Technische Fakultät
Universität Bielefeld

Ausarbeitung zum Vortrag im Rahmen der Veranstaltung
"Humanoide Roboter" des Wintersemesters 2004/2005 an der
Universität Bielefeld

Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point

Markus Henschel (mhensche@techfak)

28. Februar 2005

Inhaltsverzeichnis

0.1 Kurzbeschreibung

Gegenstand des Vortrages ist im Kern ein Controller, der mit Hilfe einer Zukunftsreferenz (future-reference) unter externer Hinzugabe einer Zero-Moment Point (ZMP) Referenzvorgabe die Trajektorie des Massenschwerpunktes eines Zweibeinigen Roboters berechnet, so dass der dazu korrespondierende Roboter dazu in der Lage ist einen stabilen Gang zu vollführen. Es werden zunächst einige Modellierungsgrundlagen erörtert, die die Vorstellung der Bewegungen dieses Massenschwerpunktes vereinfachen und ein Mathematisches Modell eingeführt welches die Dynamik eines Roboters vereinfacht beschreibt und die Basis des Controllers darstellt.

0.2 Motivation

In der humanoiden Roboterforschung gibt es zwei große alternative Ansätze zur Generierung von Laufmustern, die beide in gewisser Weise problembehaftet sind, bzw viel Wissen über die zugrundeliegende Domäne erfordern, welches jedoch ggf. nicht ad-hoc verfügbar ist. Der ZMP-basierte Ansatz erfordert genaue Kenntnisse der Dynamik des korrespondierenden Roboters wie beispielsweise der Massen, der Massenverteilungen und des Massenschwerpunktes, hängt somit also stark von der Genauigkeit des zugrundeliegenden Modells ab. Zusätzlich würde dies bedeuten, dass Veränderungen am Roboter, welche Beispielsweise durch zusätzliche Massen zur Laufzeit enstehen könnten, immer wieder zu aufwendigen Neuberechnungen in der Dynamik führen würden, was einem autonomen und vor allem menschenähnlichen Soforthandeln widerspricht. Ein alternativer Ansatz beschäftigt sich nun ausschliesslich mit der vereinfachten Darstellung des Roboters durch ein inverses Pendel welches die Dynamik des Roboters beschreibt. Die Ungenauigkeiten in der Modellierung werden dort durch ein Feedback-System ausgeglichen.

Das Problem beim ZMP-Ansatz besteht darin, dass relativ lange ZMP-Referenzen vorgegeben werden müssen, damit das System in der Lage ist durch ggf. sogar Offline-Berechnung die Massenschwerpunktstrajektorie (Center of Mass - CoM) zu ermitteln. Dies führt dazu, dass bereits sehr viel Wissen über noch zukünftig kommenden Bewegungen vorhanden sein muss damit zur Jetzt-Zeit überhaupt eine Bewegung berechnet werden kann. Ein Segmentierungsansatz führt zu Interpolationsproblemen an den Segmentgrenzen und ist somit auch nur bedingt praktikabel, da es zu nicht flüssigen inhumanen Bewegungen führen kann.

Das vereinfachte Pendelmodell muss ggf. die Fussplatzierungen wechseln, sprich von den ZMP-Referenzvorgaben abweichen, was bei Aufgaben wie in (0.1) dargestellt wird, nicht akzeptabel ist. Die Lösung wäre ein Mischmodell, welches die Vorteile beider Ansätze zusammenführt und nach Möglichkeit die Probleme der seperaten Ansätze eliminiert. Dieser Ansatz wird im im Rahmen der Ausarbeitung verfolgt und erörtert.

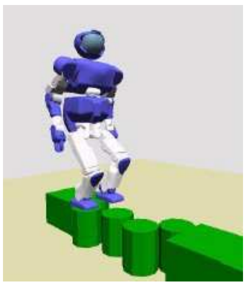

Abbildung 0.1: stepping stones

1 Grundlagen

1.1 3D-LIPM - 3D Linear Inverted Pendulum Mode

Das inverse Pendel dient als vereinfachte Dynamikgrundlage eines zweibeinigen Roboters. Abbildung (1.1) zeigt den generellen Aufbau dieses Pendels. Die Pendelbasis im Ursprung des aufgezeigten Koordinatensystems ist identisch mit dem ZMP und die Pendelspitze korrespondiert zum Massenschwerpunkt des Robotersystems. Ziel wird es im folgenden sein anhand von ZMP-Referenzvorgaben mittels einer Transformation die daraus resultierenden Beschleunigungen im Massenschwerpunkt zu ermitteln um daraus bei einem fortlaufenden System einen Bewegungstrajektorie des CoM zu bestimmen die notwendig ist um die Referenzvorgabe zu erfüllen.

