verharren Pendel und Roboter in Ruhe zueinander. In  $y$ -Richtung findet diese Dämpfung nicht statt, somit kann diese ohne externe Dämpfung ausschlagen. Auch in diesem Experiment können die bisherigen Beobachtungen verifiziert werden. Der Beschleunigungssensor zeigt wieder ein stufes Verhalten und nimmt durch seine Filterung die Bewegung verzögert gegenüber den Drucksensoren wahr. Diese können allerdings erneut nicht der gesamten ZMP-Bewegung folgen, da er bereits zum Zeitpunkt  $t = 0,4\text{ s}$ , respektive  $t = 0,8\text{ s}$ , an die Grenze seines Messbereichs stößt. Der Stromsensor zeigt erwartungsgemäß das schlechteste Verhalten und prognostiziert teils sogar die Richtung der ZMP-Bewegung falsch.

#### **Experiment 4.4.4.E - Schussbewegung**

Die bisherigen Simulationen untersuchten langsame Bewegungen des Roboters. Daher wird abschließend noch eine komplexere dynamische Bewegung aus dem Anwendungsgebiet des RoboCups untersucht. Hierzu führt der Roboter NAO in Experiment 4.4.4.E einen Distanzschuss aus, der aus einer festen Abfolge von Bewegungen besteht. Der Roboter begibt sich zunächst in einen balancierten Stand auf dem linken Bein und bewegt den rechten Fuß nach hinten. Im Anschluss bringt er diesen mit maximaler Geschwindigkeit nach vorne, um ihn dann zurückzuziehen und abzusetzen. Der Plot der Messwerte in Abbildung 4.4.18 zeigt, dass die Messungen im dynamischen Fall weniger eindeutig sind. Der Stromsensor soll nicht weiter diskutiert werden, doch die Kurven der FSR- und des Beschleunigungssensors lassen einige interessante Rückschlüsse zu. Zunächst ist zu erkennen, dass der Roboter durch das Ausholen und den folgenden Schuss, wie zu erwarten war, in Schwingung versetzt wird. Diese setzt sich auch noch während der

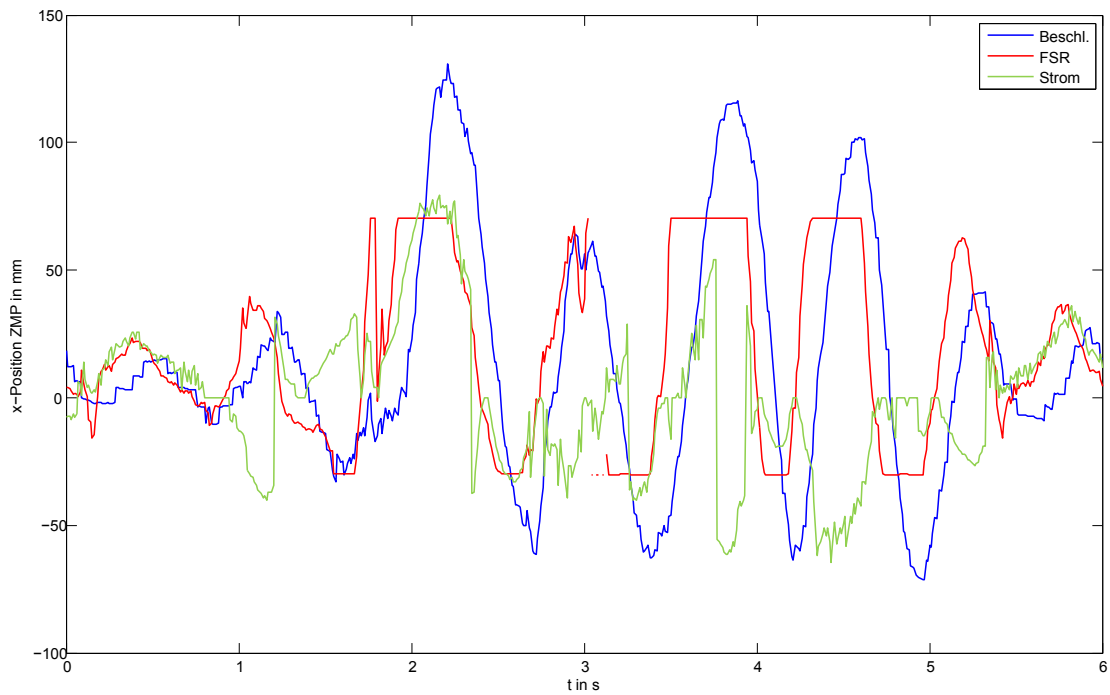


(a) ZMP-Verlauf  $x$ -Position.

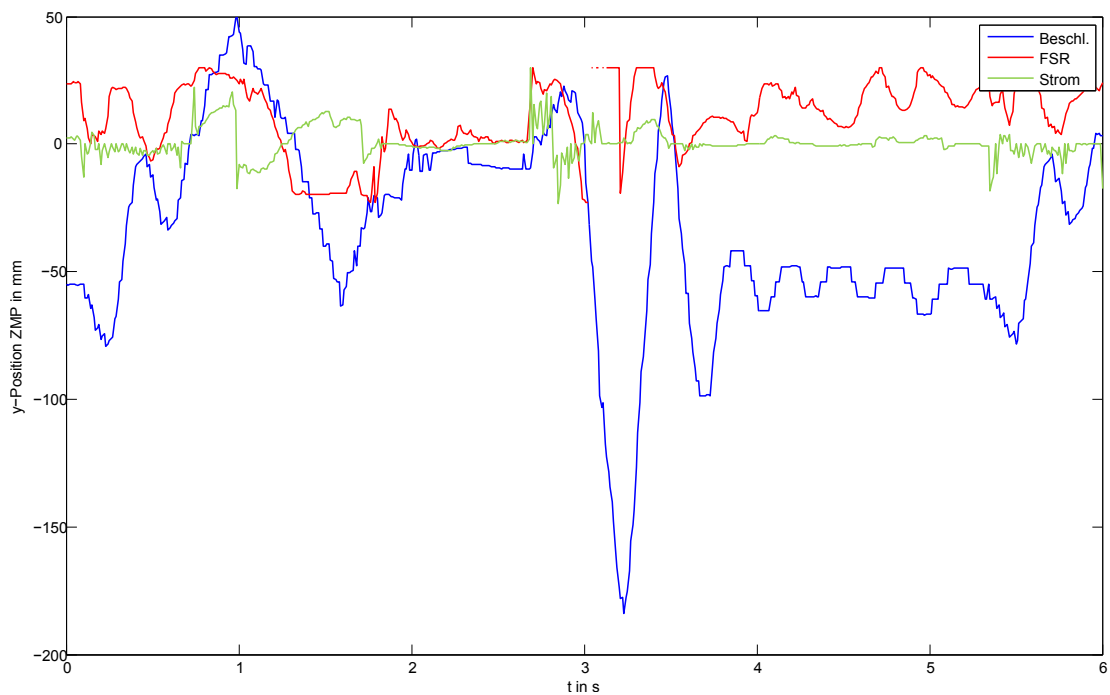


(b) ZMP-Verlauf  $y$ -Position.

Abbildung 4.4.17: Experiment 4.4.4.D - Kollision mit einer Masse.



(a) ZMP-Verlauf  $x$ -Position.



(b) ZMP-Verlauf  $y$ -Position.

Abbildung 4.4.18: Experiment 4.4.4.E - Schussbewegung.

Rückbewegung des Fußes fort und endet erst, wenn der Roboter wieder in den zweibeinigen Stand übergeht. Durch die seitliche Verlagerung des Oberkörpers entstehen auch Schwingungen in  $y$ -Richtung, die sich mit denen aus der Schussbewegung resultierenden überlagern. Die Bewegung in  $x$ -Richtung lässt sich mit den Sensoren gut nachvollziehen. Sowohl FSR- als auch Beschleunigungssensor zeichnen saubere Schwingungen auf, wobei der ZMP erneut den Messbereich der FSR-Sensoren mehrfach verlässt. Interessant ist der Zeitraum um  $t = 3\text{ s}$ , da hier die FSR-Sensoren für kurze Zeit keinen ZMP liefern, um dann wieder phasenverschoben in die Schwingung überzugehen. Da dieser Effekt auch in  $y$ -Richtung auftritt, ist dies nur dadurch erklärbar, dass der Roboter durch die Dynamik des Schusses kurz den Boden verlässt und als Folge auf keinem der 4 Sensoren eine Kraft gemessen wird. Die Analyse der seitlichen Schwingung aus Plot 4.4.18(b) zeigt eine starke Diskrepanz zwischen den Messungen der beiden Sensoren. Auffällig ist besonders das abweichende Verhalten des Beschleunigungssensors, da dieser über weite Teile des Experiments Werte liefert, die deutlich außerhalb der *Support*-Fläche liegen und zwingend zum Sturz des Roboters geführt hätten. Da dies nicht eingetreten ist, können diese als offensichtlich falsch deklariert werden. Erklärbar ist dieses Phänomen durch die Eigenbewegung des Oberkörpers, der während des Schusses seine Stellung seitlich um das Hüftgelenk verlagert. Durch die Bewegung wird offensichtlich die Orientierung des Oberkörpers oder die Lage des CoM falsch bestimmt und somit die Messung verfälscht. Die Vermutung bestätigt sich nochmals dadurch, dass dieser Effekt in  $x$ -Richtung nicht auftritt.

### 4.4.5 Beurteilung der Messverfahren

Die Experimente dieses Kapitel lassen einige Rückschlüsse auf die Praxis-Tauglichkeit der ZMP-Messung mittels der einzelnen Sensoren zu. Eine eindeutige Aussage, welches Verfahren das geeignetste ist, kann nicht getroffen werden. Doch können einige Annahmen der vorhergehenden theoretischen Diskussionen bestätigt werden. So konnte gezeigt werden, dass die Messung des Drehmoments über den Stromsensor zwar theoretisch funktioniert, aber für die Praxis zu fehleranfällig ist. Das Problem der Skalierung kann experimentell in den Griff bekommen werden. Und auch das Problem der vorzeichenbehafteten Messung könnte gelöst werden, indem die Messung besser in das NDD-Software-Framework integriert würde. Durch Verwendung des Wissens über das Kommando der Motoransteuerung kann so auf die differenzielle Berechnung der Bewegungsrichtung verzichtet werden. Das Rauschen des Sensors, teilweise durch die interne Regelung des PID-Reglers herbeigeführt, stellt jedoch ein Problem für die Anwendbarkeit des Verfahrens auf dem Roboter NAO dar.

Sowohl das Messverfahren über die FSR-Sensoren, als auch das Schätzen des ZMPs über den Beschleunigungssensor zeigen sich in der Lage den Verlauf des ZMPs ausreichend gut zu messen, und kann somit für den praxisrelevanten Einsatz empfohlen werden. Die Messung mittels der Drucksensoren erweist sich erwartungsgemäß als sehr akkurat und zeigt in allen Experimenten eine geringere Sensorverzögerung, da auf eine Vorfilterung seitens Aldebaran verzichtet wurde. Jedoch offenbart sich auch der große Nachteil der FSR-Sensoren - der geringe Messbereich der Sensoren. Die Beschleunigungssensoren decken den gesamten Fußbereich ab und bieten darüber hinaus den Vorteil, auch den

FZMP außerhalb des *Support*-Polygons bestimmen zu können. Dies ist besonders für die Reaktion auf einen Verlust der Stabilität, der in dieser Arbeit nicht untersucht wird, interessant, da gezielt ein Fuß an die Position des FZMPs gebracht werden kann, um somit die Stabilität des Systems wieder herzustellen. Die Experimente bestätigen weiterhin die Erwartungen einer höheren Fehleranfälligkeit der Beschleunigungssensor-Messung, die besonders im dynamischen Fall in erster Linie auf Fehler in der Bestimmung des CoM und der Orientierung des Oberkörpers zurückzuführen ist. Aber dennoch bescheinigen sie dem Verfahren eine ausreichende Robustheit für den Praxiseinsatz. Ein derart kritischer Effekt, wie die in Simulation 4.4.2.B untersuchte Verschiebung des  $CoM_{x,y}$ , tritt nicht auf und belegt somit die Güte der CoM-Berechnung. Es bleibt jedoch für den weiteren Verlauf der Arbeit festzuhalten, dass die ZMP-Messung aus der Beschleunigung merklich von einer verbesserten Modellierung der Oberkörperorientierung profitieren würde.

Über den Verlauf der Auswertung der Experimente hat sich wiederholt gezeigt, dass die Verfahren unterschiedliche Vor- und Nachteile besitzen. Meist konnte erst durch kombinierte Auswertung aller Sensorkurven der genaue Verlauf des Experimentes erklärt werden. Diese Beobachtung legt den Schluss nahe, dass die Messung des ZMPs signifikant von einer Sensor-Fusion profitieren würde. Die Reaktionsschnelligkeit und Genauigkeit der FSR-Sensoren könnte so mit der Fähigkeit der Messung des FZMP überein gebracht werden. Weiterhin könnten durch die Redundanz der Messungen Fehlereinflüsse reduziert werden oder gar der Ausfall eines Sensors detektiert werden. Dies würde sich merklich auf die Robustheit des NAO in der Praxis auswirken. Nach Wissen des Autors ist eine Untersuchung einer solchen Fusion noch nicht durchgeführt worden. Eine Untersuchung im Rahmen dieser Arbeit bleibt ausstehend.





---

### Fortbewegung zweibeiniger Roboter

---

Im bisherigen Verlauf der Arbeit wurde eingehend über Sensorik humanoider Roboter diskutiert und erläutert, wie diese eingesetzt werden kann, um den aktuellen Zustand des Roboters wahrzunehmen und zu bewerten. Das Ziel dieser Diskussion war es, ein Maß für das Gleichgewicht des Roboters zu entwickeln. Dies geschah nicht zum reinen Selbstzweck, sondern vor dem Hintergrund, dieses Wissen als Grundlage der Bewegungsteuerung zu verwenden. Somit können nun Algorithmen diskutiert und evaluiert werden, mit denen der Roboter stabile Fortbewegungsmuster erzeugen kann. In Abgrenzung zur allgemeinen Bewegung resultiert die Fortbewegung des Menschen immer aus der Wiederholung eines gleichbleibenden Bewegungsablaufs. Dies liegt in der Natur seines Fortbewegungsapparates begründet. Sie wird im Folgenden daher auch als *zyklische* Bewegungssteuerung bezeichnet. Das Besondere an dieser Art der Bewegung ist, dass sich ihr Zweck klar definieren lässt. Wie sich im Laufe dieses Kapitels zeigt, ist dieser sogar durch mathematische Formulierungen beschreibbar, so dass die Trajektorien der Gliedmaßen vom Roboter während der Bewegung berechnet werden können. Dies ist vergleichbar mit der unterbewussten Bewegung des Menschen beim Gang. Wir denken in der Regel nur über das Ziel oder Richtung unserer Fortbewegung nach, die Bewegung der Beine erfolgt vom Unterbewusstsein gesteuert. Die Planung komplexerer Bewegungen hingegen erfordert bewusstes Nachdenken des Menschen, solange dieser Bewegungsablauf noch nicht zur Routine geworden ist. Ähnlich verhält es sich mit autonomen Maschinen. Es ist schwierig den Nutzen einer allgemeine Bewegung so zu formulieren, dass er von einer Maschine erfasst werden kann. Folglich kann der Ablauf auch nicht selbstständig vom Roboter mathematisch geplant werden. Diese *nicht-zyklische* Bewegungssteuerung erfordert also andere Lösungsansätze, als die in diesem Kapitel diskutierte Fortbewegung. Im Kontext der SPL wird das Problem meist mittels *Special Actions* gelöst. Diese definieren feste *Key Frames*, zwischen denen eine starre Interpolation ausgeführt wird und sind somit eng mit der *Teach-In* Programmierung von Industrierobotern verwandt, vergleiche [49]. Diese Einführung dient nur der Abgrenzung - es wird im Rahmen dieser Arbeit nicht weiter auf nicht-zyklische Bewegungen eingegangen. Doch soll nicht unerwähnt bleiben, dass der Roboter NAO auch in diesem

Bereich zur Erforschung neuer Algorithmen genutzt wird. Besonders für die Planung und Ausführung von Schussbewegungen erweist sich die Unflexibilität der *Special Action* zunehmend als Problem und so wird mehr und mehr versucht Sensorinformationen für den Bewegungsablauf nutzbar zu machen. Ein Einstieg in die Thematik der *dynamischen Special Actions* kann beispielsweise anhand der Veröffentlichungen [22] und [82] erfolgen.

### 5.1 Zyklische Bewegungssteuerung

Um die im folgenden Kapitel 5.2 vorgestellten Algorithmen zur Fortbewegung des Roboters NAO in den wissenschaftlichen Kontext einordnen zu können, folgt zunächst ein Überblick über verschiedene Methoden zur Erzeugung von Laufbewegungen. Ein solcher Algorithmus wird im Kontext der Robotik oft auch als *Walking-Engine* (WE) bezeichnet. Auch wenn sich dieser Begriff eigentlich auf die konkrete Implementierung bezieht, soll er daher dennoch im Folgenden synonym mit dem Algorithmus verwendet werden. Die Diskussion der Konzepte wird jeweils durch die Vorstellung konkreter Umsetzungen ergänzt, um dem Leser die Möglichkeit zu geben sich weiter in das jeweilige Thema zu vertiefen. Wie schon bei der Vorstellung der unterschiedlichen Hardwareplattformen in Kapitel 2.3, muss auch an dieser Stelle der Hinweis erfolgen, dass diese Vorstellung keinen Anspruch auf Vollständigkeit erheben kann. Zu groß ist heutzutage die Anzahl unterschiedlicher WE. Aus diesem Grund werden bevorzugt Konzepte vorgestellt, die am Roboter NAO erprobt wurden, um weiterhin den Gedanken der Vergleichbarkeit in den Vordergrund zu stellen. Trotzdem kommt der Autor für einen Querschnitt der grundsätzlichen Konzepte und Ideen nicht umhin, auch Algorithmen außerhalb des RoboCup-Kontextes zu erwähnen, um einen repräsentativen Überblick geben zu können.

Grundsätzlich lassen sich die Algorithmen zur Fortbewegung in zwei Kategorien unterteilen, *open-loop* und *close-loop*. Das Kriterium zur Unterscheidung ist, ob die Ausführung der Bewegung mittels Sensoren überwacht und gegebenenfalls korrigiert wird. Ist dies der Fall - wird die Bewegung also durch einen Regler überwacht - handelt es sich um eine *close-loop* WE. Eine reine Steuerung der Bewegung klassifiziert demnach eine *open-loop* WE. Da eine Korrektur der *open-loop* Trajektorie zur Laufzeit per Definition nicht möglich ist, kommen hierzu in der Regel vorberechnete Bewegungsabläufe zur Anwendung. Sie sind demnach *offline*, also nicht zur Laufzeit, berechnet. Die Trajektorien der *close-loop* Algorithmen werden hingegen zur Laufzeit, also *online* berechnet. Der Rechenaufwand dieser Klasse von Algorithmen ist daher deutlich höher. Abschließend gibt es noch die Klasse der *hybriden* WE. Diese kombinieren vorberechnete Trajektorien mit einer Überwachung und Korrektur während der Ausführung. Dies muss aber nicht zwangsläufig durch einen Regler im klassischen Sinne geschehen.

Der einfachste Vertreter einer *open-loop* WE ist die wiederholte Ausführung von **periodischen Trajektorien**. Aufgrund der Ähnlichkeit zur Steuerung der Bewegung aus dem menschlichen Gehirn wird diese Klasse von Algorithmen oft auch als **Central Pattern Generator** (CPG) bezeichnet, vergleiche [73]. Allen diesen Algorithmen ist

gemein, dass sie die Trajektorie vor der Ausführung generieren und dann bei Bedarf zur Laufzeit unüberwacht abfahren. Merkwürdig unterscheiden Sie sich hingegen in der Art und Weise, wie diese Trajektorien für die Laufbewegung erzeugt werden. Ein gebräuchlicher Ansatz hierfür ist die Definition einer initialen, aber parametrierbaren, Beschreibung der Fußtrajektorie nach logischen Gesichtspunkten. Meist kommen hierzu geometrische Primitive zum Einsatz. Die geeigneten Parameter für den entsprechenden Roboter werden dann mit einem *BlackBox*-Optimierungsverfahren, üblicherweise **Evolutionären Algorithmen**, gesucht. Ein typischer Vertreter dieser Art von WE ist beispielsweise der von Hebbel anhand von 4- und 2-beinigen Robotern untersuchte Algorithmus [47]. Auch im Bereich der NAO-SPL gibt es einige Teams, die mit dieser Methode gute Ergebnisse erzielen - vergleiche ohne Anspruch auf Vollständigkeit [36], [96], [97] und [112]. Eines der Teams, das ohne Frage mit dem größten Erfolg evolutionäre Optimierung auf den Roboter NAO angewandt hat, ist das NAO-Team HTWK, denen es mittels „[...] *major algorithmic optimization*[...]“ [11] gelungen ist die Anpassung auf dem realen NAO soweit zu reduzieren, dass der Algorithmus fast ausschließlich im Simulator entwickelt werden konnte. Leider existiert über den verwendeten Ansatz keine Veröffentlichung, die auf Details eingeht. Eine weitere bemerkenswerte Applikation der *open-loop* Generierung von Lauftrajektorien ist das Verfahren des **Motion Capturing**. Dabei wird die Bewegung eines Menschen mit technischen Hilfsmitteln erfasst und auf den Roboter übertragen. Diese können natürlich aufgrund der Unterschiede in Proportion und Gewichtsverteilung zwischen Mensch und Roboter nicht eins zu eins auf dem Roboter ausgeführt werden. Doch wurden in den letzten Jahren einige Verfahren veröffentlicht, die eine Generalisierung der Bewegung und dadurch deren Übertragung auf den Roboter ermöglichen - siehe beispielsweise [78], [99] oder [120]. Obwohl dieses Vorgehen in der Regel nicht zu derart perfekt an die Hardware des Roboters angepassten Bewegungen führt, wie die evolutionäre Optimierung, so bietet dieses Vorgehen doch entscheidende Vorteile. Die Nachahmung der Bewegung des Menschen führt zu entsprechend natürlichen Bewegungsabläufen. In Kombination mit einer Weiterentwicklung der Hardware in Form von flexibleren Gelenken oder gar künstlichen Muskeln kann dieser Ansatz auch noch in Bereichen angewandt werden, die für eine *BlackBox*-Optimierung zu große Suchräume aufspannt. Durch die fortschreitende Entwicklung der Rechenleistung mobiler Hardware besteht weiterhin das Potential, diese Nachahmung in Zukunft online auszuführen. Der Mensch könnte dann die Bewegung des Roboters während der Ausführung korrigieren und es ihm somit ermöglichen aus seinen Fehlern zu lernen. Meriçli hat erste Untersuchungen zu einer solchen Online-Korrektur mit dem Roboter NAO durchgeführt und konnte bereits mit einem sehr einfachen Mechanismus vielversprechende Ergebnisse erzielen [76].

Auf der anderen Seite stehen die Algorithmen der Klasse der *close-loop* Algorithmen. Seit dem Vukobratović im Jahre 1968 mit der Erforschung des ZMPs [116] begonnen hat, versuchen Wissenschaftler auf der Theorie des Gleichgewichts basierende Algorithmen zu schaffen, die die Berechnung einer stabilen Laufbewegung ermöglichen. Hierzu ist eine mathematische Beschreibung des Systems Roboter nötig - es wird ein Modell des Roboters benötigt. Daher wird auch synonym von einer **modellbasierten** WE gesprochen. Auf der einen Seite erfordert dies eine höhere Rechenleistung während der Bewegung, da die Abläufe online generiert werden. Auf der anderen Seite ermöglicht dies eine Korrektur durch Algorithmen der Regelungstechnik, wenn die tatsächlich aus-

geführte Bewegung mittels Sensoren überwacht wird. Ein Modell ist jedoch immer zu einem gewissen Grad fehlerbehaftet. Somit gibt es zwei Ansatzpunkte, um die Resultate einer *close-loop* Steuerung zu verbessern. Dies führte in der Vergangenheit zu einer Spaltung der Forscher in zwei grundlegende Lager. Auf der einen Seite versuchten Wissenschaftler den Lauf des Roboters ohne detailliertes Wissen der Dynamik des Systems durch ausgereifte Regler zu stabilisieren und somit Unzulänglichkeiten im Modell auszugleichen, vergleiche beispielsweise [50], [59], [83] und [119]. Auf der anderen Seite versuchten Forschergruppen die Dynamik so exakt wie möglich durch das Modell zu erfassen, um die Notwendigkeit einer Korrektur durch einen Regler soweit wie möglich zu reduzieren. Beispiele hierfür sind in [34], [60], [87], [88] und [122] zu finden. Begutachtung der Veröffentlichung zeigt aber, dass die Ergebnisse nur schwer zu vergleichen sind. Die Leistungsfähigkeit einer *open-loop* WE ist nicht nur von der Modellierung und Regelung abhängig, sondern wird entscheidend auch von deren Zusammenspiel mit der Hardware bestimmt. Für die praxisnahe Forschung hat sich gezeigt, dass die humanoiden Roboter der letzten Jahre nicht genug Rechenleistung haben, um Bewegungsabläufe auf Basis komplizierter Mehrkörper-Modelle zur Laufzeit berechnen zu können. Somit wird in der Regel ein Einkörper-Modell basierend auf der Physik eines **invertierten Pendels** genutzt. Besonders hervorzuheben ist hier der Algorithmus der Forschergruppe um Kajita, die für ihre Forschungen eine linearisiertes 3D-Pendel-Modell [58] nutzen. Dieses Modell dient vielen Entwicklungen der letzten Jahre als Basis und kann daher als eine der Standard-*close-loop*-Methoden gelten. Auch die erste *close-loop* WE, die für den Roboter NAO von Team NDD entwickelt wurde [24], liegt Kajitas 3D-Pendel-Modell zugrunde. Details hierzu werden im folgenden Kapitel 5.2 genauer erörtert. Die WE des Teams NDD war derart erfolgreich, dass der Ansatz in der SPL weitere Nachahmer fand. Daher existieren derzeit mehrere WE in der SPL, die auf Kajitas Modell basieren, vergleiche beispielsweise [39], [105] und [118]. Analog zu der Anwendung des *Motion Capturing* im Bereich der *open-loop* WE findet auch bei den *close-loop* Ansätzen der Einsatz dieser Technik statt. Die Nachahmung der Bewegung liegt hier nicht im Fokus, sondern es wird stattdessen versucht, ein geeignetes Modell des Roboters zu erlernen [14]. Eine Adaption der Technik für den Roboter NAO würde interessante Vergleiche mit klassischen Modellierungsansätzen ermöglichen.

Eine dritte Klasse bilden die Algorithmen der *hybriden* WE. Hierzu gehört zum Beispiel die Generierung von komplexen Bewegungen aus sogenannten **Bewegungsprimitiven**. Dies sind Teil-Trajektorien, die entweder offline vorberechnet wurden [44] oder erneut aus *Motion Capturing* Daten generalisiert wurden [86]. Welche dieser Trajektorien zur Bewegung ausgeführt werden, wird zur Laufzeit von einem Algorithmus entschieden. Somit reduzieren diese Algorithmen die Komplexität der WE zur Laufzeit, ermöglicht es aber dennoch, in gewissem Maß adaptiv auf die ausgeführte Bewegung zu reagieren.

## 5.2 NDD-Walking-Engine

Der Überblick in Kapitel 5.1 hat zeigt, dass es eine Vielzahl Herangehensweisen gibt, um Bewegungsmuster zur Fortbewegung humanoider Roboter zu generieren. Ein Grund hierfür ist verständlicherweise die Unterschiedlichkeit der verwendeten Hardware, doch auch die Anwendung und der Fokus der Forscher ist entscheidend für die Wahl der Herangehensweise. Aus diesem Grund muss vor der Entwicklung einer *Walking-Engine* zunächst die Anforderungen durch den geplanten Einsatz des Roboters geklärt werden. Hierzu sind Faktoren zu definieren, die als Gütemaß für die Bewertung und den Vergleich der Ansätze dienen können. Im Kontext des RoboCups sind hierfür folgende Eigenschaften in Betracht zu ziehen.

**Stabilität:** Stabilität ist für den RoboCup ein entscheidendes Gütemaß. Fußball zielt darauf ab, mit Objekten der Umgebung, in erster Linie dem Ball, durch geplante Kollision zu interagieren. Doch auch ungewollte Kollisionen mit anderen Robotern sind häufig. Laut Regeln ist der Boden des Spielfeldes plan und eben. In der Realität ist dies nicht immer der Fall und bereits kleine Unebenheiten können einen Roboter von der Größe des NAO aus dem Gleichgewicht bringen. Besonders machen ihm dabei Unterschiede in der Bodenreibung zu schaffen, die in erster Linie durch die aufgeklebten Feldlinien entstehen. Ein gestürzter NAO kann zwar mittels einer *Special Action* wieder autonom aufstehen, doch ist dies zeitraubend und ein Sturz führt zu einem hohen Gefährdungspotential für die Hardware.

**Geschwindigkeit:** Wie im realen Fußball bedeutet auch im RoboCup eine hohe Laufgeschwindigkeit einen entscheidenden Vorteil gegenüber dem Gegner. In der SPL wirkt sich dies in erster Linie positiv auf das Zweikampfverhalten aus und somit auf die Möglichkeit das Spiel aktiv zu beeinflussen. Es existieren in der Liga Geschwindigkeitsunterschiede zwischen den Teams, die so groß sind, dass es gar nicht erst zum Kampf um den Ball kommt. Doch auch wenn es zu einem Zweikampf kommt, hat nach den Regeln der Roboter einen deutlichen Vorteil, der eher am Ball war.

**Manövrierfähigkeit:** Der Gang sollte die Bewegungsfreiheit des Roboters so wenig wie möglich einschränken. Der menschliche Gang ist zu quasi-omnidirektionaler Fortbewegung fähig. Aufgrund seines Gelenkaufbaus präferiert er während seiner normalen Fortbewegung zwar bestimmte Gangrichtungen, doch kann er beispielsweise Seitenschritte jederzeit ausführen, um Hindernissen auszuweichen oder auf engem Raum zu navigieren. Es ist also wünschenswert, dass auch die WE eines humanoiden Roboters die Fähigkeit zur omnidirektionalen Fortbewegung weitestmöglich erhält. Im RoboCup hilft dies, die Zeit für die Ballannäherung zu verkürzen, erlaubt aber auch sich zu positionieren ohne den Sichtkontakt zum Ball zu verlieren.

**Präzision:** Zur erfolgreichen Interaktion mit dem Ball ist eine genaue Positionierung erforderlich. Auch wenn mit der Bewegung des Fußes beim Schuss eine leichte Deplazierung ausgeglichen werden kann, so erschwert dies doch das Ball-

Handling. Schlimmer ist es aber noch, sich bei der Ballannäherung über den geplanten Punkt hinaus zu bewegen und den Ball so unbeabsichtigt zu berühren. Dies führt durch das geringe Eigengewicht des Balls in der Regel dazu, dass der Ball rollt und die Entfernung nicht mehr durch eine Anpassung des Schusses ausgeglichen werden kann. Die Folge ist eine erneute Annäherung an den Ball. Eine exakte Ausführung der geplanten Bewegung bietet also im Spiel einen Vorteil.

**Reaktionsfähigkeit:** Um auf eine Veränderung der Roboterumgebung zu reagieren, ist es nötig die Fortbewegung möglichst zu jeder Zeit variieren zu können. Bedenkt man die Natur der dynamischen Stabilität, so wird klar, dass dieses Gütemaß dem der Geschwindigkeit konträr entgegenstehen muss. Doch ist es wünschenswert, dass eine WE die Reaktionsfähigkeit des Systems nicht über dieses notwendige Maß hinaus einschränkt.

**Verschleiß:** Viele Teile des Roboters NAO bestehen aus Kunststoff. Dies gilt nicht nur für die äußere Verkleidung, sondern auch für Komponenten des Bewegungsapparates, wie beispielsweise die Zahnräder. Somit ist nicht nur darauf zu achten, dass der Roboter während des Spieles möglichst keine Schäden durch Stürze davonträgt, sondern auch, dass die Bewegung diesen nicht unnötig verschleißt. Da sich dies aber in der Praxis nicht gänzlich vermeiden lässt, muss im Umkehrschluss die *Walking-Engine* in der Lage sein, den Verschleiß des Roboters zu kompensieren und dennoch die Stabilität zu gewährleisten.

**Energieeffizienz:** Die Energieeffizienz ist besonders in der mobilen Robotik von Interesse. Durch die Tatsache, dass die Energiereserve des Roboters durch die mitgeführte Energiequelle, in der Regel eine Batterie, beschränkt ist, stellt die Energie eine Ressource dar, die beachtet werden muss. Im RoboCup muss die Laufzeit des Roboters zumindest ausreichen, um eine Halbzeit zu spielen, da die Batterie vorher nicht gewechselt werden kann.

**Laufzeit:** Die Rechenleistung mobiler Hardware ist begrenzt. Da der Roboter neben der Fortbewegung in der Regel zeitgleich auch andere Aufgaben zu bewältigen hat, wird die Rechenkapazität, die der WE zur Verfügung steht darüber hinaus weiter eingeschränkt. Die Ausführung der Laufbewegungen erlaubt während der Fortbewegung keinen Verzug, da sonst die Stabilität nicht gewährleistet werden kann. Also kann die Laufzeit des Algorithmus mehr als eine Randbedingung als ein Gütemaß der WE angesehen werden.

Auch wenn die Diskussion dieser Anforderungen auf die Anwendung des RoboCup zugeschnitten sind, so fällt bei eingehender Betrachtung auf, dass diese bis zu einem gewissen Grad generalisiert werden können. Sie können also als allgemeines Gütemaß für eine WE dienen. Lediglich der Fokus und die Gewichtung werden sich abhängig von der Aufgabe verschieben. Es zeigt sich also, dass der RoboCup eine geeignete Applikation zur Entwicklung und zum Vergleich von Algorithmen ist, die auch in nicht spielerischen Aufgabengebieten Anwendung finden können.

Unter Berücksichtigung dieser Faktoren wurde die *NAO Devils Dortmund Walking-Engine*, kurz *NDD-WE*, für den Einsatz im RoboCup entwickelt. Durch die Erfahrung

der ersten Jahre der NAO-SPL kristallisiert sich zunächst die Geschwindigkeit der Laufbewegung als der entscheidende Faktor für die *Walking-Engine* heraus. Aldebaran stellte zwar für die Erzeugung von Laufmustern eine eigene WE zur Verfügung, doch mit dieser konnten Teams auch mit gut optimierten Parametern nur verhältnismäßig langsame Höchstgeschwindigkeiten von rund 8 cm/s erreichen. Darüber hinaus ermöglichte es die *Aldebaran-WE* nicht, eine einmal geplante Laufbewegung vorzeitig kontrolliert zu beenden. Somit war die Reaktionsfähigkeit dieser Lösung sehr stark eingeschränkt und als Folge konnten nur kurze Laufbewegungen sinnvoll geplant werden. Diese *Stop-and-Go*-Fortbewegung verlangsamte weiterhin die durchschnittliche Geschwindigkeit des Laufes.

Aufgrund der Vorarbeiten am IRF besteht Expertise im Bereich der evolutionär optimierten Laufbewegungen. Doch auch wenn Hebbel [45] zeigen konnte, dass sich evolutionäre Algorithmen zur Lauf-Optimierung von 4-beinigen Robotern auf humanoide Roboter übertragen lassen, so haben diese Algorithmen doch die Einschränkung nicht regelbar zu sein. Abweichungen von der Soll-Trajektorie können also nicht korrigiert werden. Hebbels Experimente wurden ohne externe Störungen durchgeführt und zudem untersuchte er lediglich eine Geradeaus-Laufbewegung ohne Richtungsänderung. Daher fiel dieser Nachteil nicht ins Gewicht und so konnten auch ohne Regelung zufriedenstellende Ergebnisse erzielt werden. Störungen in der Bewegung finden aber während eines RoboCup-Spiels regelmäßig statt und lassen sich auch in anderen Aufgaben-Szenarien nicht generell vermeiden. Hierfür sind nicht einmal in erster Linie die externen Störungen durch Kollisionen verantwortlich, sondern Störungen, die aus der Bewegung des Roboters selber resultieren. Betrachtet man den Roboter NAO genauer, wird dies schnell offensichtlich. Der Roboter ist, ähnlich dem Menschen, mit einem verhältnismäßig hohem Schwerpunkt ausgestattet. Auch wenn der Roboter an sich robust gebaut ist und durch seine Bauform eine hohe Stabilität im statischen Fall erreicht, so führt dieser Aufbau dazu, dass der Roboter bei einem normalen Gang ein Pendel mit hoher Maße auf langem Hebel darstellt. Dadurch wirken auf die Gelenke verhältnismäßig hohe Kräfte und führen somit zu einem nicht zu vernachlässigendem zeitlichen Verschleiß des Materials. Als Folge bekommen vor allem die Gelenke, deren Zahnräder aus Plastik bestehen, mit der Zeit immer mehr Spiel. Der Roboter wird labiler und nimmt mehr und mehr die Eigenschaften eines flexiblen Körpers an. Als Resultat führt seine Bewegung verstärkt dazu, dass der Oberkörper in angeregte Schwingungen versetzt wird, die seine Fortbewegung stören. Aus diesem Grund liegt der Schluss nahe, dass es einer *Open-Loop-Walking-Engine* bedarf, wenn diese für eine längere Zeit in einem realen Szenario eingesetzt werden soll. Gleichzeitig ist eine Regelung auch in der Lage kleinen Störungen zu nivellieren und ermöglicht es somit durch Laufruhe Geschwindigkeiten zu erreichen, die mit einer reinen *Close-Loop-WE* nicht stabil wären. So war es den *NAO Devils* bereits 2009, dem zweiten Jahr der neuen SPL, möglich, Geschwindigkeiten von 25 cm/s im Vorwärtslauf zu erreichen und somit die von Aldebaran theoretisch postulierte Maximalgeschwindigkeit von 20 cm/s zu überschreiten. Durch Weiterentwicklung und Tuning der Parameter waren sogar noch höhere Geschwindigkeiten möglich und so wurde mit der NDD-WE im Jahre 2010 in Singapur der Geschwindigkeitsrekord für den NAO von 44,5 cm/s aufgestellt<sup>1</sup>, der noch bis heute Bestand hat. Geschwindigkeiten dieser Größenordnung führen aber zu geringer Präzision der Bewegung und erreichen

<sup>1</sup>Das Video dieses Laufs ist zu finden unter [http://www.youtube.com/watch?v=3a0uQ1\\_e--k](http://www.youtube.com/watch?v=3a0uQ1_e--k).



auch mit Regelung nur eine geringe Robustheit gegen Störungen. Diese Abhängigkeit liefert ein konkretes Beispiel dafür, warum eine ausgewogene gewichtete Verteilung aus den vorgestellten Fakten entscheidend für die praktische Anwendung ist.

Die Diskussion in Kapitel 5.1 hat bereits gezeigt, dass es derzeit keinen Laufalgorithmus gibt, der für jede Anwendung dem anderen überlegen ist und so kann auch die NDD-WE nicht als die einzige Lösung des Problems verstanden werden. Andere Teams konnten demonstrieren, dass es dem Roboter NAO auch mit einer *Open-Loop*-WE möglich ist ein ausreichendes Maß an Stabilität zu erreichen, um konkurrenzfähig zu sein. Jedoch kann mit Recht behauptet werden, dass die NDD-WE eine *State-of-the-Art*-Lösung darstellt. Dies wird zum einen durch den konstanten Erfolg der NDD im Wettbewerb belegt. Zum anderen haben mehrere Teams erfolgreich auf den Veröffentlichungen der NDD-WE aufgebaut und durch Anpassung eigene WE entworfen, die auf deren grundlegenden Algorithmen basieren. Aus diesem Grund widmen sich die folgenden Kapitel der Vorstellung der NDD-*Walking-Engine*. Die Algorithmen und Teile der Auswertung wurden bereits in [24] und [26] veröffentlicht. Wichtige Fakten werden aber im Folgenden zwecks einer einordnenden Zusammenfassung nochmals aufgegriffen und ergänzt. Da die Zusammenfassung zwangsläufig nicht auf alle Fakten der Umsetzung detailliert eingehen kann, sei der Leser zudem zur weitergehenden Vertiefung auf das NDD CodeRelease [25] hingewiesen.

### 5.2.1 Erzeugung von Laufmustern

Grundlage einer Laufsteuerung ist zunächst die Planung der Bewegung, die in der gewünschten Fortbewegung resultiert. Als Eingabe aus dem *Behavior*-Modul wird hierzu der sogenannte *WalkRequest* erwartet. Dieser Vorgabevektor besteht aus den Geschwindigkeiten  $v_x$ ,  $v_y$  und der Rotationsgeschwindigkeit  $\dot{\varphi}$  um die Hochachse des Roboters. Die Aufgabestellung der Bewegungsplanung ist als das Problem aufzufassen, die Trajektorien der einzelnen Körperteile so vorzugeben, dass die resultierende Bewegung zu dem gewünschten stabilen Gang führt. Die NDD-WE macht sich die Tatsache zu Nutze, dass die kinematischen Ketten untereinander nur durch den Torso verbunden sind. Da sie sich dadurch nur indirekt gegenseitig beeinflussen, können sie voneinander entkoppelt betrachtet werden<sup>2</sup>. Das Resultat ist ein mehrstufiges Planungsverfahren. Jede bereits geplante Bewegung führt zu Nebenbedingungen für die übrigen Gelenktrajektorien, wenn die Stabilität weiterhin gewährleistet sein soll. Die Abfolge wird basierend auf der Priorität der Bewegung gewählt:

1. Berechnung der Fußaufsatzpunkte.
2. Berechnung der Schwungfuß-Trajektorie.
3. Generierung der CoM-Trajektorie.
4. Berechnung der Armbewegungen.

---

<sup>2</sup>Die einzige Ausnahme stellt das *HipYaWPitch*-Gelenk dar, auf deren Behandlung aber bereits im Rahmen der Kinematikbetrachtung in Kapitel 3.1.2 eingegangen wurde.



Eine tiefgehendere Beschreibung der Funktionsweise ist der einzelnen WE-Modul Implementierungen ist im NDD-Teamreport [25] zu finden. Zur Gesamtübersicht über die Struktur der NDD-WE von der Eingabe  $(v_x, v_y, \dot{\varphi})$  bis zur Abfolge der Konfigurationen des Roboters ist diesem aber Abbildung 5.2.1 entnommen. Die Kästchen markieren die einzelnen Planungsschritte und - im Vorgriff auf den weiteren Verlauf - auch die Regelungsmodule für den Sensorrückfluss. Die Farbe der Pfeile gekennzeichnet, in welchem Koordinatensystem die jeweilige Berechnung stattfindet - Rot WKS, Blau ReKS.

Während der Fortbewegung interagieren primär die Füße des Roboters mit der Umgebung. Da sich aus diesen per Definition direkt die Position und Pose des Roboters und als Folge auch seine Geschwindigkeit ergeben, ist es naheliegend die Aufsatzzpunkte der Füße zuerst zu wählen. Die Trajektorie des Roboters, die direkt aus den Fußpunkten folgt, kann so ohne Berücksichtigung von Nebenbedingungen frei gewählt werden. Diese Aufgabe obliegt dem sogenannten *PatternGenerator*-Modul, das den jeweils nächsten Punkt aus dem Eingabevektor  $(v_x, v_y, \dot{\varphi})$  unter Berücksichtigung der kinematischen Randbedingungen berechnet. Die Frequenz der Schritte wird während eines Aufrufs der WE konstant gehalten, so dass die Geschwindigkeit des Roboters nur durch die Wahl der Schrittweite angepasst werden kann. Diese Wahl erscheint zunächst willkürlich, da die Geschwindigkeit ebenfalls durch die Frequenz beeinflusst werden kann, doch wurde sie bewusst getroffen. Es hat sich gezeigt, dass der Roboter NAO bei einigen Frequenzen stabiler und ruhiger läuft, als bei anderen. Diese Tatsache lässt sich theoretisch untermauern, wenn man bedenkt, dass es sich bei dem realen Roboter nicht um einen vollkommen starren Festkörper handelt. Vor allem die Gelenke führen Flexibilitäten ins System ein, wie Urbann et al. in [113] belegen konnten. Als Folge verhält sich der Körper wie ein Feder-Masse-System, das durch periodisches Auftreten der Füße zu einer erzwungenen Schwingung angeregt wird. Erfolgt diese Anregung nun mit der Eigenfrequenz eines Teil-, oder gar des ganzen Systems, so treten Resonanzschwingungen auf. Diese können sich nach wenigen Schritten soweit aufschwingen, dass der Roboter nicht mehr zu kontrollieren ist. Als Folge steht nur ein Teil des Frequenzbandes für die WE zur Verfügung<sup>3</sup>. Da die Resonanzfrequenz vom Roboter und dessen Verschleiß abhängig ist, wird das Ändern der Frequenz als potentiell kritisch erachtet. Weiterhin führt jede Änderung der Schrittfrequenz zu einer Veränderung in der Periode der Bewegung. Dies hat einen Ruck im System zur Folge, wenn sie abrupt durchgeführt wird. Ähnliches gilt zwar auch für plötzliche Änderungen der Schrittweite, doch können diese durch lineare Anpassung der Beschleunigung leichter begrenzt und somit algorithmisch kontrolliert werden. Störungen im System können somit reduziert werden.

Als Alternative zur reaktiven Verhaltensplanung existiert die Option den *PatternGenerator* mit einer zeitlichen Abfolge von *WalkRequests* als Eingabe zu versorgen. Diese werden vom Verhalten so berechnet, dass auf Basis der Lokalisierung explizit eine Trajektorie des Roboters vorgegeben und somit implizit auch die Position der Füße entlang dieser bestimmt wird. Dieses Vorgehen ermöglicht eine komplexere Pfadplanung des Roboters und erleichtert beispielsweise Ballannäherung und Hindernisvermeidung. Darüber hinaus bietet es den Vorteil, dass der Roboter auch Hindernisse auf dem Boden gezielt überschreiten und diesen somit optimal ausweichen kann. In der *OpenChallen-*

<sup>3</sup>Nach einer groben Abschätzung dürfte sich die Resonanzfrequenz bei ungefähr 1,5 Hz befinden, so dass durch sie das sinnvolle Frequenzband sogar geteilt wird.

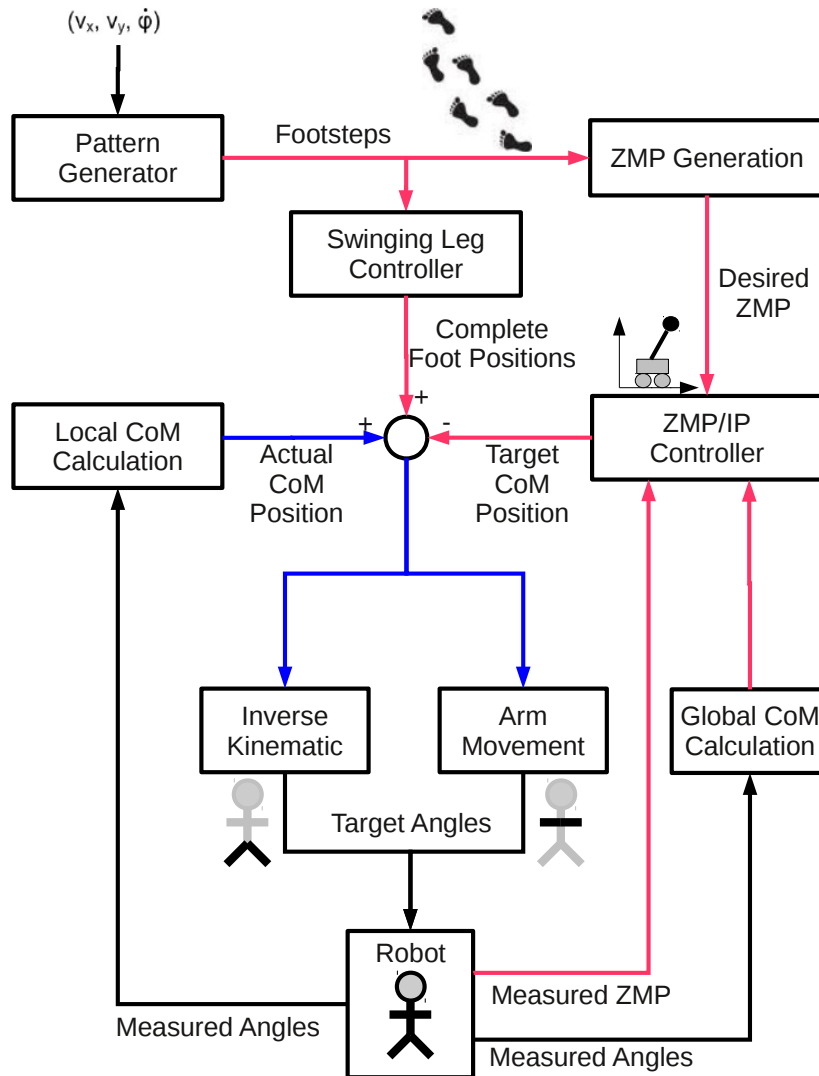


Abbildung 5.2.1: Übersicht der NDD-WE Struktur entnommen aus [25].

ge<sup>4</sup> 2010 konnten die NDD dies mit einem Hindernisparcours demonstrieren. Auf diese Weise lassen sich Forschungen in der Pfadplanung wie beispielsweise [15] ohne weiteres auch mit der *NDD-Walking-Engine* durchführen.

Nachdem die Trajektorie des Roboters durch die Aufsatzzpunkte der Füße eindeutig definiert ist, werden die Bewegungen der Füße so geplant, dass diese korrekt aufgesetzt werden. Hierzu muss gewährleistet werden, dass in der *Single-Support*-Phase des Gangs die Trajektorie des Schwungfußes in dem geplanten Aufsatzzpunkt endet. Diese Aufgabe übernimmt der *SwingLegController*. Hebbel zeigt in [45], dass die Wahl der Trajektorie des Schwungfußes einen Einfluss auf die Stabilität der Bewegung hat. Er untersucht verschiedene einfache parametrierbare Trajektorienformen, wie Halbkreise oder Rechtecke, und kommt zu dem Schluss, dass die Form der Halbellipse bei der Optimierung zu den besten Ergebnissen führt. Dies erklärt er damit, dass diese Trajektorie den Fuß während des Anhebens und vor allem des Absetzens nur in  $z$ -Richtung bewegt<sup>5</sup>. Der Fuß wird also nicht gleichzeitig parallel zum Boden bewegt, wie dies beispielsweise bei der Sinus-Trajektorie der Fall ist. Dies führt zwar zu einer Reduktion der maximalen Schrittweite bei vorgegebener Frequenz und Fußgeschwindigkeit, hat aber den Vorteil Störungen zu reduzieren, sollte der Fuß vorzeitig den Boden berühren. Die von Hebbel optimierten Parameter betreffen beispielsweise Schrittweite und -höhe, führen jedoch lediglich zu einer symmetrischen Veränderung der Trajektorie. Um diesen Nachteil auszugleichen wird für die Berechnung des *SwingLegControllers* die Schwungtrajektorie mittels einer Bézier-Kurve abgebildet. Über die gewählten 4 Kontrollpunkte kann ebenfalls die Schritthöhe angepasst werden, das Verhalten der Füße beim Abheben und Aufsetzen aber unabhängig voneinander beeinflusst werden.

