ZERO-MOMENT POINT WALKING CONTROLLER FOR HUMANOID
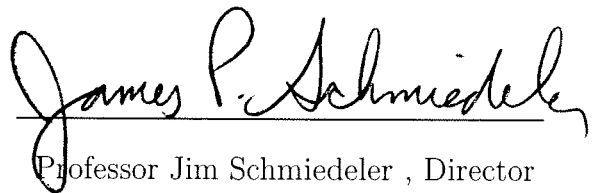
WALKING USING DARWIN-OP

A Thesis

Submitted to the Department of Aerospace

and Mechanical Engineering of the University of Notre Dame

in Partial Fulfillment of the Requirements

for the Degree of

Bachelor of Science in Mechanical Engineering

by

Joseph Rudy

Professor Jim Schmiedeler , Director

Undergraduate Program in

Notre Dame, Indiana

April 2014

ZERO-MOMENT POINT WALKING CONTROLLER FOR HUMANOID

WALKING USING DARWIN-OP

Abstract

by

Joseph Rudy

The design and implementation of gait generation for stable steady-state humanoid walking is a complex problem due to the high dimensional control space as well as the inherently unstable motion. Moreover, humanoids must have a fast walking gait that models a human gait. This thesis aims to implement a stable walking controller using the computed Zero-Moment Point (ZMP) method with a simplified mass model of a humanoid robot, namely a cart-table model. From this simplified model, a center of mass (CoM) trajectory can be obtained and tracked using the stance foot to achieve the desired ZMP. A swing leg trajectory can be obtained through walking parameters such as stride length, stride width, and ground clearance. Using this controller, a more human-like gait can be achieved in a humanoid for faster steady-state walking.

CONTENTS

α	fraction of time spent in initial and terminal swing
b	conversion constant from time to stride length
$\mathbf{c}$	center of mass
${}^{j}C^{i}$	describes rotation from outer body, i, to inner body, j
$\mathbf{c}_{N/\mathcal{F}}$	center of mass relative to the $\mathcal{F}$ frame
c_{g}	ground clearance in [m]
$\mathbf{d}$	absolute value of difference between $\mathbf{p}$ and $\mathbf{f}$
δ	ZMP reference offset
$\Delta\mathbf{p}$	${}^{5}\mathbf{p}^{swing} - {}^{5}\mathbf{p}^{hip}$
η	ratio of stride width to stride length
$\mathbf{f}$	location of the $\mathcal{F}$ frame relative to the N frame
$\mathcal{F}$	local foot reference frame
f_{l}	foot length, 0.104 m
f_{w}	foot width, 0.066 m
$\mathbf{f}^{ref}$	foot reference locating the location of previous swing foot
G_{I}	integral gain for cart-table controller
G_{p}	preview gains for cart-table controller
G_{x}	state feedback gains for cart-table controller
h_{p}	height of 3D pendulum in CoM tracking
I_{p}	$p \times p$ identity matrix

K_{bp}	proportional gain for body pitch controller
K_{kf}	proportional gain for knee flexion controller
N	global inertial reference frame
ν	CoM offset
$\mathbf{p}$	location of ZMP
$\mathbf{p}^{hip}$	location of swing hip relative to $\mathcal{F}$ frame
$\mathbf{p}^{ref}$	ZMP reference for the desired ZMP
$\mathbf{p}^{swing}$	vector pointing from stance to swing foot
$\mathbf{s}$	foot path vector
s_w	stance width, 0.074 m
$\mathbf{S}$	stride vector
T	step period
T_s	sampling time, 8 ms
T_{DS}	period of double support
T_{SS}	period of single support
z_c	constant height of the cart off the ground in cart-table model

ACKNOWLEDGMENTS

CHAPTER 1

INTRODUCTION

The design and implementation of gait generation for stable steady-state humanoid walking is a complex problem due to the high dimensional control space as well as the inherently unstable motion. Moreover, humanoids must have a fast walking gait and a gait that models the human gait. Many humanoid walking controllers use the center of mass (CoM) to ensure static stability throughout the gait. Static stability implies that the projection of the CoM onto the support surface is inside the support polygon of the stance foot or feet throughout the gait. These CoM walking controllers have a very long time period between steps, or a slow cadence, and do not model human walking very well. However, a walking controller which can provide dynamic stability allows for a faster cadence and a more human-like gait. Dynamic stability implies that a point, called the Zero-Moment Point (ZMP), is inside the support polygon throughout the gait.