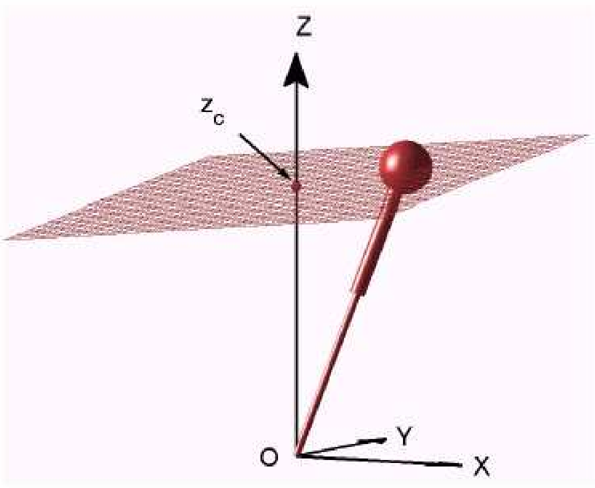

Abbildung 1.1: inverted pendulum

Die Position des CoM relativ zur Basis kann mit den folgenden Trigonometrischen Gleichungen ermittelt werden.

$$x = r \cdot \sin\theta_x, y = -r \cdot \sin\theta_y, z = r \cdot \sqrt{1 - (\sin\theta_x)^2 - (\sin\theta_y)^2} \tag{1.1}$$

Das Tupel (θ_x, θ_y, r) beschreibt den CoM eindeutig im Raum. Im folgenden bezeichnen wir mit (τ_x, τ_y, f) die zugehörigen Kräfte in die jeweiligen Bewegungsrichtungen. Wir können nun die Bewegungsgleichung des CoM aufstellen.

$$m \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = (J^T)^{-1} \begin{pmatrix} \tau_x \\ \tau_y \\ f \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -mg \end{pmatrix} \tag{1.2}$$

Die Inverse der Transponierten Jacobimatrix transformiert die Kräfte die in der Pendelbasis wirken in Kräfte die im CoM wirken; der zweite Term berücksichtigt geschickt die Erdanziehung. Unter Hinzunahme der konkreten Jacobimatrix

$$J = \begin{pmatrix} 0 & r\sin\theta_y & \sin\theta_y \\ -r\cos\theta_x & 0 & -\sin\theta_x \\ \frac{-r\cos\theta_x, \sin\theta_x}{\sqrt{1-(\sin\theta_x)^2-(\sin\theta_y)^2}} & \frac{-r\cos\theta_y, \sin\theta_y}{\sqrt{1-(\sin\theta_x)^2-(\sin\theta_y)^2}} & \sqrt{1-(\sin\theta_x)^2-(\sin\theta_y)^2} \end{pmatrix} \tag{1.3}$$

können nun die Bewegungsgleichungen in die relevanten Bewegungsrichtungen (x,y) ermittelt werden.

Diese Gleichungen ergeben sich zu

$$m(-z\ddot{y} + y\ddot{z}) = \frac{\sqrt{1 - (\sin\theta_x)^2 - (\sin\theta_y)^2}}{\cos\theta_x}\tau_x - mgy \tag{1.4}$$

in x-Richtung und

$$m(-z\ddot{x} + x\ddot{z}) = \frac{\sqrt{1 - (\sin\theta_y)^2 - (\sin\theta_x)^2}}{\cos\theta_y}\tau_y - mgy \tag{1.5}$$

in y-Richtung.