Nachdem nun die Fußpunkte und -trajektorien dem *WalkRequest* entsprechend optimal gewählt wurden, erfolgt die Planung der restlichen Gliedmaßen. Hierbei muss nun die Randbedingung der Stabilität beachtet werden. Durch die Fußpunkte sind die *Support*-Flächen des Roboters entlang der Trajektorie bereits vorgegeben. Da der Roboter NAO mit der NDD-WE in der Lage sein soll möglichst hohe Geschwindigkeiten zu erreichen, ist es unumgänglich für den Gang dynamische Stabilität zu fordern. Das Stabilitätskriterium muss also zwangsläufig der ZMP sein. Daher wird logischerweise für die weitere Bewegung die gewünschte ZMP-Trajektorie vorgegeben, anstatt direkt die einzelnen Gliedmaßen zu planen. Der Referenz-ZMP-Verlauf erreicht die optimale Stabilität für die vorgegebene Support-Polygone. Diese Aufgabe übernimmt das *ZMP-Generation*-Modul. Der Gang des Roboters wechselt fortwährend zwischen dem linken und rechten Fuß als Standbein. Folglich muss der ZMP zwischen diesen beiden *Single-Support*-Phasen im ReKS von Links nach Rechts wandern. Dies geschieht überwiegend während der *Double-Support*-Phase. Der Verlauf des ZMPs kann aber nicht direkt beeinflusst werden, sondern ergibt sich vielmehr aus der Bewegung und Beschleunigung des CoM. Aufgrund der Massenträgheit kann dieser aber nicht abrupt beliebig beschleunigt werden. Folglich muss die CoM-Trajektorie im Voraus berechnet werden, so dass sich als Resultat die gewünschte ZMP-Trajektorie einstellt. Dies wird mit Hilfe des Konzepts eines *Preview-Controllers* erreicht und vom Modul *ZMP/IP-Controller* berechnet. Da

<sup>4</sup>Die *OpenChallenge* ist ein Wettbewerb im RoboCup, bei dem die Teams in einer offenen Demonstration wissenschaftlicher oder technischer Herausforderungen gegeneinander antreten.

<sup>5</sup>Dies gilt natürlich auch für das Rechteck, doch wird durch diese Form die Schrittweite zu sehr reduziert.

dies aber sehr eng mit dem eigentlichen Regelungskonzept verwunden ist, wird hierauf erst in Kapitel 5.2.2 genauer eingegangen. Zur Veranschaulichung zeigt Abbildung 5.2.2 dennoch bereits an dieser Stelle exemplarisch den resultierenden Verlauf eines CoM zu einer vorgegebenen ZMP-Trajektorie. Es ist zu beachten, dass eine Trajektorie immer eine zeitliche Komponente beinhaltet, die in diese Darstellung natürlich nicht einfließen kann. Somit kann diese Art von Diagramm immer nur den Pfad der Bewegung abbilden. In der Abbildung ist ein abrupter Wechsel des ZMPs vom rechten auf den linken Fuß abgebildet. Dieses Verhalten ist theoretisch ideal, da der ZMP sich zu jeder Zeit auf der Mittelachse des Standfußes befindet. Folglich ist zu jedem Zeitpunkt ein maximales Maß an Stabilität gewährleistet. Dies ist aber nur mit einer unendlich hohen Beschleunigung des CoM zu erreichen. Die Trajektorie ist somit zwar ideal, kann aber in der Praxis mit dem Roboter nicht angesteuert werden. Derart extreme Beschleunigungsänderungen führen zudem zu einem hohen Ruck, der für Unruhe im System des Roboters sorgt. Folge können ungewollte Stabilitätsprobleme sein und daher sollte der Verlauf der ZMP-Trajektorie ruckbegrenzt sein. Um in der Parametrisierung flexibel zu sein, wird auch hier erneut der Verlauf der Trajektorie über eine Bézier-Kurve mit 4 Kontrollpunkten vorgegeben.

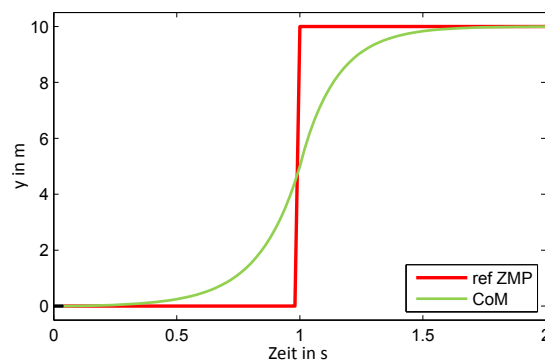


Abbildung 5.2.2: CoM-Verlauf um eine gegebene ZMP-Trajektorie zu erreichen.

Der *ZMP/IP-Controller* verwendet die Rückwärtstransformation des Roboters, um die Trajektorien der Gelenke so zu berechnen, dass die Bewegung der einzelnen Festkörper in der gewünschten Bewegung des Gesamt-CoM resultiert. Eine Sonderstellung nimmt die Bewegung der Arme ein. Diese kann optional vorgegeben werden. In dem Fall stellt sie eine weitere Randbedingung für die Bewegung dar. Diese Designentscheidung wurde aufgrund einer praktischen Beobachtung getroffen, die auch theoretisch leicht erklärt werden kann. Je nach Bodenbeschaffenheit und Laufgeschwindigkeit des Roboters läuft der Roboter bei Ansteuerung eines reinen Vorwärtslaufs in  $x$ -Richtung eine Kurve. Erklärt werden kann dies, da im ZMP zwar die Drehmomente in der  $x$ - und  $y$ -Richtung gleich Null sind, aber nicht zwangsläufig auch um die  $z$ -Achse. Als Folge kann es vorkommen, dass der Roboter sich bei der Bewegung ungewollt um seine Hochachse dreht. Auf dieses Phänomen wird Kapitel 6.2 detaillierter eingehen. Indem durch eine asymmetrische Bewegung der Arme ein ausgleichendes Drehmoment in Gegenrichtung erzeugt wird, kann dieser Effekt ausgeglichen werden,

## 5.2.2 Regelung der Bewegung

Sobald geklärt ist, wie die CoM-Trajektorie aus der ZMP-Trajektorie folgt, ist das im vorhergehenden Kapitel beschriebene Verfahren in der Lage, den Roboter NAO mittels einer *Open-Loop*-WE fortzubewegen. Die geplanten Bewegungen sind in der Theorie stabil und können unter geeigneten Bedingungen auch in der Praxis einen Roboter im Gleichgewicht halten. Unzulänglichkeiten in der Berechnung der CoM-Bewegung könnten im *Open-Loop*-Fall durch Parametrisierung der ZMP- und der Fußtrajektorie ausgeglichen werden. Diese Anpassung an das reale System könnte beispielsweise eine evolutionäre Optimierung übernehmen. Allerdings wurde der verfolgte Ansatz ausgewählt, da er einen entscheidenden Vorteil gegenüber einem solchen Verfahren bietet. Die Bewegung des Roboters ergibt sich nicht implizit aufgrund einer unbekanntenen Optimierung, sondern die logischen Zusammenhänge sind explizit bekannt. Durch die mathematische Beschreibung kann bei Abweichungen von der geplanten Bewegung mittels eines Reglers korrigierend eingegriffen werden. Voraussetzung ist allerdings, dass mittels geeigneter Sensorinformationen der Zustand des Systems überwacht und dadurch der Regelkreis geschlossen werden kann. So kann eine höhere Stabilität auch in unvorhergesehenen Situationen erreicht und darüber hinaus systematische Fehler des Systems zu einem gewissen Grad ausgeglichen werden.

Die vorangegangene Beschreibung der Bewegungsplanung lässt noch offen, wie die korrespondierende CoM-Trajektorie zur ZMP-Trajektorie berechnet wird. Hierzu verwendet die NDD-WE ein vereinfachtes Modell der Roboterdynamik, das den Körper lediglich auf seinen CoM reduziert. Man spricht von der Reduktion auf ein *Einkörper*-Modell. Dieser Ansatz wird von vielen ZMP-basierten WE-Algorithmen verfolgt und ist bekannt als *Inverted-Pendulum*. Die Idee basiert auf der Beobachtung, dass während der *Single-Support*-Phase nur ein Bein den Boden berührt. Somit lässt sich durch Vereinfachung des Modells die Bewegung des Roboters als die eines invertierten starren Pendels auffassen. Kajita et al. schränken diese Pendelbewegung in [56] noch weiter ein, indem für den CoM nur Bewegungen in einer bodenparallelen Ebene zugelassen werden. Dies kann erreicht werden, indem während der Pendelbewegung durch Streckung des Standbeines die Höhe des CoM entgegen der normalen Pendelbewegung konstant gehalten wird. Kajita prägt hierfür den Begriff des *3D-Linear-Inverted-Pendulum-Modell*, kurz 3D-LIPM. Gegenüber dem einfachen *Inverted-Pendulum* wird die Dynamik des Systems noch weiter vereinfacht und ist nun mit ausschließlich linearen Gleichungen beschreibbar. Im Folgenden wird Kajitas Herleitung der 3D-LIPM Gleichungen nachvollzogen, um ihre Implikationen besser verstehen zu können.

Die Ebene, in der der CoM sich bewegt, ist bestimmt durch ihren Normalenvektor  $(k_x, k_y, -1)$  und den Schnittpunkt  $z_c$  mit der  $z$ -Achse der ReKZ. Wird weiterhin davon ausgegangen, dass der Boden eben ist, gilt durch die Parallelität in jedem Fall  $k_x = k_y = 0$ . Sei nun  $m$  die Masse des Pendels,  $g$  die Erdbeschleunigung und  $M_x$  und  $M_y$  die Drehmomente um die  $x$ - und  $y$ -Achsen, so ist die Dynamik des 3D-LIPM nach Kajita [55] beschrieben durch:

$$\ddot{y} = \frac{g}{z_c} y - \frac{1}{m z_c} M_x, \quad (5.2.1)$$

$$\ddot{x} = \frac{g}{z_c} x + \frac{1}{m z_c} M_y. \quad (5.2.2)$$

Es sei angemerkt, dass Kajita die Gültigkeit dieser Gleichungen auch im Falle eines schrägen Bodens belegen kann. Hierzu müssen allerdings weitere Einschränkungen angenommen werden. Da dies für die Anwendung im RoboCup irrelevant ist, sei hierzu nur auf Kajitas Ausarbeitung in [56] verwiesen. Durch die Bedingung des 3D-LIPM kann die Position  $p = (p_x, p_y)$  des ZMPs auf dem Boden berechnet werden durch:

$$p_x = -\frac{M_y}{m g}, \quad (5.2.3)$$

$$p_y = \frac{M_x}{m g}. \quad (5.2.4)$$

Einsetzen der Gleichungen (5.2.3) und (5.2.4) in die Gleichungen (5.2.1) und (5.2.2) liefert erneut die folgenden ZMP-Gleichungen, die bereits vom *Cart-Table*-Modell bekannt sind - vergleiche Kapitel 4.4.2:

$$p_x = x - \frac{z_c}{g} \ddot{x}, \quad (5.2.5)$$

$$p_y = y - \frac{z_c}{g} \ddot{y}. \quad (5.2.6)$$

### 5.2.2.1 Preview-Regler

Unter der Voraussetzung  $z_c = const$ , folgt aus den Formeln (5.2.5) und (5.2.6), dass für das 3D-LIPM die Position des ZMPs lediglich von der Position und Beschleunigung des CoM in der Pendelebene abhängt. Weiterhin ist ersichtlich, dass die  $x$ - und  $y$ -Komponenten unabhängig sind und somit getrennt voneinander betrachtet werden können. Die Ansteuerung einer gewünschten ZMP-Trajektorie reduziert sich dann auf die Bewegungen des CoM auf einer entsprechenden Trajektorie für jede Richtung. Das System kann also entkoppelt und zugleich vereinfacht werden, indem es durch zwei Systeme repräsentiert wird. Daher wird im Folgenden o.B.d.A lediglich der Regler in  $x$ -Richtung betrachtet. Die Herleitung für den  $y$ -Regler folgt analog.

Bereits in Kapitel 5.2.1 wurde diskutiert, dass aufgrund der Dynamik des Systems der geforderte Verlauf des ZMPs nicht allein durch abrupten Positionswechsel des CoM zum aktuellen Zeitpunkt erreicht werden kann. Vielmehr ist es nötig den CoM bereits vor dem ZMP in Bewegung zu versetzen. Der zukünftige Verlauf des ZMPs muss also zwingend in die Planung der CoM-Trajektorie einbezogen werden. Dies führt zu einer sogenannten *Preview*-Phase in der Bewegungsplanung der NDD-WE. Da in diesem Zeitfenster der Verlauf des ZMPs feststehen muss, impliziert dies unweigerlich eine gewisse zeitliche Verzögerung für Änderungen im Bewegungsablauf<sup>6</sup>. Die grundlegende

<sup>6</sup>Es stellt sich die Frage, ob dies nicht zu einer Einschränkung der *Reaktionsfähigkeit* des Systems führt. Eine Diskussion dieser Problematik ist in Kapitel 5.4.4 zu finden.

Idee, vorausschauende Informationen in einen Regler zu integrieren, wurde erstmalig von Sheridan im Jahre 1966 eingeführt [98]. Das Prinzip ist unter dem Namen *Preview-Control* bekannt geworden und seitdem auf unterschiedliche Probleme der Regelungstechnik erfolgreich angewendet worden. Kajita wandte das Konzept erstmalig in [55] auf zweibeinige Roboter an und übertrug hierzu den Regler von Katayama [63] für ein „[...]discrete-time system subject to a time-varying demand and an unmeasurable constant disturbance.“<sup>7</sup> Im Gegensatz zu der hier vorgestellten NDD-WE behandelt Kajita die adaptiven Sensorinformationen nicht direkt, sondern durch die Einführung eines *Auxiliary-ZMPs*. Mit diesem wird der Referenz-ZMP in Echtzeit angepasst, sollte es der Ausgleich unvorhergesehener externer Kräfte oder Fehler im Modell des Roboters zusätzliche Drehmomente erfordern. Dieser *Auxiliary-ZMP* führt jedoch dazu, dass das System mit der Zeit divergiert und erfordert zusätzliche stabilisierende Maßnahmen. Hierzu führt Kajita eine virtuelle Zeitverschiebung der Referenz-ZMP Trajektorie ein, die ausführlich in [57] beschrieben ist. Durch diese Anpassung kann das System aber nur für kurze Zeitspannen betrieben werden, bevor ein *Reset* erforderlich wird, der die virtuelle Zeitverschiebung neu startet. Dies erschwert insbesondere den Ausgleich konstanter externer Fehlerquellen, beispielsweise einer nicht modellierten Neigung der Lauffläche, da in diesem Fall der *Auxiliary-ZMP* periodisch angepasst werden muss. Aus diesem Grund nutzt die NDD-WE eine andere Adaption des *Preview*-Reglers von Katayama, die eine natürlichere Integration der Sensorinformationen in das System ermöglicht. Durch Anwendung von Grundlagen der Regelungstechnik ist es möglich ein *Observer*-Modell anzuwenden, dass die Sensoren direkt in das System integriert. Hierauf wird in Kapitel 5.2.2.2 weiter eingegangen.

Zunächst wird mit dem Entwurf des NDD-*Preview*-Reglers analog zu dem Entwurf von Kajita verfahren. Seine Herleitung wird im Folgenden aufgegriffen und kurz skizziert, um zu verstehen ab welchem Punkt die NDD-WE eine anderen Lösungsansatz verfolgt. Weitere Details können [57] entnommen werden. Das System wird weiterhin durch das bereits beschriebene *Cart-Table*-Modell abgebildet. Der energetische Zustand des Systems ist eindeutig durch  $[x, \dot{x}, \ddot{x}]^T$  beschrieben. Betrachtet man die ZMP-Gleichung (5.2.5) genauer, so erkennt man, dass das System sich auch vollständig durch den Zustandsvektor  $[x, \dot{x}, p_x]^T$  beschreiben lässt. Kajita erweitert nun das *Cart-Table*-Modell um einen Wagen, auf dem sich das gesamte 3D-LIPM bewegen kann. Damit kann nun die Geschwindigkeit  $\dot{p}$  des ZMP als Systemeingang gewählt werden:

$$\dot{p} = v. \quad (5.2.7)$$

Durch diese Wahl wird die Führungsgröße des Regelalgorithmus in die Systembeschreibung integriert und die resultierenden Gleichungen signifikant vereinfacht. Kajitas formt damit die Systemgleichung um, zur Systembeschreibung:

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ p_x \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{g}{z_c} & 0 & -\frac{g}{z_c} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ p_x \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v_x, \quad (5.2.8)$$

$$p_x = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ p_x \end{bmatrix}. \quad (5.2.9)$$

<sup>7</sup>Katayama's wählte als Beispiel System ein Kraftwerk.



Katayama betrachtet diese Art von System als LQ<sup>8</sup>-Problem und schlägt daher vor, dass System mittels eines LQR<sup>9</sup> durch einen optimalen *Servo*-Regler zu lösen. Nach [63] führt die Diskretisierung der Gleichungen (5.2.8) und (5.2.9) in die Zeitschritte  $\Delta t$  zu:

$$\mathbf{x}(k+1) = \mathbf{A} \mathbf{x}(k) + \mathbf{b} v(k), \quad (5.2.10)$$

$$p(k) = \mathbf{c} \mathbf{x}(k), \quad (5.2.11)$$

mit

$$\mathbf{x}(k) = [x(k\Delta t), \dot{x}(k\Delta t), p(k\Delta t)]^T, \quad (5.2.12)$$

$$v(k) = v(k\Delta t), \quad (5.2.13)$$

$$p(k) = p(k\Delta t). \quad (5.2.14)$$

Um für einen gegebenen Referenz ZMP Verlauf  $p^{ref}$  eine Regelung zu erreichen wird analog zu Kajita das Bewertungsmaß  $J$  definiert:

$$J = \sum_{j=1}^{\infty} \left\{ Q_e \left[ p^{ref}(j) - p(j) \right]^2 + R v^2(j) \right\}. \quad (5.2.15)$$

mit nicht-negativen Gewichten  $Q_e$  zur Beschreibung der Abweichungen durch den Führungsfehler und  $R$  zur Minimierung des Regelungseinflusses. Die Optimierung dieses Gütemaßes versucht also den Regelfehler zu minimieren und eine bestraft dabei gleichzeitig die eine hohe Änderungsrate der Führungsgröße. Zur Minimierung von  $J$  wendet Kajita nun den folgenden Regler mit  $N$  *Preview*-Schritten an:

$$v(k) = -\mathbf{K} \mathbf{x}(k) + \sum_{j=1}^N G_d(j) p^{ref}(k+j), \quad (5.2.16)$$

wobei  $\mathbf{K}$  und  $G_d(i)$  nach Katayama [63] gegeben sind durch:

$$\mathbf{K} = [R + \mathbf{b}^T \mathbf{P} \mathbf{b}]^{-1} \mathbf{b}^T \mathbf{P} \mathbf{A}, \quad (5.2.17)$$

$$G_d(j) = [R + \mathbf{b}^T \mathbf{P} \mathbf{b}]^{-1} \mathbf{b}^T \left[ [\mathbf{A} - \mathbf{b} \mathbf{K}]^T \right]^{j-1} \mathbf{c}^T Q_e. \quad (5.2.18)$$

Die Größe  $\mathbf{P}$  ist dabei bestimmt durch die Lösung folgender zeitdiskreten Ricatti-Differenzialgleichung:

$$\mathbf{P} = \mathbf{A}^T \mathbf{P} \mathbf{A} + \mathbf{C}^T Q_e \mathbf{C} - \mathbf{A}^T \mathbf{P} \mathbf{B} [R + \mathbf{B}^T \mathbf{P} \mathbf{B}]^{-1} \mathbf{B}^T \mathbf{P} \mathbf{A}. \quad (5.2.19)$$

Es muss für den NAO sichergestellt werden, dass die eindeutige, stabilisierende Lösung  $\mathbf{P}$  dieser existiert. Die Bedingung hierfür ist durch folgenden Kriterium gegeben:

„The pair (A,B) must be stabilizable (that is, all eigenvalues of A outside the unit disk must be controllable). In addition, the associated symplectic pencil must have no eigenvalue on the unit circle. [...] Sufficient conditions for this to hold are (Q, A) detectable when S=0 and R>0, or [...]. [75]“

---

<sup>8</sup>Linear-Quadratic

<sup>9</sup>linear-quadratic regulator



Die *Steuerbarkeit* eines linearen zeitinvarianten Systems beschreibt dessen Eigenschaft von jedem Ausgangszustand in endlicher Zeit durch eine Steuerfunktion  $u(t)$  in jeden beliebigen Endzustand überführbar zu sein. Dies ist mittels des Kriteriums nach Kalman leicht nachzuweisen [72], da für die Steuerbarkeitsmatrix  $\mathbf{S}_S$  des Systems  $(A, B)$  gilt:

$$\text{Rang}(\mathbf{S}_S) = 3, \quad (5.2.20)$$

mit

$$\mathbf{S}_S = [\mathbf{b} \quad \mathbf{A}\mathbf{b} \quad \mathbf{A}^2\mathbf{b}] = \begin{bmatrix} 0 & 0 & -\frac{g}{z_c} \\ 0 & -\frac{g}{z_c} & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad (5.2.21)$$

solange sichergestellt ist, dass  $-\frac{g}{z_c}$  nicht den Wert Null annimmt. Die geforderte Nichtexistenz der Eigenwerte kann mit dem Algorithmus aus [3] nachgewiesen werden, wenn geeignete Parameter  $\Delta t$  and  $z_c$  vorgegeben sind. Zur Vertiefung der Problematik sei auf die Ausarbeitung von Fassbender und Benner [29] und Mao-Lin [74] verwiesen. Es bleibt aber festzuhalten, dass die Lösung von der variablen Systemgröße  $z_c$  abhängt. Daher wird offline für mögliche CoM-Höhen in diskreten Schritten mit Hilfe des MATLAB DARE<sup>10</sup>-Algorithmus die Lösung vorberechnet.

Kajita schreibt abschließend den *Pattern-Generator* so um, dass die Verstärkungen in die 3D-LIPM Systembeschreibung eingeschlossen werden:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}^P \mathbf{x}(k) + \mathbf{b} u(k), \\ p(k) &= \mathbf{c} \mathbf{x}(k), \end{aligned} \quad (5.2.22)$$

mit  $\mathbf{A}^P = \mathbf{A} - \mathbf{b} \mathbf{K}$ . Der Systemeingang  $u(k)$  ist gegeben durch den Referenz-ZMP  $p^{ref}$  und die *Preview-Verstärkungen* zu:

$$u(k) = \sum_{j=1}^N G_d(j) p^{ref}(k+j). \quad (5.2.23)$$

Anders, als Gleichung (5.2.9) nahelegt, wird demnach nicht das Modell  $\mathbf{A}$  auf das System angewendet und der ZMP mittels der Eingangsgröße  $v(k)$  entlang der Referenz-Trajektorie bewegt. Vielmehr wird  $\mathbf{A}^P$  an die Regelgröße angepasst. Die Steuergröße  $u(k)$  integriert zudem bereits die notwendige Anpassung, um das System durch die *Preview-Schritte* so zu regeln, dass der gewünschte ZMP erreicht wird.

Die Wahl der Anzahl  $N$  der *Preview-Schritte* wirkt sich auf die Gleichungen (5.2.15) und (5.2.16)<sup>12</sup> aus. Ihre Auswirkung ist in Abbildung 5.2.3 exemplarisch dargestellt. Während das Bewertungsmaß  $J$  einen unendlichen Horizont aufspannt, kann der reale Regler lediglich eine endliche Anzahl Schritte betrachten. Dies resultiert in einer Abweichung von der optimalen Lösung. Größere  $N$ , und somit eine längere *Preview-Phase*, liefern eine bessere Annäherung an das Optimum. Das Resultat ist eine bessere Führung des Systems. Dies wird aber durch eine zusätzliche Verzögerung erkauft und beeinträchtigt die Reaktionsfähigkeit des Roboters. Für kleine Größen  $N$  des *Preview-Fensters*

<sup>10</sup>Discrete-time Algebraic Riccati Equation<sup>11</sup>

<sup>12</sup>Weiterhin beeinflusst sie auch die späteren Gleichungen (5.2.33) und (5.2.32).

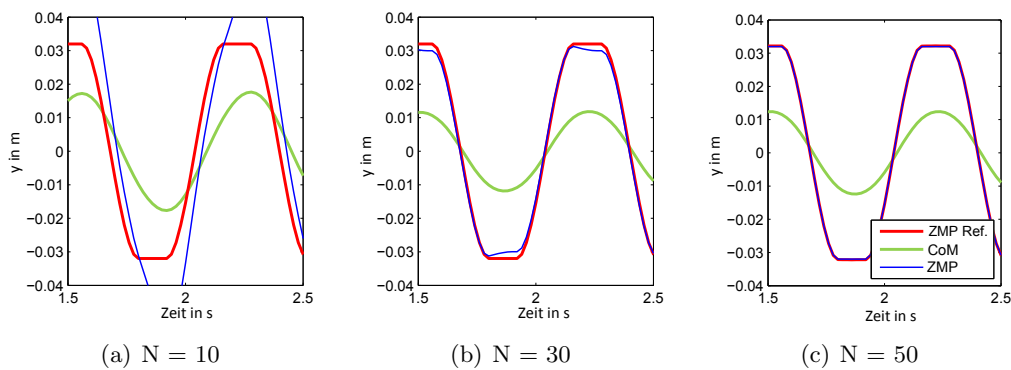


Abbildung 5.2.3: Referenz-ZMP, berechneter ZMP und CoM für unterschiedliche  $N$  der *Preview*-Verstärkung.

bewegt der CoM sich jedoch nicht früh genug und führt somit zu größeren Regelungsabweichungen. Für den Roboter NAO zeigen *Preview*-Fenster mit einer Länge  $N$  größer als ein Doppelschritt in der Praxis lediglich insignifikante Verbesserungen der Regelabweichung. Diese Wahl führt beim realen Roboter nur zur einer kaum merklichen Verzögerung, da schnelle Änderungen der ZMP-Trajektorie aufgrund der dynamischen Beschränkungen des NAO ohnehin eine Instabilität des Systems zur Folge hätten.

### 5.2.2.2 Integration des Beobachter-Modells

Der vorgestellte *Preview*-Regler ist in der Lage Laufmuster zu berechnen, die die Bedingung eines stabilen ZMP-Verlaufs erfüllen. Untersuchungen von Kajita zeigen aber, dass der Regler so durch seinen zeitliche Verzug bedingt nicht ausreichend schnell reagieren kann, um die Stabilität auch unter Störeinflüssen zu gewährleisten. Solche entstehen aber sowohl durch Unzulänglichkeiten im Modell, als auch durch externe Kräfte auf den Roboter. Stöße sind im Kontext des RoboCups unvermeidbar. Doch auch das dem Regler zu Grunde liegende Modell kann immer nur eine Abstraktion der Realität sein. Das 3D-LIPM wurde darüber hinaus bewusst vereinfacht, um zu linearen Gleichungen zu führen. Als Folge ergeben sich besonders in der stabilitätskritischen *Single-Support*-Phase nicht modellierten Abweichungen des ZMPs aus der Dynamik des Schwungbeines. Dieser Effekt nimmt mit zunehmender Laufgeschwindigkeit und -frequenz zu, da die größeren Beschleunigungen des Beines unweigerlich zu größeren Abweichungen der Drehmomente führen. Diese könnten nur durch Einführung eines komplexeren gekoppelten Pendel-Modells Beachtung finden, vergleiche beispielsweise [53] oder [106], würde aber zu einer merklichen Laufzeit-Erhöhung der NDD-WE führen<sup>13</sup>. Doch auch mit einem exakteren Modell des Roboters treten zwangsläufig Abweichungen von der Realität auf. Verstärkt wird dieses Phänomen, da der Roboter sich unweigerlich durch Verschleiß oder Defekte mit der Zeit verändert, das Modell diese Änderungen aber nicht widerspiegeln kann. Da zudem äußere Störeinflüsse, beispielsweise durch Kollisions-

<sup>13</sup>Es sei an dieser Stelle angemerkt, dass mit der Einführung des NAO V4 durch den Atom-Prozessor vermutlich genügend Rechenleistung zur Verfügung steht, um die Vorteile durch direkten Vergleich beider Modelle zu untersuchen.

sionen, in der Realität nicht ausgeschlossen werden können, ist es in der Praxis nicht ausreichend, allein auf dem *Preview*-Regler basierend einen Lauf anzusteuern. Kajita schlägt hierzu in die Integration der Sensorrückkopplung durch einen *Auxiliary*-ZMPs in das Regelsystem vor, um so solche Störungen des Systems ausgleichen zu können. Dessen Anwendung führt aber dazu, dass das System mit der Zeit divergiert und erfordert daher weitere stabilisierende Maßnahmen. Dies erreicht Kajita durch die Anwendung von "virtual timeshifting". Details könne seiner Ausarbeitung [57] entnommen werden. Diese Lösung kann aber nur für kurze Zeiträume angewandt werden, bevor Teile des Systems zurückgesetzt werden müssen. Dies erschwert aber besonders die Anpassung an Modellfehler, da diese durchgehend bestehen und so das System in Oszillationen versetzen können. Aus diesem Grund wird für die NDD-WE ein anderer Weg verfolgt, um eine unmittelbare Regelung des System zu ermöglichen.

Ohne eine Überwachung durch Sensoren können Abweichungen der Ist- von der Soll-ZMP-Trajektorie nicht festgestellt werden. Wie der reale ZMP des Systems mit Hilfe von Sensoren des Roboters NAO gemessen werden kann, wurde bereits ausführlich in Kapitel 4.4 diskutiert. Somit besteht die technische Möglichkeit, die Ist-Trajektorie des ZMPs zu messen. Um für den Regler von Nutzen zu sein, muss allerdings ein Rückfluss der Sensoren in die Regelstrecke stattfinden. Durch die Verzögerung der *Preview*-Phase ist es aber nicht möglich, dieses Feedback unmittelbar in die Regelung zu integrieren.

Betrachtet man das Ursprungssystem eingehender aus regelungstechnischer Sicht, erkennt man, warum dies ein Problem darstellt. Hierzu ist die *Beobachtbarkeit* des Ursprungssystems zu untersuchen. Am einfachsten geschieht dies, indem das Kriterium von Hautus [72] überprüft wird. Dies besagt, dass das System  $(\mathbf{A}, \mathbf{C})$  genau dann vollständig beobachtbar ist, wenn

$$\text{Rang} \begin{bmatrix} \lambda \mathbf{I} - \mathbf{A} \\ \mathbf{C} \end{bmatrix} = n, \quad (5.2.24)$$

für alle Eigenwerte  $\lambda_i (i = 0, 1, 2, \dots, n - 1)$  der Matrix  $\mathbf{A}$  erfüllt ist.  $\mathbf{A}$  hat die drei Eigenwerte  $\lambda_1 = \sqrt{\frac{g}{z_c}}$ ,  $\lambda_2 = -\sqrt{\frac{g}{z_c}}$ ,  $\lambda_3 = 0$ . Für alle diese müsste somit die Gleichung:

$$\text{Rang} \begin{bmatrix} \lambda_i & -1 & 0 \\ -\frac{g}{z_c} & \lambda_i & \frac{g}{z_c} \\ 0 & 0 & \lambda_i \\ 0 & 0 & 1 \end{bmatrix} = 3 \quad (5.2.25)$$

gelten, tut dies aber nur für  $\lambda_3 = 0$ . In den beiden anderen Fällen reduziert sich der Rang der Matrix auf 2. Folglich sind diese Eigenwerte nicht beobachtbar. Physikalisch wird dies klar, bedenkt man den Einfluss der Linearisierung durch das 3D-LIPM. Die Einführung einer variablen Pendellänge<sup>14</sup> führt zwar zu einer Vereinfachung des Zustandsraumes des Systems, doch entkoppelt dies auch die Variablen. Als Folge erfüllt das Systems nicht das Kriterium *vollständiger* sondern nur *teilweiser* Beobachtbarkeit. Zur Lösung der Problematik wird im Folgenden ein *Observer* für den *Preview*-Regler entwickelt.

<sup>14</sup>Dies ist die direkte Folge der Forderung  $z_c = \text{const.}$

Im Kontext der Regelungstechnik stellt ein Beobachter, englisch *Observer*, ein System dar, das aus bekannten Eingangs- und Ausgangsgrößen eines Referenzsystems nicht messbare Größen oder Zustände rekonstruiert. Zu diesem Zweck bildet er das beobachtete System als Modell nach und nutzt einen Regler, um die nicht messbaren Zustandsgrößen nachzuführen. In Abbildung 5.2.1 ist der Beobachter nur implizit enthalten, da er über verschiedene Regelungsmodulare verteilt ist. Daher ist in Abbildung 5.2.4 die Ergänzung des Systems nochmals schematisch abstrahiert dargestellt.

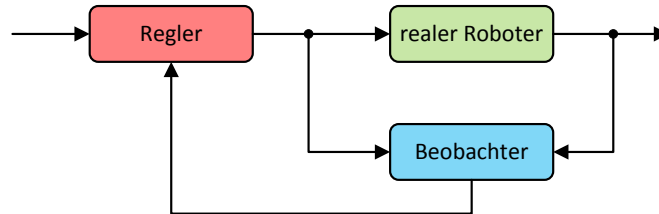


Abbildung 5.2.4: *Observer*-Integration in die NDD-WE.

Ein herkömmlicher Regler sieht den realen Roboter nur als Empfänger der CoM-Position und Quelle der ZMP-Messung. Die im Folgenden vorgestellte Modifikation des *Preview*-Regler hingegen nutzt ein Modell des Roboters parallel zur Regelstrecke, um das zu regelnde System abzubilden. Dieser Beobachter übernimmt die Aufgabe der Zustandsrückführung.

Der Entwurf eines Beobachters teilt sich in zwei Hauptaufgaben, Schätzen des Systemzustands und Berechnen einer stabilisierenden Rückkopplung. Nach [72], Kapitel 8 wird zur Integration eines Beobachters die allgemeine Form einer Regelstrecke

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t),\end{aligned}\tag{5.2.26}$$

um eine zusätzliche Eingangsgröße  $u_B$  erweitert:

$$\begin{aligned}\frac{d}{dt}\hat{\mathbf{x}}(t) &= \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{u}_B(t), \\ \hat{\mathbf{y}}(t) &= \mathbf{C}\hat{\mathbf{x}}(t),\end{aligned}\tag{5.2.27}$$

wobei nun  $\hat{\mathbf{x}}(k)$  die Schätzung von  $\mathbf{x}(k)$  basierend auf der aktuellen Messung darstellt. Diese Eingangsgröße wird auf die Differenz zwischen der gemessenen Ausgangsgröße der Regelstrecke und der Ausgangsgröße des Modells überführt:

$$\mathbf{u}_B(t) = \mathbf{L}(\mathbf{y}(t) - \hat{\mathbf{y}}(t)).\tag{5.2.28}$$

Zentraler Punkt der Zustandsschätzung ist demnach die Matrix  $\mathbf{L}$ , auf deren Basis auf die Unterschiede zwischen erwartetem und gemessenem Systemzustand geschlossen werden kann.

Kombination der Gleichungen (5.2.26) bis (5.2.28) liefert dann die allgemeine Form eines Luenberger-Beobachters:

$$\frac{d}{dt}\hat{\mathbf{x}}(t) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{L}\mathbf{C}(\mathbf{x}(t) - \hat{\mathbf{x}}(t)).\tag{5.2.29}$$

Im Fall des zweibeinigen Laufens ist die messbare Ausgangsgröße des Systems die ZMP Position  $p^{sensor}(k)$ . Die Übertragung des Beobachters (5.2.29) auf die diskrete Systemgleichung (5.2.8) ergibt demnach:

$$\hat{\mathbf{x}}(k+1) = \mathbf{A}\hat{\mathbf{x}}(k) - \mathbf{L}[p^{sensor}(k) - \mathbf{c}\hat{\mathbf{x}}(k)] + \mathbf{b}u(k). \quad (5.2.30)$$

Um ein passendes  $\mathbf{L}$  zu finden wird analog zum Entwurf des *Preview*-Reglers die *LQR*-Methode angewandt. Ähnliche Randbedingungen sind für  $\mathbf{L}$  nötig, welches bestimmt wird, so dass durch die Matrix

$$\tilde{\mathbf{A}}_L = \mathbf{A} - \mathbf{L}\mathbf{c} \quad (5.2.31)$$

asymptotische Stabilität erreicht wird.

Der resultierende *Observer*-basierte Regler, gegeben durch Gleichung (5.2.32), besteht aus einer integralen Anpassung basierend auf dem Regelfehler, einem proportionalen Feedback des Zustands und der Einbeziehung zukünftiger Aktionen basierend auf den *Preview*-Vorgaben:

$$u(k) = -G_I \sum_{i=0}^k [\mathbf{C}\hat{\mathbf{x}}(i) - p^{ref}(i)] - \mathbf{G}_x \hat{\mathbf{x}}(k) - \sum_{j=1}^N G_d(j) p^{ref}(k+j). \quad (5.2.32)$$

Die Verstärkungen  $G_I$ ,  $\mathbf{G}_x$  und  $G_d$  werden so gewählt, dass analog zu Kajita [55] das Gütemaß  $J$  optimiert wird durch:

$$J = \sum_{j=1}^{\infty} \left\{ Q_e [p(j) - p^{ref}(j)]^2 + \Delta \mathbf{x}^T(k) \mathbf{Q}_x \Delta \mathbf{x}(k) + Ru^2(j) \right\}, \quad (5.2.33)$$

wobei  $\Delta \mathbf{x}$  den inkrementelle Zustandsvektor  $\Delta \mathbf{x}(k) = \mathbf{x}(k) - \mathbf{x}(k-1)$  bezeichnet. Analog zu Gleichung (5.2.15), werden sowohl der Regelfehler und exzessive Wechsel von Zustand und Regelung mit den Gewichten  $Q_e$ ,  $\mathbf{Q}_x$  and  $R$  bestraft, so dass ein Regler, der  $J$  optimiert, eine ruhige Regelung des Systems bevorzugt. Zur Vereinfachung sei

$$\tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{c}\mathbf{b} \\ \mathbf{b} \end{bmatrix}, \tilde{\mathbf{I}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tilde{\mathbf{F}} = \begin{bmatrix} \mathbf{c}\mathbf{A} \\ \mathbf{A} \end{bmatrix}, \tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q}_e & 0 \\ 0 & \mathbf{Q}_x \end{bmatrix}, \tilde{\mathbf{A}} = [\tilde{\mathbf{I}}, \tilde{\mathbf{F}}]. \quad (5.2.34)$$

Die optimalen Verstärkungen sind dann beschrieben durch:

$$G_I = \left[ R + \tilde{\mathbf{B}}^T \tilde{\mathbf{P}} \tilde{\mathbf{B}} \right]^{-1} \tilde{\mathbf{B}}^T \tilde{\mathbf{P}} \tilde{\mathbf{I}}, \quad (5.2.35)$$

$$\mathbf{G}_x = \left[ R + \tilde{\mathbf{B}}^T \tilde{\mathbf{P}} \tilde{\mathbf{B}} \right]^{-1} \tilde{\mathbf{B}}^T \tilde{\mathbf{P}} \tilde{\mathbf{F}}, \quad (5.2.36)$$

$$G_d(j) = - \left[ R + \tilde{\mathbf{B}}^T \tilde{\mathbf{P}} \tilde{\mathbf{B}} \right]^{-1} \tilde{\mathbf{B}}^T \left[ \tilde{\mathbf{A}}_c^T \right]^{j-1} \tilde{\mathbf{P}} \tilde{\mathbf{I}}, j = 1, 2, \dots, N, \quad (5.2.37)$$

mit der Regelungs-Matrix  $\tilde{\mathbf{A}}_c$ , definiert durch

$$\tilde{\mathbf{A}}_c = \tilde{\mathbf{A}} - \tilde{\mathbf{B}} \left[ R + \tilde{\mathbf{B}}^T \tilde{\mathbf{P}} \tilde{\mathbf{B}} \right]^{-1} \tilde{\mathbf{B}}^T \tilde{\mathbf{P}} \tilde{\mathbf{A}}. \quad (5.2.38)$$

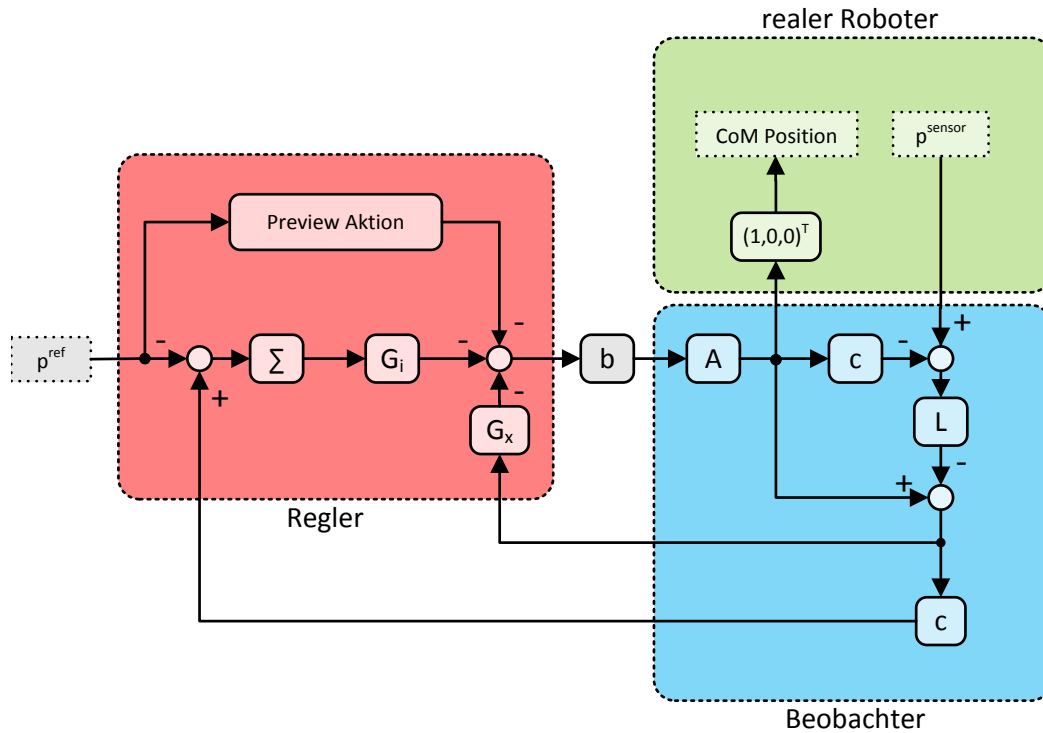


Abbildung 5.2.5: Konfiguration des Regelungssystems.

$\tilde{\mathbf{P}}$  ist hierbei die Lösung der folgenden Riccati-Differentialgleichung analog zu Gleichung 5.2.19

$$\tilde{\mathbf{P}} = \tilde{\mathbf{A}}^T \tilde{\mathbf{P}} \tilde{\mathbf{A}} - \tilde{\mathbf{A}}^T \tilde{\mathbf{P}} \tilde{\mathbf{B}} \left[ R + \tilde{\mathbf{B}}^T \tilde{\mathbf{P}} \tilde{\mathbf{B}} \right]^{-1} \tilde{\mathbf{B}}^T \tilde{\mathbf{P}} \tilde{\mathbf{A}} + \tilde{\mathbf{Q}}. \quad (5.2.39)$$

Der resultierende Regelalgorithmus ist in Abbildung 5.2.5 dargestellt.

Zur Verdeutlichung der Leistungsfähigkeit dieses *Observer*-basierten Reglers seien an dieser Stelle im Vorgriff zur Auswertung bereits in Abbildung 5.2.6 Beispiel-Trajektorien zu finden. Diese verdeutlichen den Fall, dass sich nach dem Zeitpunkt  $t = 1,5$  s ein konstanter Fehler zum gemessenen ZMP addiert. Dieser Fehler könnte beispielsweise aus einer externen Kraft oder einer Unebenheit des Bodens resultieren. Es zeigt sich an der Verschiebung des berechneten ZMP-Verlaufs, dass der Regler diese Abweichung in das Modell integriert. Als Folge stimmt der real gemessene ZMP wieder mit dem Referenz-ZMP überein. Zum Zeitpunkt  $t = 3,0$  s wird die Fehleraddition aufgehoben und der Regler passt den ZMP-Verlauf an, indem er den CoM nachzieht. Fehler in der Messung des ZMPs können so nicht ausgeglichen werden, sondern führen zu einer ungewünschten Abweichung des berechneten ZMPs. Diese Tatsache verdeutlicht nochmals die Wichtigkeit verlässlicher und exakter Sensoren zur Messung des ZMPs.

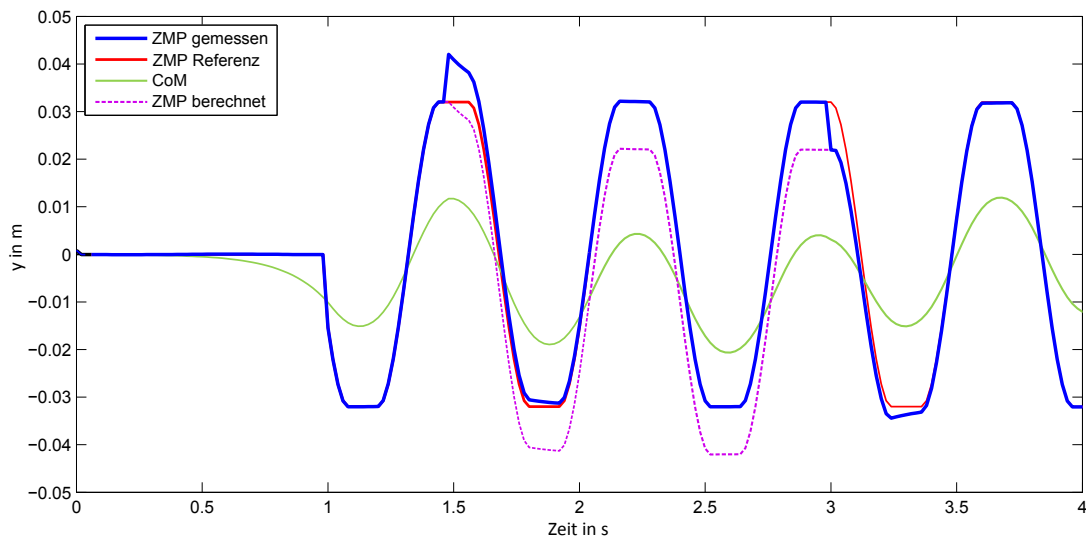


Abbildung 5.2.6: Reaktion des Reglers auf einen konstanten ZMP-Fehler, beginnend bei  $t = 1,5$  s und endend bei  $t = 3,0$  s

### 5.3 Parametrisierung

Die vorgestellte NDD-WE basiert auf einem mathematischen Modell, doch ist bei der bisherigen Diskussion aufgefallen, dass Freiheitsgrade im System verbleiben. Durch diese ist das Verhalten des Systems weiterhin beeinflussbar. Da nicht für jeden dieser Freiheitsgrade die Möglichkeit existiert, sein Optimum durch ein mathematisches Modell zu bestimmen, sind diese als Parameter implementiert. Sie dienen der manuellen Konfiguration der NDD-WE und sind in einer Config-Datei ausgelagert<sup>15</sup>. Mit diesen Parametern ist es möglich den Lauf an nicht modellierte Umweltbedingungen, wie beispielsweise die Bodenbeschaffenheit beim RoboCup Turnier, anzupassen. Darüber hinaus können auch Eigenheiten individueller Roboter, wie eine Dekalibrierung, gezielt ausgeglichen werden. Beides sind zwar Faktoren, die durch den entworfenen Regler bis zu einem gewissen Maß korrigiert werden können, wie die Auswertung in Kapitel 5.4.1 zeigen wird. Doch verhält sich das System ruhiger, wenn diese bereits im Voraus durch die Parameter in das Modell integriert wurden. Da viele Parameter ausschließlich der Kalibrierung des Roboters dienen, erscheint es wenig sinnvoll an dieser Stelle auf alle Parameter im Detail einzugehen. Vielmehr wird für eine Auflistung aller Parameter und eine detaillierte Beschreibung auf den NDD-Teamreport [25] verwiesen. Einige Parameter sind jedoch entscheidend für das Systemverhalten und sollen daher der Vollständigkeit halber erläutert und ihr Einfluss auf das System zumindest kurz betrachtet werden.

Nicht direkt mit dem Regler verbunden, aber dennoch von erheblichem Einfluss, ist die jeweilige **Stiffness** der Gelenke. Da diese das maximale Drehmoment der Motoren beeinflussen, haben sie direkten Einfluss auf die Regelbarkeit des Systems. Die *Stiffness*

<sup>15</sup>Es ist aber ohne weiteres möglich diese auch zur Laufzeit anzupassen.

bestimmt indirekt, wie schnell und in welchem Maße die Position des CoM angepasst werden kann. Um dieses Verhalten zu optimieren, wäre es logisch, die *Stiffness* aller Motoren auf den Maximalwert zu setzen. Im Gegenzug führt dies aber auch dazu, dass die Gelenke ihre Position sehr hart anfahren und eine hohe Kraft aufwenden können, um darin zu verharren. Folglich übt beispielsweise ein verfrühtes Aufsetzen des Fußes einen Ruck auf das System Roboter aus, da die Motoren im Bein noch immer mit aller Kraft versuchen, ihre Endposition anzufahren. Der Roboter stößt sich vom Boden ab. Über diesen Parameter kann also implizit die Dämpfung des Systems eingestellt werden. Wie bereits Honda mit dem ASIMO zeigen konnte, können sich Flexibilitäten in den Gelenken positiv auf das Gesamtsystem auswirken, vergleiche Kapitel 2.3.1. Weiterhin hat der Langzeiteinsatz im RoboCup gezeigt, dass die Kräfte, die durch einen hohe *Stiffness* auf die Zahnräder wirken, für deren Verschleiß nicht förderlich sind. So führt beispielsweise eine Ansteuerung der *HipRoll*-Gelenke mit maximaler *Stiffness* dazu, dass nach verhältnismäßig kurzer Zeit ein Zahn des entsprechenden Getriebes am Endanschlag bricht.

Eine besondere Stellung kommt der **Höhe des CoM** zu Gute. Da die Höhe  $z_c$  direkt in die Systemdynamik einfließt, ist die Lösung der Systemgleichung abhängig von seiner Wahl. Somit kann dieser Parameter nicht zur Laufzeit geändert werden, da die Riccati-Differentialgleichung (5.2.19) initial mittels MATLAB gelöst und in den WE-Code eingebunden wird. Somit kann nur die Parametrisierung einer diskreten Anzahl von möglichen CoM-Höhen in die WE eingebunden werden. Der Faktor  $z_c$  geht reziprok in die Dynamik-Gleichungen (5.2.1) und (5.2.2) des Systems ein. Geringere CoM-Höhen führen also dazu, dass eine Auslenkung des CoM weniger Einfluss auf den ZMP hat. Der Roboter wird stabiler gegen Störungen. Gleichzeitig führt dies aber dazu, dass der CoM in einem größeren Umfang bewegt werden muss, um die gewünschte ZMP-Trajektorie zu erreichen. Als Folge sinkt die *Energieeffizienz* des Laufes und der *Verschleiß* steigt. Weiterhin hat die Höhe des CoM einen Einfluss darauf, wie hoch der Fuß des Schwungbeins durch das Kniegelenk angehoben werden kann. Somit ist für eine gewünschte Höhe der Fußtrajektorie eine minimal erreichbare CoM-Höhe vorgegeben.

Der Entwurf des Reglers wurde bislang losgelöst von der realen Hardware durchgeführt. Es wurde folglich davon ausgegangen, dass der reale ZMP des Systems zu jeder Zeit gemessen werden kann. In der Realität benötigen die Sensoren des Roboters NAO aber durch Übertragung über den internen *I<sup>2</sup>C* Bus eine gewisse Anzahl an Taktzyklen<sup>16</sup>, die experimentell auf rund 3 Zyklen abgeschätzt wurde. Dies ist nicht der Fall für die Inertial-Sensoren, da diese Teil des DCM sind. Jedoch hat die Analyse der ZMP-Verläufe in Kapitel 4.4.4 aufgezeigt, dass sie einer noch größeren Verzögerung durch die interne Filterung unterliegen. Diese unbekannt Latenz kann dem Regler über den Parameter **SensorDelay** bekannt gemacht werden. Zum Ausgleich ist in den *Observer* ein Kalman-Filter integriert, das aus der Prädiktion einen entsprechenden aktuellen Wert extrapoliert.