Das bisherige Pendelmodell erlaubt noch willkürliche Bewegungen, die auch zu willkürlichen Roboterbewegungen - speziell Neigungen - korrespondieren, von denen die meisten allerdings vom Roboter nicht anfahrbar sind. Zusätzlich ermöglicht eine Restriktion dieser Bewegungen noch eine wichtige Vereinfachung des Dynamikmodells. Wie schon in (0.1) gezeigt legen wir eine Bewegungeebene fest auf der sich die Pendelspitze also der CoM bewegen darf. Diese Bewegungsebene wird durch

$$z = k_x x + k_y y + z_c \tag{1.6}$$

beschrieben, wobei die k_i die Neigung der Ebene beschreiben und das z_c dem Schnittpunkt der Ebene mit der z-Achse entspricht. Bringt man nun diese Ebenenbeschreibung mit Gleichung (1.4) und (1.5) zusammen und formt sie ensprechend der Richtungsbeschleunigungen um ergeben sich die folgenden zwei Gleichungen:

$$\ddot{y} = \frac{g}{z_c}y - \frac{k_x}{z_c}(x\ddot{y} - \ddot{x}y) - \frac{1}{mz_c}\tau_x \tag{1.7}$$

$$\ddot{x} = \frac{g}{z_c}x - \frac{k_y}{z_c}(x\ddot{y} - \ddot{x}y) - \frac{1}{mz_c}\tau_y. \tag{1.8}$$

Im Falle einer horizontalen Ebene ($k_x = k_y = 0$) vereinfachen sich diese Formeln zu

$$\ddot{y} = \frac{g}{z_c}y - \frac{1}{mz_c}\tau_x \tag{1.9}$$

und

$$\ddot{x} = \frac{g}{z_c}x - \frac{1}{mz_c}\tau_y \tag{1.10}$$

Für den Fall einer nicht horizontalen Ebene kann man mit dem Aufstellen der Zusatzbedingung

$$\tau_x x + \tau_y y = 0, \tag{1.11}$$

die gleiche vereinfachte Dynamik erzwingen wie bei einer horizontalen Ebene. Durch geschickte Multiplikation der zweiten Ableitung der Ebenengleichung mit Teilumformungen der ursprünglichen Gleichungen kann man die folgende Formel ermitteln, die mit der Zusatzbedingung die Mittelteile von Gleichung (1.7) und (1.8) entfallen lässt:

$$x\ddot{y} - \ddot{x}y = \frac{-1}{mz}(\tau_x x + \tau_y y) \tag{1.12}$$

Aus der ZMP-Theorie stammend kann man die Position des Zero-Moment Points in Abhängigkeit der wirkenden Drehmomente bestimmten. Die folgenden Gleichungen beschreiben die x und y-Komponente des ZMPs.

$$p_x = -\frac{\tau_y}{mg} \tag{1.13}$$

$$p_y = \frac{\tau_x}{mg}. \tag{1.14}$$

Bringt man diese Beschreibung mit den Gleichungen (1.9) und (1.10) zusammen, so ergeben sich nun die Beschleunigungen des Massenschwerpunktes in Abhängigkeit der ZMP-Referenzvorgaben. Diese Gleichungen werden wir später als die ZMP-Gleichungen referenzieren.

$$\ddot{y} = \frac{g}{z_c}(y - p_y) \tag{1.15}$$

$$\ddot{x} = \frac{g}{z_c}(x - p_x) \tag{1.16}$$

Die Berechnung der ZMP-Trajektorie bei gegebenem Massenschwerpunkt ist einfach, Laufmustergenerierung entspricht jedoch dem inversen Problem. Im folgenden Betrachten wir die ZMP-Referenzen als gegeben (Beispielsweise von einem höhereren kognitiven System ermittelt) und berechnend Anhand dieser Vorgaben die CoM-Trajektorie, zuvor wollen wir aber noch kurz das Cart-Table-Modell als Vorstellungshilfe einführen.

1.2 Cart-Table Modell

Abbildung (1.2) zeigt ein direkt zu den ZMP-Gleichungen korrespondierendes Cart-Table Modell. Das Cart ($\widehat{=}$ CoM) bewegt sich auf der Tischoberfläche und der (massenlose) Tisch ist ab einer bestimmten Position des Carts nicht mehr dazu in der Lage sich aufrecht zu halten. Weist jedoch das Cart eine bestimmte Beschleunigung auf - so dass es schneller beschleunigt als die Massenträgheit wirkt - kann der Tisch noch eine Weile aufrecht stehen. Da auch in diesem Fall die Momente im ZMP Null sein müssen, ergibt sich folgende Gleichung, die der Gleichung (1.10) entspricht.

$$\tau_{zmp} = mg(x - p_x) - m\ddot{x}z_c = 0 \tag{1.17}$$

Für unseren Fall würden zwei solcher Carts benötigt um die x und y-Bewegung zu modellieren.