Der Faktor **sensorControlRatio** bietet eine weitere Option zur Parametrisierung des eigentlichen Reglers. Mit diesem wird die Differenz zwischen dem Referenz- und dem

---

<sup>16</sup>Dass es sich bei der Verzögerung wirklich primär um die Übertragungslatenz handelt, lässt sich belegen, da mit Steigerung der Taktfrequenz des DCM von 50 Hz der Version V2 auf die 100 Hz der Version V3 sich diese Zeitspanne nahezu halbiert hat.



gemessenen ZMP gewichtet. Ein Wert von 1 entspricht der vollen beschriebenen *Close-Loop* Regelung. Ein Wert von 0 umgeht diese vollständig und der Roboter operiert *Open-Loop*. Motiviert wurde diese Entscheidung durch die Beobachtung, dass in der Realität eine zu harte Regelung das System NAO in Schwingungen versetzen kann<sup>17</sup>, die durch den Regler nicht mehr kontrolliert werden können. Somit war ein Faktor zu Dämpfung der Regelung wünschenswert. Mit der Reduktion der Sensorverzögerung durch Einführung des NAO V3 tritt dieser Effekt deutlich seltener auf. Dennoch kann er bei besonders beanspruchten Robotern oder stark federnden Böden trotzdem auftreten.

Die **Schrittfrequenz** ist neben der Schwerpunkthöhe der entscheidendste Parameter der WE, da er alle anderen Parameter beeinflusst. Grundsätzlich ist die Geschwindigkeit des Roboters durch die Schrittfrequenz beeinflussbar, doch wurde bereits erörtert, warum dies in der Praxis über die Schrittweite geschieht. Die maximale Schrittweite ist aber durch die kinematischen Grenzen des Roboters NAO eingeschränkt. Folglich kann ab der maximalen Schrittweite die Geschwindigkeit lediglich über eine Erhöhung der Frequenz erreicht werden. Umgekehrt gilt aber auch, dass eine hohe Schrittfrequenz durch die Grenzen der Dynamik des Systems, die maximal erreichbare Schrittweite eingrenzen kann. Als Folge ergibt sich eine Pareto-Front für die maximal erreichbare Geschwindigkeit in Abhängigkeit von der Frequenz und Schrittweite. Aufgrund der Eigenanregung des Roboters kann aber nicht das gesamte Frequenzband ausgenutzt werden. Laufexperimente legen nahe, dass zumindest eine Eigenfrequenz des Systems bei rund 1,5 Hz liegt. Bei Annäherung an diesen Bereich, sowohl von unten, als auch von oben, wird der Roboter zusehends instabiler und kann durch die Regelung nicht mehr unter Kontrolle gebracht werden. Die genaue Eigenfrequenz des Systems NAO ist von Roboter zu Roboter verschieden und kann mit den vorhandenen Mitteln nicht exakt experimentell ermittelt werden. Somit bleibt das Frequenzband größer als 2 Hz und kleiner als 1 Hz, um sinnvolle Lauffrequenzen einzustellen. Die Erfahrung im Praxisbetrieb bestätigt hierbei den Sachverhalt, der sich auch schlussfolgern lässt. Frequenzen über 2 Hz führen zu Bewegungen, die eine gute *Präzision* in der Ausführung der Schritte zur Folge haben. Frequenzen unter 1 Hz führen hingegen zu höheren maximalen Laufgeschwindigkeiten, was zusätzlich dadurch begünstigt wird, dass der CoM nicht so weit ausgelenkt werden muss. In Konsequenz kommt mehr Laufruhe ins System.

Die Dauer eines Schrittes ist über die Schrittfrequenz implizit gegeben. Welchen prozentualen Anteil daran die *Double-Support*-Phase einnimmt bestimmt der Parameter **doubleSupportRatio**. Dieser beeinträchtigt das Verhältnis der Dauer von *Single*- zu *Double-Support*-Phase. Eine 0 bedeutet es existiert keine *Double-Support*-Phase, eine 1 bedeutet, es existiert keine *Single-Support*-Phase. Es ist offensichtlich, dass ein Verhältnis von 1 dazu führt, dass keine Fortbewegung mehr möglich ist, da hierzu zwingend eine *Single-Support*-Phase notwendig ist. Weiterhin ist es logisch, dass eine lange *Double-Support*-Phase zu einer Reduktion der maximalen Geschwindigkeit führt. Verkürzt man die *Double-Support*-Phase hingegen, so reduziert sich bei Annäherung an die Null die Zeit für den Übergang der *Support*-Polygone der *Single-Support*-Phase, um dann gänzlich zu verschwinden. Folglich kann die Referenz-ZMP-Trajektorie nicht mehr vollständig durch stabilisierendes Gebiet geplant werden. Der ZMP geht zwischendurch

<sup>17</sup>Grund ist nicht modelliertes Spiel in den Gelenken. Demnach tritt dies besonders bei älteren Robotern auf, deren Gelenke besonders stark verschlissen sind.

zwangsläufig in den FZMP über. Graf konnte in [38] und [39] jedoch zeigen, dass es sogar ein Vorteil für den Lauf ist, die *Double-Support*-Phase gänzlich zu eliminieren. Hierbei wird in Kauf genommen, dass der ZMP sich kurzzeitig außerhalb des *Support*-Polygons befindet. Dies bedeutet nicht, dass der Roboter sofort stürzt, sondern der Roboter wird vielmehr instabil und beginnt zunächst zu kippen. Durch die bewusste Führung des ZMPs begibt der Roboter sich in eine kontrollierte Fallbewegung, die von dem neuen Standfuß gebremst wird. Somit stellt dies quasi den Grenzfall zur Flugphase des Rennens dar, die der Roboter NAO aufgrund seiner Hardware-Restriktionen nicht erreichen kann. Die kurzfristige zusätzliche Beschleunigung durch die Fallbewegung wirkt sich positiv auf die erreichte Laufgeschwindigkeit aus. Derselbe Effekt kann mit der NDD-WE bestätigt werden. Wenn auf die *Double-Support*-Phase verzichtet wird, führt dies zu höheren Laufgeschwindigkeiten, ohne die Stabilität negativ zu beeinflussen. Dies erfordert allerdings die Anpassung anderer Parameter, wie der Schwungfuß Trajektorie und der Stiffness, um den Ruck beim Auffangen der Fallbewegung durch den neuen Standfuß zu kompensieren.

### 5.4 Evaluation

Bisher wurde lediglich das mathematische Konzept der Regelung hergeleitet und seine theoretische Reaktion auf eine Abweichung der Messgröße anhand eines Beispiels demonstriert. Es ist aber noch zu zeigen, dass der Regler wirklich in der Lage ist, dem Roboter NAO zu einem stabilen Gang zu verhelfen. Dies wird nun anhand geeigneter Experimente nachgeholt. Schwerpunkt der Auswertung ist die Reaktion des Reglers im Falle von unvorhergesehenen Einflüssen auf das System, da Zustandsrückführung die primäre Änderung gegenüber Kajitas Algorithmus darstellt. Kapitel 5.4.1 beschränkt sich zunächst bewusst auf die Simulation. Durch das Modell wird so zwar vom realen NAO abstrahiert, doch hat eben dies für die Experimente den Vorteil, unbekannte Störeinflüsse auf das System zunächst nicht zu betrachten. So kann die bisher nur theoretisch postulierte Funktionsweise der WE belegt und gezielt die Reaktion des Reglers auf bestimmte Störeinflüsse untersucht werden. Im Anschluss wird Kapitel 5.4.2 demonstrieren, dass sich die Erkenntnisse aus der Simulation auf den realen Roboter übertragen lassen.

Alle Experimente wurden mit der Version V2 des NAO durchgeführt. Der für die WE signifikante Unterschied zwischen dieser und der Version V3 ist die niedrigere Taktfrequenz des DCM von 50 Hz und somit eine erhöhte Verzögerung der Sensoren des V2 gegenüber des V3. Es wird jedoch gezeigt werden, dass die NDD-WE bereits mit der geringeren Frequenz praxisrelevante Verbesserungen gegenüber einem *Open-Loop*-Ansatz aufweist. Bedauerlicherweise existierte, aufgrund der Austauschpolitik seitens Aldebarans, nie gleichzeitig Zugriff auf Version V2 und V3. Eine direkt vergleichende Untersuchung des Einfluss der Taktfrequenz muss daher entfallen. Weiterhin ist die Qualität der FSR-Sensoren, oder vielmehr die Umsetzung ihrer Montage durch Aldebaran, in der Version V2 für den praktischen Einsatz unbrauchbar gewesen. Die Sensoren sprachen durch Verkanten der Bodenplatte wechselnd stark über oder fielen gänzlich aus. Daher muss der ZMP für alle Experimente gemäß der in Kapitel 4.4.2 vorgestellten

Methode mit Hilfe des Beschleunigungssensors gemessen werden, auch wenn der Vergleich in Kapitel 4.4.4 eine mögliche Verbesserung der Ergebnisse mit FSR-Sensoren nahelegt. Die Simulation verwendet hierzu einen Beschleunigungssensor, der den untersuchten Charakteristiken des realen Sensors des NAO nachempfunden ist.

### 5.4.1 Simulation

Die simulierten Laufexperimente wurden mit dem Simulator *SimRobot* [69] durchgeführt. Dieser verwendet ein Mehrkörper-Modell des Roboters NAO, das auf Basis der kinematischen Angaben von Aldebaran erstellt wurde. Dessen Berechnung übernimmt die *Open Dynamics Engine* (ODE)<sup>18</sup> mit einer Frequenz von 1 kHz. Bereits Hebbel konnte mit seinen Laufexperimenten des Roboters AIBO [68] und später dem zweibeinigen Kondo zeigen, dass der Simulator sich sehr realitätsnah verhält. Auch Urbann et al. konnten in [113] bestätigen, dass das Mehrkörpermodell des SimRobot trotz seiner Einfachheit bereits ausreicht den realen NAO in guter Näherung abzubilden. Folglich stellt dieser Simulator eine geeignete Wahl für die Evaluation dar. Die Experimente wurden gewählt, um einige typische Störungseinflüsse für einen humanoiden Roboter zu demonstrieren, erheben aber keinen Anspruch an Vollständigkeit. Alle Experimente wurden mit einer Schrittfrequenz von 2 Hz für einen Doppelschritt und einer CoM-Höhe von  $z_c = 250$  mm durchgeführt.

#### Experiment 5.4.1.A - Unebener Boden

Mit dem ersten Experiment soll die Auswirkung einer Störung durch unebenen Boden auf dem Roboter NAO demonstriert werden. Hierzu wird dem Roboter eine Platte mit einer Höhe von 1 cm in den Weg gelegt. Auf diese läuft er mit einer Geschwindigkeit von  $v_x = 200$  mm/s zu. Schrittweite und -höhe sind so gewählt, dass der Roboter nicht mit einem direkten Schritt die höhere Ebene erklimmen kann. Es kann also garantiert werden, dass er in jedem Fall auf die Kante tritt und eine Störung des Gangs stattfindet. Der Aufbau des Experiments ist in Abbildung 5.4.7 zu finden. Das Experiment wird 50 Mal wiederholt und die Startposition jeweils leicht variiert, damit der Roboter zu unterschiedlichen Laufphasen mit der Platte kollidiert. Die Ergebnisse sind Tabelle 5.4.1 zu entnehmen und demonstrieren den Vorteil eindeutig mit einer Verbesserung der Stabilität bei 76 % der Läufe.

	Sturz	Stabiler Lauf	Verhältnis
Open-Loop	48	2	4 %
Observer	10	40	80 %

Tabelle 5.4.1: Experiment 5.4.1.A - Unebener Boden.

<sup>18</sup><http://www.ode.org>

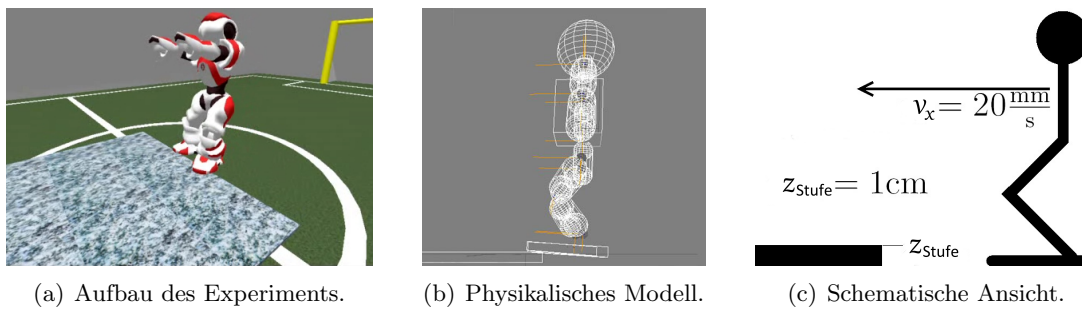


Abbildung 5.4.7: Experiment 5.4.1.A - Unebener Boden.

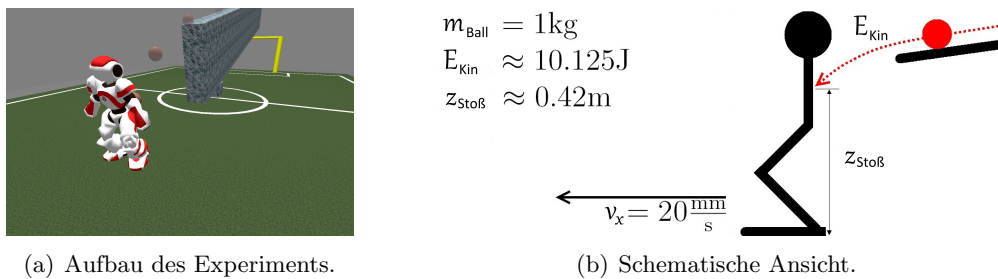


Abbildung 5.4.8: Experiment 5.4.1.B - Externe Kräfte.

### Experiment 5.4.1.B - Externe Kräfte

Ziel des zweiten Experiments im Simulator ist die Reaktion des Systems auf externe Kräfte, die direkt auf den Oberkörper wirken - beispielsweise Stöße durch andere Roboter. Hierzu wird im Simulator eine Szene erzeugt, die eine Kollision des Roboters mit einer Kugel erzwingt, siehe Abbildung 5.4.8. Zur Beschleunigung läuft die Kugel einen schräge Rampe hinab. Das Experiment ist derart aufgebaut, dass die Kugel den NAO zwischen den Schultern trifft. Ihre Startzeit wird in geringem Maß randomisiert, damit die Kollision nicht immer zum exakt selben Zeitpunkt stattfindet. Das Experiment wird ebenfalls 50 Mal wiederholt. Die Kugel wiegt 1 kg, was in etwa einem Viertel der Masse des NAO entspricht. Der Aufprall stellt somit in Kombination mit der Länge der Beschleunigungsphase eine erhebliche Störung des Systems dar. Dies belegen auch die Resultate in Tabelle 5.4.2, da die Kollision ohne Regelung den Roboter immer zu Fall bringt. Wieder zeigt sich eindeutig der Vorteil der Stabilisierung durch den Observer, die in diesem Experiment sogar einen Steigerung der stabilen Läufe von 0 auf 94 % zur Folge hat.

	Sturz	Stabiler Lauf	Verhältnis
Open-Loop	50	0	0 %
Observer	6	44	94 %

Tabelle 5.4.2: Experiment 5.4.1.B - Externe Kräfte.

### Experiment 5.4.1.C - Modellfehler

Ziel des dritten Experiments ist es, einen Fehler im Modell des Roboters nachzustellen. Hierzu wird das SimRobot-Modell des NAO verändert, indem einer der FüÙe durch einen Fuß mit doppelter Höhe ersetzt wird. Ein Verdeutlichung ist in Abbildung 5.4.9 zu finden. Dies führt zu einem dazu, dass der Roboter mit einer leichten Schrägstellung läuft. Zum anderen unterschätzt er aber auch mit dem Fuß den Abstand zum Boden, so dass dieser als Schwungfuß immer verfrüht den Boden berührt. Das System ist also periodisch einem Stoß ausgesetzt. Der Roboter läuft erneut in  $x$ -Richtung, allerdings diesmal nur mit einer Geschwindigkeit von  $v_x = 100 \text{ mm/s}$ . Das Experiment wird 10 Mal wiederholt. Die Ergebnisse sind in Tabelle 5.4.3 zu finden. Da es sich diesmal nicht um einen einmaligen, sondern einen andauernden Fehler handelt, wird nun die Zeit bis zum Sturz ausgewertet. Das Ergebnis des Experiments fällt noch eindeutiger zu Gunsten des *Observers* aus. Ohne Feedback der Sensoren schaukelt sich der Roboter nach 4 bis 5 Doppelschritten in jedem Fall soweit auf, dass es zu einem Sturz kommt. Wird der Regelkreis geschlossen, so ist der Regler in der Lage, den unbekanntem Fehler im Modell auszugleichen und das Experiment wird jeweils nach 60 Sekunden ohne Sturz abgebrochen.

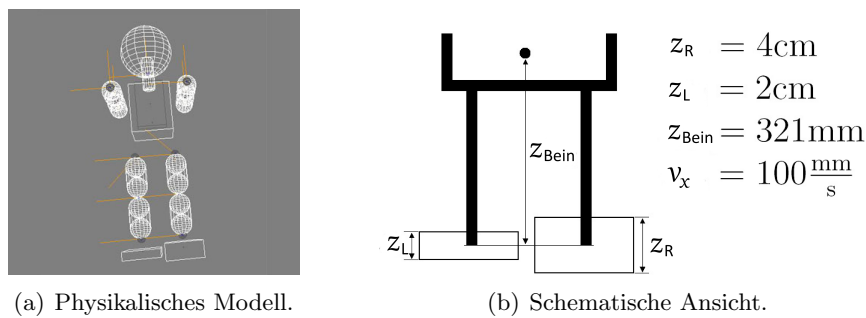


Abbildung 5.4.9: Experiment 5.4.1.C - Modellfehler.

Lauf	1	2	3	4	5
Open-Loop	6.98 s	6.96 s	7.16 s	10.34 s	8.26 s
Observer	> 60 s	> 60 s	> 60 s	> 60 s	> 60 s
Lauf	6	7	8	9	10
Open-Loop	9.32 s	8.36 s	6.98 s	8.34 s	6.98 s
Observer	> 60 s	> 60 s	> 60 s	> 60 s	> 60 s

Tabelle 5.4.3: Experiment 5.4.1.C - Modellfehler.

### 5.4.2 Realer Roboter

Die bisherigen Experimente im Simulator belegen einen deutlichen Vorteil der Integration des *Observers* entgegen dem *Open-Loop*-Ansatz. Doch auch, wenn der Simulator

ein verhältnismäßig realistisches Modell des Roboters NAO nachbildet, so ist dies nicht der reale Roboter. Urbann's Experimente zeigen, dass sich selbst durch die Integration der Motor-Dynamiken in seinem *MoToFlex* Simulator [113] nicht alle sichtbaren Effekte des realen Roboters abbilden lassen. Somit kann eine Evaluation der NDD-WE am realen Roboter nicht ausbleiben, soll die Praxistauglichkeit des entwickelten Reglers gezeigt werden. Alle Experimente werden mit einer CoM-Höhe von  $z_c = 250$  mm durchgeführt. Experimente am physikalischen Roboter sind immer zeitaufwendig und durch ihre Natur nicht deterministisch. Im Folgenden kann daher nur exemplarisch auf einzelne Experimente und ihre Auswertungen eingegangen werden und nicht deren Wiederholbarkeit in den Vordergrund gestellt werden. Da der Spieleinsatz letztendlich die einzig umfassende Bewertung zulässt, sei dem Leser darüber hinaus empfohlen, die Praxistauglichkeit der NDD-WE anhand von Archiv-Videos der RoboCup-Spiele der *NAO Devils Dortmund*<sup>19</sup> zu beurteilen.

Zur Veranschaulichung der Leistungsfähigkeit ist als Einstieg in die praktischen Experimente in Abbildung 5.4.10 die CoM-Trajektorie für einen Lauf, bestehend aus einer Abfolge mehrerer unterschiedlicher *WalkRequests*, abgebildet. Von links nach rechts, wird der NAO zunächst in Laufrichtung angesteuert, gefolgt von einer Überlagerung der reinen Translation mit einer Drehung in Links-Richtung, einer Umkehrung der Translationsrichtung in einen Rückwärtsgang, der wiederum mit einer Drehung in Links-Richtung überlagert wird. Am Ende bremst der Roboter die Translation in  $x$ -Richtung und die Rotation ab, um sich mit Seitenschritten fortzubewegen. Dies dient der Demonstration der Fähigkeit der WE sowohl Translationen als auch Rotationen ausführen zu können, zeigt aber zugleich, dass durch deren Überlagerung auch der NAO zu einem vollständig omnidirektionalen Lauf befähigt wird. Weiterhin wird bereits deutlich, dass der Regler in der Lage ist, den Roboter auch bei unmittelbaren Richtungsänderungen weiterhin zu stabilisieren.

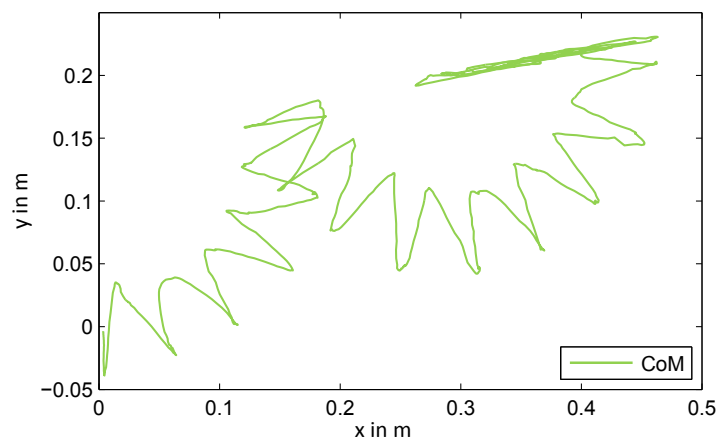


Abbildung 5.4.10: CoM-Trajektorie während eines omnidirektionalen Laufs.

<sup>19</sup>Zu finden auf <http://www.nao-devils.de/> unter dem Menüpunkt *Media*.

### Experiment 5.4.2.A - Close-Loop Lauf

Ziel des ersten Experiments ist es nochmals die Notwendigkeit von Tests am realen Roboter aufzuzeigen, indem Ergebnisse der Simulation mit dem physikalischen Roboter verglichen werden. Hierzu wird der Roboter mit einer Schrittdauer von  $t = 2,5$  s und mit einer Geschwindigkeit von  $v_x = 50 \frac{\text{mm}}{\text{s}}$  angesteuert. Dies führt auch ohne den Einsatz des Beobachters zu einem stabilen Lauf im Simulator. Wie in Abbildung 5.4.11 zu sehen ist, führen die gleichen Laufparameter auf dem realen NAO dazu, dass der Referenz-ZMP und der tatsächliche ZMP über die Zeit auseinanderlaufen. Der Roboter schwingt sich bereits nach vier Schritten so weit auf, dass er seitlich über seine Fußkante kippt. Gleichzeitig zeigt sich, dass der *Observer* bei anfänglich sehr ähnlicher ZMP-Trajektorie in der Lage ist, den Lauf im kritischen Moment zu stabilisieren und fortzusetzen.

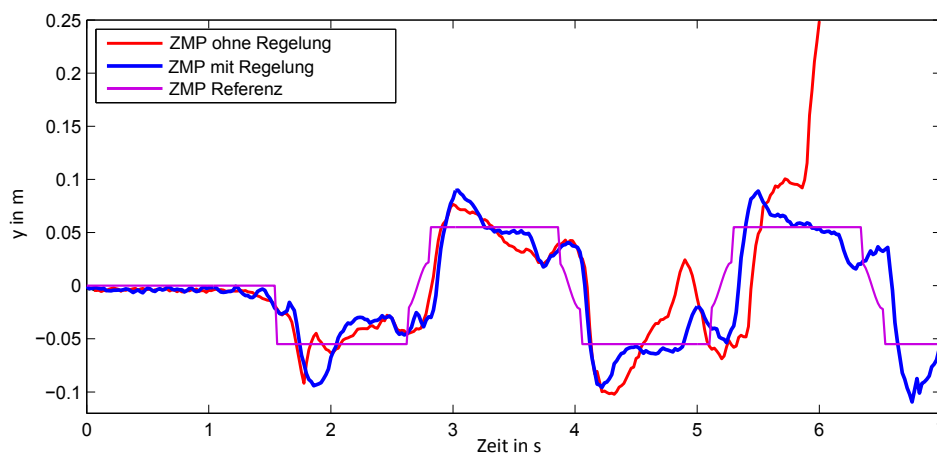


Abbildung 5.4.11: Experiment 5.4.2.A - *Close-Loop*-Lauf.

### Experiment 5.4.2.B - Externe Kräfte

Einer der Hauptgründe einen regelungsbasierter Ansatz zu verfolgen war die Tatsache, dass das System so auf unvorhergesehene externe Einflüsse reagieren kann. Diese Eigenschaft wurde bereits in Experiment 5.4.1.B in der Simulation nachgewiesen, doch muss noch die Übertragbarkeit der Ergebnisse auf den realen NAO belegt werden. Da es schwierig ist, einen Versuchsaufbau analog zu Abbildung 5.4.8(a) mit dem realen Roboter nachzuahmen wird zur Demonstration ein manuelles Experiment gewählt. Der Roboter wird bei einer Laufgeschwindigkeit von  $v_x = 50 \frac{\text{mm}}{\text{s}}$  und einer Doppelschrittdauer von  $t_{step} = 1$  s mit der Hand seitlich angestoßen. Die resultierenden ZMP-Verläufe sind in Abbildung 5.4.12 für einen Lauf exemplarisch dargelegt. Die Kollision bei  $t = 4$  s ist ebenso unmittelbar am ZMP-Verlauf zu erkennen, wie der resultierende Überschwinger bei  $t = 5$  s. Der CoM scheint unbeeinträchtigt. Dies ist erklärbar, da die Kollision zwar zu einer CoM-Verschiebung führt, aber nicht die Gelenke beeinflusst. Diese ist somit vom Roboter nicht wahrnehmbar. Ohne Regler folgt der Sturz des NAO. Wie zu erkennen ist, kann der Beobachter der NDD-WE die Schwingung ausgleichen und der NAO setzt seinen Gang wie geplant fort. Das Ergebnis dieses Experiments ist zwar



wiederholbar, doch durch seine manuelle Natur ist eine statistische Auswertung nicht aussagekräftig.

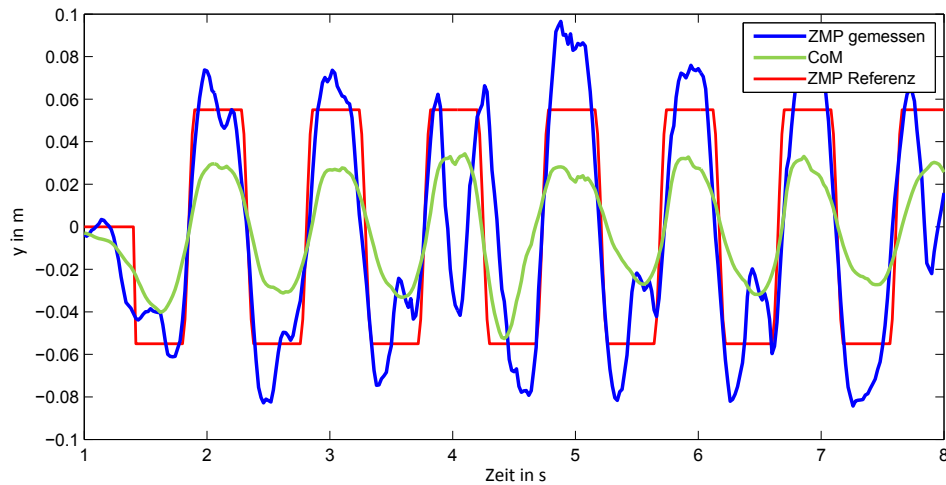


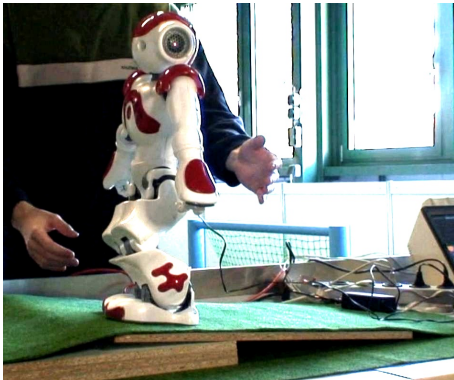
Abbildung 5.4.12: Experiment 5.4.2.B - Externe Kräfte.

### Experiment 5.4.2.C - Unebener Boden

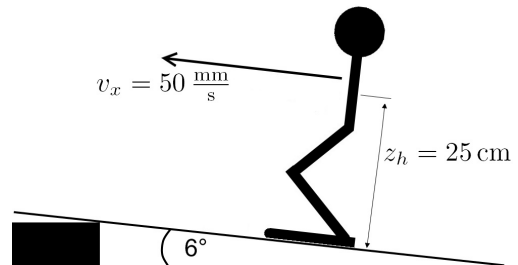
Das letzte Experiment untersucht die Reaktion des Reglers auf eine Störung durch unebenen Boden. Hierzu wird der Roboter auf eine Rampe mit einem Winkel von  $6^\circ$  gestellt und läuft dann mit einer Geschwindigkeit von  $v_x = 50 \frac{\text{mm}}{\text{s}}$  und einer Doppelschritt-Dauer von  $t_{step} = 1 \text{ s}$ . Ein Video des Versuchs kann auf der Internetseite des Teams NAO Devils betrachtet werden<sup>20</sup>. Wie in einem Standbild des Versuchsvideos in Abbildung 5.4.13(a) zu erkennen ist, hat die Rampe am Ende einen Überhang, der den Lauf des Roboters stört. Die Platte ist zudem nicht befestigt, so dass der Aufbau sich wie eine Wippe verhält, wenn der Roboter NAO deren Ende erreicht. Dies führt nochmals zu einer weiteren Störung des Laufs. Während der Roboter die Rampe entlangläuft ist sein ganzer Körper durch die Schräge geneigt. Abbildung 5.14(a) zeigt den ZMP-Verlauf in  $x$ -Richtung für den Zeitraum in dem der Roboter das Ende der Rampe erreicht. Kurz nach dem Zeitpunkt  $t = 13 \text{ s}$ , an dem die Wippe anfängt sich zu bewegen, kann beobachtet werden, dass der gemessene ZMP merklich abweicht und der Roboter im Folgenden in Schwingung gerät. Um  $t = 15 \text{ s}$  scheint der Roboter die Wippe zu verlassen und setzt dadurch das System einer weiteren Störung aus. Abbildung 5.14(b), in der die Neigung aufgetragen ist, zeigt ebenso die initiale Störung der Orientierung des Oberkörpers nach  $t = 13 \text{ s}$  und wie sich diese in den folgenden Sekunden aufschwingt. Der Regler ist jedoch in der Lage diese Störung auszugleichen und das Aufschwingen zu verhindern, so dass ab  $t = 16 \text{ s}$  in Abbildung 5.14(a) wieder der normale ZMP-Verlauf zu finden ist. Auch bei dieser Auswertung scheint der CoM von der Störung unbeeinflusst zu sein. Die Wippe führt zwar zu eine Bewegung des Roboters, die Motoren verhindern aber einen Veränderung der Konfiguration desselben. Deshalb kann auch in diesem Fall die CoM Bewegung vom Roboter nicht wahrgenommen werden.

<sup>20</sup><http://www.irf.tu-dortmund.de/NAO-devils/download/2009/experiments/wipptest.wmv>



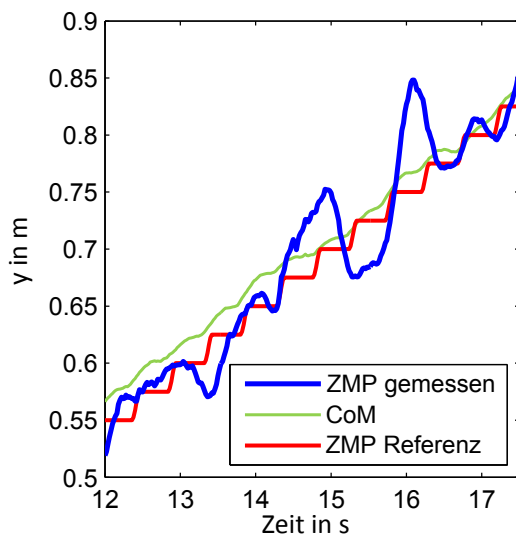
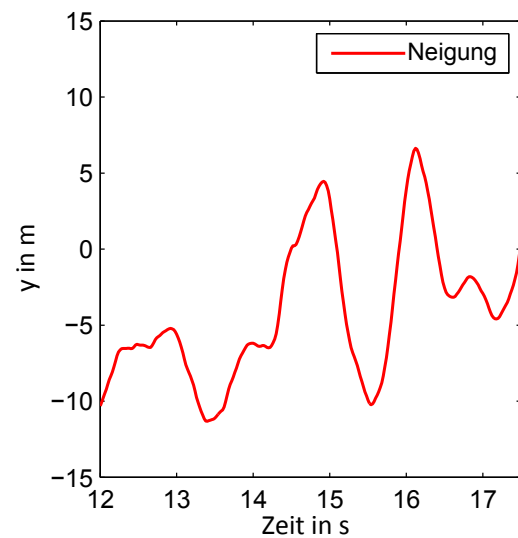


(a) Versuchsaufbau.



(b) Schematische Darstellung.

Abbildung 5.4.13: Experiment 5.4.2.C - Unebener Boden.

(a) ZMP- und CoM-Trajektorie in  $x$ -Richtung.

(b) Orientierung des Oberkörpers.

Abbildung 5.4.14: Experiment 5.4.2.C - Ergebnisse.

### 5.4.3 Energieeffizienz

Nachdem mit den bisherigen Experimenten aufgezeigt werden konnte, dass die NDD-WE einen Vorteil für die Stabilität des Roboters bietet, soll nun abschließend auf einen Faktor eingegangen werden, der bisher nicht beachtet wurde - die Energieeffizienz des Laufs. Sicherlich ist für einen Einsatz in einem realen Szenario die Laufzeit des Roboters wichtig, doch für den RoboCup reicht eine vollständig geladene Batterie des NAO aus, um den Roboter das Spiel über mit Strom zu versorgen. Die Wahl der Laufparameter beeinflusst diese Tatsache nicht. Daher wurde dieses Gütemaß in den bisherigen Experimenten nicht betrachtet und andere Kriterien rückten in den Vordergrund. Durch die fortschreitende Optimierung des Laufes hat sich aber gezeigt, dass die Energieeffizienz sehr wohl einen signifikanten Einfluss auf das Spiel haben kann - wenn auch lediglich indirekt. Um Schäden an den Motoren vorzubeugen, begrenzt Aldebaran deren maximale Leistung in Abhängigkeit der aktuellen Betriebstemperatur. Das genaue Verhalten ist undokumentiert<sup>21</sup>, doch beginnend bei 65° C wird die maximale *Stiffness* der Motoren reduziert. Dass dies potentiell kritischen Einfluss auf das System haben kann, ist offensichtlich, bedenkt man, dass es sich bei der *Stiffness* der Gelenke um einen wichtigen Parameter der NDD-WE handelt. Kritisch ist dabei vor allem, dass die Änderung geschieht, ohne die anderen Parameter entsprechend angleichen zu können. Als Resultat wird der Lauf des NAO in der Regel instabil, da die maximale Kraft der betroffenen Gelenke begrenzt wird. Folglich ist der Regler nicht mehr in der Lage, den CoM entlang der erforderlichen Trajektorie zu bewegen.

Die Bestimmung der Motortemperaturen geschieht beim Roboter NAO auf einem Umweg, da er keine Sensoren besitzt, die diese messen können. Um die Temperatur dennoch abschätzen zu können, integriert Aldebaran die Stromaufnahme der Motoren über die Zeit und berechnet somit die Energieaufnahme. Aus dieser wird die Temperatur letztlich geschätzt. Der Effekt der *Stiffness*-Reduktion ist also direkt mit der Energieeffizienz des Laufes korreliert, so dass diese mit zunehmender Geschwindigkeit der Bewegung zu einem wichtigeren Faktor wird. Bei der NDD-WE neigt besonders ein Gelenk zu Überhitzung - das Kniegelenk. Dies wird verständlich, wenn man bedenkt, dass aufgrund des 3D-LIPM-Ansatzes dieses Gelenk in jeder *Single-Support*-Phase die Höhe des Schwerpunktes reguliert und dadurch einer ständigen Belastung ausgesetzt ist. Bei Überlegungen zur Untersuchung dieses Phänomens kommt man schnell zu dem Schluss, dass die Höhe des CoM einen entscheidenden Einfluss auf die Kraft hat, die von den Kniemotoren aufgebracht werden muss. Die Höhe  $z_c$  des CoM über dem Boden verändert sich zwar per Definition nicht während des Laufs, doch führt die fortschreitende Translation des CoM dazu, dass dieser sich relativ zum FKS bewegt. Dies kann aber nur durch eine Streckung des Beins geschehen, für die das Gelenk *KneePitch* verantwortlich ist. Eine Betrachtung der irrealen Grenzfälle  $z_c = \infty$  und  $z_c = 0$  offenbart, dass das Bein stärker gestreckt werden muss, je tiefer der CoM ist. Abhängig von der CoM-Höhe verändert sich zudem der Winkelbereich, in dem das Kniegelenk arbeitet. Da dies den Hebelarm beeinflusst, mit dem der CoM über den Oberschenkel auf das Gelenk wirkt, erhöht dieser Effekt die Energieaufnahme. Zur Überprüfung ist in Abbildung 5.4.15 exemplarisch der Verlauf der Energieaufnahme eines *KneePitch*-Gelenks für zwei un-

---

<sup>21</sup>Nicht einmal die Existenz diese Verhalten ist in [1] dokumentiert, konnte aber von Technikern bestätigt werden.

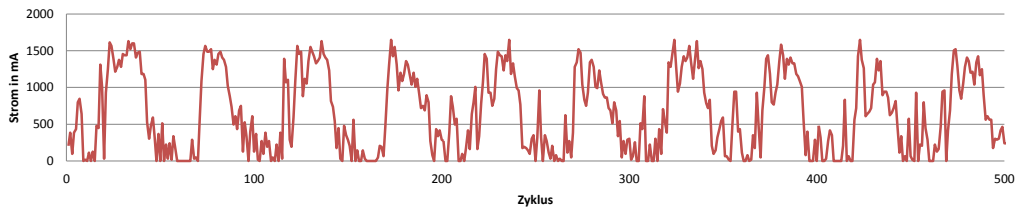
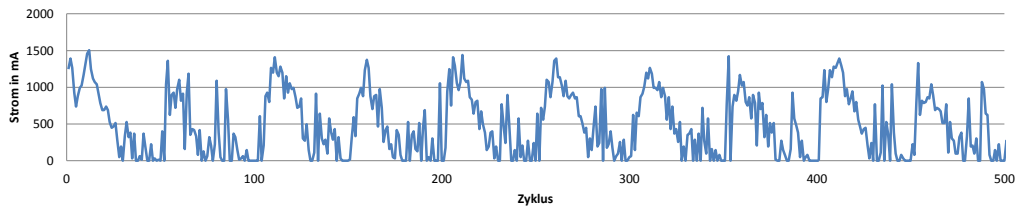
terschiedliche Schwerpunkthöhen von  $z_c = 250$  mm und  $z_c = 280$  mm zu finden. Es zeigt sich erneut, dass die Stromsensoren sehr verrauschte Werte liefern und sich im Grunde nicht für eine quantitative Analyse eignen. Trotzdem kann bereits mit bloßem Auge erkannt werden, dass zwar die Größenordnung der Energieaufnahme in beiden Fällen identisch ist, aber die Spitzenaufnahme bei der hohen Schwerpunktposition merklich geringer ausfällt. Zur Unterstützung dieser Beobachtung findet sich in Tabelle 5.4.4 die mittlere Energieaufnahme von je 8 Laufauswertungen mit  $v_x = 200$  m/s mit hohem und niedrigem Schwerpunkt, jeweils über eine Dauer von 5 Sekunden. Selbst das Mittel der Werte ist stark verrauscht. Der tendenzielle Einfluss der Schwerpunkthöhe kann aber dennoch bestätigt werden. Trotz aller Schwankung ist die größte mittlere Energieaufnahme mit hohem Schwerpunkt niedriger, als die geringste mit niedrigem. Diese Werte sind zur Verdeutlichung jeweils fett gedruckt.

Es kann und soll an dieser Stelle keine genaue Analyse des Einflusses aller Faktoren auf die Energieeffizienz durchgeführt werden. So ist klar, dass diese Untersuchung natürlich nicht den Energieverbrauch der anderen Gelenke mit einbezieht. Da eine tiefere Position des Schwerpunkts dazu führt, dass der CoM durch den Trajektorien-Generator weiter in  $x$ - und  $y$ -Richtung ausgelenkt werden muss, ist jedoch zu erwarten, dass hierdurch die Energieaufnahme des gesamten Systems weiter ansteigen wird. Das Beispiel dient vielmehr der Verdeutlichung, welchen indirekten Einfluss die CoM-Höhe auf das System haben kann und soll die Notwendigkeit motivieren das Gütemaß *Energieeffizienz* in Zukunft detaillierter zu untersuchen. Es bleibt jedoch festzuhalten, dass mit der Anhebung der CoM die Überhitzung des Kniegelenks im Spielbetrieb verhindert werden konnte.

Diese Analyse zeigt nochmals Verknüpfung und gegenseitige Beeinflussung der unterschiedlichen Gütemaße auf. Ein höherer Schwerpunkt beeinflusst zwangsläufig wieder die Stabilität des Laufs negativ, jedoch nicht mehr, als die Überhitzung der Motoren. Die Optimierung einer WE ist also eine überaus komplexe Aufgabe, die sich nur abschließend beurteilen lässt, wenn alle Gütemaße in Betracht gezogen werden.

Lauf	Durchschnitt/mA	Lauf	Durchschnitt/mA
1	539	1	722
2	<b>567</b>	2	712
3	429	3	676
4	282	4	657
5	354	5	657
6	491	6	<b>634</b>
7	286	7	734
8	490	8	639
Hoher CoM		Niedriger CoM	

Tabelle 5.4.4: Durchschnittliche Energieaufnahme eines *KneePitch*-Gelenks über einen Zeitraum von 5 Sekunden beim Lauf in  $x$ -Richtung mit  $v_x = 200$  mm/s.

(a) Lauf in  $x$ -Richtung mit niedrigem CoM.(b) Lauf in  $x$ -Richtung mit hohem CoM.Abbildung 5.4.15: Vergleich der Leistungsaufnahme eines *KneePitch*-Gelenks bei unterschiedlicher Schwerpunkthöhe.

#### 5.4.4 Diskussion und Ausblick

Nachdem nun die Algorithmen der NDD-WE eingehend vorgestellt wurden und die Funktion des vorgestellten Regelungskonzepts in Experimenten belegt werden konnte, steht es noch aus Resümee zu ziehen. Hierzu werden die erarbeiteten Gütekriterien als Grundlage dienen. Sowohl die Untersuchungen in der Simulation, als auch deren Übertragung auf den realen Roboter zeigen, dass die Güte der **Stabilität** durch den *Observer* deutlich verbessert werden konnte. Dennoch belegen Videos der Spiele der NDD, dass in der Praxis die Stabilität nicht ausreicht, um den Roboter stets im Gleichgewicht zu halten. Die Vielzahl dieser Stürze resultiert jedoch aus Kollisionen mit Robotern oder Objekten, wie dem Torpfosten. Häufig „*verhakt*“ sich der Roboter mit einer seiner Extremitäten und stürzt, da die WE die Bewegung dennoch weiter ausführt. Eine Detektion einer solchen Situation durch die WE würde also zu einer deutlichen Verbesserung der Stabilität führen. Stanton et al. [103] stellen hierzu einen erfolgversprechenden Algorithmus vor, der trotz des starken Rauschens der Stromsensoren aus diesen einen Kontakt des NAO mit einem Objekt detektieren kann. Ist die Ursache einer extern verursachten Instabilität bekannt, könnten zudem aktive Gegenmaßnahmen eingeleitet werden. Morisawa et al. zeigen beispielsweise in [80] mittels des HRP-4 den Vorteil von Ausfallschritten auf. Eine Übertragung eines solchen Konzepts auf den NAO steht derzeit noch aus. Als weiteres entscheidendes Gütemaß wurde die **Geschwindigkeit** angeführt. Der Geschwindigkeitsrekord der NDD-WE im Geradeauslaufen wurde aktuell<sup>22</sup> von keinem Team geschlagen. Somit kann die Geschwindigkeit als hervorragend beurteilt werden. In den ersten Jahren war die **Präzision** der NDD-WE ein Manko. Dies lag aber nie an der WE selbst, sondern vielmehr an deren Ansteuerung. In der neuesten Revision, die unmittelbar *FootStep*-Trajektorien als *Wal-*

<sup>22</sup>Stand RoboCup 2011.

*kRequests* verwendet, hat sich die Präzision deutlich verbessert, wie Tasse mit der *Open Challenge 2010* belegen konnte. Die Einführung des *AfterPreview*-Koordinatensystems war hierbei ein entscheidender Schritt zur Steigerung der Präzision. Eng verbunden mit der Präzision ist die **Manövrierfähigkeit**, die mit Abbildung 5.4.10 bereits belegt wurde. Der Roboter NAO ist mit der NDD-WE trotz seiner kinematischen Anomalie des geteilten Hüftgelenks zu einem vollständig quasi-omnidirektionalen Lauf fähig. Durch den *Preview*-Regler wird die **Reaktionsfähigkeit** des Systems begrenzt. Dieser Überlegung kann aber leicht begegnet werden, wenn man sich die Natur der Fortbewegung nochmals ins Gedächtnis ruft. Durch den Wunsch nach dynamischer Stabilität des Gangs, ist eine gewisse Trägheit des Systems per Definition gegeben. Diese wird durch den Regler in Abhängigkeit des gewünschten *Preview*-Fensters zwar erhöht. Doch bei hoher Schrittfrequenz kann die *Preview*-Phase soweit reduziert werden, dass eine Beeinträchtigung des Systems subjektiv nicht festzustellen ist. Dem negativen Einfluss der Verzögerung auf die Entscheidungen des *Behaviors* wird durch das *After-Preview*-Koordinatensystem begegnet. Dieses transformiert Wahrnehmungen in das ReKS nach Beendigung der *Preview*-Phase und ermöglicht es somit, den resultierenden Zeitversatz implizit in die *Behavior*-Entscheidungen zu integrieren. Ein objektiver Vergleich des Einflusses der WE auf den **Verschleiß** fällt schwer. Aus Erfahrung kann aber bestätigt werden, dass der größte Verschleiß des Roboters aus Stürzen und der folgenden Aufstehbewegung resultiert. Der Verschleiß durch die WE konnte zudem durch Erhöhung der Schrittfrequenz merklich reduziert werden. Dies ist einfach erklärt, bedenkt man, dass der CoM bei niedriger Schrittfrequenz größere Strecken bewegt werden muss, da dem System Dynamik fehlt, um den ZMP durch die Beschleunigung zu verschieben. Als Resultat werden besonders die *HipRoll*-Gelenke stärker beansprucht und schlagen öfters an ihren physikalischen Grenzen an<sup>23</sup>. Dies ist belegt, da den Getrieben dieser beiden Gelenke bei niedriger Schrittfrequenz deutlich häufiger die äußeren Zähne gebrochen sind. Eine ausführliche Untersuchung der **Energieeffizienz** des Laufes war nicht Zielsetzung dieser Arbeit. Der kurze Exkurs hat aber dennoch aufgezeigt, dass die Parameter der WE durchaus einen Einfluss auf die Energieaufnahme des Systems haben. Eine detaillierte Untersuchung dieser Abhängigkeit dürfte daher interessante Ergebnisse liefern, bleibt aber ausstehend. Die **Laufzeit** der NDD-WE erfüllt alle Vorgaben. Die Zykluszeit des *Motion*-Threads beträgt analog zum DCM 10 ms. Die Berechnungen der WE, inklusive der kinematischen, benötigen relativ konstant rund 3 ms. Folglich hat der *Real-time*-Scheduler des Betriebssystems keine Schwierigkeiten diese Zykluszeit zu garantieren. Zudem lässt die WE genug CPU-Zeit, um die Berechnungen der *Cognition* und des *Behaviors* mit einer ausreichenden Updaterate zu ermöglichen.

Zusammenfassend lässt sich festhalten, dass die NDD-WE alle aufgestellten Anforderungen an eine WE erfüllt. Die Tatsache, dass der zugrunde liegende Algorithmus im RoboCup viele Nachahmer gefunden hat, spricht ebenfalls dafür, dass die NDD-WE eine gute Referenzlösung für das Laufproblem NAO darstellt. Dennoch soll nicht unerwähnt bleiben, dass Problematiken existieren, die von dem vorgestellten Ansatz nicht erfasst werden. Diese lassen sich grob in die Klasse *Unzulänglichkeit des Modells* einordnen. Auch wenn der Roboter NAO in Wirklichkeit kein komplett starres Festkörper-System darstellt, so gilt diese Annahme für einen fabrikneuen NAO doch in ausreichender Nä-

<sup>23</sup>Dies wird zwar in der Theorie durch Aldebaran Arbeitsraumbegrenzung verhindert, doch in der Praxis passiert es bei höheren Geschwindigkeiten dennoch gelegentlich.

herung. Die NDD-WE kann den Roboter trotz der Vernachlässigung der Schwungbeindynamik stabil fortbewegen. Durch Verschleiß werden die Gelenke des Roboters mit der Zeit aber immer nachgiebiger. Als Resultat steigt die Flexibilität des Systems und es überlagern sich weitere nicht-modellierte Störungen durch Schwingungen. Dies führt bei einigen älteren Robotern dazu, dass diese auch vom Beobachter nicht mehr stabilisiert werden können. Das Problem wurde durch einen separaten Regler gelöst, dessen Aufgabe es lediglich war, die Neigung des Oberkörpers um seine  $y$ -Achse zu korrigieren<sup>24</sup>. Dies kann aber lediglich als Notlösung zur Umgehung der Problematik angesehen werden, da es sich nicht um eine saubere Integration in das Regelungskonzept handelt. Dennoch ist von einigen Teams bekannt, dass sie ebenfalls verschiedene Regler zur Stabilisierung einsetzen. Eine derartige Integration in die NDD-WE in Zukunft wissenschaftlich anzugehen wird daher als interessant erachtet. Urbann et al. haben in [113] bereits erste Untersuchungen der Flexibilitäten des Roboters durchgeführt und konnten zeigen, dass sich mit deren Integration in ein Simulationsmodell einige auftretende Effekte modellieren und somit erklären lassen. Mit diesem Wissen ließe sich der Systemzustand des Roboters besser schätzen. Auch ohne das grundlegende 3D-LIPM zu verändern, würde dieses Wissen helfen, das Beobachtermodell, und damit die ausgleichende Regelung, zu verbessern. Besonders im Hinblick auf die höhere Rechenleistung des NAO V4 bietet sich auch die Ersetzung des 3D-LIPM durch ein fortgeschritteneres Modell zur Weiterentwicklung der WE an. Der NAO ist also prädestiniert zukünftig, die in Kapitel 5.1 eingangs ausgeführte Unstimmigkeit, ob nun eine bessere Modellierung oder eine bessere Regelung zu einer besseren WE führt, an einer Plattform vergleichend zu erforschen.

Obwohl es sich um einen mathematisch basierten Ansatz handelt, spiegelt die Anzahl der WE-Parameter die Tatsache wieder, dass auch bei der modellbasierten Entwicklung eine Vielzahl Freiheitsgrade offen bleiben. Einige sind zum Ausgleich der Modell-Unzulänglichkeiten als Korrekturparameter aufgenommen worden. Doch Kapitel 5.3 zeigt auf, dass viele Parameter auch das physikalische Verhalten der Systems beeinflussen. Da der Suchraum durch die Vielzahl der Parameter groß ist und zudem deren Einfluss, wie beispielsweise bei den *Stiffness*-Parametern, ungeklärt ist, liegt es nahe diese in Zukunft weiter zu untersuchen. Mittels einer automatischen Anpassung könnte die WE beispielsweise für einzelne NAOs optimiert werden. Erste Ansätze einer solchen automatischen Kalibrierung wurden vom NAO-Team HTWK vorgeführt, doch existiert keine Veröffentlichung über deren Vorgehen. Abschließend ist die Parametrisierung der ZMP-Trajektorie besonders hervorzuheben. Diese beeinflusst als Führungsgröße zwar nicht unmittelbar den Regler, doch maßgeblich die Regelung des Systems. Bisher wurde diese immer als gegeben betrachtet. Doch kann nicht ohne weiteres beurteilt werden, welche ZMP-Vorgabe optimal ist. Das stabilisierende Gebiet ist durch die geplanten Fußpunkt zwar gegeben, doch lässt es viel Spielraum für mögliche ZMP-Trajektorien. Darüber hinaus hat Graf gezeigt, dass es sogar von Vorteil sein kann, wenn der ZMP das stabilisierende Gebiet zwischenzeitlich verlässt. Untersuchungen des Einflusses des Referenz-ZMP auf das Laufverhalten sind am realen Roboter nur extrem schwierig durchzuführen, da viele Effekte den untersuchten überlagern. Doch fortgeschrittene Simulatoren, wie Urbann's *MoToFlex*, begünstigen solche Untersuchungen.