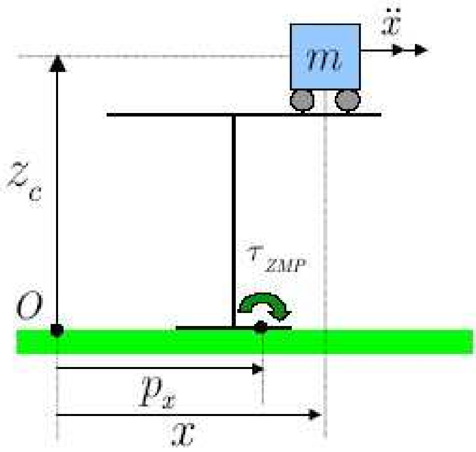

Abbildung 1.2: cart table

2 Preview Controller

2.1 Einleitung

Es gibt bereits verschiedene Ansätze nun Anhand der ermittelten ZMP-Gleichungen CoM-Trajektorien zu berechnen, unter anderem einen FFT-Ansatz nach Takanishi et al. oder einen Diskretisierungsansatz nach Kagami, Nishiwaki et al. die jedoch an einer gemeinsamen Schwäche leiden. Sie benötigen jeweils lange ZMP-Referenzbereiche, so dass - wie in der Motivation angedeutet - ggf. Offlineberechnung oder Segmentierungen notwendig wären, was wiederum zu Nichtautonomität und Nichtreaktivität oder Segmentierungsproblemen führt. Eine Alternative in der nun folgende Preview-Controller.

2.2 Grundidee

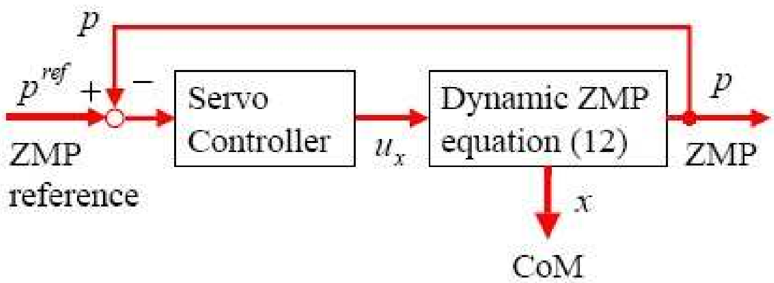

Abbildung 2.1: servo controller

Grundidee des Controllers (Abbildung (2.1)) ist zunächst aus gegebenen ZMP-Referenzvorgaben mit Hilfe eines Servo-Controllers Stellgrößen (u_x) für die ZMP-Gleichungen zu ermitteln, so dass der resultierende ZMP (ggf. modifiziert durch den Controller, deshalb Rückführung notwendig) der Vorgabe folgt. Die ZMP-Gleichungen liefern dann den CoM, bzw. im Verlauf die für die entsprechenden ZMP-Referenzen notwendige CoM-Trajektorie. In diesem Zusammenhang müssen die ZMP-Gleichungen in ein striktes dynamisches System übersetzt werden, welches die Basis für die mathematische Controllermodellierung liefert.

2.3 Anforderungen an den Controller

Abbildung (2.2) zeigt den Verlauf einer CoM-Trajektorie (grüne Linie) relativ zu einer gegebenen ZMP-Referenz. Bei 1.5s findet ein Supportwechsel bei den Füssen statt. Es ist erkennbar, dass der CoM-Trajektorienverlauf bereits vor dem Supportwechsel der Füsse verändert wer-

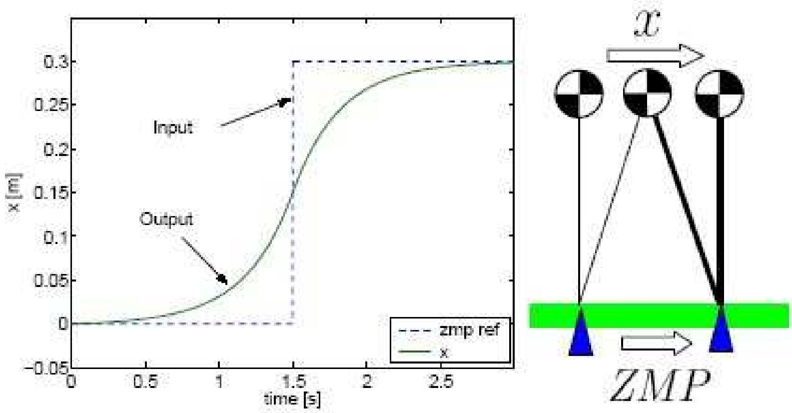

Abbildung 2.2: ZMP and CoM trajectory

den muss, d.h. die aktuelle Ausgabe des Controllers muss unter Hinzunahme von zukünftigen ZMP-Referenzen (future-input) berechnet werden. Dies ist keine Kausalitätsverletzung sondern entspricht einem Blick in die Zukunft wie beim Autofahren, wo man durch das sehen Beispielsweise noch kommender Hindernisse schon vor Ankunft Maßnahmen ergreifen kann um diesen auszuweichen.

2.4 Performance Index

Katayama et al. haben 1985 bereits einen Performance Index aufgestellt der die Performanz des Preview Servo Controllers beschreibt. Im Vorfeld musste dafür das Beschreibungssystem diskretisiert werden.

$$J = \sum_{i=k}^{\infty} (Q_e e(i)^2 + \Delta x^T(i) Q_x \Delta x(i) + R \Delta u(i)) \tag{2.1}$$

Der Index besteht aus drei wesentlichen Komponenten, das $e(k) \equiv p(k) - p^{ref}(k)$ entspricht dem Servo-Error; der Differenz der ZMP-Vorgabe und dem resultierenden ZMP. $\Delta x(k) \equiv x(k) - x(k-1)$ modelliert einen Zustandsvektor der Auskunft über die Zustandsveränderung (feedback) in der Welt zwischen zwei Diskretisierungsschritten gibt. $\Delta u(k) \equiv u(k) - u(k-1)$ ist die in der Grundidee skizzierte Stellgröße mit der die ZMP-Gleichungen den CoM berechnen. Die $Q_e, R > 0$ sind skalare Gewichte, das Q_x eine nicht negativ definite symmetrische Matrix. Mit diesen drei Parametern lässt die Wichtigkeit der drei Komponenten im Gesamtzusammenhang skalieren.

2.5 Controller

Ziel ist es nun, diesen Index zu minimieren, was der folgende Controller optimal erreicht.

$$u(k) = \underbrace{-G_i \sum_{i=0}^{k} e(k)}_{\text{Servo Error}} - \underbrace{G_x x(k)}_{\text{Zustandsfeedback}} \underbrace{- \sum_{j=1}^{N} G_p(j) p^{ref}(k+j)}_{\text{future reference}} \tag{2.2}$$

Die $G_k, k \in \{i, x, p\}$ charakterisieren die Gewinne die aus den jeweiligen Komponenten zum Gesamtergebnis beitragen. Sie entsprechen also nur Gewichtungen die aus den Gewichten des Performance-Indexes J und den Systemparametern der Diskretisierungsansatzes ermittelt werden. Dieser Controller minimiert nun J für N Zukunftsschritte, wobei zu beachten ist, dass der Gewinn (engl. gain) der future-reference Komponente exponentiell abfällt wie in Abbildung (2.3) aufgezeigt. Nach zwei Sekunden ist der Gewinn bereits zu vernachlässigen

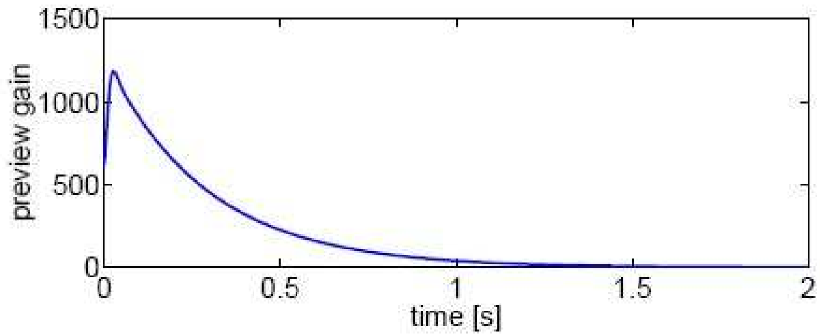

Abbildung 2.3: preview gain

und schon nach einer Sekunde ist er auf einem sehr geringen Niveau. Es reicht also eine relativ kurze future-reference des ZMP. Damit sind die Probleme der Segmentierung, der Offline Berechnung und der dazugehörenden Schwierigkeiten bereits gelöst.