---

<sup>24</sup>Die Anregung des Oberkörpers durch die Schritte zu einer Schwingung in dieser Richtung führt bei verschlissenen *Hip*- und *Ankle*-Winkeln zur signifikantesten Störung.

---

### Propriozeption der Bewegung

---

Die bewusste Bewegung eines Roboters ist das Resultat eines Aktionsbefehls des *Behavior*-Moduls. Damit ist eine Bewegung immer zielgerichtet, erfüllt also einen Zweck. Bislang wurde diskutiert, wie Bewegungsabläufe geplant werden können und Konzepte vorgestellt, die es dem Roboter ermöglichen, das Gleichgewicht während der Bewegung zu überwachen und bei Abweichungen den geplanten Ablauf mittels Regelungsalgorithmen stabilisierend zu korrigieren. Diese Stabilität der Bewegung ist für die Erfüllung ihres Zwecks wichtig, doch in der Regel eine reine Nebenbedingung. Um entscheiden zu können, ob eine Bewegung aber den eigentlichen Zweck erfüllt, müssen dem Roboter noch weitere Werkzeuge an die Hand gegeben werden. Er muss nicht nur die Abweichung von den angesteuerten Gelenkbewegung wahrnehmen, sondern auch seine resultierende Bewegung im Raum verfolgen können. Diese folgt zwar in der Regel aus der Bewegung im Gelenkwinkelraum, doch wird sich im Laufe des Kapitels zeigen, dass aus der Dynamik des Roboters auch Bewegungen resultieren können, die keine Änderung der Konfiguration zur Folge haben. Zudem können externe Kräfte, beispielsweise durch Kollision mit einem Objekt, dazu führen, dass ungewollte Bewegungen ausgeführt werden oder Bewegung im Gelenkwinkelraum nicht zu der gewünschten Bewegung im Weltkoordinaten-System (WKS) führt. Aus diesem Grund werden im Folgenden Algorithmen vorgestellt, die dem Roboter die Propriozeption seiner Eigenbewegung ermöglichen, um dadurch die Ego-Modellierung verbessern. Diese stellt ein entscheidende Grundlage für die *Behavior*-Planung dar. Hierzu werden teils Sensoren benötigt, die der Roboter NAO ab Werk nicht besitzt oder Algorithmen untersucht, die auf der CPU des Roboters nicht in Echtzeit ausgeführt werden können. Dies verletzt die eigentliche Intention der Arbeit, ist jedoch als Ausblick in die Entwicklung der Robotik zu verstehen, da beides in zukünftigen Generationen des NAO kein Hindernis mehr darstellen muss. Zunächst wird in Kapitel 6.1 die Wahrnehmung der Fortbewegung diskutiert und in Kapitel 6.2 der Einsatz von Sensoren untersucht, um diese zu korrigieren. Abschließend stellt Kapitel 6.3 fortgeschrittene Filteralgorithmen zur Schätzung der Pose des Roboters im Raum vor, die das Ziel haben, die Verfolgung der Eigenbewegung zu verbessern.

## 6.1 Odometrie

Die Einleitung in Kapitel 1 erläutert, dass Methoden der Lokalisierung nicht primär Thema dieser Arbeit sein sollen. Dennoch sind logischerweise die Bereiche Bewegung und Lokalisierung eines mobilen Roboters untrennbar miteinander verknüpft. Eine Bewegung hat immer die Änderung der Pose und oft auch der Position des Roboters zur Folge. Demnach ist eine klare Trennung beider Bereiche nur schwer möglich. Ohne in die Tiefen der Lokalisierung abzugleiten, erfolgt an dieser Stelle ein kurzer Exkurs mit dem geklärt wird in wieweit fehlerhaft ausgeführte Bewegungen ein Problem für die Ortsbestimmung des Roboters darstellen können. Eine tiefergehende Befassung mit dieser Thematik kann u. a. anhand der Ausarbeitungen [31] oder [109] erfolgen.

In der Lokalisierung wird zwischen *lokaler* und *globaler* Lokalisierung unterschieden. Der Themenbereich der globalen Lokalisierung befasst sich mit dem Problem, die aktuelle Position des Roboters in seiner Umwelt unabhängig von Vorwissen zu bestimmen. Die vorherigen Positionen sind also unbekannt. Dies kann nur mit Hilfe von exterozeptiver Sensorik und der Erkennung bekannter Punkte, sogenannter *Landmarken*, geschehen und ist daher nicht direkt mit der Bewegung des Roboters verknüpft. Aus diesem Grund wird hierauf nicht weiter eingegangen. Die Aufgabenstellung der lokalen Lokalisierung sieht hingegen anders aus. Sie ist in eigenen Worten definiert als:

Finde die Position  $P_i$  des Roboters im WKS zum aktuellen Zeitpunkt  $t_i$  unter der Voraussetzung, dass mindestens eine vorhergehende Position  $P_{i-1}$  des Roboters zum Zeitpunkt  $t_{i-1}$  bekannt ist.

Wenn der Roboter sich selbstständig von  $P_{i-1}$  nach  $P_i$  begeben hat<sup>1</sup>, kann dieser Übergang nur das Resultat der eigenen Fortbewegung sein. Somit ist die Kenntnis über die ausgeführte Bewegung sehr hilfreich bei dieser Art der differenziellen Lokalisierung. Auf dieser Idee basiert die in der Schifffahrt gebräuchliche Methode der *Koppelnavigation*, englisch *dead reckoning*. Bei dieser Art der Navigation wird zur Positionsbestimmung der Kurs und die Geschwindigkeit des Schiffes mit geeigneten Hilfsmitteln gemessen. Ausgehend von einem Startpunkt lässt sich so die aktuelle Position des Schiffes durch Integration über die Zeit berechnen. Dieses Verfahren zur Lokalisierung ist in der Robotik unter dem Begriff *Odometrie*, englisch *odometry* geläufig. Eine einheitliche Definition der Methode gibt es nicht, aber [100] umschreibt sie beispielsweise wie folgt:

„In the context of autonomous vehicles, odometry usually refers to the use of data from the actuators (wheels, treads, etc.) to estimate the overall motion of the vehicle.“

Aufgrund der Einfachheit dieses Verfahrens findet es bei vielen mobilen Robotern Verwendung und ist daher in Lehrbüchern bereits ausführlich diskutiert und in der Vergangenheit Thema etlicher Forschungsarbeiten gewesen. Allerdings lässt sich anhand der zitierten Definition schon erahnen, dass diese Untersuchungen meist im Kontext der rollenden Roboter oder Fahrzeuge durchgeführt wurden. Zwar gibt es auch einige wenige Untersuchungen, die sich mit der Odometrie laufender Roboter beschäftigen,

---

<sup>1</sup>Das sogenannte *Kidnapped-Robot-Problem* wird bewusst nicht betrachtet.



doch zumeist hatten diese 4 oder mehr Beine, vergleiche beispielsweise [46] oder [70].

Da sich die Fortbewegung von humanoiden Robotern grundlegend von der rollender - und ebenfalls merklich von der viel-beiniger Roboter - unterscheidet, folgt an dieser Stelle zunächst eine theoretische Betrachtung und die Übertragung der Methodik der Odometriemessung auf einen zweibeinigen Roboter. Anschließend findet sich im folgenden Kapitel eine experimentelle Evaluation der Odometriemessung und eine Diskussion der Möglichkeiten, Abweichungen der realen Bewegung von der Odometrie sensorisch zu erfassen.

### 6.1.1 Messung der Odometrie zweibeiniger Roboter

Um die Odometrie zur Aktualisierung der Position des Roboters im Raum nutzen zu können, ist es nötig, diese Messen zu können. Sucht man in der Literatur nach der Messung der Odometrie eines mobilen Roboters, so findet man den Hinweis, dass dies mittels eines *Hodometers*<sup>2</sup>, englisch als *odometer* bezeichnet, geschieht. Wie die Zusammensetzung aus den griechischen Wörtern *hodós* - Weg - und *métron* - Maß - nahelegt, handelt es sich dabei um ein Gerät, das die zurückgelegte Wegstrecke eines Roboters misst. Entsprechend wird dieses in [109] auch definiert als:

„Odometers are sensors that measure the revolution of a robot’s wheels. As such they convey information about the change of state.“

Das Zitat legt nahe, dass die Art des Odometers direkt mit der Fortbewegungsart des jeweiligen Roboters zusammenhängen muss. In der Literatur wird sich zumeist darauf beschränkt, das Odometriekonzept für rollende Roboter zu erläutern, für die sich ein Hodometer mittels eines *Radencoders*, englisch *wheel encoder* recht einfach realisieren lässt. Dieser bestimmt auf optische, elektrische oder mechanische Weise die Umdrehung der Räder des Roboters. Ein detaillierter Überblick dieser Sensoren ist dem Buch [101] zu entnehmen. Wenn die Räder eines mobilen Roboters sicheren Kontakt mit dem Boden haben, führt aufgrund der Haftreibung eine Bewegung der Räder direkt zu einer Fortbewegung des Roboters. In der Theorie wird davon ausgegangen, dass die Räder des Roboters ständigen Bodenkontakt haben und weder das *sliding constraint*, noch das *rolling constraint* verletzt werden, vergleiche [101]. In diesem Fall kann die Rotation der Räder über deren Kinematik direkt mit der Fortbewegung verknüpft werden. Die Informationen der Radencoder können also direkt in die Fortbewegung des Roboters umgerechnet werden und somit kann diese unmittelbar gemessen werden. Für eine eingehendere Betrachtung der Odometrie rollender Roboter sei auf [100] verwiesen.

Die Fortbewegung eines laufenden Roboters folgt einem anderen Schema. Aufgrund diskreter Veränderung der Position, nochmals verdeutlicht in Abbildung 6.1.1, ist offensichtlich, dass die Bewegung eines Aktuators nicht in jedem Fall die Fortbewegung des Roboters zur Folge haben muss. Das Konzept der Odometriemessung rollender Roboter, die Bewegung der Aktuatoren unmittelbar in die Fortbewegung des Roboters zu

---

<sup>2</sup>Im Sprachgebrauch üblich ist auch die aus dem englischen eingedeutschte Form des *Odometers*. Streng genommen gibt es diesen Begriff aber nicht in der deutschen Sprache.

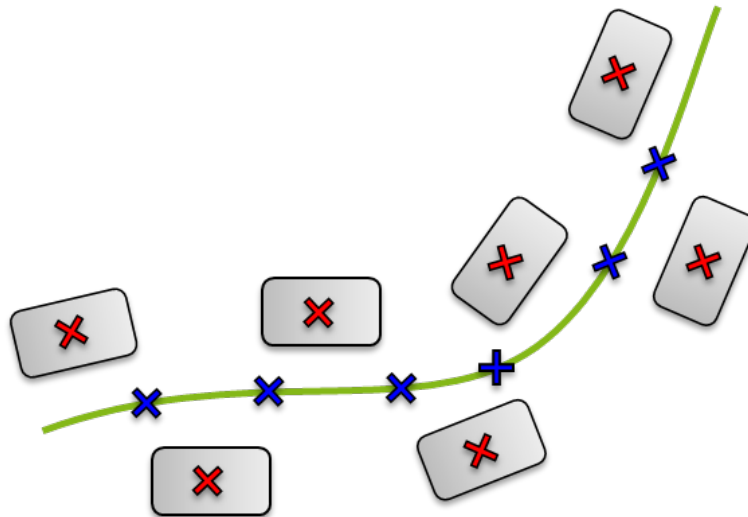


Abbildung 6.1.1: Diskrete Positionsänderung bei kontinuierlichem CoM-Verlauf.

überführen, kann also nicht unangepasst übernommen werden. Grundsätzlich bestehen drei Methoden, um mittels Sensorik die Odometrie laufender Roboter zu erfassen.

**Inertialsensoren** Die Inertialsensoren des Roboters sind in der Lage, Beschleunigungen des Oberkörpers zu messen. Die Position des Roboters wurde in Kapitel 2.1 zwar als Ursprung des Fußkoordinaten-Systems (FKS) definiert, diese lässt sich aber mittels der Kinematik auch aus der Position und Orientierung des Oberkörpers berechnen. Somit kann die Fortbewegung des Roboters theoretisch nach dem *Weg-Zeit-Gesetz* mittels Integration der Inertialsensor-Messwerte über die Zeit bestimmt werden. Der Roboter bewegt während des Ganges jedoch seinen Oberkörper fortwährend relativ zum Standfuß, ohne damit eine Fortbewegung zu erzielen. Dies ist das Resultat der periodischen Hin- und Herbewegung in  $y$ -Richtung, um den ZMP gemäß des invertierten Pendelmodells zu verschieben. Hinzu kommt, dass in der Praxis die Ansteuerung von schnellen Bewegungen auf dem NAO nicht ohne Ruck geschehen können, so dass dieser ständig Störbeschleunigungen ausgesetzt ist. In der Theorie löschen sich diese für die Fortbewegung irrelevanten Beschleunigungen durch die Summierung aus. Auch die Fehler des Sensors durch Rauschen würden sich in gewissem Maße aufheben, da es sich primär um ein gleichverteiltes weißes Rauschen handelt. Aber zu dieser Fehlerquelle kommt noch ein weiterer Störeinfluss. Für die Berechnung geschieht auf Basis der Beschleunigungswerte in Weltkoordinaten. Der Inertialsensor misst diese aber im Koordinatensystem des Oberkörpers. Diese Beschleunigungen sind aber nur identisch, wenn der Oberkörper senkrecht und parallel zum Boden steht. Dies wird versucht durch das 3D-LIPM zu erreichen, doch gelingt dies in der Praxis nicht immer. Folglich unterliegt die Sensormessung einer Rotation. Dies geschieht aber auf Basis der Lage des Oberkörpers im Raum. Diese wird entweder ebenfalls über die Inertialsensoren oder alternativ aus der Kinematik berechnet. Die Fehler der Sensoren potenzieren sich demnach weiter. Die resultierenden Fehler sind zwar systematischer Natur, doch ist ihr Ausmaß nicht bekannt, da es abhängig vom aktuellen Systemzustand ist. Folglich können sie nicht durch

explizite Betrachtung ausgeglichen werden. Da die Fehler somit nicht rein statistischer Natur sind, ist nicht mehr gegeben, dass deren Zeitmittelwert Null ist. Gleichen sich die Fehler aber nicht mehr aus, divergiert als Folge die Odometriemessung mit der Zeit. Diese Methode der Odometriemessung hat sich daher nach ersten Versuchen am NAO als kaum praktikabel erwiesen. Auf eine weiterführende experimentelle Untersuchung wird daher an dieser Stelle verzichtet.

Auch wenn Odometriemessung unmittelbar aus den Messungen des Beschleunigungssensors nicht in praxistauglichen Ergebnissen resultiert, berichtet Team B-Human in [90] davon, die Bewegungsschätzung mittels geeigneter Filter und unter Zuhilfenahme der Gyrometerdaten deutlich verbessert zu haben. Untersuchung der reinen Odometrie fanden aber nicht statt, da diese bei B-Human ein integraler Bestandteil des globalen Lokalisierungs-Moduls ist. Im späteren Verlauf dieses Kapitel wird aber in 6.3 die Idee der Kalman-Filterung unter weiterführender Prämisse aufgegriffen und genauer evaluiert. Doch selbst mit fortgeschrittener Kalman-Filterung zeigt sich bereits in der Simulation ein Drift der Position über die Zeit. In der Realität ist demnach nochmal eine signifikante Steigerung des Effekts zu erwarten. Zudem muss an dieser Stelle nochmals darauf hingewiesen werden, dass der NAO nur über ein 2-Achsen Gyrometer verfügt und somit Rotationen um die Hochachse nicht messen kann. Mit dieser Methode ist es also möglich die Position des NAO zu bestimmen - nicht jedoch die Orientierung. Doch eben diese ist für die Odometrie inhärent wichtig, wie Kapitel 6.2 aufzeigen wird.

**Kamera** Eine weitere Methode zur Schätzung der Fortbewegung mobiler Roboter ist die Verwendung von Kamerabildern. Die Fortbewegung des Roboters führt zu einer Bewegung der Kamera und diese wiederum dazu, dass sich Objekte und Landmarken relativ zur Kamera bewegen. Folglich verschiebt sich die Position innerhalb des Bildes. Mittels eines geeigneten *Feature*-Detektors können Bewegungen im Bild nachverfolgt werden und so, beispielsweise über die Berechnung des *Optical-Flows*, wiederum Rückschlüsse auf die Bewegung des Roboters gezogen werden. Diese Methode findet in der mobilen Robotik Anwendung und zeigt bei rollenden Robotern in geeigneten Umgebungen gute Resultate, wie der Vergleich von DeSouza und Kak belegt [27]. Grund hierfür ist, dass bei diesen jede Bewegung der Kamera auch aus einer Fortbewegung des Roboters resultiert, solange man von einer guten Dämpfung der Reifen ausgehen kann. Doch wie bereits bei der Diskussion der Inertialsensoren festgestellt wurde, ist dies bei humanoiden Robotern nicht der Fall, da der Oberkörper während der Fortbewegung relativ zur Position des Roboters bewegt wird. Damit ergibt sich analog die Akkumulation-Problematik der Fehler, die bei der Bestimmung der Bewegung aus Kamerabildern noch größer ist. Aus diesem Grund ist nicht bekannt, dass diese Methode bis zum heutigen Zeitpunkt erfolgreich auf humanoide Roboter zur Odometriemessung angewandt wurde. Vielmehr werden einzelne Bilder der Kamera mit erkennbaren, möglichst eindeutigen *Features* dazu verwendet, um auf Basis geeigneter Filteralgorithmen eine globale Lokalisierung durchzuführen. Eine detailliertere Beschreibung der visuellen globalen Lokalisierung des Teams NAO-Devils Dortmund (NDD) ist beispielsweise in [23], [25] und [54] zu finden.

**Kinematik** Die Position eines laufenden Roboters wurde als Ursprung des FKS definiert. Da dies sich während des Ganges nur beim Übergang in eine *Single-Support*-Phase verschiebt, geht die Fortbewegung des Roboters immer mit einer Veränderung der Fußposition einher - oder genauer gesagt mit der Änderung des Bodenkontaktes eines der Füße. Somit kann die Positionsänderung eines humanoiden Roboters inkrementell über die Abfolge der Fußaufsatzpunkte beschrieben werden, wie Abbildung 6.1.1 verdeutlicht. Diese können nicht direkt vom Roboter wahrgenommen werden. Doch beim Übergang von der *Single*- in die *Double-Support*-Phase kann die Position des neuen Fußpunktes relativ zum Standfuß der vorherigen *Single-Support*-Phase bestimmt werden. Dies geschieht mit Hilfe der Sensorinformationen aus den Gelenkwinkelencodern. Aus deren Stellung kann mittels der Kinematik des Roboters das inkrementelle Update der Bewegung berechnet werden. Die Berechnung der neuen Orientierung erfolgt nach demselben Vorgehen. Das Kinematikverfahren ähnelt also in Teilen der Odometrieermessung rollender Roboter, die Berechnung erfolgt jedoch nicht unmittelbar, sondern wird sukzessive dem Fortbewegungsmuster angepasst.

In der Praxis liefert die Methode der Kinematik die erfolgsversprechendsten Ergebnisse und wird daher im Folgenden eingehender untersucht. Für eine Beurteilung der Genauigkeit dieses Messverfahrens, muss zunächst eine Abschätzung des maximal möglichen Odometriefehlers erfolgen, der aus dem Kinematikfehler resultiert. Die Problematik wird zunächst theoretisch betrachtet. Eine praktische Untersuchung folgt im Anschluss. Es sei an dieser Stelle bereits vorgreifend angemerkt, dass Schmitz et al. in [95] vergleichbare Beobachtungen zur Fehlerursache mittels eines *Motion-Capturing*-Systems, jedoch an einem anderen Robotermodell, gemacht haben und somit die Resultate bekräftigen. Ihre Ergebnisse sind vergleichbar, jedoch versuchen sie in erster Linie den Fehler durch einen Lernalgorithmus auszugleichen. Der Fokus an dieser Stelle liegt auf dem Verständnis der Ursache.

Bereits in Kapitel 4.1.1 wurde eine Untersuchung durchgeführt, wie Ungenauigkeiten in der Gelenkwinkelstellung zu einer fehlerhaften Berechnung der Kinematik führen. Hierbei stellte sich heraus, dass nicht alle Gelenke den gleichen Einfluss haben, sondern sich durch die kinematische Verknüpfung gegenseitig beeinflussen. Die Fehlerbetrachtung für die Fußpositionen kann analog zur Berechnung der Lage im Raum durchgeführt werden. Die Betrachtung erfolgt daher nicht mehr im Detail. Dennoch ist es wichtig, auf die Unterschiede einzugehen, da in die Odometrieberechnung nicht nur der Fehler des Standbeins, sondern auch des Schwungbeins eingeht. Die Analyse erfolgt nur für einen Fehler in  $x$ -Richtung, die Betrachtung in  $y$ -Richtung kann nach dem gleichen Schema erfolgen. Betrachtet man zunächst die Abschätzung der Grenze des Odometriefehler für einen Schritt in  $x$ -Richtung, so ist zu erkennen, dass aus den Fehlern  $\delta_i = \pm 1,5^\circ$  der jeweiligen Gelenkwinkelencoder sowohl ein Kinematikfehler  $\Delta_{Stand}$  des Stand-, als auch  $\Delta_{Schwung}$  des Schwungbeines entsteht, wie Abbildung 6.1.2 zu entnehmen ist. Im schlimmsten Fall addieren sich diese auf, da der Fehler des Standbeines sich auf die Position des Hüftgelenkes des Schwungbeines auswirkt. Eine Fehlerbetrachtung analog zu Kapitel 4.1.1 unter Beachtung des Stand- und Schwungbeines liefert eine ungefähre obere Grenze von 25 mm für einen Schritt in  $x$ -Richtung. Dies erscheint zunächst

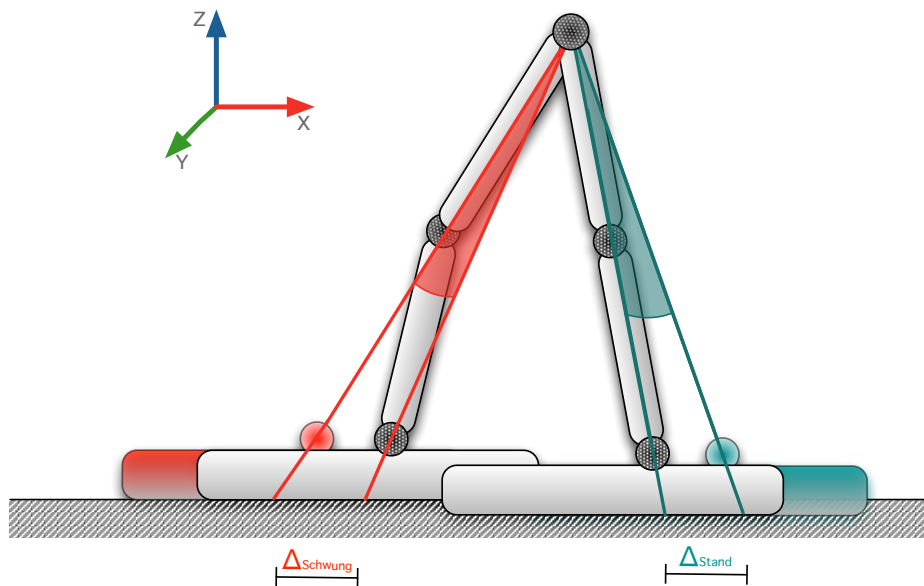


Abbildung 6.1.2: Resultierende Abweichung  $\Delta_x$  aus dem Winkelfehler des Schwung- und Standbeins.

recht groß, zieht man die Schrittweite der WE in Betracht. Doch stellt dies eine obere Grenze für den Fall dar, dass sich alle Winkelfehler addieren. Für eine praxisnähere Abschätzung muss die Natur der Fehler nochmals genauer betrachtet werden.

Die bisherige Untersuchung bezieht sowohl systematische als auch stochastische Fehler gleichermaßen mit ein. Die stochastischen Fehler sind aber aufgrund ihrer Natur gleichverteilt. Somit summieren sie sich über mehrere Schritte im Allgemeinen nicht auf und können für den Odometriefehler einer längeren Folge von Schritten vernachlässigt werden. Ähnlich verhält es sich mit den systematischen Fehlern der Hüfte. Betrachtet man eine Folge von zwei Schritten, so heben sich diese gegenseitig auf. Dies lässt sich am anschaulichsten an einem konkreten Beispiel erläutern. Wenn davon ausgegangen wird, dass aufgrund eines systematischen Fehlers in einem der Hüftgelenke dieses den Winkel in jedem Fall überschätzt, so führt dies dazu, dass es als Schwungbein den Fußaufsatzpunkt weiter entfernt annimmt und somit die Schrittweite überschätzt. Im nächsten Schritt - wenn dieses Bein Standbein ist - unterschätzt dieser Fehler jedoch die Schrittweite. Somit mittelt sich der systematische Fehler der Hüfte in jedem Doppelschritt heraus und kann für eine Abschätzung über eine längere Wegstrecke ebenfalls ignoriert werden. Anders verhält es sich hingegen mit dem systematischen Fehler der Fußgelenke. Der Fehler des Standfuß-Gelenks wirkt sich über die gesamte Beinlänge auf des Odometriefehler aus. Wenn dieses Gelenk im nächsten Schritt Schwungbein-Gelenk ist, wirkt sich der Fehler nur noch über die Fußhöhe aus. Folglich mittelt sich der Fehler nur zu einem gewissen Anteil aus. Der systematische Anteil des Gelenkwinkelfehlers wurde bei ausreichender Sorgfalt mit  $\pm 1^\circ$  abgeschätzt. Durch die Kinematik ergibt sich damit für lange Schrittfolgen eine ungefähre obere Grenze des Odometriefehlers durch die Gelenkwinkencoder pro Schritt von 6 mm.

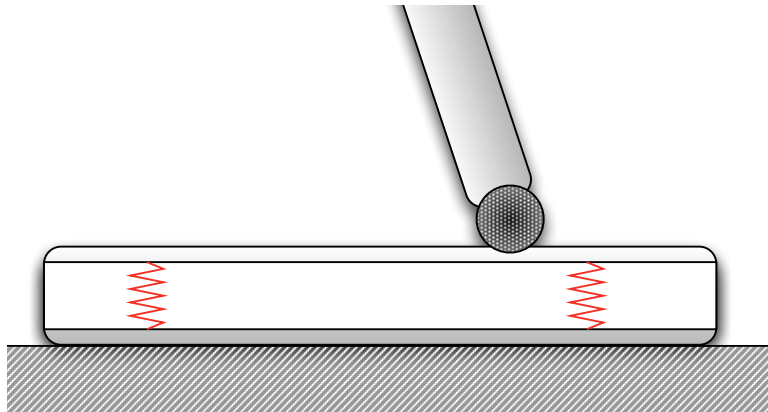


Abbildung 6.1.3: Modellierung des Fußes mit Federn.

Betrachtet man den NAO eingehender, fällt zusätzlich zu den Gelenkwinkelencodern noch eine weitere Fehlerquelle für die Berechnung der neuen Fußposition auf. Für die theoretische Modellierung wurde der NAO als System von Festkörpern angenommen, die mit Gelenken verbunden sind. Auch wenn dies in guter Näherung für die meisten Körperteile des NAO gilt, nehmen die Füße eine Sonderrolle ein. Damit Kraft von der starren Fußplatte auf die FSR-Sensoren übertragen werden kann, ist diese leicht beweglich gegen den eigentlichen Fuß gelagert. Der Fuß sollte daher in der Praxis besser als zwei Festkörper betrachtet werden, die durch Federn an den Eckpunkten verbunden sind. Verdeutlicht wird dies in Abbildung 6.1.3. Betrachtet man nun einen Schritt des NAO, so führt die Verlagerung des Schwungbeines nach vorne zu einer Verschiebung des CoP in die gleiche Richtung. Folge ist eine Stauchung der vorderen und Streckung der hinteren Federn. Auch wenn die Federkonstanten sehr groß sind, führt dies unweigerlich dazu, dass die Oberfläche des Fußes um einen gewissen Winkel zum Boden geneigt wird. Dieser systematische Fehler wirkt sich über die gesamte Beinlänge auf die Odometrie aus. Das Resultat ist immer, dass der reale Fuß weiter entfernt aufgesetzt wird, als durch die Kinematik angenommen. Geht man nun davon aus, dass durch den Fehler der Gelenke die Odometrie ebenfalls unterschätzt wird, summieren sich die Effekte der Fußplatte und der Gelenkwinkelencoder. Eine Abschätzung dieses Fehlers ist nur schwer möglich, da der Neigungswinkel des Fußes nicht gemessen werden kann. Zudem ist er auch von der Schrittweite abhängig, da eine größere Verlagerung des Schwungfußes nach vorne zu einer größeren Hebelkraft auf den Standfuß führt. Gleiches gilt für die Geschwindigkeit, da durch eine höhere Dynamik mehr Kraft auf den Fuß ausgeübt wird. Läuft der Roboter auf einem weichen Boden, tritt das gleiche Phänomen nochmals in Wechselwirkung des Fußes mit dem Untergrund auf - durch die Gewichtsverlagerung sinkt der vordere Teil des Fußes stärker in den Boden ein, als die Ferse. Auch die Beschaffenheit des Bodens kann also einen Einfluss haben. Da die Federkonstanten  $D_{Fu\ddot{u}ss}$  aber groß sind und sich der Fuß daher nur mit großer Kraft bewegen lässt, kann davon ausgegangen werden, dass im Mittel ein Winkel von  $\pm 1^\circ$  eine realistische obere Grenze darstellt. Der Einfluss des Fußsohlenfehlers des Standbeins beträgt somit nochmals rund 4 mm pro Schritt. Der gesamte Fußfehler in  $x$ -Richtung resultiert damit in Summe zu:

$$\Delta_x = 10 \text{ mm.} \quad (6.1.1)$$



Wie sehr sich der bisher berechnete absolute Fehler relativ zur zurückgelegten Strecke auswirkt ist von der Schrittlänge abhängig, d.h. für größere Schrittweiten ist der Einfluss des Fehlers auf die Odometrie geringer. Daher wird diese Betrachtung experimentell an einem konkreten Beispiel fortgeführt und welches in Kapitel 6.1.2 zu finden ist.

Es ist anzumerken, dass für dieses Odometrieverfahren der Zeitpunkt benötigt wird, an dem der Schwungfuß wieder auf dem Boden aufsetzt. Hierzu können wahlweise die Informationen der Fußdrucksensoren verwendet werden oder die *Walking-Engine* kann den theoretischen Zeitpunkt des Wechsels vorgeben. In beiden Fällen entsteht ein weiterer Fehler in der Odometrie. Im Falle der FSR-Sensoren liegt dies in der Latenz begründet, durch die Sensorinformationen erst verspätet zur Verfügung stehen. Im Falle der WE ist davon auszugehen, dass der geplante Aufsatzzeitpunkt nicht exakt der Realität entspricht. Der Fehler ist in beiden Fällen aber als gering anzunehmen und wird nicht weiter untersucht. Für diese Arbeit wurde der Methode des theoretischen Zeitpunkts der Vorzug gegeben.

### 6.1.2 Experimentelle Odometriemessung

Zur weiteren praktischen Begutachtung des kinematischen Verfahrens wird dieses anhand eines Experiments genauer untersucht. Der Roboter startet jeweils aus dem Stand in der Mitte des Fußballfeldes. Wiederholt werden unterschiedliche Bewegungsansteuerungen durchgeführt:

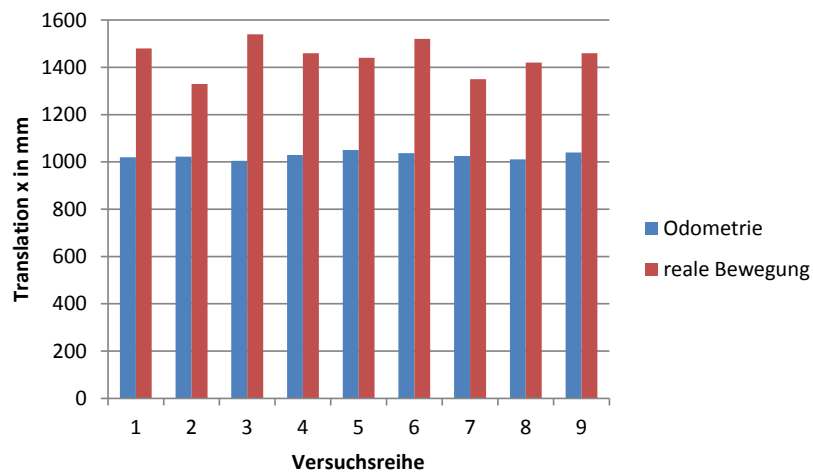
- Translatorische Bewegung in  $x$ -Richtung mit 100 mm/s und 200 mm/s.
- Translatorische Bewegung in  $y$ -Richtung mit 50 mm/s.
- Rotatorische Bewegung um die  $z$ -Achse mit 0,4 rad/s (rund  $22^\circ$ /s).

Die Ansteuerung der gewünschten Bewegung wird jeweils solange vom Roboter durchgeführt, bis der *OdometryOnly-Selflocator*<sup>3</sup> die gewünschte Strecke überschreitet. Die gemessene Bewegung kann nicht für jeden Schritt mit der tatsächlich ausgeführten verglichen werden. Niemüller et al. haben hierzu zwar den Nutzen eines *Motion Capturing* Verfahrens aufgezeigt [84] und auch Burchardt et al. zeigen in [13], dass bereits eine einfache Deckenkamera in Kombination mit der SSL-Software [123] brauchbare Ergebnisse liefert. Jedoch steht keines dieser visuellen Trackingsysteme am Institut zur Verfügung. Somit wird nur im Anschluss die tatsächlich zurückgelegte Strecke und Orientierungsänderung des NAO manuell als Referenzwert bestimmt. Alle Experimente wurden mit einer Schrittdauer von 0,5 s für eine Periode, also einen Doppelschritt, durchgeführt.

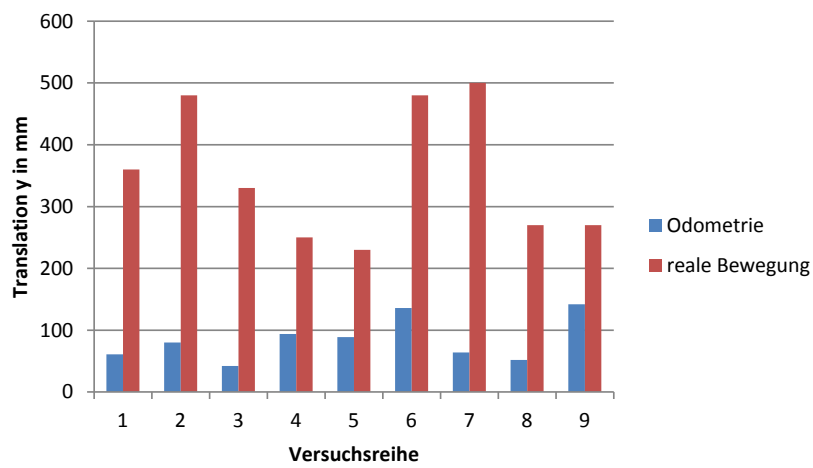
**Experiment 6.1.2.A - Translation in  $x$ -Richtung** Zur Untersuchung der Odometrie läuft der NAO, beginnend beim Anstoßpunkt, aus dem Stand in positiver  $x$ -Richtung. Die bisher durchgeführten theoretischen Überlegungen zeigen auf, dass die Schrittweite

---

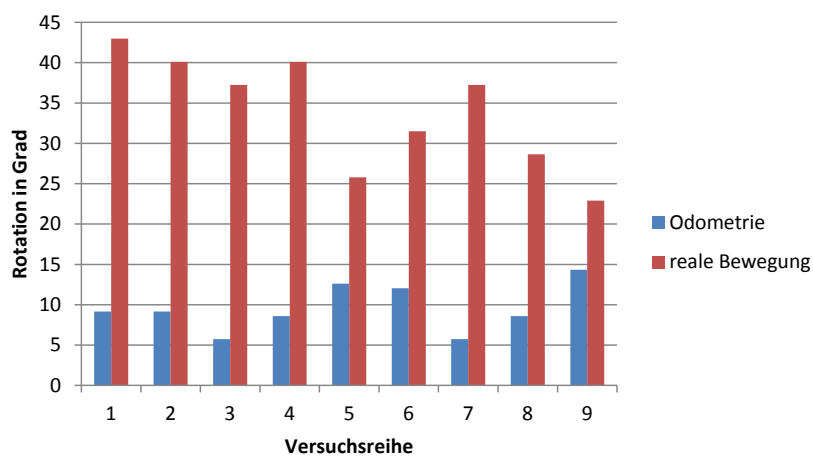
<sup>3</sup>Ein Teilmodul der NDD-Lokalisierung, das nur Odometrieinformationen nach dem beschriebenen kinematischen Verfahren auswertet.



(a) Translation in  $x$ -Richtung.



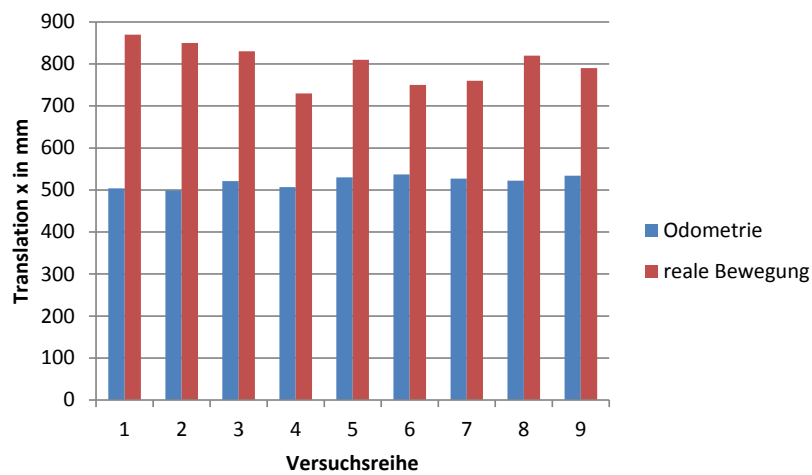
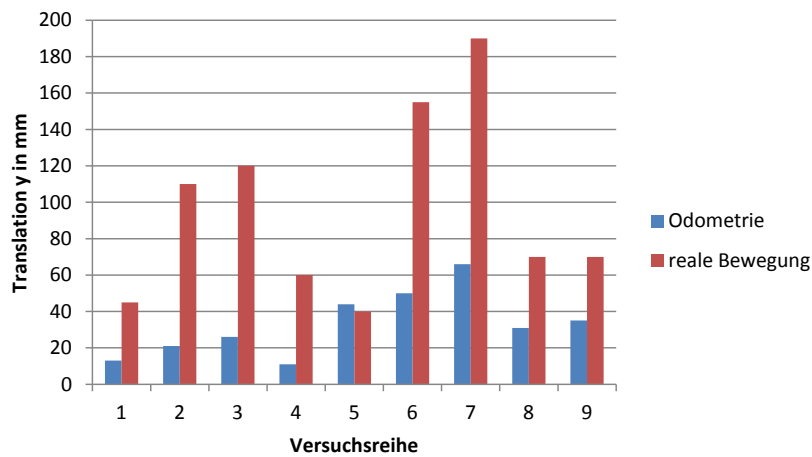
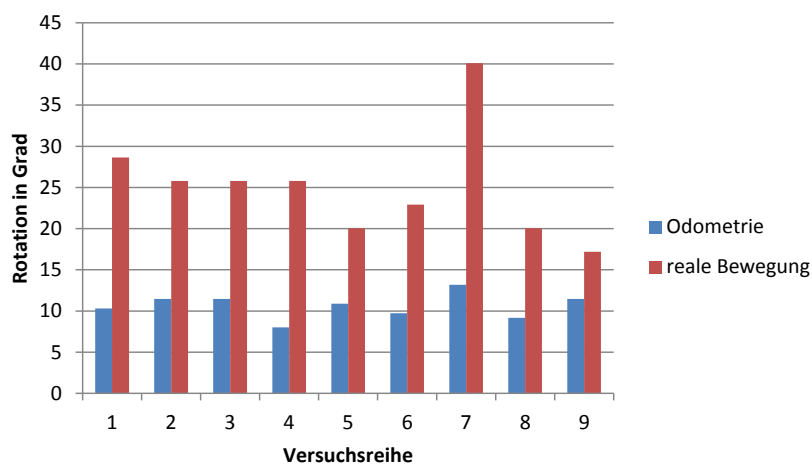
(b) Translation in  $y$ -Richtung.



(c) Rotation um  $z$ -Achse.

Abbildung 6.1.4: Experiment 6.1.2.A - Lauf in  $x$ -Richtung mit  $v = 200$  mm/s.



(a) Translation in  $x$ -Richtung.(b) Translation in  $y$ -Richtung.(c) Rotation um  $z$ -Achse.Abbildung 6.1.5: Experiment 6.1.2.A - Lauf in  $x$ -Richtung mit  $v = 100 \text{ mm/s}$ .

einen Einfluss darauf hat, wie stark sich der Sensorfehler auf die Odometrie auswirkt. Daher wird dieses Experiment mit unterschiedlichen Geschwindigkeiten wiederholt. Die Auswertung des Experiments mit  $v_x = 200$  mm/s ist in Abbildung 6.1.4 zu finden. Eine Betrachtung der Werte in 6.1.4(a) zeigt, dass die Odometriemessung die tatsächlich in  $x$ -Richtung zurückgelegte Strecke in jedem Fall unterschätzt. Der Roboter bewegt sich also weiter fort, als er denkt. Die Auswertung ergibt einen mittleren Fehler von 41 Prozent. Bei einer Schrittfrequenz von 4 Einzelschritten pro Sekunde folgt für den NAO eine Schrittweite von 50 mm. Der in Kapitel 6.1.1 abgeschätzte Fehler  $\Delta_x$  für eine Schrittfolge dürfte daher mit 10 mm im Höchstfall einen Einfluss von 20 Prozent auf die Messung haben. Da die Messung aber mindestens um 30 Prozent und maximal sogar um über 50 Prozent abweicht, liegt die Vermutung nahe, dass zumindest ein weiterer Fehler Einfluss auf die Messung nimmt.

In einem zweiten Teil des Experiments wird der Lauf wiederholt, allerdings mit einer Geschwindigkeit von  $v_x = 100$  mm/s. Damit die Schrittzahl in dergleichen Größenordnung bleibt wie im ersten Teil des Experiments, hält der Roboter nun aber an, nachdem der *OdometryOnly-Selflocator* angibt, er habe eine Strecke von 500 mm in  $x$ -Richtung zurückgelegt. Die Auswertung dieses zweiten Laufs in Abbildung 6.1.5 zeichnet ein ähnliches Bild. Auch bei niedrigerer Geschwindigkeit bewegt sich der Roboter weiter, als die Messung der Odometrie ergibt. Der Wert liegt im Mittel sogar mit einem Faktor von 1,54 noch höher als beim ersten Teil des Experiments. Dies stützt die im vorangehenden Kapitel aufgestellte Theorie, dass ein konstanter Winkelfehler bei kleinerer Schrittweite einen größeren Einfluß auf den Fehler der Odometriemessung hat. Dennoch kann auch in diesem Fall der Fehler nicht vollständig durch die Winkelfehler-Theorie erklärt werden, da bei einer Schrittweite von 25 mm dieser nur einen Einfluss von 40 Prozent im *Worst Case* auf die Strecke haben sollte. Dieser Wert wird aber immer überschritten und im Maximum weicht die Messung um über 70 Prozent ab. Zudem ist in beiden Durchläufen eine starke Streuung der zurückgelegten Strecke zu verzeichnen. Auch diese lässt sich nicht durch einen rein systematischen Fehler erklären.

Bei der weiteren Betrachtung der Ergebnisse in 6.1.4(b) und 6.1.5(b) fällt auf, dass auch eine Fortbewegung in  $y$ -Richtung stattgefunden hat, obwohl der Roboter sich nur in  $x$ -Richtung bewegen sollte. Ein Fehler  $\Delta_y$  kann durch die Gelenkwinkelencoder natürlich auch entstehen, wenn keine Bewegung in diese Richtung vollzogen wird. Durch das fehlende Kniegelenk ist zu erwarten, der Fehler in  $y$ -Richtung geringer als der Fehler in  $x$ -Richtung ist. Doch selbst wenn  $\Delta_y = \Delta_x$  angenommen würde, so dürfte sich dieser bei ungefähr 20 Schritten maximal im Bereich von 120 mm bewegen. Für  $v_x = 100$  mm/s stimmt dies im Mittel mit einer Abweichung von 95 mm in Näherung sogar recht gut. Die Abweichungen für den schnellen Lauf liegen jedoch mit einem Mittel von 352 mm erheblich darüber. Eine plausible Begründung hierfür legt eine Betrachtung der rotatorischen Abweichungen in Abbildung 6.1.4(c) und 6.1.5(c) nahe. Über die Folge von 20 Schritten summiert sich beim NAO in beiden Versuchsreihen ein Fehler von bis zu  $45^\circ$  in der Orientierung auf. Dies deckt sich mit der visuellen Beobachtung, die während der Durchführung des Experimentes gemacht werden konnte. Der NAO rotiert bei jedem Schritt um seine Hochachse. Eine Diskussion dieses Effekts und seines Einflusses findet im folgenden Kapitel 6.2 statt. Es sei jedoch vorweggenommen, dass ein Rotationsfehler der Odometrie sich über die Zeit fortpflanzt und somit ebenfalls zu

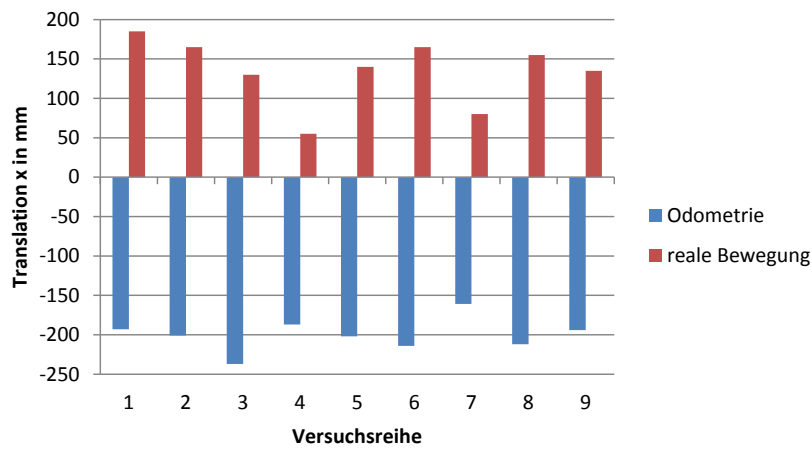
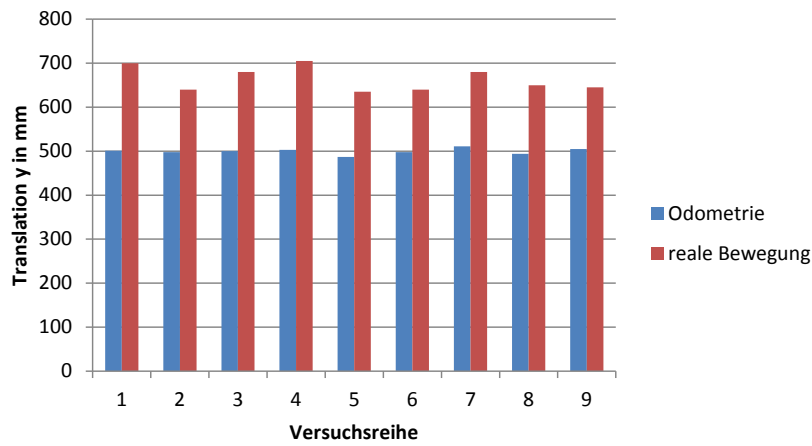
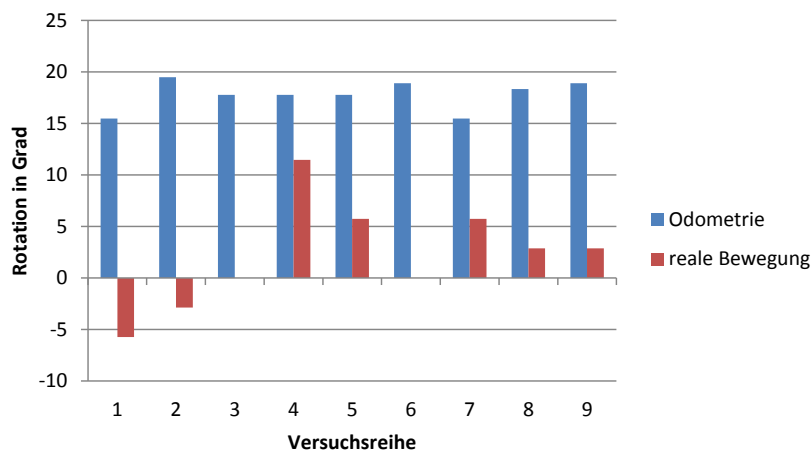
einer translatorischen Abweichung in  $y$ -Richtung führt. Da der Rotationsfehler bei der höheren Geschwindigkeit im Mittel merklich höher als bei 100 mm/s liegt, erklärt dies auch, warum sich bei diesem Lauf deutlich größere Abweichungen in der  $y$ -Richtung ergeben.

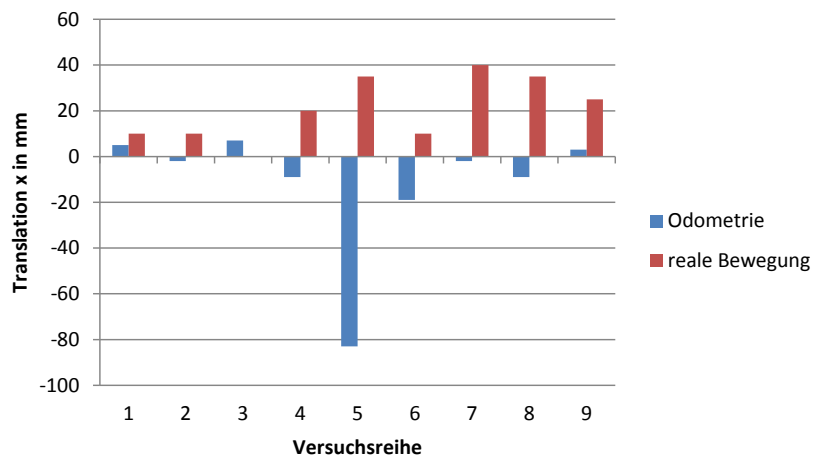
**Experiment 6.1.2.B - Translation in  $y$ -Richtung** Zur Untersuchung der Odometrie läuft der NAO, beginnend beim Anstoßpunkt, aus dem Stand wiederholt in positive  $y$ -Richtung. Die Auswertung dieser Läufe, dargestellt in Abbildung 6.1.6, zeigt ein ähnliches Bild wie Experiment 6.1.2.A. Die Bewegung in Laufrichtung wird von der Odometrie in jedem Fall unterschätzt, allerdings in einem Bereich, der noch durch Fehler in der Winkelmessung zu erklären wäre. Allerdings zeichnet die Messung der Orientierung und der  $x$ -Translation ein auffälliges Bild. Abbildung 6.1.6(c) weist für den Roboter in der Messung der Odometrie durch die Gelenkwinkelsensoren einen verhältnismäßig gleichbleibenden Fehler in der  $z$ -Rotation auf. Der reale Roboter scheint aber während des Experiments praktisch keine Rotationsveränderung durchgeführt zu haben. Gleichzeitig misst die Odometrie eine Abweichung in  $x$ -Richtung, siehe Abbildung 6.1.6(a), die der realen Bewegung entgegen verläuft. Die Erklärung dieses Phänomens lässt sich leider nicht direkt anhand der Ergebnisse, wohl aber anhand der Beobachtungen während des Experimentes erklären. Auch in diesem Experiment rotierte der Roboter während des Laufs um seine  $z$ -Achse. Allerdings wechselte er während des Experiments die Richtung, so dass sein Pfad eine S-Kurve beschreibt und sich die Orientierung über die Zeit wieder ausgleicht. Da bei der Seitwärtsbewegung das linke und das rechte Bein auf derselben Achse bewegt werden, entstehen bei dieser Fortbewegung keine Drehmomente um die Hochachse des Roboters. Somit können die Drehungen des Roboters nicht analog zu Experiment 6.1.2.A erklärt werden. Eine plausible Erklärung lässt sich aber aus der Betrachtung der kinematischen Struktur des Roboters finden. Es wird bei der Fortbewegung immer versucht, den Fuß des Roboters parallel zum Boden aufzusetzen. Damit bei kleinen Abweichungen in der Bewegung keine Störung durch Verkanten oder eine schräge Bodenberührung des Fußes entsteht, wird der Fuß bereits kurz vor dem geplanten Aufsetzen parallel zum Boden gehalten und die Position des Fußes am Ende der Trajektorie nur noch in  $z$ -Richtung bewegt. Anschaulich gesagt wird versucht den Fuß *von oben* aufzusetzen. Doch eben dies ist bei einer reinen Seitwärtsbewegung nur eingeschränkt möglich, da der NAO nur ein *KneePitch*-, aber kein *KneeRoll*-Gelenk besitzt. Daher kann es bei einer translatorischen Bewegung auf der  $y$ -Achse durchaus vorkommen, dass der Fuß bei einer zu frühen Berührung des Bodens schräg aufgesetzt wird. Bei einer weiteren Ausführung der Bewegung kann dies zu einer Rotation des Roboters durch Zusammenziehen der Beine führen.

Zieht man den Versuchsaufbau genauer in Betracht, wird diese Erklärung noch plausibler. Der Roboter startet auf dem Anstoßpunkt des Feldes und bewegt sich in  $y$ -Richtung fort, also entlang der Mittellinie. Diese ist mit 5 cm breitem weißem Klebeband auf dem Feld markiert, das eine glattere Oberfläche als der Teppich hat. Die Reibungseigenschaften der beiden Bodenbeläge unterscheiden sich also merklich. Somit kann es durchaus vorkommen, dass beide Füße den Grund mit unterschiedlicher Haftreibung berühren und somit der Rotationseffekt beim Zusammenziehen der Füße verstärkt wird. Mit dieser Theorie lassen sich dann auch die Ergebnisse der Abbildung

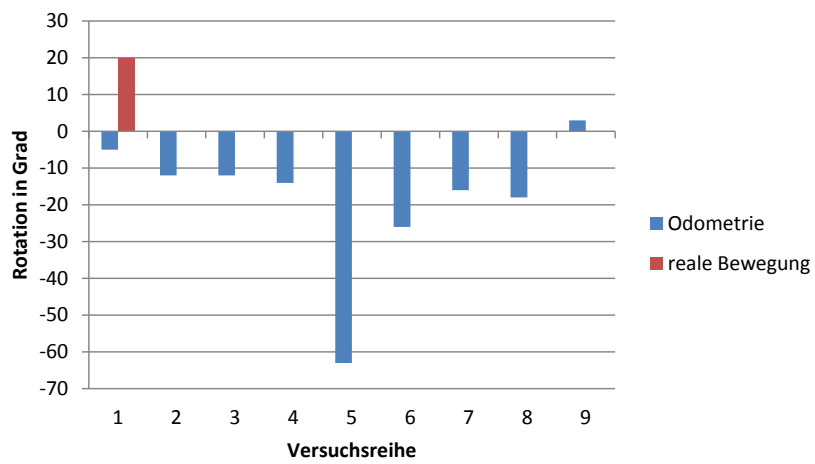
6.1.6(a) plausibel darlegen. Die Odometrie misst für jeden Lauf eine gleichbleibende Abweichung in negativer  $y$ -Richtung. Dies deutet darauf hin, dass der reale Roboter die von der WE geplante Winkel in der Zeit nicht erreichen kann und somit eine systematische Abweichung in der Ansteuerung entsteht. Sonst könnte dies über die Sensoren nicht wahrgenommen werden. Die reale Bewegung des Roboters resultiert aber in jedem Lauf in einer positiven Abweichung in  $x$ -Richtung. Der systematische Drift in negativer  $x$ -Richtung muss also durch eine Rotation gegen den Uhrzeigersinn überlagert werden, so dass sich über die Zeit ein Fehler in positiver Richtung aufsummiert.

**Experiment 6.1.2.C - Rotation in  $z$ -Richtung** Zur Untersuchung der Odometrie rotiert der NAO wiederholt, beginnend beim Anstoßpunkt, aus dem Stand auf der Stelle im Uhrzeigersinn. Die Ergebnisse dieses Experiments zeigt Abbildung 6.1.7. Erneut unterschätzt die Odometriemessung die ausgeführte Bewegung in fast jedem der Versuchsreihen, liegt aber näher an der Realität. Die Auswertung der  $x$ -Translation zeigt ebenfalls teilweise eine erhebliche Bewegung des Roboters, die eigentlich nicht hätte stattfinden sollen. Da auch bei diesem Experiment die Messung der Odometrie in vielen Fällen eine Abweichung in die entgegengesetzte Richtung aufweist, lässt dies erneut den Schluss zu, dass während der Bewegung Rotationen stattgefunden haben, die sich über die Zeit ausgeglichen haben, aber zu einem akkumulierten Fehler in der Translation führten. Die Auswertung in 6.1.7(b) zeichnet ein ähnliches Bild und stützt diese These.

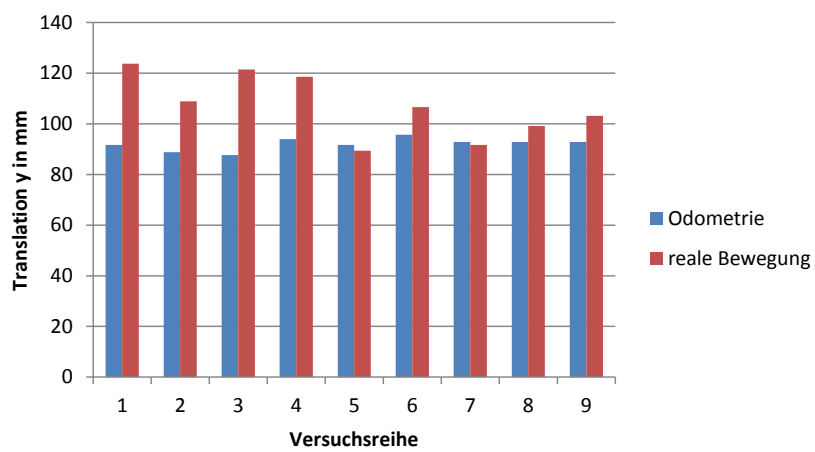
(a) Translation in  $x$ -Richtung.(b) Translation in  $y$ -Richtung.(c) Rotation um  $z$ -Achse.Abbildung 6.1.6: Experiment 6.1.2.B - Lauf in  $y$ -Richtung mit  $v = 50$  mm/s.



(a) Translation in  $x$ -Richtung.



(b) Translation in  $y$ -Richtung.



(c) Rotation um  $z$ -Achse.

Abbildung 6.1.7: Experiment 6.1.2.C - Drehung um  $z$ -Achse mit  $v = 0.4$  rad/s.

## 6.2 Messung odometrischer Fehler mittels optischer Sensoren

Die Ergebnisse des vorangegangenen Kapitels legen nahe, dass die Odometriemessung über die Gelenkwinkelsensoren des NAO einen vielversprechenden Ansatz darstellt. In der Praxis zeigt sich jedoch, dass bei der Messung Fehler auftreten, die über die Stellung der Gelenkwinkelsensoren nicht erfasst werden können. Einige Gründe, wie beispielsweise Fehler in der Messung selbst oder das Phänomen der flexiblen Füße, wurden bereits eingehend diskutiert, doch dies sind nur einige Möglichkeiten für Abweichungen. Die Fehler haben unterschiedlicher Ursachen und müssen daher getrennt betrachtet werden.

**Kalibrierung:** Einige systematische Fehler können durch eine korrekte Kalibrierung der Sensoren ausgeglichen werden. Hierzu gehören beispielsweise die Fehler, die aus einer abweichenden Rotation beim Einbau der Gelenkwinkelencoder resultieren. Wenn diese bekannt sind, können sie bei der Auswertung der Informationen der Gelenkwinkel berücksichtigt und damit ausgeglichen werden. Dies erfordert eine exakte Kalibrierung, die aufgrund fehlender Referenzwerte mit dem NAO nur schwer zu erreichen ist.

**Modellierung:** Einige Fehler, die nicht mit den Sensormessungen zusammenhängen, entstehen durch eine Abweichung des Modells von der Realität. Ein Modell kann nie die Realität exakt abbilden, doch kann es durch Verbesserung der Realität weiter angenähert werden. So kann beispielsweise die Bewegung der Fußplatte des Standbeines abgebildet werden, indem eine Abschätzung über den resultierenden Winkelfehler gemacht wird. Das Modell kann dann um einen entsprechenden passiven Freiheitsgrad ergänzt und so die Schrittlänge entsprechend korrigiert werden. Ein exakter Ausgleich kann hierdurch in der Praxis nicht gelingen, da der Fehler von zu vielen dynamischen Einflüssen abhängt. Auch diese wären durch ein entsprechendes Modell berechenbar, doch könnte der NAO dessen Berechnung während der Laufzeit nicht in Echtzeit bewältigen. In der Praxis ist diese Modellanpassung vom Roboter abhängig, da die Federkonstante der Sohlenaufhängung vom Materialverschleiß beeinflusst wird. Weiterhin hat auch die Umgebung Einfluss, da beispielsweise ein Teppichboden die Schrägstellung der Sohle durch Einsinken weiter begünstigt. Dennoch kann durch eine Beachtung einer ungefähren Abweichung im Modell eine Reduktion des Einflusses erreicht werden.

**Erfassung durch Sensoren:** Einige Fehler sind nicht systematisch genug, um sie geeignet durch deterministische Modellierung zu erfassen und auszugleichen. Unter diese Klasse fallen beispielsweise die Beobachtungen, die in Experiment 6.1.2.A gemacht werden konnten. Die Rotation um den Standfuß ist zwar ein Vorgang, der physikalischen Gesetzmäßigkeiten folgt und somit deterministischer Natur ist. Jedoch sind die dynamischen Zusammenhänge dabei so komplex, dass diese sich nicht exakt genug modellieren lassen, um eine solche Vorhersage zu machen. Man sagt, das System verhalte sich *deterministisch chaotisch* und hat daher scheinbar stochastischen Charakter. Somit kann der Fehler nicht sinnvoll aus der Bewegung abgeleitet werden. Will man ihn korrigierend berücksichtigen, braucht der NAO Sensoren, die diese Abweichung messen können.

Die Beobachtungen der durchgeführten Experimente legen nahe, dass der Einfluss der Störungen der letzten Klasse größer ist, als die Fehler, die sich durch Kalibrierung oder Modellierung ausgleichen lassen. Diese Störungen haben unterschiedliche Ursachen und Auswirkungen auf die geplante Bewegung, doch ist ihnen gemein, dass sie nicht zu einer Bewegung der Gelenke führen und demnach nicht mit den Gelenkwinkelencodern gemessen werden können. Zur Odometrikorrektur wird also ein anderer Sensor benötigt.

Die bisherigen theoretischen Betrachtungen der Fortbewegung gingen davon aus, dass das *Sliding Constraint* nicht verletzt wird, der Standfuß also an seiner Position relativ zum Boden verharrt. Solange die Haftreibung  $F_{H_{Boden}}$  zwischen Boden und Sohle des Roboters groß genug ist, um die aus der Fortbewegung resultierenden Kräfte zu kompensieren, ist dies auch der Fall. Die Beobachtung der Experimente 6.1.2.A bis C legen jedoch nahe, dass der NAO diese Bedingung in der Praxis nicht immer einhält. Sobald die Kraft, die die Haftreibung  $F_{H_{Boden}}$  aufbringen kann, überschritten ist, gerät der Fuß des Roboters in eine Bewegung parallel zum Boden und es wirkt nur noch die Gleitreibung  $F_{G_{Boden}}$  auf den Roboterfuß. Da im Allgemeinen

$$F_{G_{Boden}} > F_{H_{Boden}}, \quad (6.2.1)$$

gilt, führt die Kraftdifferenz dazu, dass von einem auf den anderen Zeitpunkt die Kraft

$$F_a = F_{H_{Boden}} - F_{G_{Boden}}, \quad (6.2.2)$$

auf den Roboter wirkt. Dies beschleunigt den NAO ruckartig und verursacht eine Rutschbewegung. Abhängig von der Ursache der ursprünglich auf den Roboter wirkenden Kraft, wird der Roboter von der Gleitreibung gebremst und kommt nach einer ungewollten Gleitbewegung wieder zur Ruhe.

Um die weiteren Betrachtungen zu motivieren, ist zunächst ein Einschub nötig, um die Auswirkungen unterschiedlicher Odometriefehler auf die Fortbewegung eines humanoiden Roboters zu klären. Betrachtet man deren Folge über die Zeit, wird klar, dass hierbei zwischen *translatorischen* und *rotatorischen* Fehlern zu unterscheiden ist. Ein einmaliger translatorischer Fehler, dem der Roboter beispielsweise durch eine Fehleinschätzung des Fußpunktes unterliegt, führt zu einer konstanten Abweichung der Roboterposition über die Zeit. Die Anzahl der folgenden Schritte ist dabei irrelevant, solange diese keine weiteren Fehler verursachen. Ein einmalig gemachter rotatorischer Fehler hingegen führt zwar ebenfalls zu einer konstanten Abweichung, jedoch in der Orientierung des Roboters. Somit hat jeder der Folgeschritte eine größer werdende Abweichung der Position zur Folge. Ein einmaliger rotatorischer Fehler integriert sich also über die Zeit zu einem translatorischen Fehler auf. Veranschaulicht wird dies in Abbildung 6.2.8(a) und 6.2.8(b). Dies führt zu den aus der Robotik bekannten *Unsicherheitsellipsen* der Position über die Zeit, wenn von einer reinen Lokalisierung durch Odometrie ausgegangen wird - vergleiche beispielsweise [16].

Das Rutschen des Roboters kann unterschiedliche Ursachen haben und muss nicht aus der Bewegung des Roboters selbst resultieren - auch externe Quellen sind denkbar. Als Grundlage der im nächsten Kapitel folgenden Untersuchung geeigneter Sensoren werden nun abschließend die verschiedenen Ursachen dargelegt und diskutiert.



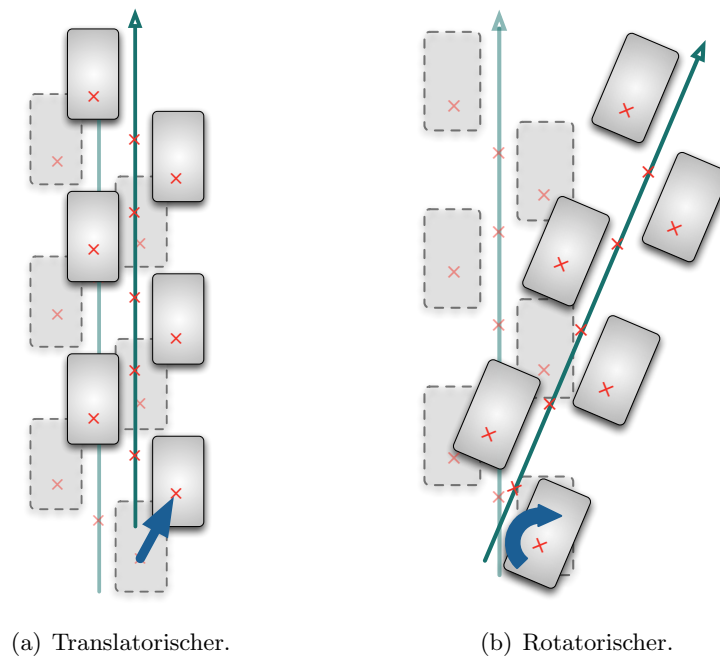


Abbildung 6.2.8: Odometriefehler über die Zeit.

**Rotation durch Schwungbein** Dieses Phänomen ist für den Roboter NAO typisch und soll daher in seiner Entstehung untersucht werden. Zur Verdeutlichung wird von einer ausschließlichen Bewegung in  $x$ -Richtung ausgegangen. Beim Wechsel von der *Double-Support*- in die nächste *Single-Support*-Phase wird zunächst das Schwungbein angehoben und dann in  $x$ -Richtung beschleunigt. Dies erzeugt eine Schwingkraft  $\mathbf{F}_S$  in Laufrichtung. Da der Roboter aus starren Festkörpern besteht wirkt diese Kraft über den Körper als Hebel auf den Standfuß. Das Resultat ist ein Drehmoment  $M_z$  in Abhängigkeit vom Fußabstand  $\mathbf{r}$ :

$$\mathbf{M}_z = \mathbf{r} \times \mathbf{F}_S. \quad (6.2.3)$$

Bei niedrigen Beschleunigungen wird die durch das Drehmoment auf den Boden wirkende Scherkraft  $F_z$  durch die Haftreibung  $F_H$  kompensiert. Da bei der NDD-WE höhere Geschwindigkeiten durch eine größere Schrittweite erreicht werden, muss folglich das Schwungbein stärker beschleunigt werden, um in der vorgegebenen Zeit den gewünschten Fußpunkt zu erreichen. Dies führt dazu, dass durch die Beschleunigung auch  $F_S$  größer wird. Als Resultat reicht ab einer gewissen Laufgeschwindigkeit die Haftreibungskraft nicht mehr zur Kompensation von  $F_S$  aus und der Roboter beginnt, um den Standfuß zu rotieren. Ohne experimentelle Verifikation kann angenommen werden, dass es sich bei dem Drehpunkt um den *Center of Pressure* (CoP) handelt, da dieser das Zentrum der Druckverteilung darstellt. Am Ende der Schwungphase muss das Bein wieder abgebremst werden, wodurch eine Kraft in negativer  $x$ -Richtung erzeugt wird. Diese könnte eine ausgleichende Rotation in Gegenrichtung erzeugen, allerdings sind die Beschleunigungs- und Verzögerungsphasen des Beines in der Praxis nicht identisch in Länge und Kraft. Grund hierfür ist der Wunsch, den Fuß *sanfter* aufzusetzen, damit der Ruck auf den Roboter nicht so groß ist, wenn der Fuß vorzeitig den Boden

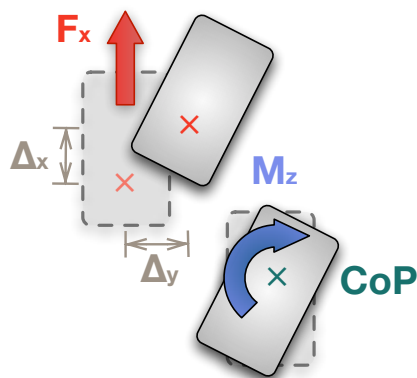


Abbildung 6.2.9: Drehmoment erzeugt durch Schwungfuß.

berührt. Aber auch wenn die Verzögerung die gleiche Kraft erzeugen würde, würden sich die beiden Rotationen nicht ausgleichen, da nach Ende der Verzögerungsphase, das Schwungbein den Boden berührt und dadurch die rotatorische Bewegung abbremst. Aus der Rotation resultiert auch eine Abweichung in  $x$ - und  $y$ -Richtung, wie in Abbildung 6.2.9 verdeutlicht wird.

In der Theorie führt der Roboter bei symmetrischer Ansteuerung in der nächsten *Single-Support*-Phase eine Rotation in entgegengesetzter Richtung aus. Somit würde sich der Fehler in der Rotation und in der  $y$ -Translation über zwei Schritte summiert aufheben. Es ist zu beachten, dass die Abweichung in  $x$ -Richtung sich in diesem Fall nicht ausgleicht und sich daher über die Zeit aufsummiert. Als Resultat legt der Roboter eine größere Strecke zurück, als geplant. Dies erklärt zum einen die größeren Abweichungen in  $x$ -Richtung bei Experiment 6.1.2.A, erklärt zum anderen aber auch, warum der Roboter seine Geschwindigkeit in der Regel unterschätzt, wie die Untersuchungen von Urbann et al. in [113] aufzeigen. In der Praxis ist die Stärke des Drehmoments von vielen dynamischen Faktoren abhängig, so dass die Rotation meist nicht gänzlich durch den nächsten Schritt kompensiert wird. Die Folge ist ein Fehler in der Rotation und  $y$ -Translation über die Zeit. Zum Verständnis dieses Phänomens sei hier das Beispiel der unterschiedlichen Bodenbeschaffenheit angeführt. Wenn beispielsweise der Standfuß des NAO beim RoboCup auf einer aufgeklebten Feldlinie steht, so ist der Haftreibungskoeffizient für diesen Schritt grundverschieden von dem des nächsten Schrittes auf Teppich. Aber bereits leichte Unterschiede in der Oberflächenbeschaffenheit des Untergrunds können zum gleichen Effekt führen, ebenso wie leichte dynamische Abweichungen in der Bewegung der Gliedmaßen.

Mit Wissen über den Effekt kann der Roboter diesen zwar nicht unterbinden, doch könnte er zumindest seine Trajektorie entsprechend anpassen oder ausgleichend auf diesen reagieren. So könnte beispielsweise einer ungewollten Rotation durch gezielte Anpassung der Armbewegung für den nächsten Schritt entgegen gewirkt werden. Durch die Komplexität der dynamischen Vorgänge erfordert deren Modellierung einen Detailgrad, der vom Roboter nicht zur Laufzeit berechnet werden kann. Folglich bleibt nur der Einsatz von Sensoren, um die realen Auswirkungen messen zu können.

**Bewegung durch vorzeitigen Bodenkontakt** Wenn der Roboter, beispielsweise durch Verschleiß der Gelenke, mit dem Schwungbein verfrüht den Boden berührt, kann er in der Regel nicht darauf reagieren<sup>4</sup>. Daher wird er für einige Taktzyklen die geplante Bewegung noch fortsetzen, was durch den ungewollten Kontakt mit dem Boden zu einer unplanmäßigen Gleitbewegung des Roboters führen kann. Eine Modellierung des Fehlers ist nicht möglich, da kein Wissen über den Zeitpunkt des Kontakts besteht. Wenn die ungewollte Bewegung hingegen mit einem Sensor gemessen wird, kann der Roboter entsprechend darauf reagieren und gegebenenfalls seine zukünftige Bewegung anpassen, um diese auszugleichen.

**Kollision mit Objekt** Kollidiert der Roboter während des Ganges mit einem festen Objekt, kann dies dazu führen, dass er fällt. Ist dies nicht der Fall, hat die Kollision zur Folge, dass der Roboter seine Bewegungen nicht wie geplant ausführt. Im Falle des NAO reicht das Drehmoment der Motoren in der Regel aus, um die geplante Bewegung im Gelenkwinkelraum auszuführen, auch wenn das Objekt die Trajektorie in Weltkoordinaten blockiert. Damit ist aber zwangsläufig das Resultat, dass der Roboter sich relativ zum Objekt bewegt - entweder in der das Objekt oder seine eigene Position verschiebt. Dies ist im Umfeld des RoboCups beispielsweise bei der Kollision des Roboters mit einem anderen NAO oder einem Torpfosten zu beobachten. Betrachtet man letztere als Beispiel für eine Frontalkollision, so läuft der Roboter auf der Stelle. Dies führt zu einer signifikanten Translation des Standfußes, da der Schwungfuß den Roboter vom Tor wegschiebt. Eine Messung dieser Bewegung wäre hilfreich, da so der Roboter zumindest innehalten könnte, um einen Sturz zu verhindern oder sich gezielt neu zu lokalisieren. Kollidiert der Roboter hingegen nur teilweise mit einem Objekt wird er sich in der Regel noch fortbewegen, allerdings anders als geplant. So führt beispielsweise die Berührung zweier NAOs im Vorwärtsgang mit den Armen dazu, dass diese sich um die eigene  $z$ -Achse drehen, bis die Berührung beendet ist. Ein solcher Rotationsfehler führt zu einem Fehler in der Lokalisierung, der erst über die Zeit wieder aufgelöst werden kann. Fehlentscheidungen des Roboters ließen sich also auch in diesem Fall durch Messung der Rotation vermeiden.

**Stoß durch Objekt** Im Gegensatz zum vorhergehenden Fall kann eine Kollision auch von einem externen Objekt ausgelöst werden. Ist die Kraft stark genug, um den NAO in Bewegung zu bringen, so führt sie in der Regel jedoch zu dessen Sturz. Dennoch könnte, je nach Richtung aus der die Kraft ausgeübt wurde, der Roboter zunächst noch über den Boden verschoben werden. Die Messung einer solchen unvorhergesehenen translatorischen Bewegung würde dem Roboter die Möglichkeit geben, Rückschlüsse auf die Ursprungsrichtung der Kraft zu ziehen. Einem Sturz könnte somit aktiv entgegenwirkt werden oder zumindest die Fallrichtung und Geschwindigkeit des Sturzes abgeschätzt werden, um diesen durch einen Ausfallschritt abzufangen.

---

<sup>4</sup>Eine Detektion mit Hilfe der FSR-Sensoren wäre denkbar. Allerdings muss diese zuverlässig sein und daher einer gewissen Filterung unterliegen. Als Folge würde die Reaktion des Roboters erst verspätet einsetzen können.

Alle diskutierten Bewegungsursachen resultieren aus einer Kraft, die die Haftreibung übersteigt und führen zu einer Gleitbewegung des Roboters. Dabei bleibt in der Regel die Fußsohle des Roboters während der Translation im Kontakt mit dem Boden, solange nicht sehr starke Störkräfte auf den Roboter dazu führen, dass dieser den Kontakt mit dem Boden gänzlich verliert. Da bisher immer als Voraussetzung angenommen wurde, dass sich der Roboter auf planem Boden fortbewegt, stellt diese Bewegung also die Verschiebung zweier ebenen Flächen gegeneinander dar. Eben solche Bewegungen zu verfolgen ist die Aufgabe optischer Sensoren, wie sie in Computermäusen Anwendung finden. Aus diesem Grund werden solche im weiteren Verlauf des Kapitels auf ihre Eignung untersucht, den odometrischen Fehler zu messen, wenn sie unter der Sohle eines humanoiden Roboters angebracht werden. Grundlegende theoretische Überlegungen und Experimente hierzu wurden von Kerner und Hegele bereits veröffentlicht [21], werden aber an dieser Stelle erneut aufgegriffen, um eingehender diskutiert und um weiterführende Untersuchungen ergänzt zu werden.

Es bleibt zu erwähnen, dass die Idee optische Sensoren zur Messung der Odometrie mobiler Roboter einzusetzen nicht grundsätzlich neu ist. So sind diese beispielsweise schon erfolgreich bei rollenden Robotern der *Middle-Size-League* (MSL) des RoboCup eingesetzt worden [10]. Allerdings konnte bislang noch keine Untersuchung eine erfolgreiche Übertragung der Idee auf eine Anwendung mit realem Praxisbezug nachweisen. Dies liegt wohl darin begründet, dass mit dieser Methode die Odometrie und nicht der *Fehler* der Odometrie gemessen werden soll. Bei einem rollenden Roboter sollten nach Möglichkeit nur die Räder Kontakt zum Boden haben. Da diese sich drehen, sind sie für die Anbringung solcher Sensoren ungeeignet. Da aus Gründen der Manövrierfähigkeit ein Fahrzeug, und folglich auch ein rollender Roboter, ein Mindestmaß an Bodenfreiheit besitzt, muss der optische Sensor eine gewisse Höhe über dem Boden haben. Für diese Anwendung wurden Maussensoren jedoch nicht konzipiert. Da Roboter für den Außeneinsatz zudem meistens mit einer Federung ausgestattet sind, variiert die Bodenhöhe mit der Zeit. Dies führt in Kombination mit hohen Geschwindigkeiten und strukturierten Untergründen dazu, dass diese Art der Odometriemessung in der Praxis der klassischen Methode der Messung über die Räder unterlegen ist. Eine Anwendung auf humanoide Roboter wurde jedoch bislang nicht eingehend untersucht. Da der Rutschfehler und nicht die eigentliche Bewegung gemessen werden soll, sind nur kürzere Strecken zu messen. Dies lässt erwarten, dass die Fehlerrobustheit steigt. Zudem ist durch die Art der humanoiden Fortbewegung gewollt, dass der Standfuß beim Gehen direkten Kontakt mit dem Boden hat. Die Voraussetzungen für den Einsatz optischer Sensoren zur Odometriemessung sind also bei laufenden Robotern besser, als bei rollenden.

### 6.2.1 Einführung in die Grundlagen optischer Sensoren

Die Beobachtung der Entstehung odometrischer Fehler legt nahe, dass zumindest die durch Rutschen entstandenen Abweichungen mittels optischer Sensoren gemessen werden können. Im Folgenden wird daher der Einsatz von optischen Maussensoren untersucht. Als Einführung in die Thematik kann im Rahmen dieser Arbeit keine vollständige Erörterung der Funktionsweise dieser Sensoren gegeben werden. Um für die weiteren Betrachtungen relevante Schlussfolgerungen dennoch nachvollziehen zu können, ist es jedoch unumgänglich in einem rudimentären Exkurs zumindest einige relevante Fakten darzulegen. Zur Vertiefung der Thematik ist beispielsweise die Dissertation von Horn [51] geeignet, die sich konkret mit dieser Klasse von Sensoren befasst.

Seit dem ersten kommerziellen Erfolg einer optischen Computermaus im Jahre 1999 ist die optische Bewegungsmessung eine gebräuchliche Methode zur Wahrnehmung der Bewegung einer Maus relativ zum Untergrund. Ihr Vorteil liegt in erster Linie in der Präzision des Sensors und der Tatsache, dass für die Messung kein direkter Kontakt zum Boden bestehen muss. Daher haben optische Mäuse die mechanischen mittlerweile gänzlich vom Markt verdrängt. Es gibt verschiedene Techniken, mit denen die Messung der zurückgelegten Strecke bewerkstelligt wird. Doch allen ist gemein, dass sie auf einem *CMOS*<sup>5</sup>-Sensor beruhen, der auf die eine oder andere Art und Weise ein monochromes Abbild der Oberfläche erzeugt. Diese CMOS-Chips haben in der Regel nur eine geringe Auflösung von rund  $15 \times 15$  Pixeln, um preisgünstig zu sein. Um mit der Kamera Bilder aufzunehmen, wird der Untergrund mit einer Lichtquelle angestrahlt, deren Reflexion von dem Sensor erfasst wird. Die Positionen markanter Bildbereiche, sogenannter *Features*, werden mit Hilfe eines geeigneten *Feature-Detection*-Algorithmus in diesen Bildern verfolgt, um so Rückschlüsse auf die Bewegung der Maus relativ zum Untergrund zu ziehen. Zur Reduktion des Hardwareaufwand für dieses *Tracking*, wird von einer rein translatorischen Bewegung der Maus ausgegangen und keine Informationen über eine Rotation des Sensors geliefert. Da mit der Computermaus der Mauszeiger nur in der Ebene des Monitors verschoben und nicht rotiert wird, ist diese Vereinfachung durchaus zweckdienlich. Wie jede optische Kamera verwendet auch der Maussensor eine Linse, um das Licht zu fokussieren. Da diese in der Regel unbeweglich ist<sup>6</sup>, resultiert eine feste Fokusebene und somit ein bestimmter Abstand zum Boden, der vom Sensor eingehalten werden muss. Die Empfindlichkeit einer Maus wird durch deren Auflösung in *dots-per-inch*, abgekürzt als DPI, angegeben. Durch eine hohe Auflösung kombiniert mit einer hohen Datenrate von bis zu 12 Megapixel/s arbeiten moderne Hightech-Mäuse, laut Datenblatt, zuverlässig bei Beschleunigungen von bis zu 30 g und Geschwindigkeiten bis zu 4 m/s. Die Zuverlässigkeit einer Maus hängt aber nicht nur von ihrer Auflösung ab, sondern sehr von der Qualität der *Features*, die diese im Bild verfolgen kann. Somit ergibt sich eine Abhängigkeit von der Beschaffenheit des Untergrundes, die die Hersteller versuchen durch Verbesserung der Beleuchtung zu verringern. Dadurch sind in den letzten Jahren unterschiedliche Beleuchtungstechniken entwickelt und angewendet worden, über die ein kurzer Überblick gegeben werden soll.

---

<sup>5</sup>Complementary Metal Oxide Semiconductor

<sup>6</sup>Einige Mäuse des Herstellers *Roccat*, können die Fokusebene des Sensors verschieben. Diese wird aber nicht automatisch angepasst, sondern wird durch eine Treibereinstellung fest eingestellt.

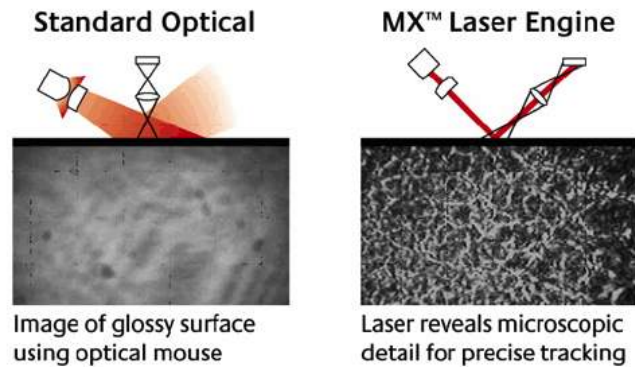


Abbildung 6.2.10: Vergleich einer LED- und Laser-Maus, entnommen aus [71].

**LED** Die erste Maus, die eine LED zur Beleuchtung einsetzte und keinen speziellen Untergrund benötigte<sup>7</sup>, verwendete die von Microsoft entwickelte *IntelliEye*<sup>TM</sup>-Technik. Eine Fortentwicklung dieser Technik durch unterschiedliche Hersteller findet sich in einer Vielzahl moderner Computermäuse und zählt aufgrund der kostengünstigsten Fertigung derzeit zu den verbreitetsten Techniken. Einzelheiten in der technischen Umsetzung können sich je nach Hersteller unterscheiden, doch alle verwenden als Lichtquelle eine LED, die neben dem CMOS-Sensor angebracht ist und den Untergrund beleuchtet. Die Reflexion des sichtbaren Lichts wird von einer Linse gebündelt und durch den Sensor ausgewertet, vergleiche Abbildung 6.2.10<sup>8</sup> links. Die ersten Mäuse verwendeten monochromes rotes Licht, um einen guten Kontrast zu erzeugen, doch je nach Fertigung kommen derzeit auch andere Farben zum Einsatz. Die Technik hat Schwierigkeiten auf Oberflächen, die wenig Struktur im sichtbaren Bereich der Auflösung enthält.

**Laser** Eine Weiterentwicklung der LED-Technik wurde von Logitech im Jahre 2004 mit der Computermouse *MX 1000 Laser* erstmals für den Massenmarkt vorgestellt. Als Lichtquelle kommt eine Infrarot-Laser-Diode zum Einsatz, die den Untergrund beleuchtet. Bei der Reflexion des Laserlichts entsteht ein Kohärenz-Muster, das auch auf unformen Oberflächen *Features* bildet, die besser zu verfolgen sind, als das rein visuelle Abbild einer LED. Diese Tatsache verdeutlicht Abbildung 6.2.10, die einen Vergleich zwischen dem Bild einer LED und einer MX-Laser Maus zeigt. Als Resultat ermöglicht der Einsatz von Laser-Dioden in Mäusen nicht nur eine höhere DPI, sondern auch eine Messung bei höheren Geschwindigkeiten und Beschleunigungen. Daher kommt diese Technologie vermehrt in Mäusen für den *Gaming*-Bereich zum Einsatz, schlägt sich aber aufgrund der Hardware in einem höheren Preis und Stromverbrauch nieder. Mäuse mit einem solchen Sensor sind auf nahezu jedem Untergrund einzusetzen - lediglich Glas und Oberflächen, die strukturbedingt inhärente Höhenunterschiede aufweisen, stellen ein Problem dar.

<sup>7</sup>Die Idee optische Sensoren in Mäusen zu verwenden kam bereits Ende der 80er Jahre auf. Die technische Umsetzung erforderte jedoch immer eine spezielle Mausunterlage.

<sup>8</sup>Copyright ©2009: Logitech, Inc.

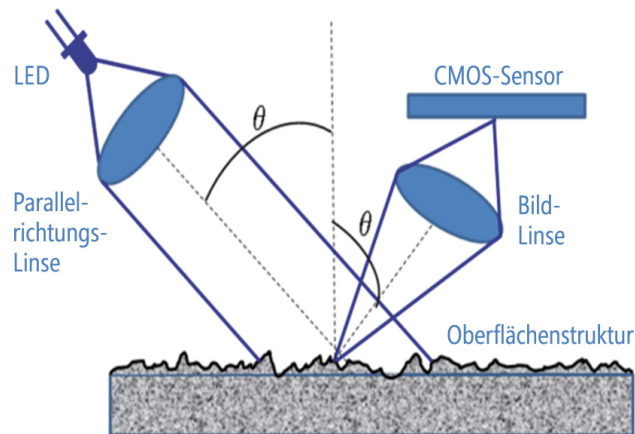


Abbildung 6.2.11: BlueTrack Schematik entnommen aus [77].

**BlueTrack-Technology** Von Microsoft entwickelt kommt die *BlueTrack-Technology*, im Folgenden mit *BlueTrack* abgekürzt, derzeit exklusiv in Mäusen dieses Herstellers zum Einsatz. Wie der Name nahelegt, kommt blaues Licht zur Erhellung des Untergrundes zum Einsatz. Hierzu wird eine LED verwendet, die inkohärentes Licht ausstrahlt. Microsoft hat jedoch einige Änderungen an dem Aufbau einer herkömmlichen LED-Maus vorgenommen, so dass es sinnvoll ist, auf diese genauer einzugehen. Wenn man Microsofts schematische Übersicht, zu finden in Abbildung 6.2.11<sup>9</sup>, genauer betrachtet, fällt zunächst auf, dass das Licht der LED durch eine Parallelrichtungs-Linse fokussiert wird. Dies hat den Vorteil im Vergleich zur herkömmlichen LED-Technologie einen schärferen Kontrast zu erzeugen, im Gegensatz zur Laser-Technologie aber dennoch einen größeren Bereich auszuleuchten. In Kombination mit einer *specular optic* sorgt dies dafür, dass sowohl Bewegungen auf Oberflächen mit einer großen Tiefenstruktur, wie Teppich oder Granitplatten, als auch glänzende Oberflächen erkannt werden. Auf Spiegeln oder Glas funktionieren auch diese Mäuse nicht.

**Darkfield Laser Tracking™** Von Logitech entwickelt, kommt die *Darkfield Laser Tracking™* Technologie, im Folgenden kurz *Darkfield*, derzeit exklusiv in Mäusen dieses Herstellers zum Einsatz. Es handelt sich dabei um eine Übertragung einer Technik der Dunkelfeldmikroskopie, bei der das Licht am Objektiv vorbei geleitet wird. Licht gelangt nur durch Streuung am Objekt in das Objektiv und so entstehen auch bei durchsichtigen Präparaten weiße Strukturbilder auf schwarzem Untergrund, siehe Abbildung 6.2.12<sup>10</sup>(c) In Anlehnung an diese Technologie verwendet Logitech zwei Laser, um auch auf transparenten Flächen ein Bild der Oberflächenstruktur erzeugen zu können. Abbildung 6.2.12 verdeutlicht den Vorgang. Aus dieser Schematik geht aber nicht hervor, dass die zwei unterschiedlichen Laser nur exklusiv verwendet werden:

„To achieve dark field illumination, Logitech mice with Darkfield Laser Tracking use two lasers to more effectively collect microscopic details of the tracking surface. When the mouse is used on a regular surface – such

<sup>9</sup>Copyright ©2012: Microsoft.

<sup>10</sup>Copyright ©2009: Logitech, Inc.



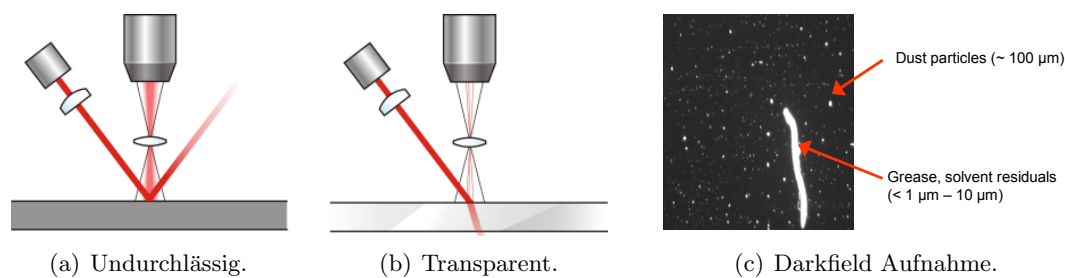


Abbildung 6.2.12: Darkfield Laser Tracking<sup>TM</sup>, entnommen aus [71].

as a formica table or a piece of paper – the texture of the surface provides plenty of detail for the lasers to track and requires only one of the lasers to be used. But for high-gloss surfaces that don't have enough details, such as a glass tabletop, the mouse sensor views the surface itself as black and, instead, tracks the dust and other residuals that appear on the surface.“ [71]

Dank dieser Technologie arbeiten die Mäuse demnach auch auf Glas. Logitech gibt allerdings darüber hinaus an, dass diese Mäuse auf anderen Untergründen ebenfalls bessere Ergebnisse liefern. Aus diesem Grund tragen derzeit alle Darkfield-Mäuse den Produktnamen *Anywhere*.

Der Anschluss von Mäusen erfolgt kabelgebunden oder per Funkübertragung. Heutzutage kommt als Anschluss praktisch immer USB zum Einsatz, der alte PS/2-Standard findet keine Anwendung mehr. Eine Ausnahme findet sich lediglich bei Mäusen, die das *BlueTooth*-Funkprotokoll unterstützen und daher mit einem integrierten Empfänger im Computer direkt kommunizieren können. *BlueTooth* wird aber nur noch von einigen wenigen Mäusen, die für den mobilen Einsatz gedacht sind, verwendet und so kommuniziert auch die Mehrzahl der Funk-Mäuse über einen proprietären USB-Empfänger mit dem Computer. Die Übertragung mittels USB erlaubt es, die Informationen der Mäuse im Intervall von 1 ms auszuwerten. Diese Frequenz wird heutzutage auch von einigen USB-Funk-Empfängern unterstützt, so dass praktisch kein Latenzunterschied mehr zwischen Funk- und Kabel-Anbindung bestehen muss.

Abschließend folgt eine kurze Überlegung die Messgüte betreffend, da diese bei den Experimente wieder aufgegriffen wird. Der CMOS-Sensor einer Maus hat in der Regel eine quadratische Form und besitzt nur eine recht geringe Bildauflösung. Da von der Maus nur *Features* verfolgt werden können, die sich in dem neuen Bild wiederfinden, ist zu bedenken, dass die Überlappung der Bildausschnitte einen Einfluss auf die Qualität der Messung hat. Daher macht es theoretisch einen Unterschied, ob eine Translation in reiner *x*- bzw. *y*-Richtung oder als Überlagerung beider stattfindet. Eine grafische Darstellung ist in Abbildung 6.2.13 zu finden und verdeutlicht, dass der Bildausschnitt zur *Feature*-Verfolgung bei Überlagerung der Achsen signifikant kleiner ist. Es ist jedoch zu erwarten, dass der Einfluss dieses Phänomens sich in der Praxis mit höheren Abtastfrequenzen verringert, da sich bei gleicher Geschwindigkeit die Fensterverschiebung zwischen zwei Messungen entsprechend verkleinert.



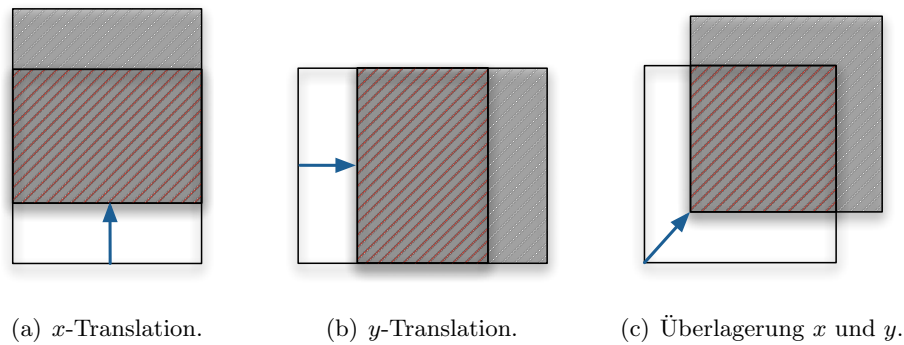


Abbildung 6.2.13: Überlagerung des Bildfeldes bei Bewegung.

### 6.2.2 Messung rotatorischer und translatorischer Bewegung mittels optischer Sensoren

Maussensoren sind nicht in der Lage rotatorische Bewegungen zu messen. Da sich aber in der bisherigen Betrachtung der Odometriefehler gezeigt hat, dass eben diese den größten Einfluss haben, sollte ein entsprechender Sensor diese messen können. Durch einen Punkt ist lediglich die Position einer Fläche im zweidimensionalen Raum<sup>11</sup> eindeutig bestimmt. Erst durch Hinzunahme eines zweiten Punktes ist auch die Orientierung gegeben. Somit ist es aus zwei bekannten Positionen des Fußes ebenfalls möglich, die Rotation der Bewegung zu berechnen. Zwei Sensoren sind also zur Messung des Odometriefehlers ausreichend. Es werden daher nun die mathematischen Grundlagen für einen entsprechenden Algorithmus zur Berechnung hergeleitet.

Der Sensor einer optischen Maus misst die Veränderung der Bewegung in  $x$ - und  $y$ -Richtung, indem er inkrementell *counts*, also die Verschiebung von Bildpunkten, aufaddiert. Abbildung 6.2.14 zeigt exemplarisch die Bewegung eines Fußes in Form einer  $x$ -Translation von 100 mm und einer Rotation um  $20^\circ$  im Uhrzeigersinn.  $\mathbf{S}_0$  und  $\mathbf{S}_1$  seien die Ausgangsvektoren der Position zweier optischer Sensoren im FKS des Roboters. Wenn von einer starren Verbindung der Sensoren zur Fußsohle ausgegangen wird, ist deren Position relativ zum Ursprung  $0$  des FKS bekannt. Diese sei bestimmt durch die Vektoren  $\mathbf{u}$  und  $\mathbf{v}$ . Die  $z$ -Komponente wird während der Bewegung als konstant angenommen, da davon ausgegangen wird, dass der Fuß flach auf dem Boden aufliegt.

Durch die Position von  $\mathbf{S}_0$  und  $\mathbf{S}_1$  ist der Vektor  $\mathbf{c} = \mathbf{S}_0 - \mathbf{S}_1$  definiert. Die Messung der Sensoren liefert  $\Delta_{xy}(\mathbf{S}_0)$  und  $\Delta_{xy}(\mathbf{S}_1)$  aus denen die Positionen  $\mathbf{S}'_0$  und  $\mathbf{S}'_1$  nach der Bewegung durch Addition ermittelt werden können. Diese Update enthält sowohl den rotatorischen, als auch translatorischen Anteil. Durch  $\mathbf{S}'_0$  und  $\mathbf{S}'_1$  kann nun auf den Vektor  $\mathbf{d} = \mathbf{S}'_0 - \mathbf{S}'_1$  geschlossen werden. Der Winkel zwischen diesem und dem Ursprungsvektor  $\mathbf{c}$  bestimmt den Rotationswinkel  $\Theta$  zum alten FKS:

$$\Theta = \arccos \left( \frac{\mathbf{c} \cdot \mathbf{d}}{|\mathbf{c}| |\mathbf{d}|} \right). \quad (6.2.4)$$

<sup>11</sup>Dies reicht zur Bestimmung der Pose im Kontext dieser Arbeit aus, siehe Kapitel 2.1



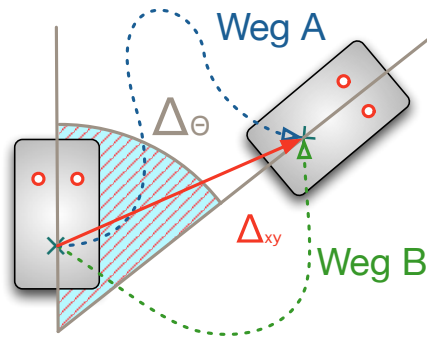


Abbildung 6.2.15: Unterschiedliche Gleitwege.

Die Translation des Fußes  $\mathbf{T}$  kann abschließend durch Subtraktion berechnet werden:

$$\mathbf{T} = \mathbf{O}' - \mathbf{O}. \quad (6.2.10)$$

Durch Kombination der Translation  $\mathbf{T}$  mit anschließender Rotation um  $\Theta$  ergibt sich abschließend die Transformation der beiden Koordinatensysteme.

Bei der Betrachtung der Rechnung fällt auf, dass der Fehler in der Rotation  $\Theta$  sich über die Länge der Vektoren  $\mathbf{v}$  und  $\mathbf{u}$  als Fehler auf die Translation fortpflanzt. Die Rechnung hätte auch umgekehrt durchgeführt werden können, indem zunächst die Translation berechnet wird. In diesem Fall hätte sich der Fehler auf die Rotation fortgepflanzt. Da aber die Translation einen geringeren Einfluss auf den Odometriefehler hat, wird bevorzugt, den Fehler in der Rotation so gering wie möglich zu halten.

Weiterhin fällt auf, dass bei der theoretischen Betrachtung das Zentrum der Rotation keine Rolle spielt. In der Tat ist dies irrelevant für die Messung der Orientierung. Ebenso ist es irrelevant, auf welchem Wege sich die Sensoren von  $\mathbf{S}_0$  nach  $\mathbf{S}'_0$ , bzw.  $\mathbf{S}_1$  nach  $\mathbf{S}'_1$ , fortbewegt haben. Da die Updates inkrementell erfolgen heben sich die Bewegungen theoretisch über die Zeit auf und nur die Endposition der Messung ist relevant. Dies ist in Abbildung 6.2.15 für zwei Wege A und B exemplarisch verdeutlicht. In der Praxis hat die Länge des Weges dennoch einen Einfluss auf die Messung, da ein längerer Weg bedeutet, dass sich Messfehler der *counts* aufsummieren können. Findet die Bewegung zudem in dergleichen Zeit statt, so ist eine höhere Geschwindigkeit notwendig, wodurch sich die Qualität der Bewegungsschätzung verringern kann.

### 6.2.3 Bewegungsmessung am NAO

Bisherige Betrachtungen waren rein theoretischer Natur. Um zu zeigen, dass diese Art der Messung auch in der Praxis einen Vorteil bietet, müssen Experimente am realen Roboter durchgeführt werden. Die folgenden Experimente bis einschließlich Kapitel 6.2.4 basieren auf Daten, die im Rahmen seiner Diplomarbeit [48] von Hegele aufgenommen worden sind. Zudem sind Teile der Auswertung bereits in [21] veröffentlicht. Der NAO ist nicht mit optischen Sensoren unter den Füßen ausgestattet. Um dennoch den praktische Nutzen dieser Klasse von Sensoren und ihren möglichen Einfluss auf das Design zukünftiger Roboter untersuchen zu können, wird an dieser Stelle eine Ausnahme von der Grundidee der Übertragbarkeit der Ergebnisse gemacht - der NAO wird um entsprechende Sensoren erweitert. Da die Platinen von modernen mobilen Mäusen sehr klein in ihren Abmessungen sind, wäre im Fuß des NAO genug Platz, diese zu integrieren. Der I2C-Bus ist bereits in den Fuß durchgeleitet. Es wäre also ohne weiteres möglich die Sensordaten direkt über den Bus an den DCM zu übermitteln und die Sensoren über die interne Stromversorgung zu speisen. Jedoch würde dies voraussetzen, dass die Buskommunikation offen ist. Da diese aber nicht dokumentiert ist und zudem bauliche Änderungen dieser Art nur von Aldebaran mit dem nötigen Wissen um die Konstruktion durchgeführt werden können, muss auf eine Integration der Sensoren verzichtet werden. Es bleibt aber festzuhalten, dass diese technisch möglich wäre.