2.6 Preview Vergleich

Im folgenden werden zwei Ergebnisse des Controllers mit verschiedenen Preview-Zeiten verglichen um die Funktionsfähigkeit der Modellierung zu zeigen. Bei einer Preview Zeit von 1.6s

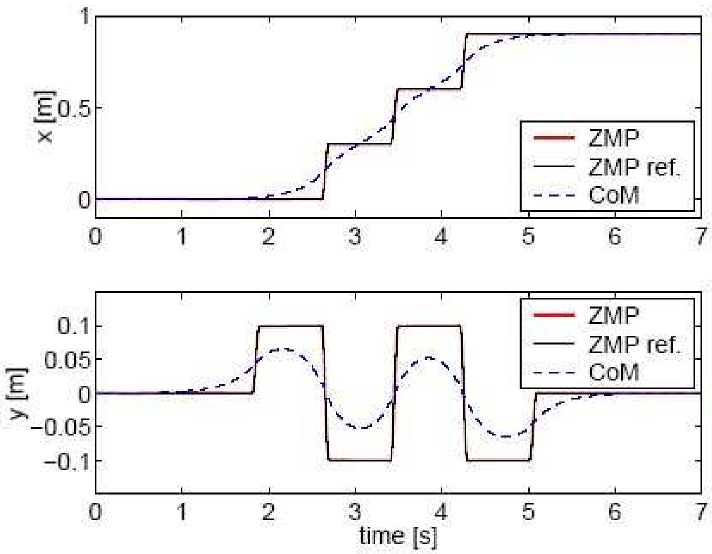

Abbildung 2.4: 1.6s preview time

(Abbildung (2.4)) wird der ZMP-Referenz in beide Bewegungsrichtungen (x,y) gut gefolgt. Die Performance ist optimal und das Systemverhalten ist stabil. Bei einer Preview Zeit von

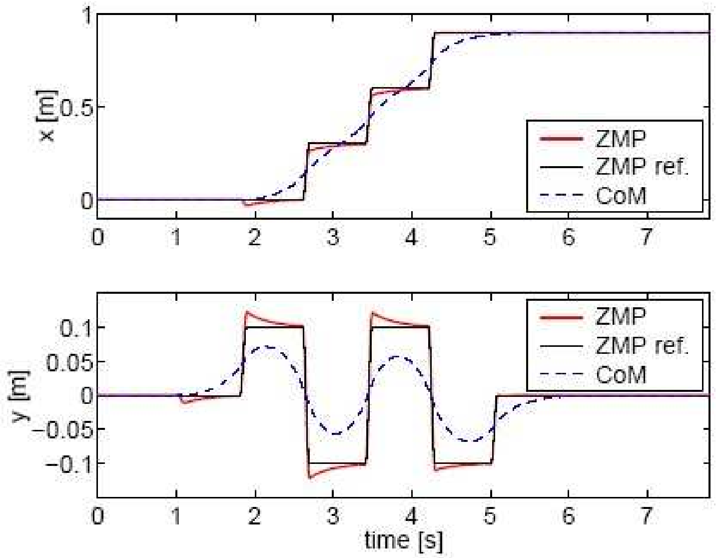

Abbildung 2.5: 0.8s preview time

0.8s (Abbildung (2.5)) erzeugt der Controller einen Overshoot in x-Richtung und einen Undershoot in y-Richtung. Die Performance ist relativ schlecht, das System aber aufgrund des Zustandsfeedbacks immer noch stabil.

2.7 Vereinfachung komplexer Robotersysteme

Zusätzlich zu dem Gewinn den wir durch den Preview-Controller erhalten haben, dass nur noch geringe ZMP-Referenzbereiche notwendig für CoM-Trajektorienberechnung sind, besteht immer noch das Problem, dass wir die komplexe Dynamik des Roboters, die im Bewegungsverlauf und unter Fremdeinwirkung (Lasten) ständig variiert, bisher noch aufwendig Neuberechnen müssen. Die folgende Darstellung zeigt, dass der Preview-Controller zusätzlich noch dazu in der Lage ist, Modellierungsfehler in der Dynamik auszugleichen. Wir können dann den Massenschwerpunkt durch (wie in der folgenden Gegenüberstellung verwendet) das Beckensegment approximieren und erreichen damit immer noch äußerst zufriedenstellende Ergebnisse. Die Aufgetretenen maximalen Fehler in x-Richtung betragen 2.3cm und in y-

Als Basis dienten hierfür die physischen Daten des HRP-2P (Abbildung (2.6)) und als Softwaresystem wie im gesamten Zusammenhang die zugehörige OpenHRP-Software. Abbildung (2.7) zeigt die berechnete ZMP-Trajektorie im Vergleich mit dem komplexen Dynamikmodell des HRP-2P ohne die Preview Komponente des Controllers.