Es muss demnach ein anderer Weg für die Experimente gewählt werden und so werden die Sensoren extern an den Roboter montiert. Hierzu wurde eine Art *Schuh* entwickelt, der es erlaubt die Sensoren in einer Ebene mit den Sohlen des Roboters anzubringen. Es wurde darauf geachtet, für die Fertigung ein Material zu verwenden, das eine ähnliche Oberflächenbeschaffenheit und Reibungsdynamik besitzt, wie die Sohle des NAO. Ein Foto ist in Abbildung 6.2.16(a) zu sehen. Da dies eine Modifikation des Roboters darstellt, die zu einer deutlichen Veränderung der Dynamik führt, handelt es sich nicht um einen Vorschlag für den praxisrelevanten Einsatz im RoboCup. Besonders kritisch ist die Tatsache, dass der Schuh die Bodenplatte des Fußes feststellt und folglich die FSR-Sensoren außer Funktion setzt. Vielmehr sind die Experimente als *Proof of Concept* zu verstehen, mit denen nachgewiesen werden soll, ob es sinnvoll ist, solche Sensoren in Zukunft beim Entwurf neuer Roboter in Betracht zu ziehen.

Durch die Designwahl vergrößern die Schuhe unweigerlich die Auflagefläche des Roboters. Um diesen Einfluss so gering wie möglich zu halten, wurden die Platinen der Sensoren am Institut für Roboterforschung von Hegele [48] eigens entworfen. Da aber keine Expertise in der Entwicklung von optischen Sensoren besteht, wird als Basis ein *Sensor-Kit* verwendet, wie es für die kommerzielle Entwicklung und Fertigung von Computermäusen angeboten wird. Der führende Hersteller solcher Sensoren ist die Firma *Avago Technologies*, die unterschiedliche Modelle mit LED- oder Laser-Technik im Angebot hat. Es ist anzumerken, dass zu dem Zeitpunkt, zu dem diese Untersuchungen durchgeführt wurden, zwar bereits Mäuse mit *BlueTrack* und *Darkfield* Technologie angekündigt, aber noch nicht kommerziell erhältlich waren. Daher sei bereits vorweggenommen, dass in Kapitel 6.2.5 noch ein nachträglicher Vergleich dieser Sensoren folgen wird. Um einen möglichen Einsatz in einem zukünftigen Robotermodell zu begünstigen, fiel die Wahl auf das Modell ADNS-5030, das laut Herstellerangabe zu der Zeit

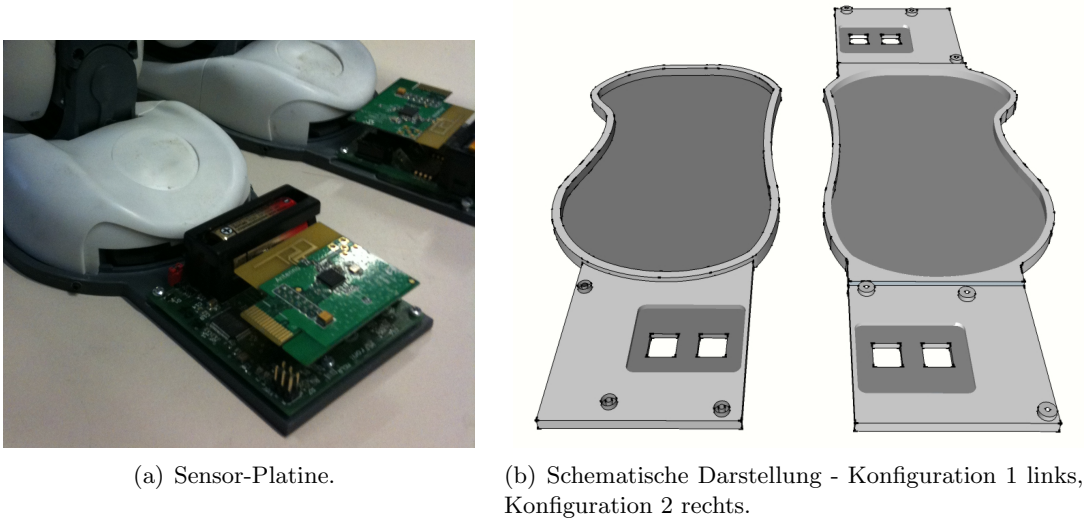


Abbildung 6.2.16: Sensorschuhe des NAO.

der kleinste optische Maussensor auf dem Markt gewesen ist. Dieser verwendet zwar lediglich LED-Technik, doch bietet diese Wahl auch einen Vorteil in den Kosten und im Energieverbrauch<sup>12</sup>. Da beides für den kommerziellen Einsatz in der mobilen Robotik interessante Faktoren sind, wird so gleichzeitig untersucht, ob bereits *low-cost* Sensoren für einen robusten Einsatz in der Praxis ausreichend geeignet sind. Aufgrund seiner geringen Abmessungen von  $9,1 \times 9,9$  mm können zwei dieser Sensoren nebeneinander auf einer Platine untergebracht werden. Dies würde beim Einbau in die Füße eine weitere Platzersparnis bedeuten. Der Sensor besitzt eine Auflösung von 1000 DPI und arbeitet laut Hersteller zuverlässig bis zu einer Geschwindigkeit von rund  $0,35$  m/s und einer Beschleunigung von bis zu  $2$  g. Zur Auswertung der Sensorinformationen kommt der Mikrocontroller MRF24J40 der Firma *Microchip* zum Einsatz. Ein Foto der Platine ist in Abbildung 6.2.17(a) zu finden.

Da die Sensoren nicht direkt mit dem Bussystem des NAO verbunden werden können besteht nur die Möglichkeit diese über eine externe Schnittstelle zu verbinden. Weil sowohl LAN als auch WLAN aus Gründen der Latenz nicht für diese Kommunikation geeignet sind, bleibt noch der Einsatz der USB-Schnittstelle. Eine externe Kabelanbindung würde den Roboter merklich in seiner Bewegungsfreiheit einschränken, somit muss die Übermittlung der Daten drahtlos über einen USB-Funkempfänger geschehen. Hierzu wurde für die Sensorplatinen ein Anschluss für ein *ZigBee*-Funkmodul vorgesehen, welches in Abbildung 6.2.17(a) zu finden ist. Mittels einer eigens entworfenen USB-Empfänger-Platine, siehe Abbildung 6.2.17(b), ist damit der Anschluss von bis zu 4 Sensor-Platinen gleichzeitig möglich.

<sup>12</sup>Es muss jedoch angemerkt werden, dass die Leistungsaufnahme einer Laser-Diode zwar höher ist, diese jedoch sehr klein gegenüber der Energieaufnahme der Motoren ist.

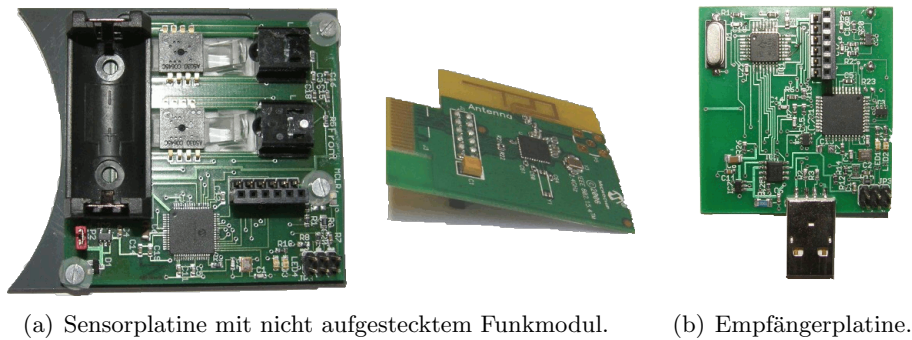


Abbildung 6.2.17: Sensor- und Empfängerplatine.

### 6.2.3.1 Kalibrierung der Sensoren

Die fertigen Platinen wurden zunächst kalibriert, um eine zuverlässige Messung zu ermöglichen. Hierzu wurden die Sensoren mit Hilfe eines Industrieroboters vom Typ KR6 der Firma KUKA, wiederholt über verschiedene Untergründe verfahren. Die Wiederholungsgenauigkeit des KR 6 gibt der Hersteller mit  $\pm 0,05$  mm an. Daher kann dessen Bewegung im Rahmen der zu untersuchenden Anwendung als fehlerfrei angenommen werden und bestimmt den Referenzwert. Es soll nicht auf alle Details dieser Kalibrierung eingegangen werden, doch lässt die Untersuchung des Sensorverhaltens auf unterschiedlichen Untergründen interessante Rückschlüsse auf die Zuverlässigkeit der Messung zu und wird daher ausschnittsweise angeführt.

Der optische Sensor misst die zurückgelegte Strecke in *counts*. Diese können über die DPI-Zahl theoretisch in die reale Strecke zurückgerechnet werden. Allerdings zeigen bereits erste Untersuchungen, dass es sich dabei durch eine Abhängigkeit vom Untergrund nur um eine ungefähre Umrechnung handelt. Die tatsächliche Bewegung des Mauszeigers hängt zum einen von der Auflösung des Monitors ab und erfordert zum anderen keine derart exakte Präzision. Der Benutzer schließt in diesem Fall den Regelkreis und sorgt durch seine Erfahrung für die korrekte Bewegung des Zeigers. Für die genaue Bestimmung der Bewegung des Roboters muss also zunächst ein exaktes Verhältnis von gemessenen *counts* zur tatsächlichen Strecke bestimmt werden. Exemplarisch sei hier die Kalibrierungsmessung eines Sensors bei unterschiedlichen Geschwindigkeiten und Untergründen angeführt, deren Auswertung in Abbildung 6.2.18 zu finden ist. Untersucht wurden Papier - als Referenzwert - und typische Untergründe im häuslichen Umfeld, wie kurzfloriger Teppich, Linoleum und Laminat. Die Werte wurden jeweils über 20 Kalibrierungsdurchläufe gemittelt. Die Auswertung zeigt eine Variation in den gemessenen *counts/cm*, die sowohl von der untersuchten Geschwindigkeit, als auch von der Bodenbeschaffenheit abhängt. Wie erwartet, liefert der Sensor die besten Resultate auf Papier, da es eine gute Strukturierung in der Oberfläche besitzt. Unerwartet schlecht schneidet Linoleum ab, obwohl es ebenfalls eine gute Struktur besitzt. Offensichtlich ist aber die Oberfläche zu glänzend, so dass die Reflexion die Messung verschlechtert. Ebenfalls reflektierend ist das Laminat, doch scheint dies durch die Maserung wieder ausgeglichen zu werden, da es konstant gute Werte liefert. Stark geschwindigkeitsab-

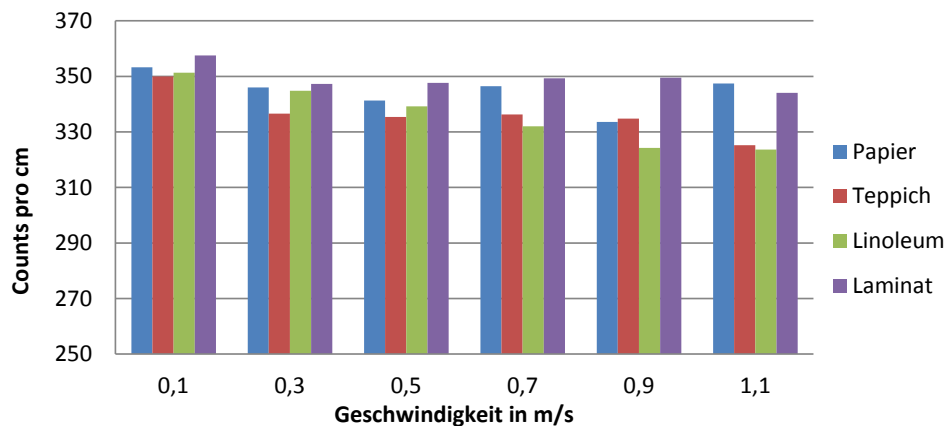


Abbildung 6.2.18: Sensorkalibrierung auf unterschiedlichen Untergründen.

hängig ist die Messung auf Teppichboden. Dies liegt vermutlich daran, dass der Boden relativ zur Höhe des Sensors große Höhenunterschiede in der Struktur aufweist und es so verstärkt zu Messfehlern kommt. Um gute Messwerte zu erreichen genügt es also nicht, die Messwerte mit einem Kalibrierungsfaktor zu versehen, sondern man benötigt eine Kalibrierung für jeden Untergrund, die idealerweise auch die Geschwindigkeit in Betracht ziehen würde. Da aber die Geschwindigkeit, die beim Rutschen erreicht wird, nicht bekannt ist, kann diese in der Kalibrierung nicht berücksichtigt werden. Eine Kalibrierung in Abhängigkeit vom Boden erscheint sinnvoll, doch setzt dies voraus, dass der Roboter Wissen darüber besitzt, auf welcher Art von Untergrund er sich bewegt. Festzuhalten bleibt, dass der LED-Sensor eine starke Oberflächenabhängigkeit in der Messung besitzt.

Interessant ist die Tatsache, dass der Sensor auch noch bei Geschwindigkeiten von über 1 m/s vergleichsweise gute Werte liefert und somit auch noch weit außerhalb der Herstellerspezifikation von rund 0,3 m/s operieren kann. Angenommen, der Sensor sei an der Fußspitze des Roboters angebracht, so müsste sich der Roboter, bei einer Fußlänge von rund 10 cm, innerhalb einer Sekunde zweimal auf der Ferse um seine eigene Achse drehen, um eine solche Geschwindigkeit des Sensors zu erreichen. Dies wird in der Praxis nicht eintreten, womit die Messgeschwindigkeit des Sensors als ausreichend beurteilt werden kann. Als Resultat dieser Betrachtung erscheint es sinnvoll, die Sensoren nur auf Messungen im niedrigen Geschwindigkeitsbereich von 0,1 m/s zu eichen.

### 6.2.3.2 Evaluation der Sensor-Schuhe

Nachdem durch die Kalibrierung sichergestellt ist, dass die Sensoren zuverlässig Translationen auf den Testuntergründen messen, wird nun die Bewegungsmessung des NAO mit diesen genauer untersucht. Aus Abbildung 6.2.16 ist bereits ersichtlich gewesen, dass für die Experimente die Sensorplatinen an unterschiedlichen Positionen in den Schuhen montiert werden können. Grund hierfür sind die vorangegangenen theoretischen Überlegungen in Kapitel 6.2.2, die aufgezeigt haben, dass die Position der Sensoren



Einfluss auf die Rechnung hat. Die Sensoren messen, jeweils unabhängig von der Position, die Bewegung  $\Delta_{xy}$  mit einem Fehler, der unter anderem durch das Rauschen der Sensoren entsteht. Somit ist die gemessene Orientierung des Vektors  $\mathbf{d}$  in der Praxis fehlerbehaftet. Bei geringem Abstand der Sensoren wirkt sich ein identischer absoluter Fehler jedoch, relativ gesehen, stärker auf die Orientierung aus, als bei weiter entfernten Sensoren. Da diese Orientierung von  $\mathbf{d}$  aber nach Formel (6.2.4) die Berechnung der Rotation bestimmt, ist offensichtlich, dass bei sehr nahe liegenden Sensoren der Fehler durch den Einfluss von Sensorrauschen dominant werden könnte. Daher werden im weiteren Verlauf jeweils die Ergebnisse eines Sensorsetup mit beiden Sensoren vorne - im Folgenden als Konfiguration 1 bezeichnet - und einem Setup mit einem Sensor vorne und einem Sensor hinten - Konfiguration 2 - gegenübergestellt. Zur Verdeutlichung sind in Abbildung 6.2.19 jeweils die Abstandsvektoren  $\mathbf{c}$  der Konfiguration 1 (roter Pfeil) respektive Konfiguration 2 (grüner Pfeil) eingezeichnet. Für Konfiguration 1 ergibt sich damit ein Abstand der Sensoren von 17 mm gegenüber 232 mm bei Konfiguration 2.

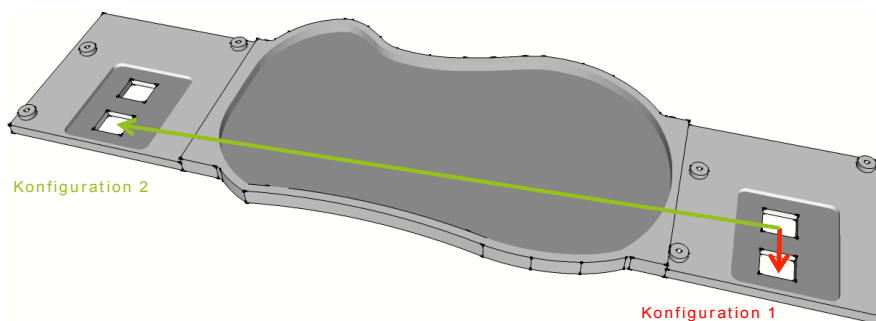


Abbildung 6.2.19: Einfluss der Sensorposition auf die Messung.

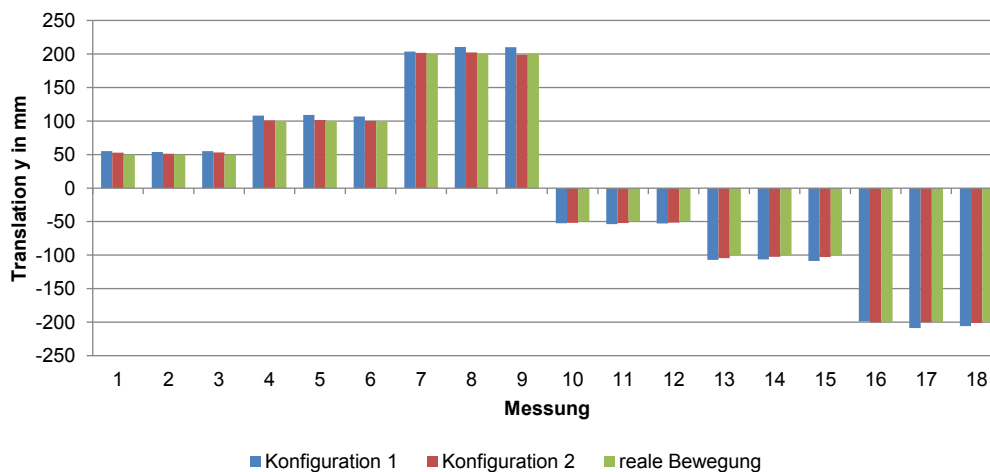


Abbildung 6.2.20: Experiment 6.2.3.2.A - Manuelle Translation in  $y$ -Richtung.

Zur Verifikation des in Kapitel 6.2.2 entwickelten Algorithmus zur Messung der Gleitbewegung wird der Roboter in einem ersten Experiment manuell wiederholt über einen grünen Teppich geschoben. Hierzu wurden jeweils Flachwinkel auf dem Boden an der Start- und Endposition des Roboters befestigt, um zu gewährleisten, dass die Wiederholbarkeit der Messung im Rahmen einer manuellen Durchführung gegeben ist. Der



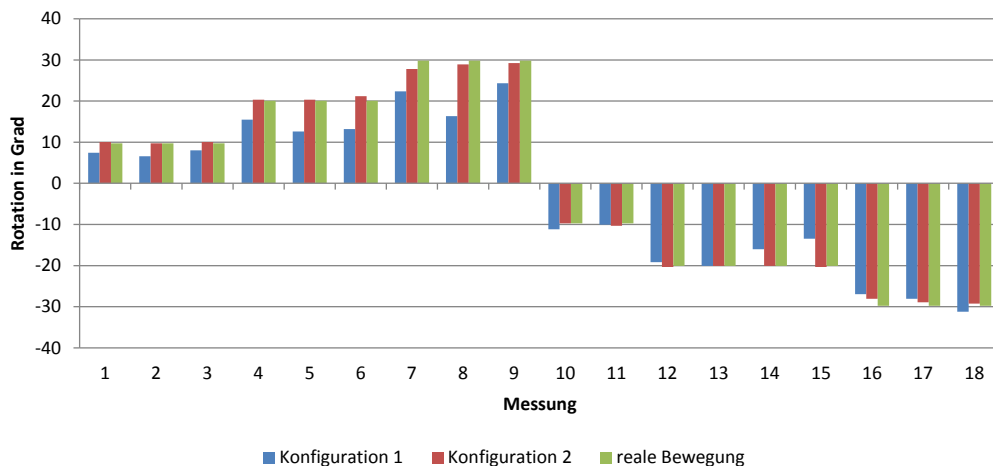
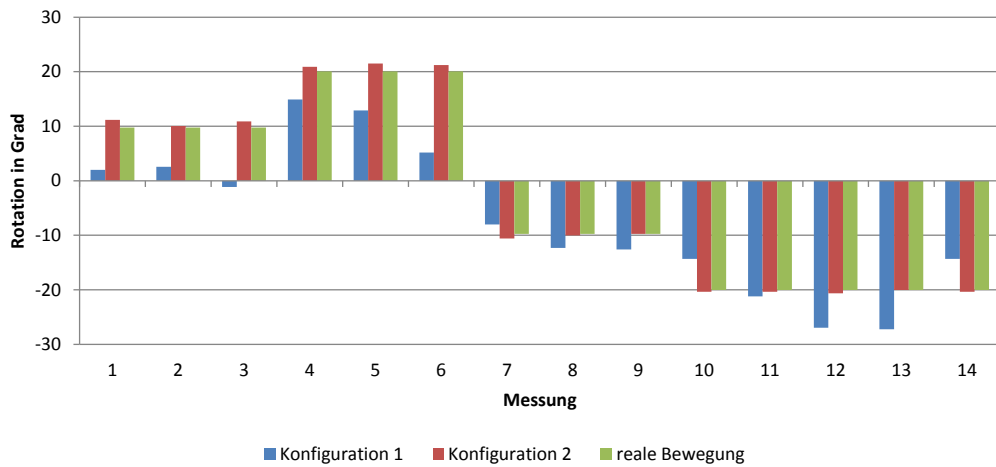


Abbildung 6.2.21: Experiment 6.2.3.2.B - Manuelle Rotation.

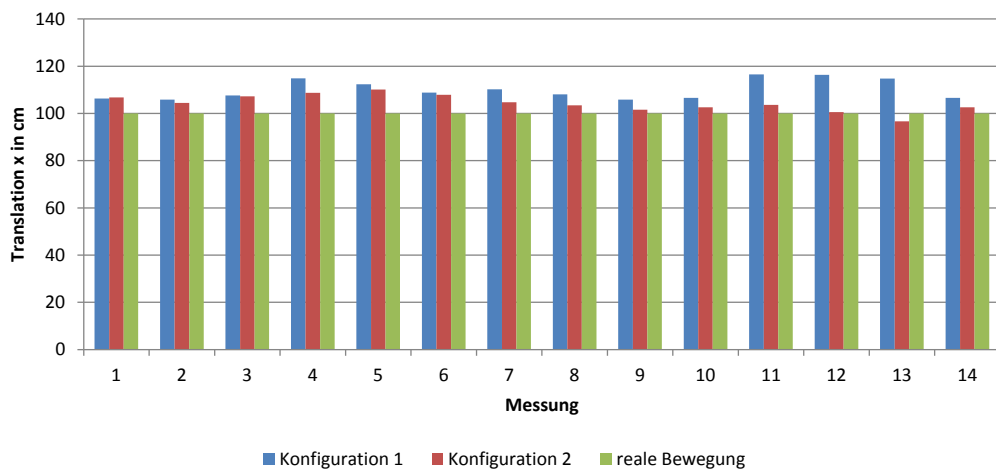
NAO ist hierzu am rechten Fuß mit der Konfiguration 1 und am linken mit Konfiguration 2 ausgestattet. Jeder Sensor ist separat durch den Industrieroboter KUKA kalibriert worden, um Unterschiede zwischen den Sensoren auszuschließen und vergleichbare Ergebnisse zu erreichen. Ausschnitte dieser Messungen sind in den Abbildungen 6.2.21 und 6.2.22 zu finden.

Experiment 6.2.3.2.A untersucht zunächst eine reine Translation des Roboters in  $y$ -Richtung um eine Strecke von 5 cm, respektive 10 cm und 20 cm. Diese Bewegung kann bereits mit einem einzelnen Sensor gemessen werden. Daher ist zu erwarten, dass die Ergebnisse durch den Einsatz der Schuhe, unabhängig von der Sensorkonfiguration, durch Mittlung der Messwerte verbessert werden können. Die Ergebnisse in Abbildung 6.2.20 bestätigen zunächst, dass die reale Bewegung von den Sensoren verhältnismäßig exakt aufgezeichnet wird. Bei genauerer Betrachtung fällt jedoch auf, dass der Fehler der Konfiguration 1 in jedem Fall größer ist, als der der Konfiguration 2. Da dieser Fehler zum einen größer als die Kalibrierungsungenauigkeit ist und zum anderen der Abstand der Sensoren nicht in diese Rechnung einfließt, muss es eine weitere Ursache zugrunde liegen. Der Teppich weist durch seinen Gebrauch als Fußballfeld stellenweise unterschiedliche Abnutzung, ebenso wie leichte Unebenheiten auf. Diese Unterschiede in der Bodenbeschaffenheit können zu Abweichungen in der Messung führen. Da diese Bereiche lokal begrenzt sind, liegt der Schluss nahe, dass bei Konfiguration 1, durch die Nähe der Sensoren zueinander, eine solche Anomalie jeweils beide Sensoren gleichzeitig beeinflusst hat. Bei Konfiguration 2 hingegen scheint durch die Lokalität jeweils nur ein Sensor beeinflusst worden zu sein. Durch Mittlung der Messwerte wird der Effekt folglich abgeschwächt. Es kann also als erstes Ergebnis festgehalten werden, dass die Sensorschuhe in der Lage sind Translationen mit einer guten Genauigkeit zu messen, es jedoch einen Vorteil bietet, die Sensoren nicht lokal zu gruppieren.

Die nächste Messung in Form von Experiment 6.2.3.2.B misst eine reine Rotation des Roboters. Hier zeigt die Auswertung der Daten in Abbildung 6.2.21 einen deutlichen Vorteil der Konfiguration 2 gegenüber Konfiguration 1 auf. Wie erwartet, beeinflusst das Sensorrauschen die Konfiguration 1 stärker als Konfiguration 2. Es zeigt sich je-



(a) Gemessene Rotation.



(b) Gemessene Translation in  $x$ -Richtung.

Abbildung 6.2.22: Experiment 6.2.3.2.C - Manuelle kombinierte Bewegung.

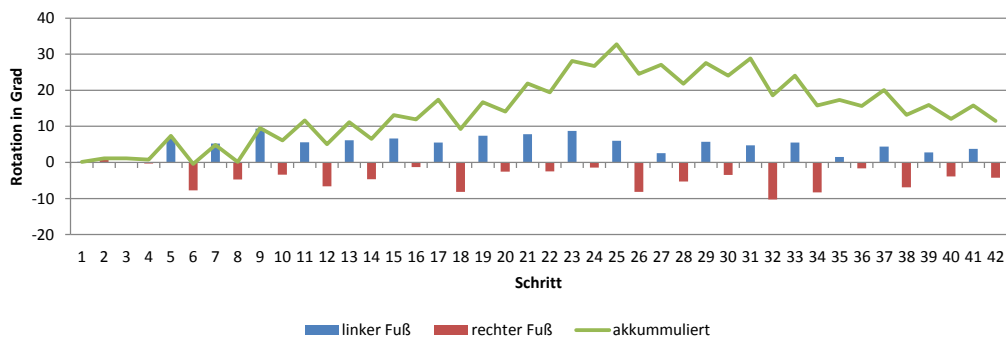
doch, dass das Rauschen der Sensoren wohl so signifikant ist, dass sich der Fehler auch bei reiner Rotation nicht durch Integration über die Zeit ausgleicht. Die Messung von Konfiguration 2 ist hingegen erneut genau genug, um eine Eignung für die Praxis zu belegen.

Um darzulegen, dass auch bei Überlagerung der Bewegung der Algorithmus in der Lage ist, Rotation und Translation voneinander zu trennen, wird abschließend in Experiment 6.2.3.2.C eine Bewegung sowohl in  $x$ - als auch in  $y$ -Richtung untersucht. Wie die Diskussion um Abbildung 6.2.13 nahelegt, weist der Plot der Ergebnisse in Abbildung 6.2.22 eine Verschlechterung beider Konfigurationen gegenüber Experiment A und B auf. Der Einfluss der Messfehler auf Konfiguration 1 ist jedoch deutlich größer, so dass teilweise nicht einmal mehr die Richtung der Rotation korrekt gemessen wird. Im Vergleich zu den Experimenten A und B sind die Ergebnisse der Konfiguration 2 hingegen nur minimal schlechter. Somit liefert diese Messung zum einen die praktische Verifikation des theoretischen Algorithmus und belegt zum anderen die Notwendigkeit, die Sensoren entsprechend weit voneinander entfernt zu platzieren.

## 6.2.4 Laufbewegung

Die manuellen Experimente haben gezeigt, dass die entwickelten Sensorschuhe in der Lage sind, gleitende Translationen und Rotation des Roboters zu messen. Folglich sollte sich durch ihren Einsatz die Odometriemessung positiv beeinflussen lassen. Daher wird dies nun an einer autonomen Laufbewegung des Roboters untersucht. Aufgrund der vorangegangenen Ergebnisse wird in den folgenden Experimenten nur Konfiguration 2 für beide Füße des Roboters verwendet. Es ist zu bedenken, dass das zusätzliche Gewicht und die Vergrößerung der Füße das Laufverhalten des Roboters beeinflusst. Daher sind die Laufparameter der NDD-WE für dieses Experiment entsprechend angepasst, um trotzdem einen stabilen Lauf zu gewährleisten.

Im folgenden Experiment 6.2.4.A wird der reine Vorwärtslauf des Roboters NAO mit einer Geschwindigkeit von  $v_x = 100$  mm/s untersucht. Der Roboter erhält wiederholt vom selben Startpunkt aus einen *Walkrequest* in positiver  $x$ -Richtung und läuft solange, bis seine Odometriemessung angibt, dass er sich 2 Meter in  $x$ -Richtung fortbewegt hat. Beim Lauf kann die reale Bewegung des Roboters nicht während jedes Schrittes überwacht werden, da hierzu am IRF keine geeignete externe Sensorik zur Verfügung steht. Eine manuelle Messung scheidet aus, da hierzu der Lauf unterbrochen werden müsste. Somit kann zur Evaluation im Folgenden erneut lediglich die gesamte Lauftanz herangezogen werden. Hierzu wird nach dem Anhalten des Roboters die Summe der gemessenen Fehler mit der Abweichung der Odometriemessung von der realen Endposition des Roboters verglichen. Dennoch liefern gerade die Sensormessungen der einzelnen Schritte aufschlussreiche Informationen, über das Zustandekommen des Gesamtfehlers. Aus diesem Grund sind diese exemplarisch für einen Lauf in Abbildung 6.2.23 aufgeführt, um dem Leser ein besseres Verständnis zu ermöglichen. Es ist klar zu erkennen, dass die Sensoren während jedes Schrittes eine Rotation des Roboters aufzeichnen. Auch wenn über die Qualität der Messung mangels Referenzwertes keine Aussage getroffen werden kann, so bestätigt sich doch zumindest die visuelle Beobachtung



(a) Rotationsmessung einzelner Schritte.

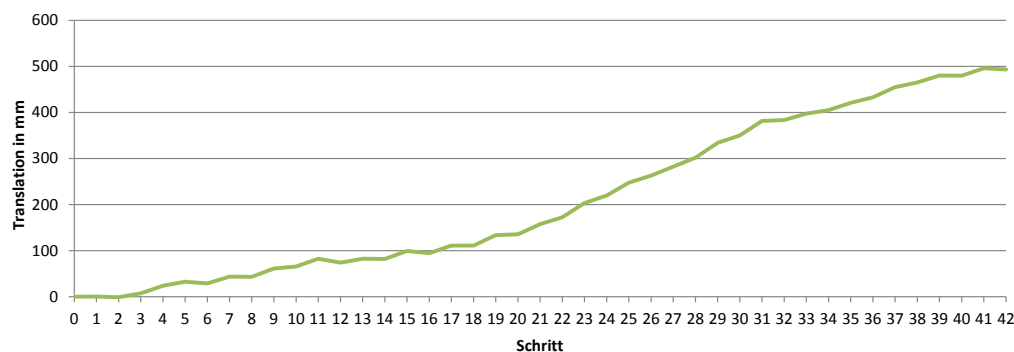
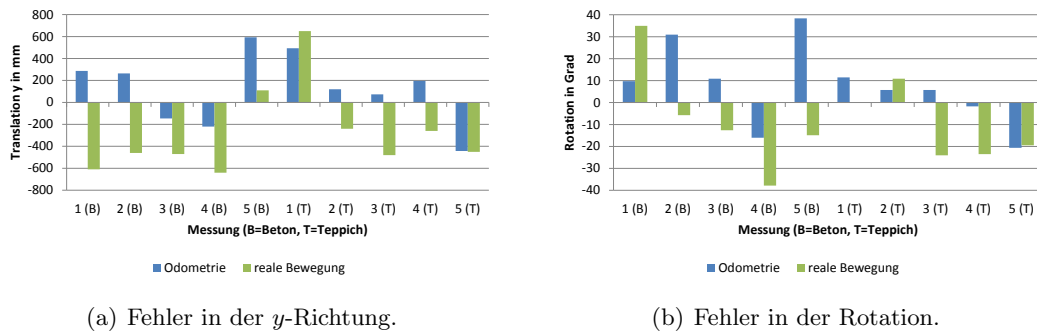
(b) Akkumulierter Fehler in  $y$ -Richtung.

Abbildung 6.2.23: Exemplarische Schrittauswertung eines Laufs.

der Drehung des Roboters aus Kapitel 6.1.2. Weiterhin fällt auf, dass diese Rotation zu Beginn der Bewegung nicht stattfindet, sondern erst nach einigen Schritten abrupt einsetzt. Dies ist ein Indiz für die Korrektheit der Annahme, dass die Beschleunigung des Schwungbeines Ursache für die Rotation ist. Während der ersten Schritte befindet sich der Roboter noch in der Beschleunigungsphase der NDD-WE und die Kraft durch die Beschleunigung des Schwungbeines reicht daher nicht aus, um die Haftreibung der Füße zu überwinden. Dieses Phänomen tritt am Ende des Laufes nicht mehr auf, da die Abbremsphase der NDD-WE im Verhältnis zur Beschleunigungsphase deutlich kürzer ist. In diesem Lauf konnte durch die Messung der optischen Sensoren ein Fehler von rund 50 cm gemessen werden. Dieser Wert liegt nahe genug an der realen Abweichung von 65 cm, um zumindest die Korrektheit der Tendenz der Messung zu belegen. Somit kann Abbildung 6.2.23(a) zur Verifikation der Vermutung herangezogen werden, dass sich die Rotationen nicht in jedem Fall durch den nächsten Schritt ausgleichen. Folglich akkumuliert sich der Rotationsfehler über die Zeit und führt - wie erläutert - zu einem Translationsfehler in  $y$ -Richtung, wie Abbildung 6.2.23(b) belegt.

Abbildung 6.2.24 zeigt einen repräsentativen Ausschnitt der wiederholten Messung des Laufexperiments 6.2.4.A. Es wurde auf grünem Teppich und zum Vergleich auf Zementboden ausgeführt. Die Ergebnisse offenbaren einen erheblichen Unterschied der autonomen im Gegensatz zur manuellen Fortbewegung. Die autonome Messung zeigt nicht nur eine größere Abweichung der Messung von der Realität auf, sondern es wird

(a) Fehler in der  $y$ -Richtung.

(b) Fehler in der Rotation.

Abbildung 6.2.24: Experiment 6.2.4.A - Lauf in  $x$ -Richtung mit  $v_x = 100$  mm/s.

teilweise sogar die Richtung der Translation oder Rotation falsch gemessen. In diesem Falle würde die Odometriemessung durch die optischen Sensoren sogar noch verschlechtert. Die Beobachtung des Experiments hält eine schlüssige Erklärung für dieses Phänomen bereit. Die Schuhe sind bewusst so entworfen, dass die Sensoren nur bei Bodenkontakt eine Bewegung messen, um zu vermeiden, dass beim Anheben des Fußes im Schwungbein noch Bewegung gemessen wird. Doch auch, wenn die NDD-WE versucht den Schwungfuß parallel zum Boden aufzusetzen, so ist der NAO in der Praxis dazu nicht immer in der Lage. Daher berühren die Sensoren teilweise erst zeitlich versetzt den Boden und nehmen Bewegungen des Fußes ungleich wahr. Dieser Effekt wird durch die Anbringung der Sensoren vorne und hinten am Fuß nochmals verstärkt. Bereits in Kapitel 6.2 wurde diskutiert, dass genau dies Ursache für eine Rotation des Roboters sein kann, die in diesem Fall nicht, oder falsch, gemessen wird, da nur ein Sensor Bodenkontakt hat. Führt man diesen Gedankengang weiter fort, kann es zu einem ähnlichem Phänomen auch beim Standfuß kommen. Während der *Single-Support*-Phase verlagert die NDD-WE den ZMP vom hinteren zum vorderen Ende des Fußes. Da dies den CoP verlagert, resultiert dies in einer unterschiedlichen Anpresskraft der Sensoren. Gerade bei nachgiebigen Böden, wie einem Teppich, kann dies in Kombination mit der geringen Leistung des verwendeten ADNS-5090 Sensors dazu führen, dass bei einer Bewegung beide Sensoren abweichende Strecken messen. Es bleibt also als Ergebnis des Experiments festzuhalten, dass es von Vorteil wäre, die Sensoren näher an den Bereich des Fußes zu verlagern, in dem sich der CoP beim Lauf bewegt.

## 6.2.5 Vergleich optischer Sensoren

Die Ergebnisse der bisherigen Experimente mit optischen Sensoren zeigen deutlich, dass der Ansatz potenziell in der Lage ist Fehler der Odometrie zu messen. Allerdings erwies sich der bisher untersuchte ADNS-5030 Sensor als nicht ausreichend geeignet, um trotz der bei autonomer Fortbewegung auftretenden Störungen eine Verbesserung zu erzielen. Da die Ergebnisse aber eine Notwendigkeit aufzeigen, die Sensoren mit geringerem Abstand zu platzieren, folgt nun zum Abschluss des Kapitels ein Vergleich der Robustheit verschiedener modernerer optischer Sensoren. Primäres Ziel ist es, die Zuverlässigkeit der autonomen Messung zu verbessern. Es ist bereits erwähnt worden, dass zum Zeitpunkt der Experimente der vorangegangenen Kapitel 6.2.3 und 6.2.4 Sensoren mit

fortgeschrittenerer Messtechnologie noch nicht kommerziell erhältlich waren. Mittlerweile können jedoch alle in Kapitel 6.2.1 vorgestellten Technologien verglichen werden, um einen repräsentativen Überblick zu liefern. Lediglich von der Untersuchung der herkömmlichen LED-Technik wird abgesehen, da diese sich bereits als unzureichend robust erwiesen hat. Da *BlueTrack* und *Darkfield* exklusiv von Microsoft, respektive Logitech, hergestellt werden, ist es nicht möglich, diese in Form eines zum ADNS-5030 vergleichbaren Sensorkits zu erwerben. Es werden daher im Folgenden die Untersuchungen an gekauften Mäusen durchgeführt. Bei der Auswahl wird Wert darauf gelegt, Mäuse mit einer drahtlosen Datübertragung zu verwenden, um eine spätere Untersuchung am Roboter NAO nicht durch Kabel zu behindern. Aus diesem Grund fiel die Wahl auf die Microsoft *Wireless Mobile Mouse 4000* (1000 DPI) mit *Bluetrack*-Technologie und das Modell Logitech *Anywhere Mouse MX* (1000 DPI) mit *Darkfield Laser Tracking*. Zum Vergleich mit herkömmlichen Sensortechnologien kommt noch eine Hightech-Maus aus dem *Gaming*-Bereich zum Einsatz. Dies sind Mäuse, die primär zum Einsatz für Computerspiele entwickelt wurden und sich daher durch besondere Präzision auch bei hohen Geschwindigkeiten und Beschleunigungen auszeichnen. Durch die Überlegenheit gegenüber der LED-Technik finden in diesem Bereich lediglich Laser-Sensoren Anwendung. Die Wahl fällt hier auf die Logitech *G700* (5700 DPI). Diese lässt bei der Funkübertragung über USB eine Zykluszeit von 1 ms zu und unterscheidet sich somit in ihrer Latenz nicht von einer kabelgebundenen Maus. Die Auswertung der folgenden Experimente beruht auf Daten, die von Fondjo im Rahmen seiner Bachelorarbeit [33] am Institut aufgezeichnet worden sind.

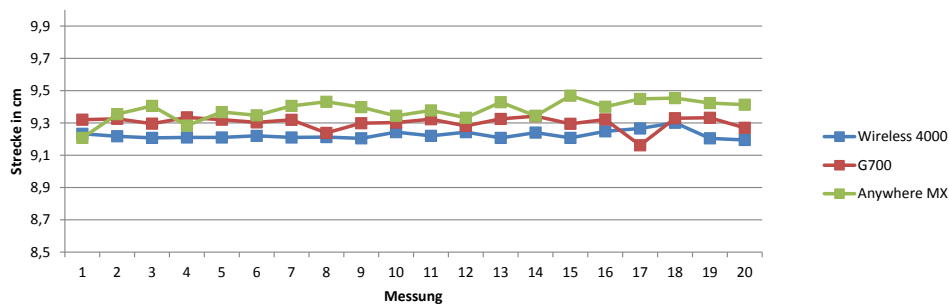
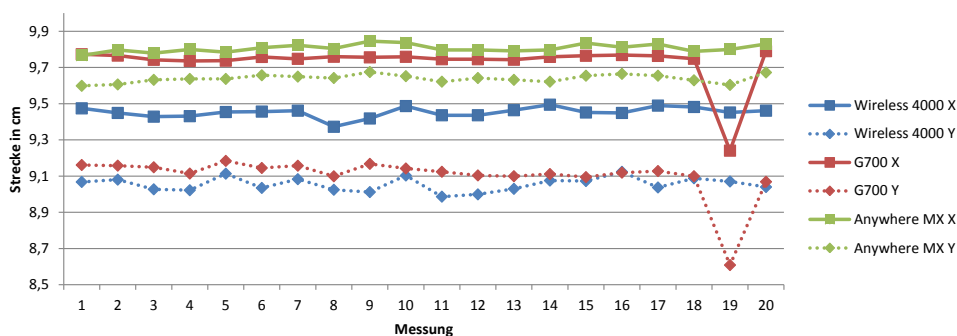
(a) Bewegung  $x$ -Richtung.(b) Bewegung  $x$ - und  $y$ -Richtung.

Abbildung 6.2.25: Experiment 6.2.5.A - Grüner Teppich.

Maus	Standardabweichung X
Wireless Mobile Mouse 4000	0,026 cm
G700	0,041 cm
Anywhere MX	0,062 cm

Tabelle 6.2.1: Experiment 6.2.5.A(a) - Grüner Teppich.

Maus	Standardabweichung X	Standardabweichung Y
Wireless 4000	0,028 cm	0,039 cm
G700	0,116 cm	0,120 cm
Anywhere MX	0,021 cm	0,022 cm

Tabelle 6.2.2: Experiment 6.2.5.A(b) - Grüner Teppich.

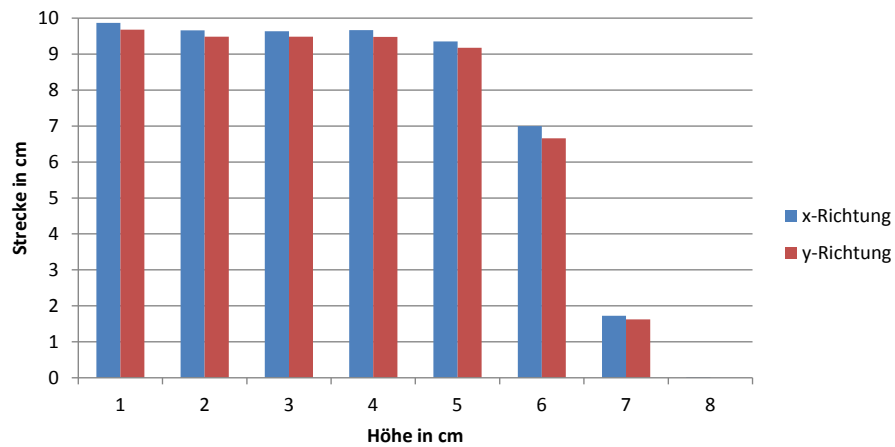
Zum Vergleich kommt erneut der KUKA KR6 Versuchsaufbau zum Einsatz, um die Mäuse wiederholt eine bekannte Strecke zu verfahren. Da der Fokus der folgenden Untersuchung auf der autonomen Bewegung des Roboters liegt, wird an dieser Stelle nur die Bewegung auf grünem Teppich ausgewertet. Da sich die Robustheit der Messung als entscheidend herausgestellt hat, wird zunächst die Wiederholbarkeit der Messung mit allen Mäusen untersucht, indem diese mehrfach eine Strecke von 10 cm bewegt werden. Exemplarisch sind in Abbildung 6.2.25(a) die Messdaten einer Bewegung in  $x$ -Richtung aufgetragen. Gemessen wurde nur an den markierten Punkten. Die lineare Verbindung soll lediglich dem menschlichen Auge die Auswertung erleichtern. Um trotz unterschiedlicher DPI-Werte die Ergebnisse vergleichen zu können, wird direkt auf die gemessene Strecke normiert. Erneut zeigt sich, dass die vom Hersteller angegebene DPI-Zahl sich von der realen Messung unterscheidet und eine Kalibrierung der Sensoren durchgeführt werden muss. Da aber nach der Kalibrierung die gemessenen Werte um einen gemeinsamen Mittelwert streuen würden, wird an dieser Stelle auf eine Durchführung verzichtet, um die Auswertung weiterhin lesbar zu halten. Bei genauerer Betrachtung fällt zunächst auf, dass alle Mäuse bereits ohne Kalibrierung verhältnismäßig gute Ergebnisse liefern. Auch die Streuung der Messwerte ist gering, so dass die Robustheit als gut angesehen werden kann. Optisch betrachtet erscheint hier die Wireless Mobile Mouse 4000 gegenüber den anderen Mäusen einen geringen Vorteil zu haben. Dies lässt sich auch statistisch durch die Standardabweichungen belegen, die in Tabelle 6.2.1 zu finden sind.

Um zu evaluieren, ob diese Ergebnisse durch den in Abbildung 6.2.13 erläuterten Effekt der Bildfensterverschiebung bei diagonaler Bewegung beeinträchtigt werden, sind in Abbildung 6.2.25(b) Messwerte einer gleichzeitigen Bewegung von 10 cm in  $x$ - und 10 cm in  $y$ -Richtung aufgetragen. Erneut zeigt sich im Graphen ein stabiler Verlauf der auch anhand der Standardabweichungen in Tabelle 6.2.2 belegt werden kann. Bereits in der vorhergehenden Messung ist aufgefallen, dass bei der G700 der Messwert 17 stärker abweicht als die sonstige Streuung. Analoges, jedoch noch stärker ausgeprägtes, Verhalten findet sich nun in Messung 19 wieder. Es ist allerdings besonders auffällig, dass dieses Phänomen sowohl in  $x$ -, also auch in  $y$ -Richtung auftritt. Da der Sensor nur inkrementelle Updates übermittelt, liegt die Vermutung nahe, dass dies auf ein Problem in der Übertragung zurückzuführen ist. Entgegen der Erwartungen zeigt sich,

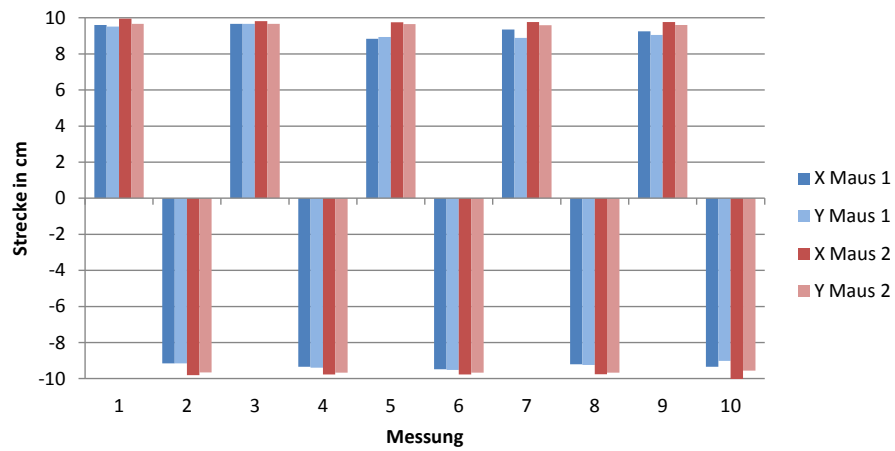
dass durch die Überlagerung der Bewegung die Messung der Anywhere MX bei dieser Evaluation stabiler geworden ist. Allerdings trat bei der Anywhere MX ein Problem auf, dass sich ohne Langzeituntersuchung nicht experimentell belegen lässt. Die Messung ist über die Dauer eines Experiments immer stabil. Jedoch berichtet Fondje davon, dass im Laufe der Untersuchungen seiner Bachelorarbeit immer wieder Messungen mit der MX auftraten, die eine signifikante Abweichung des Mittelwert aufwiesen. Somit kann es sich bei der Verbesserung in Abbildung 6.2.25(b) durchaus um das Resultat einer solchen Schwankung handeln. Es ist denkbar, dass diese die Folge eines hier nicht untersuchten Faktors ist, beispielsweise der Betriebstemperatur. Als Fazit des Experiments lässt sich jedenfalls festhalten, dass die Abstraten moderner Sensoren hoch genug sind, dass Effekt der Fensterverkleinerung kaum einen Einfluss hat. Es hat sich allerdings gezeigt, dass eine Kalibrierung der Sensoren für jede separate Achse durchzuführen ist. Dies gilt auch für jede Bewegungsrichtung, da sich hier ebenfalls, wenn auch deutlich kleinere, Abweichungen feststellen lassen.

Die Experimente bestätigen die Vermutung, dass die neuen Technologien robust und akkurat sind und auch auf nicht idealem Untergrund - wie dem verwendeten Teppichboden - verlässlich arbeiten. Es lässt sich nicht eindeutig beurteilen, welche Technologie überlegen ist, da alle Ergebnisse vergleichbar gut sind. Für die weiteren Experimente fällt die Wahl auf die Microsoft Wireless Mobile Mouse 4000, da diese zum einen über alle Messungen gute Ergebnisse lieferte und zum anderen die kleinste und leichteste der untersuchten Mäuse ist. Letzteres begünstigt die Anbringung am NAO. Aus diesem Grund folgt nun abschließend in Experiment 6.2.5.B eine weitere Vertiefung der Analyse der Eigenschaften der Wireless Mobile Mouse 4000. Entgegen dem Avago ADNS-5030 Sensor zeigt die Maus mit der *Bluetrack*-Technologie keine Abhängigkeit von der Geschwindigkeit bis 1,5 m/s, weshalb auf eine genauere Analyse verzichtet werden kann. Jedoch legen die Experimente mit dem Sensorschuh nahe, dass es sinnvoll ist, auch ohne Bodenkontakt noch bis zu einer gewissen Höhe Bewegungen messen zu können. Daher zeigt 6.2.26(a) eine Messreihe der Maus bei unterschiedlichem Abstand vom Boden. Obwohl ab einer Höhe von 1 mm über dem Boden eine Verschlechterung der Messung zu beobachten ist, so ist diese doch als vernachlässigbar gering zu bezeichnen. Ab einer Höhe von 5 mm verschlechtert sich diese rapide und führt bereits ab 7 mm dazu, dass keine Bewegung mehr von der Maus registriert wird. Da zur Messung der Rotation bekanntlich mehrere Sensoren benötigt werden, wird in Abbildung 6.2.26(b) exemplarisch noch die Vergleichbarkeit zweier identischer Wireless Mobile Mouse 4000 Mäuse untersucht. Eine Analyse der Messung offenbart, dass die untersuchte Robustheit der Messung zwar für beide Exemplare charakteristisch ist, die absolut gemessene Strecke sich aber für beide dennoch unterscheidet. Es bedarf also einer individuellen Kalibrierung jedes Sensors.





(a) Höhenabhängigkeit.



(b) Vergleich zweier Wireless Mobile Mouse 4000.

Abbildung 6.2.26: Experiment 6.2.5.B - Zuverlässigkeit Wireless Mobile Mouse 4000.

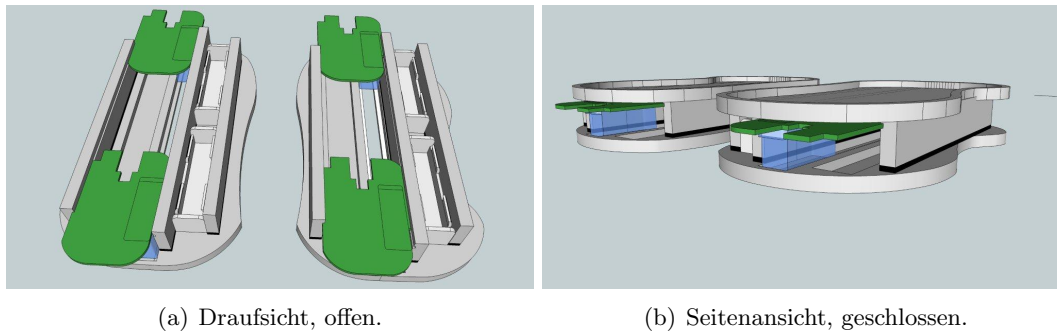


Abbildung 6.2.27: Schemazeichnung Sensorschuhe MS-4000.

### 6.2.6 Autonome Schussbewegung

Die vorangegangenen Untersuchungen zeigen, dass die Entwicklungen der letzten Jahre dazu geführt haben, dass moderne Maussensoren präziser und robuster geworden sind. Daher scheint es sinnvoll, zum Abschluss zu untersuchen, ob sich dieser Verbesserung auf die Odometriemessung im autonomen Betrieb des NAO übertragen lässt. Hierzu wird erneut ein Sensorschuh konstruiert, der unter den Füßen des Roboters NAO angebracht werden kann. Es werden pro Schuh zwei Wireless Mobile Mouse 4000 Mäuse verwendet, bei denen jeweils das Gehäuse und die überflüssige Platine für die Maustasten entfernt wird. Um den Ergebnissen des letzten autonomen Experiments Rechnung zu tragen, werden für den neuen MS-4000 Sensorschuh die Platinen unterhalb des Schuhs angebracht. Wie die Schemazeichnung in Abbildung 6.2.27 erkennen lässt, wird dies erreicht, indem die Schuhe aus zwei übereinanderliegenden Sohlen aufgebaut sind. Somit steht der Roboter NAO nun rund 2 cm höher, die Fußfläche bleibt hingegen auch mit Schuhen identisch. Die Sensoren werden so angebracht, dass sie noch bis einer Höhe von 2 mm verlässliche Werte liefern. Dies sollte ausreichen, um trotz Störungen durch nicht paralleles Aufsetzen der Füße verlässlich die Bewegung des Standbeins zu messen. Gleichzeitig sollte dadurch sichergestellt sein, dass keine Störungen durch das Schwungbein entstehen, wenn dieses angehoben oder abgesetzt wird. Zur Einsparung von Gewicht werden jeweils beide Platinen eines Fußes aus einer Batterie gespeist. Eine Kalibrierung der Sensoren wird für jeden Schuh einzeln für alle Richtungen und Achsen durchgeführt.

Zur Verifikation der Funktion wird auch mit dem MS-4000 Schuh zunächst eine Rotationsmessung ohne Bewegung des Roboters untersucht. Da primär die Untersuchung der Robustheit und Präzision der Messung im Vordergrund steht, wird diese ebenfalls mit dem KUKA Roboter durchgeführt. Dieser führt mit dem MS-4000 Schuh wiederholt eine Kreisbewegung mit unterschiedlichem Winkel durch. Der Rotationsanteil ist als Messreihe in Abbildung 6.2.28 zu finden. Eine Beschränkung auf einen Winkelbereich von  $\pm 45^\circ$  verringert nicht die Aussagekraft der Messung, da während eines DCM-Zyklus der Roboter keine größere Rotation durchführen kann. Es ist klar erkennbar, dass die Messung sehr präzise ist, vor allem wenn man denkt, dass die Rotation jeweils noch von einer Translation überlagert wird.

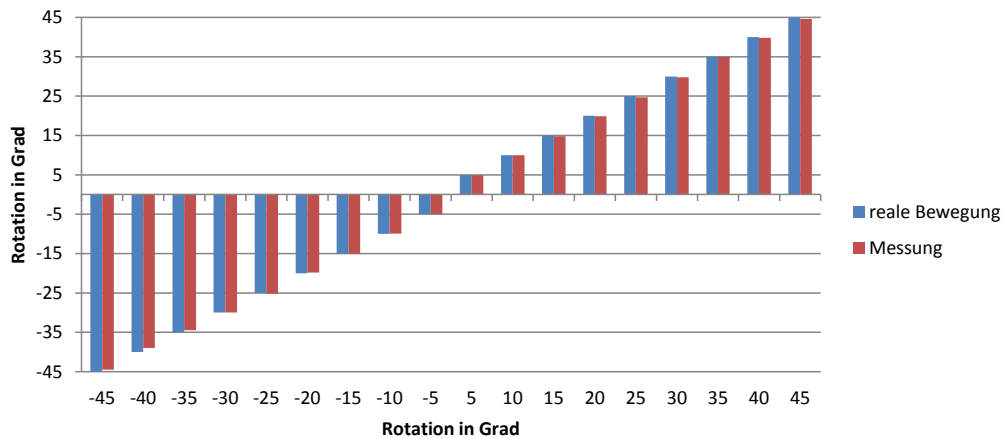
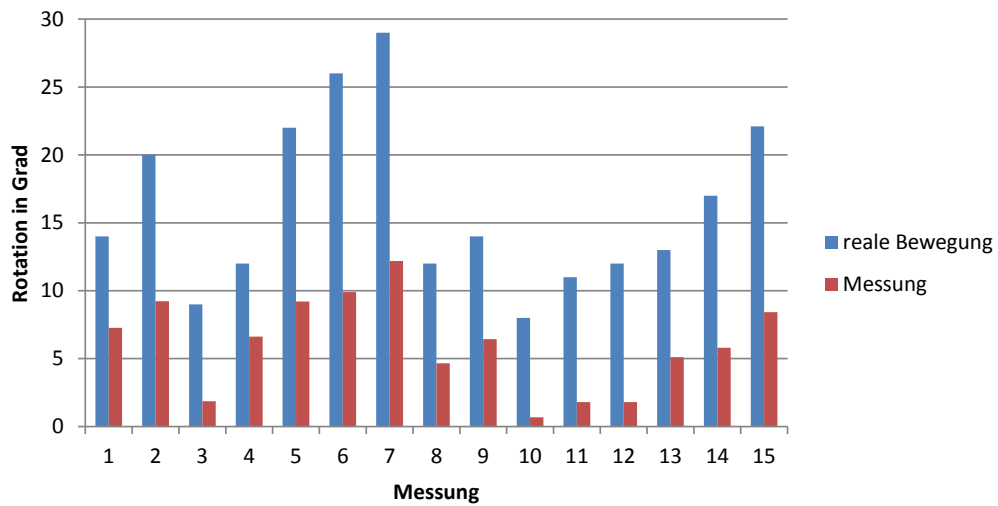


Abbildung 6.2.28: Experiment 6.2.6.A - Rotation Sensorschuh MS-4000.

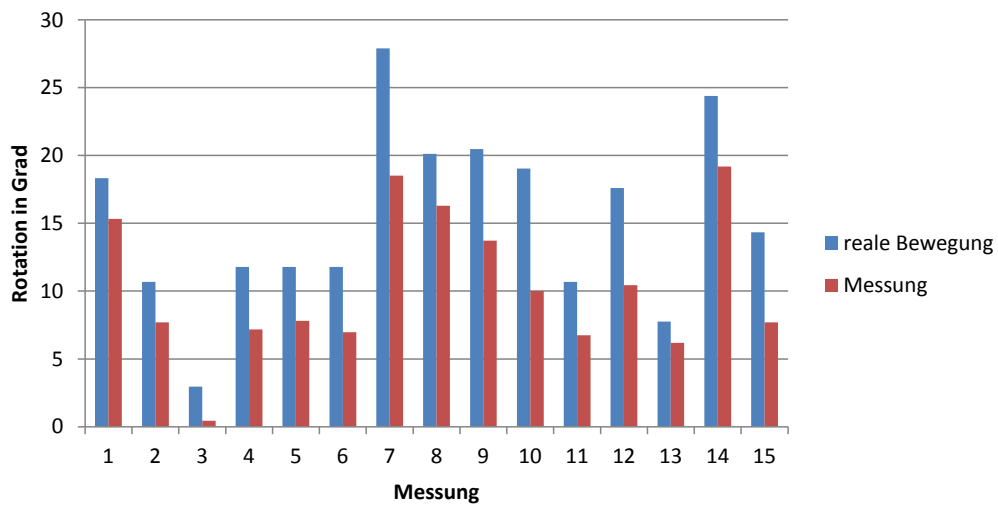
Bei Durchführung des folgenden Versuchs zeigt sich, dass der einzige freie USB-Port des Roboters NAO<sup>13</sup> nicht genügend Strom liefert, um 4 USB-Empfänger über einen passiven USB-Hub mit Strom zu versorgen. Daher kann die Aufzeichnung der Messung nicht vom Roboter NAO selbst durchgeführt werden, sondern erfolgt *remote* über einen separaten Linux PC. Als Experiment zur Untersuchung der Messung autonomer Bewegung wird diesmal ein anderer Versuchsaufbau gewählt. Da beim autonomen Lauf keine Auswertung eines einzelnen Schrittes gegen einen Referenzwert möglich ist, wird eine Bewegung des Roboters untersucht, die einem Schritt nahe kommt. Vergleicht man eine starke Schussbewegung des NAO mit der *Single-Support*-Phase, so sind die Bewegungsabläufe sehr ähnlich - lediglich beim Schuss noch extremer. Der Roboter geht von einem Stand auf zwei Beinen in einen einbeinigen Stand über und bewegt den Oberkörper zum Ausgleich seitlich, um anschließend den Schussfuß wie das Schwungbein nach vorne zu bewegen. Bereits die Auswertung dieser Bewegung in Experiment 4.4.4.E hat gezeigt, dass der Roboter durch seine Dynamik erheblichen Störungen ausgesetzt ist und sogar teilweise den Bodenkontakt gänzlich verliert. Somit scheint diese Bewegung geeignet, um Rückschlüsse auf einen Lauf zuzulassen. Abbildung 6.2.29(a) zeigt die Auswertung einer Folge von 15 Schüssen und vergleicht jeweils den vom Sensor gemessenen Wert mit einem durch manuelle Messung ermittelten Referenzwert. Im Vergleich zu Experiment 6.2.4.A lässt sich zunächst festhalten, dass zumindest die Richtung der Rotation in jedem Fall von den Sensoren korrekt wahrgenommen wurde. Jedoch ist die Qualität der Messung enttäuschend, da selbst in der besten Messung lediglich 51 Prozent der durchgeführten Rotation gemessen wurde. Dies ist besonders in Hinblick auf die sehr präzise Messung des vorhergehenden Experiments 6.2.6.A verwunderlich.

Bedenkt man, dass die Wireless Mobile Mouse 4000 von Microsoft für den Mobilbetrieb am Laptop entwickelt wurde, lässt sich eine Erklärung hierfür leicht finden. Bei Mäusen für den mobilen Einsatz ist es nicht unüblich, diese nach kurzer Zeit ohne Bewegung in einen Stromsparmodus zu versetzen. Aus diesem wacht die Maus wieder auf, sobald eine Bewegung detektiert wird. Das Resultat ist eine Totzeit der Maus. Ein solches Verhalten

<sup>13</sup>Das Mainboard des NAO besitzt lediglich zwei USB-Ports, von denen einer durch den zwingen benötigten USB-Speicher belegt ist.



(a) Schuss aus Stand.



(b) Schuss nach vorheriger Bewegung.

Abbildung 6.2.29: Experiment 6.2.6.B - Schuss aus Stand.

ist für die Wireless Mobile Mouse 4000 weder dokumentiert noch kann es durch den Treiber konfiguriert werden. Jedoch belegen die Ergebnisse in Abbildung 6.2.29(b) das Vorhandensein dieses Phänomens. Erneut werden 15 Schüsse durchgeführt. Doch wird der Roboter unmittelbar vor dem Schuss manuell auf der Stelle hin- und herbewegt. In der Tat belegen die Ergebnisse, dass sich hierdurch die Messung wesentlich verbessern lässt. Im Mittel wurde nun 64 Prozent der Rotationsbewegung durch die Sensoren aufgezeichnet. Somit besteht noch immer eine Abweichung zur Präzision der idealen Messung mit dem KUKA, doch ist diese vermutlich erneut darauf zurückzuführen, dass durch die Bewegung Störungen auftreten, die von beiden Sensoren unterschiedlich aufgenommen werden und die Rotation verfälschen. Es ist jedoch zu beachten, dass die Schussbewegung das Bein mit maximal möglicher Beschleunigung und Geschwindigkeit bewegt. Somit kann davon ausgegangen werden, dass diese Messung für einen einzelnen Schritt einen *Worst Case* im Bezug auf Störungen durch das Schwungbein darstellt.

### 6.2.7 Diskussion und Ausblick

Zusammenfassend lässt sich festhalten, dass die Anwendung von optischen Sensoren in der humanoiden Robotik ein vielversprechendes Forschungsgebiet zur Verbesserung der Odometrie ist. Dies belegen die theoretischen, sowie die experimentellen Untersuchungen. Autonome Bewegungen mit hoher Dynamik führen zwar zu Störungen, die die Messung der Sensoren negativ beeinflussen, doch bei Einsatz moderner Sensoren lässt sich auch dann noch eine signifikante Verbesserung der Messung erzielen. Erwartungsgemäß erweist sich der Roboter NAO als ungeeignete Plattform für diese Art der Experimente, da sein nicht modifizierbarer Aufbau ein Hindernis darstellt. Die Wahl der Sensoren wird hierdurch auf solche eingeschränkt, die für den Mobilbetrieb konzipiert sind. Eben dies konnte aber in Experiment 6.2.6.B klar als Problem identifiziert werden, solange die Stromsparfunktion nicht deaktiviert werden kann. Daher wird geraten für weiterführende Experimente einen modifizierbaren Roboter zu verwenden, bei dem Sensoren direkt in das System des Roboters integriert werden können. Dies würde auch das Problem der Deaktivierung der FSR-Sensoren durch die Schuhe beheben. Als weiterer Vorteil kann dann auf Basis der FSR-Sensoren geschlossen werden, welchen Messung der optischen Sensoren vertraut werden kann und bei welcher Messung kein Bodenkontakt mehr bestand. Mit diesem Wissen kann der gesamte Messbereich der Sensoren über die ganze Höhe ausgenutzt werden, da Messungen des Schwungbeins bei der Auswertung ignoriert werden könnten. Letztendlich hat sich bei Translation gezeigt, dass die Mittlung aus zwei Sensormessungen bereits bei den schlechteren Avago Sensoren zu einer merklichen Verbesserung der Messung führt. Diese Beobachtung legt den Schluss nahe, dass eine Erhöhung der Sensoranzahl und eine anschließende Fusion der Messungen zu einer weiteren Verbesserung auch im Bereich der Rotationsmessung führen wird. Experimente in diese Richtung stehen noch aus.

### 6.3 Bewegungsschätzung im Raum

Im Laufe dieser Arbeit hat sich an mehrere Stellen gezeigt, dass das Wissen über die aktuelle Pose des Roboters und seine Eigenbewegung wichtig ist, um eine stabile und zielgerichtete Bewegung des Roboters durchzuführen. Nicht nur für deren Planung, sondern auch für eine Regelung ist Wissen über den aktuellen Zustand des Systems notwendig. Fehlerfortpflanzungsanalysen haben aufgezeigt, dass eine Bestimmung der Lage des NAO einzig aus den Gelenksensoren in einem erheblichen Fehler resultieren kann. Gleiches gilt für die Position des CoM im RzKS. In Kapitel 4.1 wird diskutiert, dass die Lage des Roboters theoretisch auch durch andere Sensoren bestimmbar ist - beispielsweise den Beschleunigungssensoren oder dem Gyrometer. Jedoch erreichen diese Ansätze bei unmittelbarer Auswertung nur unzureichende Ergebnisse mit den Sensoren des NAO. Aus diesem Grund wurde von Hauschildt und Kerner in [43] ein Algorithmus vorgestellt und untersucht, der unterschiedliche Sensorinformationen mit explizitem Wissen über das System kombiniert. Dieser basiert auf den Methoden, die Hauschildt in seiner Diplomarbeit [42] untersucht hat und ermöglicht eine bessere Zustandsschätzung von Mehrkörpersystemen. Durch seine Flexibilität und Skalierbarkeit ist der Ansatz in der Lage eine beliebige Anzahl Sensorinformationen zu verwenden. Somit ist es denkbar, dass ein zukünftiger Roboter mit zusätzlichen Beschleunigungssensoren in den Extremitäten ausgestattet wird und deren Information unmittelbar in das Bewegungsmodell einfließen können. Im Folgenden wird die Idee des Algorithmus nochmals aufgegriffen, um in den Kontext der Arbeit eingegliedert zu werden und das Kapitel zum Thema *Eigenbewegung* damit zum Abschluss zu bringen. Der Leser sei aber bereits an dieser Stelle darauf hingewiesen, dass der vorgestellte Algorithmus auf dem NAO V3 nicht zeitgleich mit der *Cognition* und *Motion* in Echtzeit ausführbar ist. Eine Evaluierung am realen Roboter muss daher ausbleiben, kann aber erwartungsgemäß mit Erscheinen der leistungsfähigeren Version V4 durchgeführt werden. Da dieses Kapitel sich somit nur als Ausblick auf zukünftige Forschung verstehen kann, wird die Herleitung des Verfahrens im Folgenden lediglich skizziert. Eine Vertiefung ist anhand der Ausarbeitung von Hauschildt [42] möglich, der die Zusammenhänge detaillierter ausführt.

Die Idee, eine Fusion mehrerer Sensoren zur Verbesserung einer Messung durchzuführen, ist genau so wenig neu wie deren Anwendung auf mobile Roboter. In der Regel werden hierzu Bayes- oder Kalman-Filter eingesetzt. Auf deren Grundlagen wird nicht weiter eingegangen und daher sei bei Interesse für eine Vertiefung in diese Thematik u. a. [109] empfohlen. Gemein ist diesen Ansätzen, dass in der Regel nur sehr vereinfachte Modelle des Roboters als Grundlage für den Algorithmus verwendet werden. Zur Verbesserung der Resultate werden stochastische Informationen über die Sensoren verwendet, um eine Aussage über die Wahrscheinlichkeit der Zustände zu treffen. Der Roboter wird als eine Kette von Festkörpern modelliert, um dann mittels Darstellung in *lokalen* Koordinaten vereinfacht zu werden. Hierzu werden die Gelenke des Systems explizit betrachtet. Ein typisches Beispiel ist der in [30] vorgestellte *Articulated-Body-Algorithmus*. Nur wenig ist jedoch in Richtung komplexerer Modelle geforscht worden, wie der Vergleich von Aoustin et al. in [2] zeigt. Ein plausibler Grund hierfür ist mit Sicherheit die begrenzte Rechenleistung mobiler Hardware. Doch durch die Fortschritte der mobilen Prozessoren der vergangenen Jahre erscheint es durchaus sinnvoll, auch komplexere Modellansätze

hinsichtlich ihrer Vorteile für die autonome Anwendung zu untersuchen.

### 6.3.1 Modellierung mit globalen Koordinaten

Die Verwendung von lokalen Koordinaten hat den Vorteil, die Anzahl der Parameter auf die Anzahl der Freiheitsgrade des Systems zu reduzieren und so dessen Komplexität zu vereinfachen. Allerdings führt dies zu einem System nichtlinearer Gleichungen, wodurch die Anwendung gebräuchlicher Methoden zur Zustandsschätzung, wie Kalman-Filter, erschwert oder verhindert wird. Aus diesem Grund wird hier ein anderer Ansatz verfolgt.

Wenn Kalman-Filter in der Robotik zur Bewegungsschätzung angewendet werden, geschieht dies meist indem der Roboter durch einen einzigen Festkörper angenähert wird. Für diesen können dann die Daten der Inertialsensoren zum Bewegungsupdate genutzt werden. Für rollende Roboter führt dies zu befriedigenden Ergebnissen. Die Dynamik des periodischen Laufs humanoider Roboter erfordert jedoch Beschleunigungen und Geschwindigkeiten, die die der eigentlichen Fortbewegung übersteigen. Folglich stellt deren Überlagerung ein Problem für eine derartige Systemabstraktion dar. Meist wird diese Dynamik durch das Filter lediglich als Rauschen behandelt, was ein signifikantes Rausch-Signal-Verhältnis zur Folge hat und nützliche Informationen über den Zustand des Roboters verwirft. Der folgende Algorithmus versucht daher, diese explizit zu betrachten. Zur sinnvollen Integration dieser Messungen ist es unumgänglich, den Roboter in seiner gesamten kinematischen Struktur zu modellieren und als Grundlage für die Schätzung zu verwenden. Um dies zu erreichen, wird für den vorgestellten Algorithmus eine Darstellung in *globalen*<sup>14</sup> Koordinaten angewandt. In der Literatur finden sich hierzu beispielsweise impulsbasierte Ansätze [8] oder solche, die Lagrange-Multiplikatoren nutzen [5]. Betrachtet man diese Ansätze, so ist ihnen gemein, dass Gelenke als Nebenbedingung aufgefasst und nicht explizit modelliert werden. Dies hat zur Folge, dass sich die Komplexität des Modells nicht reduziert, sondern dieses noch um weitere Parameter ergänzt wird. Obwohl dies zunächst nicht zielführend erscheint, erlaubt dieses Vorgehen die Zustandsschätzung für jeden Festkörper zunächst entkoppelt zu betrachten. Das komplexe Problem des Mehrkörpersystems wird in einfacher zu handhabende Teilprobleme zerlegt. Die Betrachtung jedes einzelnen Festkörpers als separates Filterproblem ermöglicht es zudem weitere Sensorinformationen direkt in das Filter zu integrieren. Jeder Festkörper könnte mit einem eigenen Beschleunigungssensor ausgestattet werden und somit die Verfolgung der Lage im Raum signifikant verbessert werden. Es ist jedoch zu bedenken, dass die Entkoppelung in Teilprobleme einen deutlichen Nachteil gegenüber den lokalen Koordinatenansätzen nach sich zieht. Da jeder Festkörper durch seinen eigenen Filter geschätzt wird, divergiert das System mit der Zeit aufgrund unterschiedlicher Messfehler. Diese Problematik wird in Kapitel 6.3.3 aufgegriffen und eine Lösung in Form einer Nachstabilisierung vorgestellt.

Ein humanoider Roboter bestehe aus  $n$  Festkörpern, deren Zustand  $\mathbf{x}_i$  im WKS eindeutig bestimmt ist durch ihren Positionsvektor  $p_i$ , ihre Orientierung  $q_i$ , die lineare

<sup>14</sup>In der Literatur auch als *maximale* Koordinaten bezeichnet.

Geschwindigkeit  $v_i$  und Winkelgeschwindigkeit  $\omega_i$ :

$$\mathbf{x}_i = (p_i, q_i, v_i, \omega_i)^T. \quad (6.3.1)$$

Zur einfachen Handhabung seien  $q_i$  und  $\omega_i$  in Quaternionen-Darstellung gegeben. Eine detaillierte Darstellung der daraus folgenden Implikationen ist in [18] zu finden.

Betrachten wir das System zu diskreten Zeitpunkten  $k \Delta t$  so kann die Bewegung des Körpers zwischen diesen in Näherung immer als linear und rotatorisch gleichförmig beschleunigten Bewegung betrachtet werden. Die Transferfunktion, die den Übergang vom Zustand  $k$  zu  $k + 1$  beschreibt, folgt dann nach dem Zeit-Weg-Gesetz zu

$$\mathbf{x}_i^{k+1} = \begin{pmatrix} p_i^k + v_i^k \Delta t + \frac{1}{2} a_i^k \Delta t^2 \\ Q_{[\frac{1}{2} \Delta q_i^k]} q_i^k Q_{[\frac{1}{2} \Delta q_i^k]}^* \\ v_i^k + a_i^k \Delta t \\ \omega_i^k + \alpha_i^k \Delta t \end{pmatrix}, \text{ mit } \Delta q_i^k = \omega_i^k \Delta t + \frac{1}{2} \alpha_i^k \Delta t^2. \quad (6.3.2)$$

Hierbei bezeichnet  $a_i^k$  die entsprechenden linear Beschleunigungen,  $\alpha_i^k$  die Winkelbeschleunigungen und  $Q_{[\delta]}$  das Einheitsquaternion zur Rotation  $\delta$ , siehe [18]. Durch die Massematrix  $\mathbf{M}_i$  kann  $a_i^k$  durch die Kräfte  $\mathbf{F}_i^k$  auf den Körper ausgedrückt werden. Ebenso hängt  $\alpha_i^k$  über den entsprechenden Trägheitstensor  $\mathbf{I}_i^k$  mit dem Drehmoment  $\boldsymbol{\tau}_i^k$  auf den Körper zusammen<sup>15</sup>. Folglich kann die Transferfunktion in Abhängigkeit vom Dynamikvektor  $\mathbf{u}_i^k = (\mathbf{F}_i^k, \boldsymbol{\tau}_i^k)^T$  der externen Kräfte  $\mathbf{F}_i$  und Drehmomente  $\boldsymbol{\tau}_i$  ausgedrückt werden:

$$\mathbf{x}_i^{k+1} = f(\mathbf{x}_i^k, \mathbf{u}_i^k). \quad (6.3.3)$$

Ausgehend von Gleichung (6.3.1) und (6.3.3) kann dann der Zustand von  $n$  unabhängigen Festkörpern zum diskreten Zeitpunkt  $k \Delta t$  beschrieben werden durch:

$$\begin{aligned} \mathbf{x}^k &= (x_0^k, \dots, x_{n-1}^k)^T, \\ \mathbf{u}^k &= (u_0^k, \dots, u_{n-1}^k)^T, \\ f(\mathbf{x}^k, \mathbf{u}^k) &= (f(x_0^k, u_0^k), \dots, f(x_{n-1}^k, u_{n-1}^k))^T. \end{aligned} \quad (6.3.4)$$

Bisher werden die Festkörper unabhängig voneinander betrachtet. Durch die kinematische Struktur des Roboters unterliegt jedoch jeder der  $n$  Festkörper Nebenbedingungen, engl. *Constraints*, die aus der Kopplung der Körper durch die Gelenke entstehen. Diese können unterschieden werden in Nebenbedingungen der Position  $C_p(\mathbf{p}, \mathbf{q})$  und Nebenbedingungen der Bewegung  $C_m(\mathbf{v}, \boldsymbol{\omega})$  und werden im Folgenden getrennt betrachtet. Da sich mittels  $C_p$  und  $C_m$  alle bei humanoiden Robotern gebräuchlichen Gelenke modellieren lassen, bleiben andere Nebenbedingungen unbehandelt. Die Bewegung von  $n$  verketteten Festkörpern ist dann beschrieben durch

$$\mathbf{x}^{k+1} = f(\mathbf{x}^k, \mathbf{u}^k), \quad (6.3.5)$$

<sup>15</sup>Im Gegensatz zur Massematrix  $\mathbf{M}_i$  wird der Trägheitstensor  $\mathbf{I}_i^k$  von der Orientierung des Körpers beeinflusst und ist somit abhängig von  $k$ .



mit

$$C_p(\mathbf{p}, \mathbf{q}) - \mathbf{d}_p = \mathbf{0}, \quad (6.3.6)$$

$$C_m(\mathbf{v}, \boldsymbol{\omega}) - \mathbf{d}_m = \mathbf{0}, \quad (6.3.7)$$

wobei  $\mathbf{d}_p$  und  $\mathbf{d}_m$  die Offsets der Nebenbedingungen bezeichnen, wie sie beispielsweise aus dem vom Festkörper gegebenen festen Abstand zweier Gelenke resultieren.

### 6.3.2 Zustandsschätzung ohne Nebenbedingungen

Die bisherigen Betrachtungen dienen der Beschreibung des Systemzustands zu einem festen Zeitpunkt, gegeben durch Gleichung (6.3.4). Auf Basis dieser Gleichung wird nun eine Zustandsschätzung durchgeführt, die zunächst die Nebenbedingungen  $C_p$  und  $C_m$  außer Acht lässt. Für den Prädiktionsschritt wird zusätzlich zum aktuellen Systemzustand - gegeben durch  $\mathbf{x}^k$  und  $\mathbf{u}^k$  nach Formel (6.3.4) - das Sensorupdate  $\mathbf{z}^k$  benötigt:

$$\mathbf{z}^k = \left( z_0^k, \dots, z_{n-1}^k \right)^T, \quad (6.3.8)$$

wobei

$$\mathbf{z}_i^k = g(\mathbf{x}_i^k), \quad (6.3.9)$$

den Zustand des  $i$ -ten Festkörpers zum Zeitpunkt  $k \Delta t$  darstellt. Das Resultat ist ein nichtlineares, zeitinvariantes System. Messungen, die mehrere Festkörper betreffen, werden nicht betrachtet und daher vernachlässigt. Da sich alle in dieser Arbeit vorgestellten Sensoren eindeutig einem Festkörper zuordnen lassen, stellt die für die Anwendung keine Einschränkung dar. Eine Ausnahme bilden die Gelenkwinkelsensoren, die zwei Festkörper verbinden und somit theoretisch beide beeinflussen. Diese werden für den Algorithmus jedoch lediglich dem jeweils folgenden Festkörper zugeordnet.

Es wird nun angenommen, dass die Eingangsgröße  $\mathbf{u}_i$  des Regelungssystems und die Messungen  $\mathbf{z}_i$  entweder unbekannt oder stark verrauscht sind. Sie werden daher durch Gauß'sche Zufallsvariablen modelliert. In Konsequenz werden die Zustände  $\mathbf{x}_i$  der Festkörper ebenfalls Zufallsvariablen  $\hat{\mathbf{x}}_i$  mit unbekanntem Verteilungen. Unter dieser Annahme und unter Verwendung des Erwartungsoperators  $E(\cdot)$  sind die Momente erster Ordnung gegeben als

$$E(\mathbf{a}) = (E(a_0), \dots, E(a_{n+1}))^T = \hat{\mathbf{a}}, \quad (6.3.10)$$

mit  $\mathbf{a} \in \{\mathbf{x}, \mathbf{u}, \mathbf{z}\}$ . Weiterhin sind die Momente zweiter Ordnung beschrieben durch

$$E([\mathbf{a} - \hat{\mathbf{a}}][\mathbf{a} - \hat{\mathbf{a}}]^T) = \begin{pmatrix} \Sigma_{a_0, a_0} & \cdots & \Sigma_{a_0, a_{n-1}} \\ \vdots & \ddots & \vdots \\ \Sigma_{a_{n-1}, a_0} & \cdots & \Sigma_{a_{n-1}, a_{n-1}} \end{pmatrix} = \Sigma_a, \quad (6.3.11)$$

Zur Reduktion der Komplexität wird angenommen, dass für die Kovarianzen

$$\Sigma_{x_i, x_j} = \Sigma_{u_i, u_j} = \Sigma_{z_i, z_j} = \mathbf{0} \text{ für } i \neq j, \quad (6.3.12)$$

gilt und diese folglich vernachlässigt werden können. Diese Annahme erlaubt es, das Problem der Zustandsschätzung des Mehrkörpersystems effektiver zu bearbeiten, da es nun in  $n$  Teilprobleme für die einzelnen Festkörper zerlegt werden kann. Das Resultat dieser Vereinfachung sind Fehler in der Schätzung. Doch diese werden durch den folgenden Schritt der Projektion der Nebenbedingungen, vergleiche Kapitel 6.3.3, korrigiert. Durch diese stochastische Modellierung des Problems können nun erprobte Algorithmen zur Zustandsschätzung, wie beispielsweise *Extended Kalman-Filter* (EKF) oder *Unscented Kalman-Filter* (UKF) angewandt werden, um das Problem möglichst exakt zu lösen. Auf Details der einzelnen Filter soll nicht eingegangen werden und es wird deshalb an dieser Stelle auf eine Erklärung der Theorie verzichtet. Bei Interesse kann eine Vertiefung der Methodik des EKF und UKF beispielsweise anhand von [41] oder [67] erfolgen. Details zur Umsetzung des hier verwandten UKF beschreibt Hauschildt in seiner Ausarbeitung [42]. Eine grafische Darstellung des Ablaufs der Zustandsschätzung ohne Berücksichtigungen der Nebenbedingungen ist in Abbildung 6.3.30 zu finden.

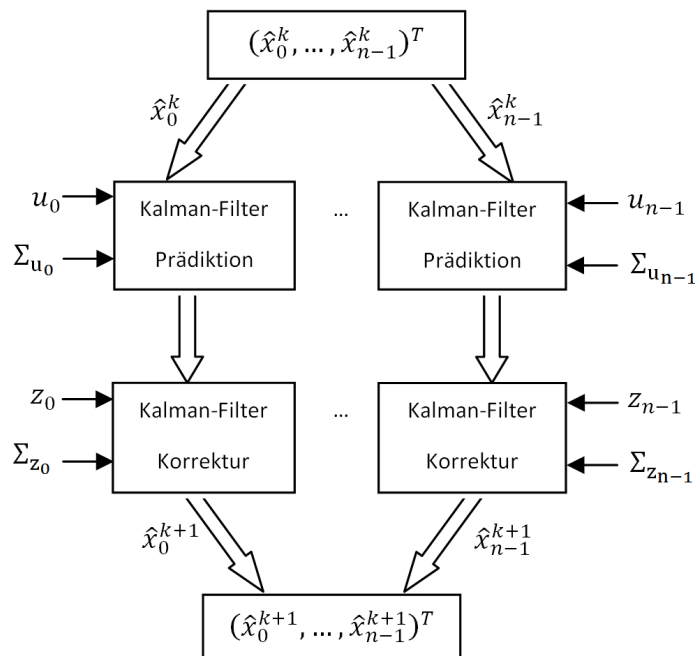


Abbildung 6.3.30: Prozess der Zustandsschätzung ohne Nebenbedingungen.

### 6.3.3 Projektion der Nebenbedingungen

Der vorhergehende Abschnitt beschreibt einen Algorithmus zur Schätzung des Zustands  $\hat{\mathbf{x}}^{k+1}$  des Systems ohne Berücksichtigung der Nebenbedingungen. Dies führt zu Fehlern in der Schätzung, da nicht das gesamte System betrachtet wird. Als Folge divergieren die Festkörper über die Zeit. Diese Ungenauigkeiten in der Prädiktion wurden aus Gründen der Effizienz in Kauf genommen. Die daraus resultierenden Fehler werden nun in einem Korrekturschritt behandelt, indem die Lösung ohne Nebenbedingungen

auf den verketteten Körper projiziert wird. Durch diesen Schritt der Nachstabilisierung wird die Lösung  $\hat{\mathbf{x}}_C^{k+1}$  für den Zustand unter den gegebenen Randbedingungen berechnet. Nach Gupta und Hauser [40] lässt sich dieser Zustand durch die Minimierung des folgenden quadratischen Systems berechnen:

$$\hat{\mathbf{x}}_C = \underset{x}{\operatorname{argmin}} (\mathbf{x} - \hat{\mathbf{x}}) \mathbf{W} (\mathbf{x} - \hat{\mathbf{x}}), \quad (6.3.13)$$

formuliert, mit

$$C_p(\mathbf{p}, \mathbf{q}) - \mathbf{d}_p = \mathbf{0}, \quad (6.3.14)$$

$$C_m(\mathbf{v}, \boldsymbol{\omega}) - \mathbf{d}_m = \mathbf{0}, \quad (6.3.15)$$

wobei  $\mathbf{W}$  eine positiv definite Gewichtematrix ist. Eine Lösung kann mittels der Methode der kleinsten Quadrate gefunden werden. Jedoch ist zu beachten, dass die Nebenbedingungen der Position und Bewegung nicht in jedem Fall gleichzeitig mit der Lösung ohne Nebenbedingungen vereinbar sind. Aus diesem Grund werden  $C_p$  und  $C_m$  im Folgenden getrennt voneinander betrachtet. Die Gleichung (6.3.13) wird schrittweise zunächst für  $C_p$  und anschließend für  $C_m$  gelöst.

### Nichtlineare Projektion der Positionsbedingungen

Aus Gründen der Allgemeingültigkeit werden die Nebenbedingungen der Position  $C_p$  als nichtlinear angenommen. Unter Vernachlässigung der Nebenbedingungen  $C_m$  der Bewegung, kann Lösung der Gleichung (6.3.13) mit Hilfe der Methode der Lagrange-Multiplikatoren  $\boldsymbol{\lambda}$  gefunden werden. Damit entsteht die Gleichung

$$\hat{\mathbf{x}}_C = \underset{x}{\operatorname{argmin}} (\mathbf{x} - \hat{\mathbf{x}}) \mathbf{W} (\mathbf{x} - \hat{\mathbf{x}}) - \boldsymbol{\lambda}^T (C_p(\mathbf{p}, \mathbf{q}) - \mathbf{d}_p), \quad (6.3.16)$$

mit

$$C_p(\mathbf{p}, \mathbf{q}) - \mathbf{d}_p = \mathbf{0}. \quad (6.3.17)$$

Somit kann das optimale  $\hat{\mathbf{x}}_C$  bestimmt werden, indem

$$\mathbf{W} (\mathbf{x} - \hat{\mathbf{x}}) - \mathbf{J}_p^T(\hat{\mathbf{x}}) \boldsymbol{\lambda} = \mathbf{0}, \quad (6.3.18)$$

$$C_p(\mathbf{x}) - \mathbf{d}_p = \mathbf{0}, \quad (6.3.19)$$

gelöst wird, wobei die Jacobi-Matrix  $\mathbf{J}_p(\mathbf{x})$  von  $C_p(\mathbf{p}, \mathbf{q})$  gegeben ist durch:

$$\mathbf{J}_p(\mathbf{x}) = \frac{\partial C_p(\mathbf{p}, \mathbf{q})}{\partial \mathbf{x}}. \quad (6.3.20)$$

Zur Lösung des nichtlinearen Problems, wird die Positionsbedingung um den aktuellen Arbeitspunkt linearisiert und iteriert, bis ausreichende Konvergenz erreicht wird. Dies liefert

$$\begin{aligned} C_p(\mathbf{x} + \Delta \mathbf{x}) &\approx C_p(\hat{\mathbf{x}}) + \mathbf{J}_p(\hat{\mathbf{x}}) \Delta \mathbf{x} - \mathbf{d}_p, \\ \Delta \mathbf{x} &= (\mathbf{x} - \hat{\mathbf{x}}). \end{aligned} \quad (6.3.21)$$

Ersetzen der nichtlinearen Nebenbedingung mit dem linearisiertes Gegenstück und Umformen der Gleichung (6.3.18) nach  $\Delta \mathbf{x}$  ergibt

$$\Delta \mathbf{x} = \mathbf{W}^{-1} \mathbf{J}_p^T(\hat{\mathbf{x}}) \boldsymbol{\lambda}. \quad (6.3.22)$$

Durch Einsetzen von  $\Delta \mathbf{x}$  in Gleichung (6.3.21) folgt

$$\mathbf{J}_p(\hat{\mathbf{x}}) \mathbf{W}^{-1} \mathbf{J}_p^T(\hat{\mathbf{x}}) \boldsymbol{\lambda} = \mathbf{d}_p - C_p(\hat{\mathbf{x}}). \quad (6.3.23)$$

Es fällt auf, dass die Komplexität der Lösung für Gleichung (6.3.23) nicht von der Dimension von  $\mathbf{x}$ , sondern vielmehr von der Dimension von  $\boldsymbol{\lambda}$  abhängt. Der Aufwand das Minimierungsproblem zu lösen hängt also nicht von der Anzahl der Festkörper ab, sondern wird von der Anzahl der Nebenbedingungen bestimmt.

Die Lösung kann maßgeblich durch die Wahl der Gewichtematrix  $\mathbf{W}$  beeinflusst werden [102]. Die Wahl von  $\mathbf{W} = \mathbf{M}$  führt nach Cline und Pai [17] zu der physikalisch plausibelsten Lösung, da in diesem Fall die Positionsänderungen mit dem geringsten Energieaufwand bevorzugt werden. Alternativ kann nach Gupta und Hauser [40] als Gewichtematrix die invertierte Zustands-Kovarianzmatrix  $\mathbf{P}_x$  gewählt werden:

$$\mathbf{W} = \mathbf{P}_x^{-1}. \quad (6.3.24)$$

Dies begünstigt eine Schätzung nach der *Maximum-Likelihood*-Methode, da die Kovarianz des Update-Fehlers minimiert wird. Da dies hinsichtlich der Komplexität zu bevorzugen ist, wird letzterer Lösungsweg eingeschlagen. Durch Ausnutzung der Tatsache, dass sich die partiellen Ableitungen schreiben lassen als

$$\frac{\partial C_p(\mathbf{x})}{\partial (\mathbf{p}, \mathbf{q})^T} = \mathbf{J}_p(\mathbf{p}, \mathbf{q}) \neq \mathbf{0}, \quad (6.3.25)$$

$$\frac{\partial C_p(\mathbf{x})}{\partial (\mathbf{v}, \boldsymbol{\omega})^T} = \mathbf{J}_p(\mathbf{v}, \boldsymbol{\omega}) = \mathbf{0}, \quad (6.3.26)$$

und die Kovarianz-Matrix  $\boldsymbol{\Sigma}_x$  aufgeteilt werden kann in

$$\boldsymbol{\Sigma}_x = \begin{pmatrix} \boldsymbol{\Sigma}_{(p,q)} & \boldsymbol{\Sigma}_{(p,q),(v,w)} \\ \boldsymbol{\Sigma}_{(v,w),(p,q)} & \boldsymbol{\Sigma}_{(v,w)} \end{pmatrix}, \quad (6.3.27)$$

wobei

$$\boldsymbol{\Sigma}_{(a,b),(c,d)} = E \left( \begin{pmatrix} \mathbf{a} - \hat{\mathbf{a}} \\ \mathbf{b} - \hat{\mathbf{b}} \end{pmatrix} \begin{pmatrix} \mathbf{c} - \hat{\mathbf{c}} \\ \mathbf{d} - \hat{\mathbf{d}} \end{pmatrix}^T \right), \quad (6.3.28)$$

die Kovarianz der Zufallsvektoren  $(\mathbf{a}, \mathbf{b})^T$  und  $(\mathbf{c}, \mathbf{d})^T$  ist, kann Gleichung (6.3.23) vereinfacht werden zu

$$\mathbf{J}_p(\mathbf{p}, \mathbf{q}) \boldsymbol{\Sigma}_{p,q} \mathbf{J}_p^T(\mathbf{p}, \mathbf{q}) \boldsymbol{\lambda} = \mathbf{d}_p - C_p(\hat{\mathbf{x}}). \quad (6.3.29)$$

Somit ergibt die Schätzung  $\hat{\mathbf{x}}_C^j$  des  $j$ -ten Iterationsschrittes zu

$$\hat{\mathbf{x}}_C^0 = \hat{\mathbf{x}}, \quad (6.3.30)$$

$$\hat{\mathbf{x}}_C^j = \hat{\mathbf{x}}_C^{j-1} + \begin{pmatrix} \boldsymbol{\Sigma}_{(p,q)} \\ \boldsymbol{\Sigma}_{(p,q),(v,w)} \end{pmatrix} \mathbf{J}_p^T(\mathbf{p}, \mathbf{q}) \boldsymbol{\lambda}. \quad (6.3.31)$$

### Projektion der linearen Bewegungsbedingungen

Nachdem nun die Einhaltung der Nebenbedingungen der Position erzwungen wurde, folgen die Nebenbedingungen der Bewegung. Günstigerweise handelt es sich in erster Linie um lineare Bedingungen, so dass  $C_m(\mathbf{v}, \boldsymbol{\omega})$  als Gleichung

$$\mathbf{J}_m(\mathbf{v}, \boldsymbol{\omega})^T - \mathbf{d}_m = \mathbf{0}, \quad (6.3.32)$$

ausgedrückt werden kann, welche linear in  $(\mathbf{v}, \boldsymbol{\omega})^T$  ist. Sollten beim Mehrkörpersystem hingegen nichtlineare Nebenbedingungen auftreten, so kann auch mit diesen analog zum vorhergehenden Abschnitt verfahren werden. Für lineare Bedingungen jedoch vereinfacht sich die Gleichung (6.3.13) zu

$$\boldsymbol{\Sigma}_{(\mathbf{v}, \boldsymbol{\omega})}^{-1} \begin{pmatrix} \mathbf{v} - \hat{\mathbf{v}} \\ \boldsymbol{\omega} - \hat{\boldsymbol{\omega}} \end{pmatrix} - \mathbf{J}_m^T \boldsymbol{\lambda}_{\mathbf{v}, \boldsymbol{\omega}} = \mathbf{0}, \quad (6.3.33)$$

$$\mathbf{J}_m \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{pmatrix} - \mathbf{d}_m = \mathbf{0}, \quad (6.3.34)$$

wobei  $\boldsymbol{\Sigma}_{\mathbf{v}, \boldsymbol{\omega}}$  die Kovarianz-Matrix der Bewegung ist. Weitere Auflösung der Gleichung (6.3.33) nach  $(\mathbf{v}, \boldsymbol{\omega})^T$  und Einsetzen in Gleichung (6.3.34) resultiert in

$$\mathbf{J}_m \boldsymbol{\Sigma}_{\mathbf{v}, \boldsymbol{\omega}} \mathbf{J}_m^T \boldsymbol{\lambda} = \mathbf{d}_m - \mathbf{J}_m^T \begin{pmatrix} \hat{\mathbf{v}} \\ \hat{\boldsymbol{\omega}} \end{pmatrix}. \quad (6.3.35)$$

Die Struktur dieser Gleichung ist äquivalent zu (6.3.23) aufgebaut. Folglich kann die Lösung nach dergleichen Herangehensweise gefunden werden.

#### 6.3.4 Evaluation und Diskussion

Um ein komplexes Mehrkörpermodell als Basis für die Zustandsschätzung eines humanoiden Roboters verwenden zu können und dennoch ausgereifte Filtermethoden, wie EKF und UKF anwenden zu können, wurde im vorhergehenden Kapitel ein Algorithmus entworfen, der zunächst Nebenbedingungen vernachlässigt. Hierdurch entstehen Fehler in der Schätzung, die anschließend wieder durch eine Projektion auf die kinematische Struktur des Roboters korrigiert werden. Dies kann aber ebenfalls nicht zu einer gänzlich fehlerfreien Lösung führen, da die Nebenbedingungen der Bewegung und Position sich im Allgemeinen nicht zeitgleich durch diese Projektion erfüllen lassen. Somit steht der Nachweis aus, dass der Vorteil einer besseren Modellierung der Struktur den Nachteil der fehlerhaften Projektion aufwiegt. Die Projektion der nichtlinearen Bedingungen erfordert eine iterative Lösung. Hierdurch erhöht sich der Rechenaufwand des Algorithmus soweit, dass die Geode CPU des NAO V3 diesen nicht in Echtzeit zeitgleich mit der NDD-WE und Sensorauswertung ausführen kann. Eine Evaluierung am realen Roboter kann daher nicht im Rahmen dieser Arbeit durchgeführt werden.

Erfahrungen mit dem Algorithmus auf mobiler Hardware legen aber den Schluss nahe, dass der Atom Prozessor des NAO V4 hierzu in der Lage sein sollte.

Als *Proof of Concept* wird der Algorithmus daher an einer Simulation des Roboters NAO erprobt. Hauschildt hat die Daten im Rahmen seiner Diplomarbeit [42] aufgezeichnet und sie sind bereits in [43] veröffentlicht. Die Sensoren werden im Bezug auf die Ergebnisse in Kapitel 3.3 modelliert, um ein möglichst realistisches Update-Verhalten zu erreichen. Zum Vergleich wird ein klassischer Ansatz gewählt, der basierend auf einem Einkörpermodell mittels eines UKF die Informationen des 3-Achsen-Beschleunigungssensor und des 2-Achsen-Gyrometer nutzt, um die Position des Oberkörpers zu schätzen. Der neu vorgestellte Algorithmus nutzt zusätzlich die Informationen der Winkelencoder der Motoren und die Kinematik als Nebenbedingung. Zur Filterung kommt ebenfalls ein UKF zum Einsatz, um die Vergleichbarkeit der Ergebnisse zu gewährleisten. Details über diesen sind [42] zu entnehmen. Aufgrund seiner Implementierung wird er im folgenden als *Quaternion MultiBody Unscented-Kalman-Filter* (QMB-UKF) bezeichnet.

Die Simulation der Experimente wird erneut in *SimRobot* [69] durchgeführt. Die Referenzwerte werden daher mittels der zeit-diskreten Festkörper-Simulation ODE<sup>16</sup> erstellt. Solche Mehrkörper-Simulationen basiert auf Algorithmen, denen ähnliche Ideen zugrunde liegen, wie dem hier vorgestellte Algorithmus. Folglich ist für die Evaluation sicherzustellen, dass beim Update des Kalman-Filters ein Prädiktionsfehler entsteht, und nicht beide Algorithmen eine identische Lösung produzieren. Die ODE verwendet zur Stabilisierung der Position einen Solver nach der Methode von Baumgarte [7]. Diese unterscheidet sich soweit von dem hier beschriebenen Ansatz, dass eine Abweichung von Prädiktion und Referenzwert sichergestellt ist. Zur Anschaulichkeit sind die Auswertungsdiagramme dieses Abschnitts additiv gestaffelt aufgetragen. Der akkumulierte Gesamtfehler kann so besser verglichen werden.

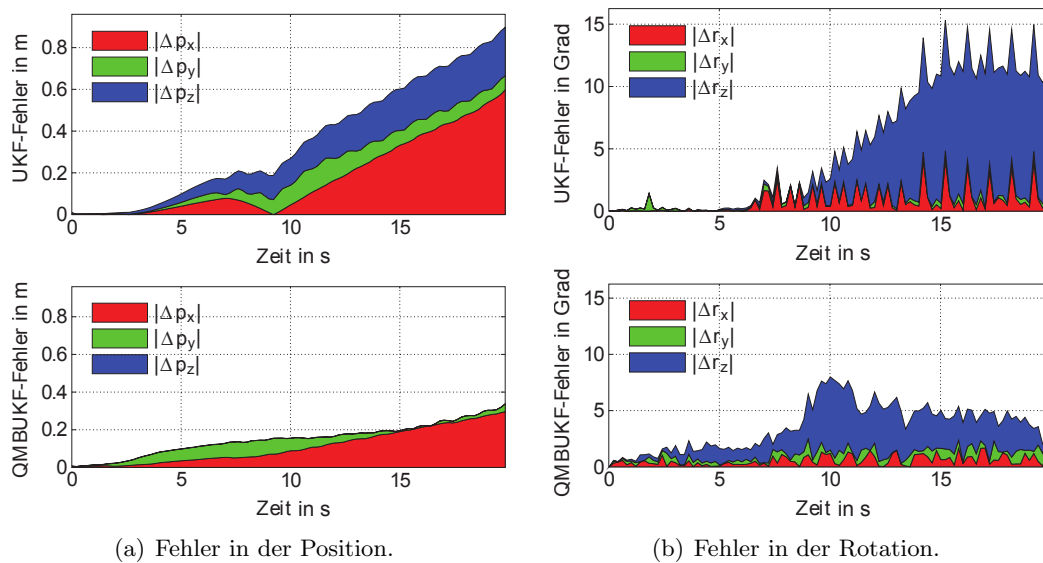
### Experiment 6.3.4.A - Lauf in $x$ -Richtung

Für ein erstes Experiment läuft der Roboter ohne Störung in  $x$ -Richtung des RzKS für eine Dauer von 20 Sekunden mit einer konstanten Geschwindigkeit von  $v_x = 0.07$  m/s. Zum Vergleich wird die Verfolgung der Trajektorie des CoM zum einen mit dem UKF und zum anderen mittels des QMB-UKF verglichen. Die Ergebnisse der Simulation sind in Abbildung 6.3.31 zu finden.