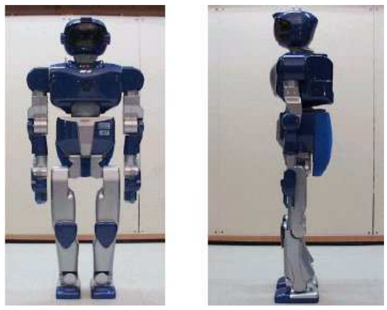

Abbildung 2.6: HRP-2P

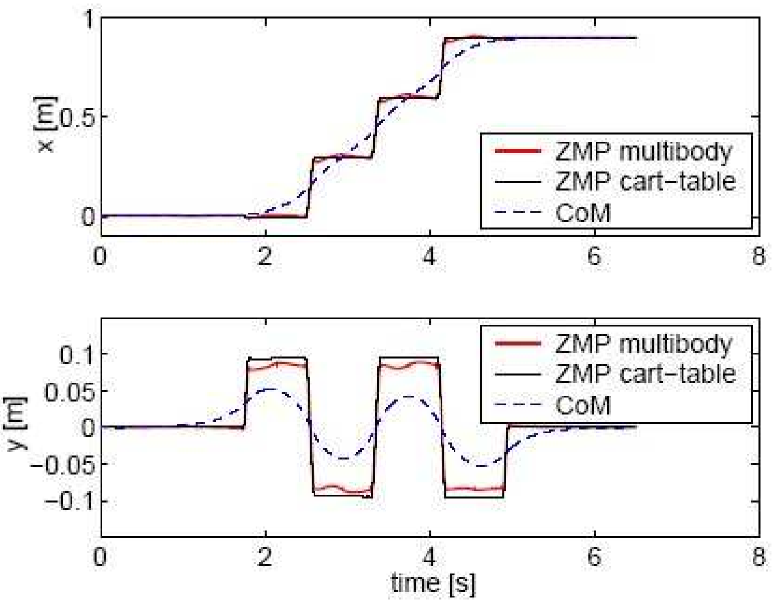

Abbildung 2.7: Approximation ohne Preview-Komponente

Richtung 1.6cm. Das erzielte Ergebnis könnte unter Umständen nicht mehr zufriedenstellend sein. Anders bei den gleichen Gegebenheiten unter Verwendung der Preview-Komponente mit einer Preview-Zeit von 0.75s. Die maximalen Fehler liegen mit 1.2cm (x) und 0.4cm (y)