Eine eingehendere Betrachtung von Abbildung 6.3.31(a) offenbart bereits deutlich den Vorteil des Mehrkörpermodells. Da der QMB-UKF das Wissen über die Bewegungen der einzelnen Extremitäten als zusätzliche Informationsquelle nutzen kann, summiert sich der Fehler in der Prädiktion über die Zeit lediglich zu rund 35 cm auf. Für den UKF sind diese Bewegungen jedoch lediglich Störungen, so dass die Fehler in der Position mit der Dauer des Experiments merklich ansteigen. Zum Ende des Experiments erreicht der akkumulierte Fehler einen Wert von rund 90 cm, also ungefähr Faktor 2,5 größer als beim QMB-UKF. Deutlicher ist noch das Ergebnis des Vergleichs der Rotationsfehler,

---

<sup>16</sup><http://www.ode.org>

Abbildung 6.3.31: Experiment 6.3.4.A - Lauf in  $x$ -Richtung.

die in Abbildung 6.3.31(b) zu finden sind. Es verwundert nicht, dass in beiden Fällen der Rotationsfehler um die  $z$ -Achse dominiert, da dem Gyrometer des NAO diese Achse fehlt. Doch ist der QMB-UKF in der Lage dies zumindest teilweise auszugleichen und somit den Gesamtfehler um nahezu Faktor 3 bis 4 gegenüber dem klassischen UKF zu reduzieren.

### Experiment 6.3.4.B - Rotation um eigene Achse

In einem zweiten Experiment wird die Rotation um die eigene  $z$ -Achse untersucht. Hierzu läuft der Roboter NAO mittels eines konstanten *Walkrequest* mit einer Geschwindigkeit von rund  $20^\circ/\text{s}$ . Die Ergebnisse des Experiments 6.3.4.B sind in Abbildung 6.3.32 aufgetragen.

Bei der Auswertung ist zu beachten, dass die Rotationsachse die Achse des Roboters ist, die nicht direkt durch das Gyrometer gemessen werden kann. Als Folge zeigt sich in Abbildung 6.3.32(a), dass der Positionsfehler des UKF über die Dauer des Experiments kontinuierlich ansteigt, sobald sich der Roboter in Bewegung gesetzt hat. Das Wissen über die Struktur und Bewegung der Festkörper erweist sich in diesem Fall als enormer Vorteil. Die fehlende Information über die Rotation kann zumindest teilweise aus der Bewegung abgeleitet werden und der Fehler des QMB-UKF im Vergleich deutlich geringer gehalten werden.

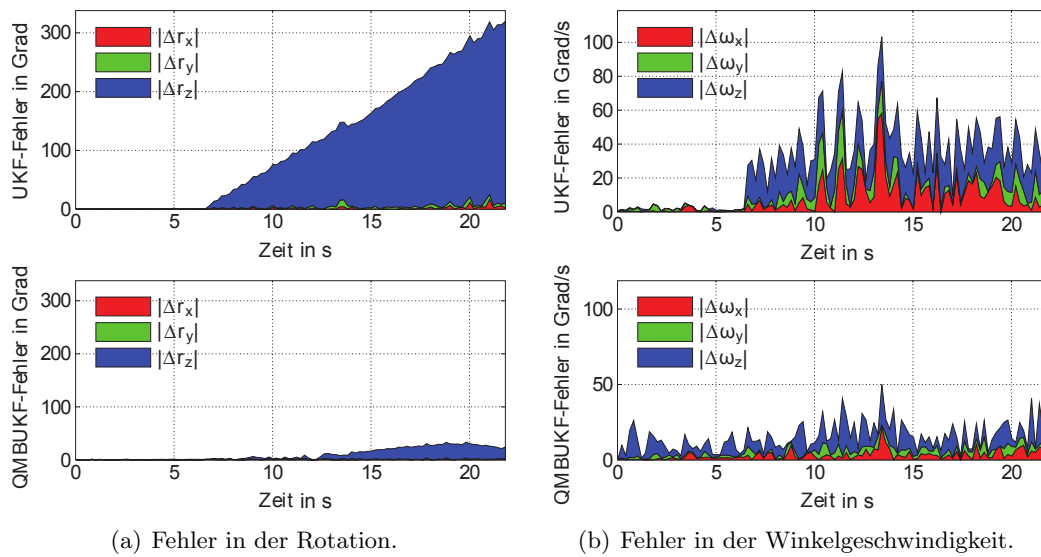


Abbildung 6.3.32: Experiment 6.3.4.B - Rotation um eigene Achse.

### Experiment 6.3.4.C - Unvorhergesehene Kollision

Da der Lauf mit konstanter Geschwindigkeit einen vorteilhaften Fall für das Kalman-Filter darstellt, untersucht das dritte Experiment die Reaktion auf eine unvorhergesehene Änderung der Bewegung. Diese Störung wird erreicht, indem der Roboter sich erneut in  $x$ -Richtung mit einer Geschwindigkeit von  $v_x = 0,07$  m/s bewegt. Diesmal kollidiert er jedoch während des Laufs mit einem Objekt und ist somit einem abrupten Wechsel der Geschwindigkeit ausgesetzt. Die Ergebnisse des Experiments finden sich in Abbildung 6.3.33.

Zunächst fällt auf, dass Abbildung 6.3.33(a) zu Beginn eine deutlich bessere Resultat des UKF gegenüber Experiment 6.3.4.A aufzeigt. Dies ist zunächst verwunderlich, da sich für diesen Teil der Versuchsaufbau nicht merklich verändert hat. Weil jedoch auch der QMB-UKF parallel leicht bessere Ergebnisse liefert, scheint es eine kleinere Änderung im Lauf zu geben, die vermutlich auf eine geänderte Anfangsposition des Experiments zurückzuführen ist. Da anscheinend bereits kleine Änderungen in der Störung der Messung merkliche Auswirkungen auf die Prädikation haben, kann dem Einkörper-Modell diese Verbesserung nicht positiv angerechnet werden, sondern bescheinigt diesem vielmehr eine mangelnde Robustheit. Eine weitere Betrachtung zeigt, dass die Kollision bei beiden Filtern zu einer Vergrößerung des Fehlers führt. Ohne ins Detail zu gehen, zeigt sich erneut, dass der QMB-UKF mit der Störung durch die größere Anzahl an Informationen besser zurecht kommt. Die Schätzung der Position und Bewegung zeigt signifikante Vorteile gegenüber dem UKF.



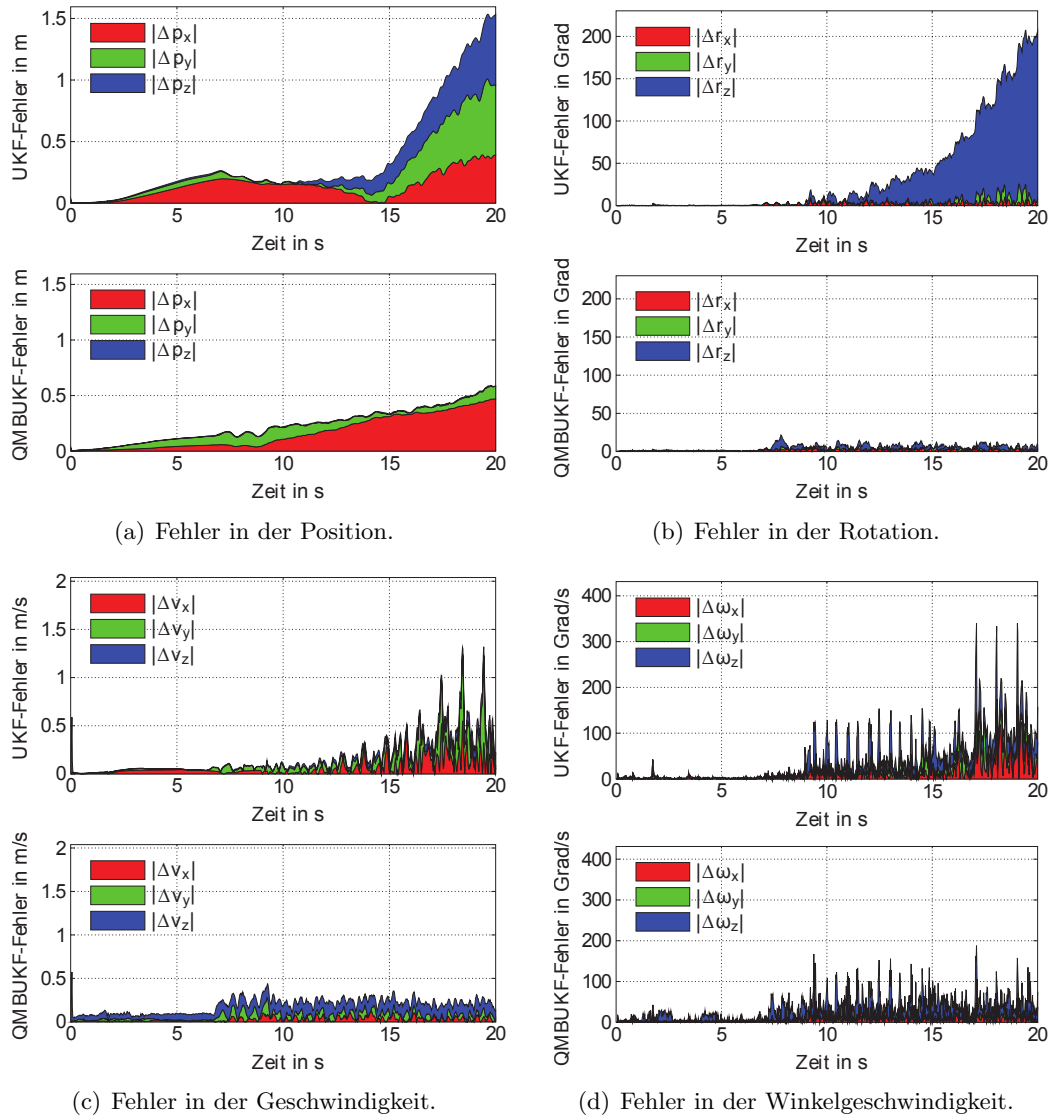


Abbildung 6.3.33: Experiment 6.3.4.C - Unvorhergesehene Kollision.



---

## Diskussion und Ausblick

---

Der Einsatz humanoider Roboter für unterschiedliche Aufgabengebiete rückt mehr und mehr in den Fokus des industriellen Interesses. Gleichzeitig kann jedoch das Problem der Bewegungssteuerung eines solchen Roboters keineswegs als gelöst gelten. Diese Arbeit erforschte daher Algorithmen, zum Einsatz propriozeptiver Sensorik, um die Präzision und Robustheit der Bewegungssteuerung zu verbessern.

Alle experimentellen Untersuchungen wurden mit dem Roboter NAO der Firma Aldebaran Robotics durchgeführt. Da diese Wahl die Interpretation der Ergebnisse maßgeblich beeinflusst, wurde dessen Hardware zunächst eingehend untersucht. Die Analyse der Sensoren zeichnet diese als repräsentativ für ihre Typenklasse aus. Keiner der Sensoren ist frei von Störeinflüssen, jedoch weist auch keiner eklatante Makel auf. Als problematisch für den Praxiseinsatz stellt sich lediglich die mechanische Aufhängung der Fußsohle heraus, da diese durch ihre Unzuverlässigkeit die FSR-Sensormessung stören kann. Im Rahmen der experimentellen Vergleiche konnte aber darauf geachtet werden, dass diese Problematik das Experiment nicht beeinflusst. Über alle Experimente betrachtet, erweist sich der Roboter NAO als geeignete Wahl für wissenschaftliche Untersuchungen mit dem Fokus der Bewegungssteuerung. Die Ergebnisse der Evaluation sind durchaus repräsentativ für einen Roboter vergleichbarer Größe. Der NAO ist mit allen wichtigen Sensoren für die Forschung im Bereich der Fortbewegung ausgestattet. Lediglich Kraft-Drehmoment-Sensoren für die Gelenke wären für die vergleichenden Experimente von Vorteil. Diese sind aber in Robotern dieser Größe und Preisklasse derzeit nicht erhältlich. Als problematisch für den Einsatz als Forschungsplattform erwies sich die Haltung Aldebarans, auf viele Hardwarekomponenten keinen direkten oder nur eingeschränkten Zugriff zu gewähren. Aldebaran hat für die Version NAO V4 teilweise Quellenoffenheit für ihr Framework NAOqi angekündigt, so dass die Problematik zukünftig weniger relevant sein dürfte.

Propriozeptive Sensorik kommt im Bereich der Bewegungsteuerung in erster Linie für die Ego-Modellierung des Roboters zum Einsatz. Ein Schwerpunkt liegt hierbei auf der Wahrnehmung des Gleichgewichtszustandes. Daher folgte nach eingehender theoretischer

scher Betrachtung eines Kriteriums für diesen, eine experimentelle Untersuchung. Der Fokus lag auf dem direkten Vergleich der Methoden, besonders im Hinblick auf Störungen der Messung und des Systems. Keines der untersuchten Messverfahren stellte sich als überlegen heraus. Sowohl aus den Messwerten der FSR- als auch der Beschleunigungssensoren kann der ZMP ausreichend genau bestimmt werden, um die Stabilität des Roboters in der Praxis zu überwachen. Die FSR-Sensoren sind robuster gegenüber Störungen, doch der Beschleunigungssensor ist in der Lage den ZMP über die gesamte Fußfläche und darüber hinaus auch den FZMP zu bestimmen. Diese Erkenntnis legt den Schluss nahe, dass die ZMP-Bestimmung von einer kombinierten Messung beider Sensoren durch eine entsprechende Fusion profitieren würde.

Zur Integration der Gleichgewichtsmessung in die Fortbewegung, wurde eine Referenz-*Walking-Engine* für den NAO entworfen. Der verwendete *close-loop* Ansatz basiert auf einem *Preview*-Regler mit linearisiertem Pendelmodell, wie er ursprünglich von Kajita et al. entwickelt wurde. Dieser wurde zur Stabilisierung um einen Beobachter erweitert, der für die Umsetzung der Sensorrückkopplung sorgt. Mit diesem erreicht der NAO Stabilität auch bei sehr hohen Laufgeschwindigkeiten. Aber auch alle anderen Kriterien für einen erfolgreichen Einsatz im Kontext des RoboCup werden durch den Regler erfüllt. Bei der praktischen Implementierung der WE bleiben Parameter offen, mit denen das Systemverhalten über den Regelungsalgorithmus hinaus beeinflusst werden kann. Diese wurden konkret am Beispiel des NAO diskutiert.

Zur Verbesserung der Ego-Modellierung wurden weiterführende Methoden erforscht, um die Eigenwahrnehmung des Roboters während der Bewegung zu präzisieren. Hierzu wurde der Einsatz optischer Sensoren zur Korrektur der Odometriemessung am NAO untersucht. Mit diesen können Fehler gemessen werden, die aus rutschender Bewegung der Füße resultieren. Die Experimente legen einen klaren Vorteil dieser Art von Sensorik für die Bewegungskontrolle nahe. Eine vollständige Integration in das System des Roboters musste aus Gründen der technischen Realisierbarkeit ebenso ausbleiben, wie eine Auswertung während einer dynamisch geregelten Laufbewegung.

Um die Ego-Modellierung über die Integration neuer Sensorik hinaus zu verbessern, wurde zudem ein Mehrkörper-Kalman-Filter entworfen, das Wissen über die kinematische Struktur des Roboters einbezieht. Dadurch können Sensorinformationen für einzelne Festkörper explizit in die Filterung integriert werden. Dies geschieht auf Basis einer separaten Schätzung einzelner Körper und anschließender Projektion auf die kinematische Struktur des Roboters. Durch die iterative Natur des Verfahrens konnte es bislang am realen Roboter nicht erprobt werden, da die Rechenleistung des NAO V3 hierzu nicht ausreicht. Untersuchungen in der Simulation zeigen aber einen erheblichen Vorteil des Verfahrens gegenüber eines klassischen Einkörper-Kalman-Filters auf.

Mit den in dieser Arbeit vorgestellten Algorithmen und Methoden lässt sich die Bewegungssteuerung des NAO soweit planen und überwachen, dass der Roboter praxistauglich im Kontext des RoboCups eingesetzt werden kann. Dies belegt auch der erlangte Vizeweltmeistertitel im Jahre 2011 durch das Team NAO Devils Dortmund. Der Einsatz außerhalb des RoboCups bringt jedoch weitere Herausforderungen für die Stabilität und Präzision der Bewegungen mit sich. Folglich kann die vom Autor begonnene Forschung mit dieser Arbeit auch nicht als vollständig abgeschlossen gelten.

---

Die gemachten Untersuchungen zeigen einige interessante Ansatzpunkte zur Weiterführung auf. Grundlage für die Regelung der Stabilität ist die Wahrnehmung des Stabilitätskriteriums. Die Auswertung der Experimente legt nahe, dass sich die Messung beim NAO bereits mit der vorhandenen Sensorik weiter verbessern ließe, wenn mittels Sensorfusion die Informationen der FSR- und Beschleunigungssensoren vereint würden. Könnte zudem durch Zugriff auf die Drehmomentrücklesung der Motoren eine weitere Stabilitätsmessung integriert werden, würde sich ein interessantes Forschungsfeld in diesem Bereich eröffnen. Nicht nur könnten so eine verbesserte Präzision und Robustheit der Messung erreicht, sondern auch der kritische *Single-Point-Of-Failure* durch Plausibilitätsvergleich verschiedener Sensorwerte eliminiert werden.

Weiteres Entwicklungspotential zeigt die Untersuchung der *Walking-Engine* auf. Die Diskussion der WE-Parameter belegt, dass einige entscheidenden Einfluss auf das Verhalten des Systems haben, aber bislang nicht eingehend untersucht wurden. Dies gilt besonders für die ZMP-Trajektorie, deren Verlauf das Führungsverhalten des Reglers unmittelbar beeinflusst. Eine Erforschung des optimalen Verlaufs dieser Stellgröße hat demnach Potential die Regelruhe des Systems weiter zu verbessern und die Stabilität zu erhöhen. Unerforscht ist zudem der Einfluss des *Stiffness*-Parameters. Die Auswirkung auf das Systemverhalten ist unklar, doch birgt der Parameter das Potential den Einfluss von flexiblen Gelenken, ähnlich dem ASIMO, zu untersuchen.

Das vorgestellte Konzept eines Beobachters für den *Preview*-Regler erweist sich als praxistauglich. Doch das zu grundlegende Pendelmodell ist durch seine Vereinfachung nicht in der Lage, alle Störeffekte abzubilden. Am realen NAO erwiesen sich diese aber teilweise als derart erheblich, dass Effekte auftreten, die nicht mehr durch Regelung ausgeglichen werden können. Untersuchungen von Urbann zeigen erste Erfolge, diese Effekte durch verbesserte Modellierung abzubilden, indem u. a. Flexibilitäten der Gelenke betrachtet werden. Eine Rückführung dieser Erkenntnis in das dem Regler zugrundeliegende Modell sollte demnach die Qualität der Regelung verbessern können. Es ist aber zu untersuchen, welchen Detailgrad kommende Roboter, wie der NAO V4, zur Laufzeit modellieren können und welchen Nutzen diese Veränderung für den Regler bringt.

Gleiches gilt für das vorgestellte fortgeschrittene Mehrkörper-Filter. Doch neben der Evaluierung des Konzeptes am realen NAO, steht die Untersuchung der Integration zusätzlicher Sensoren aus. MEMS-Sensoren sind durch die fortschreitende Entwicklung derart klein und günstig geworden, dass es ohne Weiteres möglich ist, mehrere zusätzliche in den Roboter zu integrieren. Denkbar wäre nicht nur die Anbringung von Sensoren in den Extremitäten des Roboters, sondern es könnten auch mehrere Sensoren verteilt im Torso angebracht werden, um so dessen Pose präziser bestimmen zu können.

Abschließend steht die vollständige Integration der optischen Sensoren in den NAO, oder einer vergleichbaren Plattform, aus. Dies bezieht sich zunächst auf den Einbau der Hardware. Doch auch die Einbeziehung der Sensorinformationen und deren gezielte Auswertung durch die Bewegungssteuerung des NAO birgt Gelegenheit für weitere Forschung. So könnte nicht nur reaktiv auf ungewollte Bewegung reagiert, sondern vielmehr durch Sensoren auf Eigenschaften des Bodens geschlossen und präventiv das Bewegungsmuster angepasst werden.



---

## Danksagungen

---

Ich bin den vielen Menschen zu Dank verpflichtet, die mir in der Zeit meiner Promotion zur Seite standen. Ohne ihre Unterstützung wäre diese nicht möglich gewesen. Daher möchte ich diese Gelegenheit nutzen, ihnen allen meinen Dank auszusprechen.

Ein spezielles Wort des Dankes möchte ich an meinem Doktorvater, Prof. Dr.-Ing. Uwe Schwiigelshohn, richten, der mir die Promotion erst möglich machte. Seine Unterstützung, besonders im Hinblick auf meine fachliche, berufliche und persönliche Weiterentwicklung, war eine große Hilfe in dieser Zeit. Die Jahre meiner Tätigkeit als wissenschaftlicher Mitarbeiter an seinem Lehrstuhl waren für mich eine große Bereicherung.

Weiterhin gilt mein Dank, meinen Kollegen und Mitarbeitern der Arbeitsgruppe Robotik am Institut für Roboterforschung, die durch Diskussion meine Forschung bereichert haben. Ihre Kollegialität hat die Arbeit immer wieder aufs Neue erleichtert und für Motivation gesorgt. Besonders hervorheben möchte ich hierbei Stefan Tasse und Oliver Urbann, mit denen ich zusammen an vielen meiner Veröffentlichungen gearbeitet habe.

Nicht unerwähnt bleiben sollen aber auch die zahlreichen Studenten, die ich in meiner Zeit als wissenschaftlicher Mitarbeiter begleiten durfte. Besonders die Teilnehmer der studentischen Projektgruppen haben über die Jahre entscheidend zum Erfolg des Teams NAO Devils Dortmund beigetragen und so durch ihre Arbeit am NDD-Framework wichtige Grundlagen geschaffen. Von den Studenten, deren Abschlussarbeiten ich betreut habe, möchte ich besonders Daniel Hauschildt, Maximilian Hegel, Matthias Fiegel und Cesaire Silchatom Fondjo erwähnen. Sie haben mit ihrer Arbeit wichtige Werkzeuge für die experimentelle Untersuchung geschaffen und die Diskussion mit ihnen war stets fruchtbar und brachte neue Ideen.

Darüber hinaus möchte ich meinem Freund Dr.-Ing. Karlheinz Ochs danken, der mir bei der Revision der schriftlichen Ausarbeitung mit Rat zur Seite stand.

Abschließend möchte ich aber noch meinen ganz besonderen Dank meiner Frau Tabea aussprechen, die mich in meinem Bestreben immer vorbehaltlos unterstützt hat. Ohne ihren Beistand wäre mir die Durchführung der Promotion so nicht möglich gewesen. Eine große Hilfe war darüber hinaus meine Tochter Sora, die mir mit ihrem Lächeln immer wieder die notwendige Motivation zum Abschluss der Arbeit geschenkt hat.





---

## Betreute Arbeiten im Kontext der Dissertation

---

Der Autor hat im Rahmen seiner Tätigkeit als wissenschaftlicher Mitarbeiter die folgenden studentischen Projektgruppen der Informatik betreut. Deren Teilnehmer waren maßgeblich an der Entwicklung des, im Rahmen dieser Arbeit verwendeten, NDD-Frameworks beteiligt.

PG 521 Adaptive dynamische Bewegungssteuerung und kooperative Weltmodellierung

PG 532 Verhaltensbasierte dynamische Bewegungssteuerung

Weiterhin wurden vom Autor folgende Abschlussarbeiten betreut, die im Kontext der Dissertation angesiedelt sind. Teilweise wurden Daten, die im Rahmen einer der Arbeiten aufgezeichnet wurden, als Grundlage für die Evaluierung dieser Dissertation verwendet. So dies geschehen ist, wurde an entsprechender Stelle darauf hingewiesen.

### Diplomarbeiten

Hauschildt, Daniel	Entwicklung eines Systems zur Bewegungs- und Kräfteschätzung eines humanoiden Roboters durch Festkörper-Simulation und Sensorfusion
Hegele, Maximilian	Automatisierte Kalibrierung der Odometrieinformationen zweibeiniger Roboter
Klagges, Daniel	Steuerung von zweibeinigen Laufrobotern mit Hilfe von Sensoren in den Füßen
Szypior, Patrick	Erlernen erweiterter Verhaltensnetzwerke zur Steuerung eines autonomen Roboters durch Demonstrationen eines Experten
Urbann, Oliver	Analyse und Kompensation laufbezogener Effekte zweibeiniger Roboter

### Bachelorarbeiten

Fiegel, Matthias	Sensorik zur Stabilitätsanalyse Humanoider Roboter
Fondjo, Cesaire	Bewegungsmessung humanoider Roboter mittels optischer Sensoren



---

## Bildnachweis

---

Der Wiederabdruck von geschützten Bildern im Rahmen dieser Arbeit erfolgte unter ausdrücklicher Genehmigung des jeweiligen Besitzer des Urheberrechtes. In alphabetischer Reihenfolge der Inhaber sind dies:

**AIST**

Abbildung 2.3.5

**Aldeabran Robotics**

Abbildung 2.3.7 | Abbildung 3.1.1 | Abbildung 3.1.4 | Abbildung 4.4.2

**ams AG**

Abbildung 3.3.11

**ASIMOV Robotics Pvt Ltd.**

Abbildung 2.3.8

**Honda**

Abbildung 2.3.4

**Logitech, Inc.**

Abbildung 6.2.10 | Abbildung 6.2.12

**Microsoft**

Abbildung 6.2.11



---

## Literaturverzeichnis

---

- [1] Aldebaran Robotics. *NAO User Guide, Version 1.10.52*, 2011. Available online.
- [2] Y. Aoustin, F. Plestan, and V. Lebastard. Experimental comparison of several posture estimation solutions for biped robot rabbit. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1270–1275, 2008.
- [3] W. F. Arnold and A. J. Laub. Generalized eigenproblem algorithms and software for algebraic Riccati equations. *Proceedings of the IEEE*, 72(12):1746–1754, 1984.
- [4] austriamicrosystems. AS5055 Low Power 12-Bit Magnetic Rotary Encoder, 2012. Revision 1.13.
- [5] D. Baraff. Linear-time dynamics using lagrange multipliers. In *SIGGRAPH*, 1996.
- [6] N. M. Barbour. Inertial navigation sensors. In *Advances in Navigation Sensors and Integration Technology*, chapter 2. Research and Technology Organisation, 2004.
- [7] J. Baumgarte. Stabilization of constraints and integrals of motion in dynamical systems. In *Computer Methods in Applied Mechanics and Engineering*, 1972.
- [8] J. Bender. Impulse-based dynamic simulation in linear time. *Computer Animation and Virtual Worlds*, 18(4-5):225–233, 2007.
- [9] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schäffer, D. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, and et al. The KUKA-DLR Lightweight Robot arm – a new reference platform for robotics research and manufacturing. *Joint 41th International Symposium on Robotics and 6th German Conference on Robotics*, pages 741–748, 2010.
- [10] A. Bonarini, M. Matteucci, and M. Restelli. Automatic error detection and reduction for an odometric sensor based on two optical mice. In *ICRA*, pages 1675–1680, 2005.
- [11] D. Borkmann, R. Tilgner, T. Reinhardt, S. Seering, and R. Fritzsche. Nao-Team

- HTWK 2010 SPL Team Description Paper. In J. Ruiz-del Solar, E. Chown, and P. G. Plöger, editors, *RoboCup 2010: Robot Soccer World Cup XIV*. RoboCup Federation, 2011.
- [12] A. G. Brooks. *Coordinating Human-Robot Communication*. PhD thesis, Massachusetts Institute of Technology, Cambridge, USA, 2006.
- [13] A. Burchardt, T. Laue, and T. Röfer. Optimizing particle filter parameters for self-localization. In J. R. del Solar, E. Chown, and P. G. Ploeger, editors, *RoboCup 2010: Robot Soccer World Cup XIV*, volume 6556 of *Lecture Notes in Artificial Intelligence*, pages 145–156. Springer, 2011.
- [14] R. Chalodhorn, D. B. Grimes, K. Grochow, and R. P. N. Rao. Learning to walk through imitation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2084–2090, June 2007.
- [15] J. E. Chestnutt, M. Lau, G. K. M. Cheung, J. Kuffner, J. K. Hodgins, and T. Kanade. Footstep planning for the honda asimo humanoid. In *ICRA*, pages 629–634. IEEE, 2005.
- [16] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, Cambridge, MA, June 2005.
- [17] M. B. Cline and D. K. Pai. Post-stabilization for rigid body simulation with contact and constraints. In *IEEE Intl. Conf. on Robotics and Automation*, 2003.
- [18] E. A. Coutsias and L. Romero. The Quaternions with Applications to Rigid Body Dynamics. Technical report, Sandia National Laboratories, 2004.
- [19] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1989.
- [20] S. Czarnetzki, M. Hebbel, and W. Nisticò. DoH!Bots Team Descripton for RoboCup 2007. In J. Ruiz-del Solar, E. Chown, and P. G. Plöger, editors, *RoboCup 2007: Robot Soccer World Cup XI*. Springer, 2007.
- [21] S. Czarnetzki, M. Hegele, and S. Kerner. Odometry correction for humanoid robots using optical sensors. In *RoboCup 2010: Robot Soccer World Cup XIV*, *Lecture Notes in Artificial Intelligence*, pages 48–59. Springer, 2011.
- [22] S. Czarnetzki, S. Kerner, and D. Klagges. Combining key frame based motion design with controlled movement execution. In J. Baltes, M. G. Lagoudakis, T. Naruse, and S. S. Ghidary, editors, *RoboCup 2009: Robot Soccer World Cup XIII*, pages 58–68. Springer-Verlag, Berlin, Heidelberg, 2010.
- [23] S. Czarnetzki, S. Kerner, and M. Kruse. Real-time active vision by entropy minimization applied to localization. In *RoboCup 2010: Robot Soccer World Cup XIV*, *Lecture Notes in Artificial Intelligence*, pages 266–277. Springer, 2011.

- [24] S. Czarnetzki, S. Kerner, and O. Urbann. Observer-based dynamic walking control for biped robots. *Robotics and Autonomous Systems*, 57(8):839–845, 2009.
- [25] S. Czarnetzki, S. Kerner, O. Urbann, M. Hofmann, S. Stumm, and I. Schwarz. Nao Devils Dortmund Team Report 2010. Technical report, Robotics Research Institute, TU Dortmund University, 2010.
- [26] S. Czarnetzki, Sören, and O. Urbann. Applying dynamic walking control for biped robots. In *Proceedings RoboCup 2009 International Symposium*, Graz, Austria, July 2009.
- [27] G. N. DeSouza and A. C. Kak. Vision for mobile robot navigation: A survey. *IEEE, TRANS. PAMI*, 24(2):237–267, 2002.
- [28] G. Dudek and M. Jenkin. *Computational principles of mobile robotics*. Cambridge University Press, 2000.
- [29] H. Fassbender and P. Benner. Initializing Newton’s method for discrete-time algebraic Riccati equations using the butterfly SZ algorithm. *Proceedings of the 1999 IEEE International Symposium on Computer Aided Control System Design*, pages 70–74, 1999.
- [30] R. Featherstone. Robot dynamics: Equations and algorithms. In *IEEE Int. Conf. Robotics and Automation*, pages 826–834, 2000.
- [31] L. Feng, J. Borenstein, and B. Everett. Where am I? Sensors and Methods for Autonomous Mobile Robot Localization. Technical Report UM-MEAM-94-21, The University of Michigan, 1994.
- [32] M. Fiegel. Sensorik zur Stabilitätsanalyse Humanoider Roboter. Bachelorarbeit, TU Dortmund, 2011.
- [33] C. S. Fondjo. Bewegungsmessung humanoider Roboter mittels optischer Sensoren. Bachelorarbeit, TU Dortmund, 2011.
- [34] J. Furusho and A. Sano. Sensor-based control of a nine-link biped. *International Journal of Robotic Research*, 9(2):83–98, Mar. 1990.
- [35] C. Glenday. *Guinness World Records 2005*. Jim Pattison Group, 2004.
- [36] B. Gökçe and H. L. Akin. Parameter optimization of a signal-based omnidirectional biped locomotion using evolutionary strategies. In J. Ruiz-del Solar, E. Chown, and P. G. Plöger, editors, *RoboCup 2010*, pages 362–373. Springer-Verlag, Berlin, Heidelberg, 2011.
- [37] A. Goswami. Foot rotation indicator (FRI) point: A new gait planning tool to evaluate postural stability of biped robots. In *IEEE International Conference on Robotics and Automation*, pages 47–52, 1999.

- [38] C. Graf and T. Röfer. A closed-loop 3D-LIPM gait for the RoboCup standard platform league humanoid. In E. Pagello, C. Zhou, S. Behnke, E. Menegatti, T. Röfer, and P. Stone, editors, *Proceedings of the Fifth Workshop on Humanoid Soccer Robots in conjunction with the 2010 IEEE-RAS International Conference on Humanoid Robots*, Nashville, TN, USA, 2010.
- [39] C. Graf and T. Röfer. A center of mass observing 3D-LIPM gait for the RoboCup standard platform league humanoid. In *RoboCup 2011: Robot Soccer World Cup XV*, Lecture Notes in Computer Science, pages 25 – 36. Springer Berlin / Heidelberg, 2012.
- [40] N. Gupta and R. Hauser. Kalman filtering with equality and inequality state constraints. Technical report, University of Oxford, February 2008.
- [41] K. Halvorsen, T. Söderström, V. Stokes, and H. Lanshammar. Using an extended Kalman filter for rigid body pose estimation. *Journal of Biomechanical Engineering*, 127(3):475–483, 2005.
- [42] D. Hauschildt. Entwicklung eines Systems zur Bewegungs- und Kräfteschätzung eines humanoiden Roboters durch Festkörper-Simulation und Sensorfusion. Diplomarbeit, TU Dortmund, 2009.
- [43] D. Hauschildt, S. Kerner, S. Tasse, and O. Urbann. Multi body Kalman filtering with articulation constraints for humanoid robot pose and motion estimation. In T. Röfer, N. M. Mayer, J. Savage, and U. Saranli, editors, *RoboCup 2011: Robot Soccer World Cup XV*, Lecture Notes in Computer Science, pages 338 – 348. Springer Berlin / Heidelberg, 2012.
- [44] K. Hauser, T. Bretl, K. Harada, and J. Latombe. Using motion primitives in probabilistic sample-based planning for humanoid robots. In *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2006.
- [45] M. Hebbel. *Evolutionäre Algorithmen zur Optimierung von Modellen für laufende Roboter*. PhD thesis, TU-Dortmund University, 2009.
- [46] M. Hebbel, I. Dahm, D. Fisseler, and W. Nistico. Learning fast walking patterns with reliable odometry information for four-legged robots, 2005.
- [47] M. Hebbel, R. Kosse, and W. Nisticò. Modeling and learning walking gaits of biped robots. In *Proceedings of the Workshop on Humanoid Soccer Robots of the IEEE-RAS International Conference on Humanoid Robots*, pages 40–48, December 2006.
- [48] M. Hegele. Automatisierte Kalibrierung der Odometrieinformationen zweibeiniger Roboter. Diplomarbeit, TU Dortmund, 2009.
- [49] S. Hesse. *Industrieroboterpraxis, Automatisierte Handhabung in der Fertigung*. Vieweg, Braunschweig/Wiesbaden, Germany, 1st edition, 1998.



- 
- [50] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of Honda humanoid robot. In *1998 IEEE International Conference on Robotics and Automation*, volume 2, pages 1321–1326, 1998.
- [51] J. Horn. *Zweidimensionale Geschwindigkeitsmessung texturierter Oberflächen mit flächenhaften bildgebenden Sensoren*. PhD thesis, Universität Karlsruhe, Karlsruhe, Germany, 2006. Schriftenreihe Institut für Mess- und Regelungstechnik Universität Karlsruhe (TH).
- [52] M. Husty, A. Karger, H. Sachs, and W. Steinhilper. *Kinematik Und Robotik*. Springer, 1997.
- [53] S.-H. Hyon and K. Fujimotoi. Invariant manifold of symmetric orbits for globally optimal biped locomotion robot control. In *9th IFAC Symposium on Robot Control (SYROCO)*, pages 619–626. Gifu, 2009.
- [54] G. Jochmann, S. Kerner, S. Tasse, and O. Urbann. Efficient multi-hypotheses unscented kalman filtering for robust localization. In T. Röfer, N. M. Mayer, J. Savage, and U. Saranlı, editors, *RoboCup 2011: Robot Soccer World Cup XV*, Lecture Notes in Computer Science, pages 222 – 233. Springer Berlin / Heidelberg, 2012.
- [55] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *ICRA*, pages 1620–1626. IEEE, 2003.
- [56] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa. A realtime pattern generator for biped walking. In *ICRA*, pages 31–37. IEEE, 2002.
- [57] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa. Biped walking pattern generator allowing auxiliary ZMP control. In *IROS*, pages 2993–2999. IEEE, 2006.
- [58] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. *Proceedings 2001 IEEERSJ International Conference on Intelligent Robots and Systems Expanding the Societal Role of Robotics in the the Next Millennium Cat No01CH37180*, 1(4):239–246, 2001.
- [59] S. Kajita, N. Koyachi, K. Kaneko, K. Yokoi, H. Arai, K. Komoriya, and K. Tanie. A high stability, smooth walking pattern for a biped robot. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 1, pages 65–71, 1999.
- [60] S. Kajita and K. Tani. Study of dynamic biped locomotion on rugged terrain - derivation and application of the linear inverted pendulum mode. In *IEEE International Conference on Robotics and Automation*, pages 1405–1411, Sacramento, CA, 1991.

- [61] T. Kanda, H. Ishiguro, M. Imai, and T. Ono. Development and evaluation of interactive humanoid robots. *Proceedings of the IEEE*, 92(11):1839–1850, Nov. 2004.
- [62] K. Kaneko, F. Kanehiro, M. Morisawa, K. Akachi, G. Miyamori, A. Hayashi, and N. Kanehira. Humanoid robot HRP-4 - Humanoid robotics platform with lightweight and slim body, 2011.
- [63] T. Katayama, T. Ohki, T. Inoue, and T. Kato. Design of an optimal controller for a discrete-time system subject to previewable demand. *International Journal of Control*, 41(3):677 – 699, 1985.
- [64] R. Keppler. *Zur Modellierung und Simulation von Mehrkörpersystemen unter Berücksichtigung von Greifkontakt bei Robotern*. Schriftenreihe des Instituts für Technische Mechanik. Univ.-Verlag Karlsruhe, 2006.
- [65] M. M. Kheirikhah, S. Rabiee, and M. E. Edalat. A review of shape memory alloy actuators in robotics. In J. Ruiz-del Solar, E. Chown, and P. G. Plöger, editors, *RoboCup 2010: Robot Soccer World Cup XIV*, pages 206–217. Springer-Verlag, Berlin, Heidelberg, 2011.
- [66] A. D. King. Inertial navigation - forty years of evolution. *GEC Review*, 13(3):140–149, 1998.
- [67] E. Kraft. A Quaternion-based Unscented Kalman Filter for Orientation Tracking. In *Proceedings of the Sixth International Conference on Information Fusion*, volume 1, pages 47–54, 2003.
- [68] T. Laue and M. Hebbel. Automatic Parameter Optimization for a Dynamic Robot Simulation. In L. Iocchi, H. Matsubara, A. Weitzenfeld, and C. Zhou, editors, *RoboCup 2008: Robot Soccer World Cup XII*, volume 5399 of *Lecture Notes in Artificial Intelligence*, pages 121–132. Springer, 2009.
- [69] T. Laue, K. Spiess, and T. Röfer. SimRobot - A General Physical Robot Simulator and Its Application in RoboCup. In A. Bredendfeld, A. Jacoff, I. Noda, and Y. Takahashi, editors, *RoboCup 2005: Robot Soccer World Cup IX*, number 4020 in *Lecture Notes in Artificial Intelligence*, pages 173–183. Springer, 2006.
- [70] P. Lin and D. E. Koditschek. *Legged Odometry from Body Pose in a Hexapod Robot*, pages 439–448. Springer, 2004.
- [71] Logitech. Darkfield laser tracking: The world is your mouse pad an innovation brief. White paper, Rev. 7/09, July 2009. Available online.
- [72] J. Lunze. *Regelungstechnik 2*. Springer-Verlag Berlin Heidelberg, 2006.
- [73] M. MacKay-Lyons. Central pattern generation of locomotion: a review of the evidence. *Phys Ther*, 82(1):69–83, Jan. 2002.

- 
- [74] N. Mao-Lin. Existence condition on solutions to the algebraic riccati equation. *Acta Automatica Sinica*, 34(1):85 – 87, 2008.
- [75] MathWorks Inc. *Matlab Documentation, Version 2012a*, 2012. Available online.
- [76] C. Meriçli and M. Veloso. Improving bipedwalk stability using real-time corrective human feedback. In J. Ruiz-del Solar, E. Chown, and P. G. Plöger, editors, *RoboCup 2010*, pages 194–205. Springer-Verlag, Berlin, Heidelberg, 2011.
- [77] Microsoft. Microsoft bluetrack-technologie. White paper, Rev. 6/08, June 2008. Available online.
- [78] K. Miura, M. Morisawa, S. Nakaoka, F. Kanehiro, K. Harada, K. Kaneko, and S. Kajita. Robot motion remix based on motion capture data towards human-like locomotion of humanoid robots. In *Humanoids*, pages 596–603, 2009.
- [79] M. Mori. Bukimi no tani [The uncanny valley]. *Energy*, 7(4):33–35, 1970.
- [80] M. Morisawa, K. Harada, S. Kajita, K. Kaneko, J. Solà, E. Yoshida, N. Mansard, K. Yokoi, and J.-P. Laumond. Reactive stepping to prevent falling for humanoids. In *Humanoids*, pages 528–534, 2009.
- [81] M. Morisawa, K. Harada, S. Kajita, S. Nakaoka, K. Fujiwara, F. Kanehiro, K. Kaneko, and H. Hirukawa. Experimentation of Humanoid Walking Allowing Immediate Modification of Foot Place Based on Analytical Solution. *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3989–3994, 2007.
- [82] J. Müller, T. Laue, and T. Röfer. Kicking a ball - modeling complex dynamic motions for humanoid robots. In J. Ruiz-del Solar, E. Chown, and P. G. Plöger, editors, *RoboCup 2010*, pages 109–120. Springer-Verlag, Berlin, Heidelberg, 2011.
- [83] K. Nagasaka, M. Inaba, and K. Nishtwaki. Generation of reactive stepping motion for a humanoid by dynamically stable mixture of pre-designed motions. In *IEEE SMC '99 Conference Proceedings*, volume 6, pages 902–907, 1999.
- [84] T. Niemueller, A. Ferrein, G. Eckel, D. Pirro, P. Podbregar, T. Kellner, C. Rath, and G. Steinbauer. *Providing Ground-truth Data for the Nao Robot Platform*, volume 6556 of *Lecture Notes in Artificial Intelligence*, pages 133–144. Springer, Singapore, 2011.
- [85] C. Ott, C. Baumg, J. Mayr, M. Fuchs, R. Burger, D. Lee, O. Eiberger, A. Albusch, M. Grebenstein, and G. Hirzinger. Development of a biped robot with torque controlled joints. *Electrical Engineering*, pages 167–173, 2010.
- [86] M. J. Powell, H. Zhao, and A. D. Ames. Motion primitives for human-inspired bipedal robotic locomotion: Walking and stair climbing. In *ICRA*, pages 543 – 549. IEEE, 2012.

- [87] J. Pratt, P. Dilworth, and G. Pratt. Virtual model control of a bipedal walking robot. In *IEEE Conference on Robotics and Automation*, pages 193–198, 1997.
- [88] M. H. Raibert. *Legged robots that balance*. Massachusetts Institute of Technology, Cambridge, MA, USA, 1986.
- [89] T. Röfer, T. Laue, J. Müller, O. Bösche, A. Burchardt, E. Damrose, K. Gillmann, C. Graf, T. J. de Haas, A. Härtl, A. Rieskamp, A. Schreck, I. Sieverdingbeck, and J.-H. Worch. B-human team report and code release, 2009.
- [90] T. Röfer, T. Laue, J. Müller, A. Burchardt, E. Damrose, A. Fabisch, F. Feldpausch, K. Gillmann, C. Graf, T. J. de Haas, A. Härtl, D. Honsel, P. Kastner, T. Kastner, B. Markowsky, M. Mester, J. Peter, and O. J. B-human team report and code release, 2010.
- [91] T. Röfer, T. Laue, M. Weber, H.-D. Burkhard, M. Jüngel, D. Göhring, J. Hoffmann, B. Altmeyer, T. Krause, M. Spranger, O. von Stryk, R. Brunn, M. Dassler, M. Kunz, T. Oberlies, M. Risler, U. Schwiegelshohn, M. Hebbel, W. Nisticò, S. Czarnetzki, T. Kerkhof, M. Meyer, C. Rohde, B. Schmitz, M. Wachter, T. Wegner, and C. Zarges. GermanTeam RoboCup 2005. Technical report, GermanTeam, 2005. Available online.
- [92] J. Rosen and B. Hannaford. Doc at a distance. *IEEE Spectrum*, 57:34–39, 2006.
- [93] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura. The intelligent ASIMO: system overview and integration. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and System*, volume 3, pages 2478–2483, September 2002.
- [94] P. Sardain and G. Bessonnet. Forces acting on a biped robot. center of pressure-zero moment point. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 34(5):630–637, 2004.
- [95] A. Schmitz, M. Missura, and S. Behnke. Learning footstep prediction from motion capture. In J. Ruiz-del Solar, E. Chown, and P. G. Plöger, editors, *RoboCup 2010*, pages 97–108. Springer-Verlag, Berlin, Heidelberg, 2011.
- [96] N. Shafii, S. Aslani, O. M. Nezami, and S. Shiry. Evolution of biped walking using truncated fourier series and particle swarm optimization. In J. Baltes, M. G. Lagoudakis, T. Naruse, and S. S. Ghidary, editors, *RoboCup 2009*, pages 344–354. Springer-Verlag, Berlin, Heidelberg, 2010.
- [97] N. Shafii, L. P. Reis, and N. Lau. Biped walking using coronal and sagittal movements based on truncated fourier series. In J. Ruiz-del Solar, E. Chown, and P. G. Plöger, editors, *RoboCup 2010*, pages 324–335. Springer-Verlag, Berlin, Heidelberg, 2011.
- [98] T. B. Sheridan. Three models of preview control. *IEEE Trans on Human Factors*

- in *Electronics*, 7(2):91–102, 1966.
- [99] A. P. Shon, K. Grochow, and R. P. N. Rao. Robotic imitation from human motion capture using gaussian processes. In *Proceedings of the IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2005.
- [100] B. Siciliano and O. Khatib. *Handbook of Robotics*. Springer, Berlin, Heidelberg, 2008.
- [101] R. Siegwart and I. R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. MIT Press, 2004.
- [102] D. Simon and T. L. Chia. Kalman filtering with state equality constraints. *Aerospace and Electronic Systems, IEEE Transactions on*, 38(1):128–136, Jan. 2002.
- [103] C. Stanton, E. Ratanasena, S. Haider, and M.-A. Williams. Perceiving forces, bumps, and touches from proprioceptive expectations. In T. Röfer, N. M. Mayer, J. Savage, and U. Saranlı, editors, *RoboCup 2011: Robot Soccer World Cup XV*, Lecture Notes in Computer Science, pages 301–312. Springer Berlin / Heidelberg, 2012.
- [104] STMicroelectronics. *LIS302DL - MEMS motion sensor*, 2008. Also available online.
- [105] J. Strom, G. Slavov, and E. Chown. Omnidirectional walking using ZMP and preview control for the NAO humanoid robot. In J. Baltes, M. G. Lagoudakis, T. Naruse, and S. S. Ghidary, editors, *RoboCup 2009*, pages 378–389. Springer-Verlag, Berlin, Heidelberg, 2010.
- [106] T. Suzuki. Trajectory planning of biped robot with two kinds of inverted pendulums. In *12th International Power Electronics and Motion Control Conference (EPE-PEMC)*, pages 396–401. IEEE, 2006.
- [107] S. Tasse, M. Hofmann, and O. Urbann. On sensor model design choices for humanoid robot localization. In X. Chen, P. Stone, L. E. Sucar, and T. V. der Zan, editors, *RoboCup 2012: Robot Soccer World Cup XVI*, Lecture Notes in Computer Science, page to appear. Springer Berlin / Heidelberg, 2013.
- [108] Tekscan. Flexiforce Standard Force & Load Sensors Model # A201, 2012. Rev C2811.
- [109] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.
- [110] R. Tilgner, T. Reinhardt, D. Borkmann, T. Kalbitz, S. Seering, R. Fritzsche, C. Vitz, S. Unger, S. Eckermann, H. Müller, M. Bellersen, M. Engel, and M. Wunsch. Team Research Report 2011 Nao Team HTWK Leipzig. Technical report, HTWK Leipzig, 2012.
- [111] P. Tipler, G. Mosca, M. Basler, R. Dohmen, C. Heinisch, A. Schleitner, and

- M. Zillgitt. *Physik: Für Wissenschaftler und Ingenieure*. Spektrum Lehrbuch. Spektrum Akademischer Verlag, 2009.
- [112] E. Torres and L. Garrido. Automated Generation of CPG-based Locomotion for Robot Nao. In *RoboCup 2011: Robot Soccer World Cup XV*, Lecture Notes in Computer Science, pages 382 – 392. Springer Berlin / Heidelberg, 2012.
- [113] O. Urbann, S. Kerner, and S. Tasse. Rigid and soft body simulation featuring realistic walk behaviour. In T. Röfer, N. M. Mayer, J. Savage, and U. Saranlı, editors, *RoboCup 2011: Robot Soccer World Cup XV*, Lecture Notes in Computer Science, pages 49 – 60. Springer Berlin / Heidelberg, 2012.
- [114] M. Vukobratović, B. Borova, and D. Surdilovic. Zero-Moment Point–Proper Interpretation and New Application. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, pages 237–244. IEEE, 2001.
- [115] M. Vukobratović and B. Borovac. Zero-moment point – Thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173, 2004.
- [116] M. Vukobratović and D. Juricić. Contribution to the synthesis of biped gait. In *Proc. IFAC Symp. Technical and Biological Problem on Control*. Erevan, 1968.
- [117] O. J. Woodman. An introduction to inertial navigation. Technical report, University of Cambridge, August 2007.
- [118] F. Xue, X. Chen, J. Liu, and D. Nardi. Real time biped walking gait pattern generator for a real robot. In J. Ruiz-del Solar, E. Chown, and P. G. Plöger, editors, *RoboCup 2010*. Springer-Verlag, Berlin, Heidelberg, 2011.
- [119] J. Yamaguchi, E. Soga, S. Inoue, and A. Takanishi. Development of a bipedal humanoid robot: Control method of whole body cooperative dynamic biped walking. In *ICRA*, pages 368–374, 1999.
- [120] Q. Yuan and I.-M. Chen. Human-like walking of humanoid robot based on biped kinematics and captured motion of human. In *13th IFToMM World Congress in Mechanism and Machine Science*, June 2011.
- [121] L. Zhang, C. Zhou, G. Yu, W. Hong, T. Yang, T. Hendra, C. Antonio, A. Calderon, R. E. Mohan, and P. K. Yue. Robo-Erectus Senior III (RESr-III): A Teen Size Humanoid Soccer Robot. In L. Iocchi, H. Matsubara, A. Weitzenfeld, and C. Zhou, editors, *RoboCup 2009: Robot Soccer World Cup XIII Proceedings*. RoboCup Federation, 2009.
- [122] Y. F. Zheng and J. Shen. Gait synthesis for the SD-2 biped robot to climb sloping surface. *IEEE Transactions on Robotics and Automation*, 6(1):86–96, 1990.
- [123] S. Zickler, T. Laue, O. Birbach, M. Wongphati, and M. Veloso. SSL-vision: The shared vision system for the RoboCup Small Size League. In J. Baltes, M. G. Lagoudakis, T. Naruse, and S. Shiry, editors, *RoboCup 2009: Robot Soccer World*

*Cup XIII*, volume 5949 of *Lecture Notes in Artificial Intelligence*, pages 425–436.  
Springer, 2010.