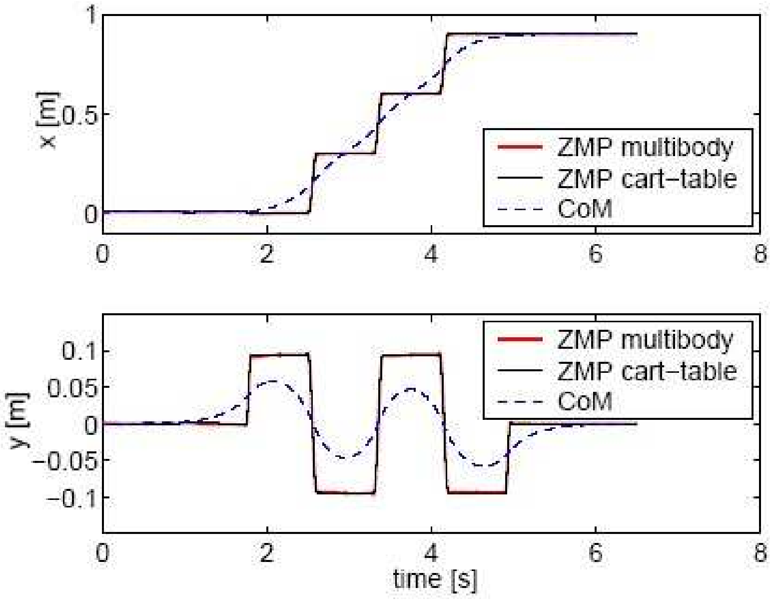

Abbildung 2.8: Approximation mit Preview-Komponente

deutlich unter der vorherigen Fehlern und sind auch wie in Abbildung (2.8) erkennbar nicht dauerhaft vorhanden wie in Abbildung (2.7). Der Controller ist somit also in der Lage Modellierungsfehler die nicht zu stark von dem tatsächlichen Massenschwerpunkt abweichen durch die Preview-Komponente zu kompensieren.

2.8 Walking on spiral stairs (Beispiel)

In einer Beispielmodellierung wurde der Versuch unternommen den HRP-2P auf einer Wendeltreppe mit gleichmäßiger Stufenhöhe aufsteigen zu lassen. Die ZMP-Referenzen wurden wie gefordert vorgegeben und sind in Abbildung (2.9) Beispielhaft dargestellt. Abbildung (2.10) zeigt vier Momentaufnahmen der OpenHRP 3D-Physikengine die die Lauffähigkeit des Algorithmusses unter Beweis stellen.

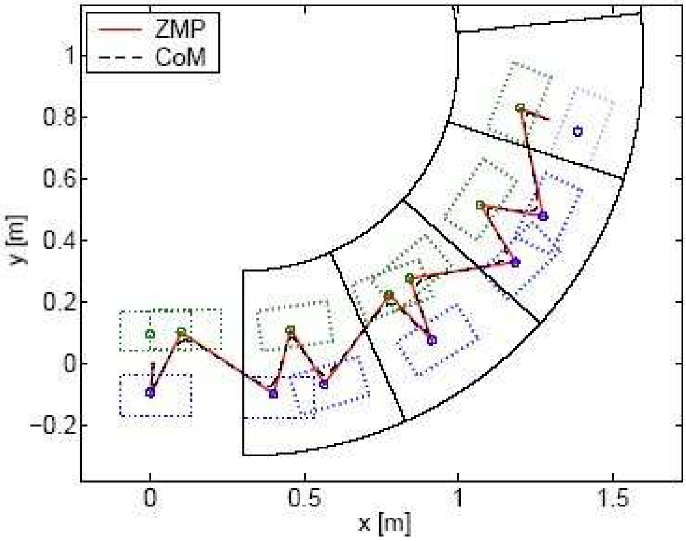

Abbildung 2.9: ZMP-Referenzen

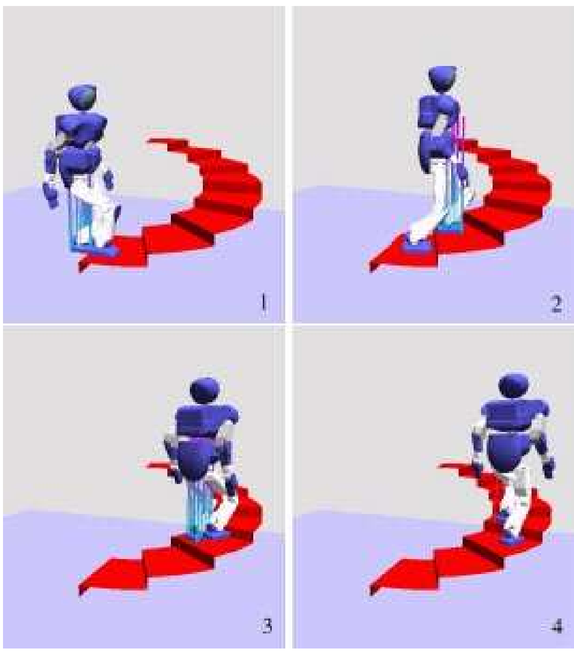

Abbildung 2.10: Momentaufnahmen des Laufvorgangs

Literaturverzeichnis

[01] Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point. ⟨Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kenzuke Harada, Kazuhito Yokoi and Hiroshia Hirukawa⟩ (2003)

[02] The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking generation. ⟨Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kazuhito Yokoi and Hiroshisa Hirukawa⟩ (2001)

[03] Zero-Moment Point - Thirty Five Years of its Life. ⟨Miomir Vukobratovic and Branislav Borovac⟩ (2004)