In order to achieve this dynamic stability, approaches have been developed based on either forward dynamics or the ZMP. This thesis explores the use of the ZMP method for a walking controller. Most of these types of walking controllers use simplified models to control the ZMP to achieve a known reference value. The intent of this thesis is to implement a ZMP walking controller in the humanoid robot DARwIn-OP (Dynamic Anthropomorphic Robot with Intelligence - Open Platform) using a simplified cart-table model and a preview controller to control the ZMP and achieve a stable walking gait.

1.1 ZMP FOR BIPEDAL ROBOTS

The Zero-Moment Point, or ZMP, is the point on the surface of the foot where a resultant force R can replace the force distribution shown in Fig. 1.1 [8]. Mathematically, the ZMP can be calculated from a group of contact points $\mathbf{p}_i$ for $i = 1, \ldots, N$ with each force vector $\mathbf{f}_i$ associated with the contact point,

$$ZMP = \frac{\sum_{i=1}^{N} \mathbf{p}_i f_{iz}}{\sum_{i=1}^{N} f_{iz}} = \frac{\sum_{i=1}^{N} \mathbf{p}_i f_{iz}}{f_z}. \tag{1.1}$$
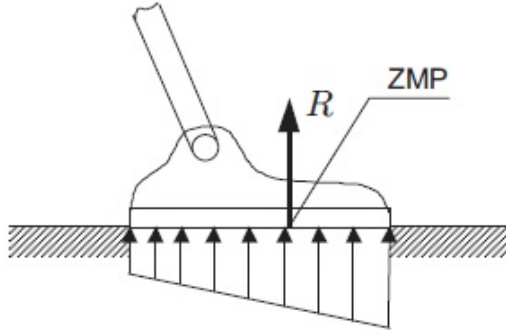
Figure 1.1. Definition of Zero-Moment Point [7]

In this definition, the ZMP can never leave the support polygon. If the floor is assumed horizontal, the torque reduces to

$$\tau_x = \tau_y = 0 \tag{1.2}$$

at the ZMP. Further derivations for the torque at the ZMP can be found in Appendix A.1. This definition is useful when pressure sensors are attached to the feet. With

2

these sensors, the center of pressure can be calculated on the feet, and the ZMP can be directly measured. The humanoid robot DARwIn-OP has pressure sensor foot attachments that can be bought separately. However, these sensors were not used, so a model-based method was employed to calculate the ZMP.

1.2   COMPUTED ZMP

Instead of calculating the resultant force from the force distribution, the position of the ZMP can be calculated from the dynamics of the system using Newton's Second Law when the mass properties and motion of the robot are known [7]. To illustrate the use of computed ZMP, the cart-table model used in the controller is presented in Fig. 1.2. This model includes a cart of mass M that rolls without slipping along a flat table with a height of z_c. The table does not rotate as long as point p, denoting the ZMP, stays inside the base, inferring that $\tau_{zmp} = 0$. If the horizontal position of the center of mass is given by x relative to origin O, the net torque around point p can be calculated from the vector equation

$$\boldsymbol{\tau}_p = \mathbf{p} \times \mathbf{R} + \boldsymbol{\tau}_{zmp}. \tag{1.3}$$

This expression is simplified to the scalar equation

$$\tau_{zmp} = -Mg\left(x - p\right) + Mz_c\ddot{x}. \tag{1.4}$$

Using Eq. 1.2 , Eq. 1.4 can be simplified and solved to give an expression for the ZMP,

$$p = x - \frac{z_c}{g}\ddot{x}. \tag{1.5}$$
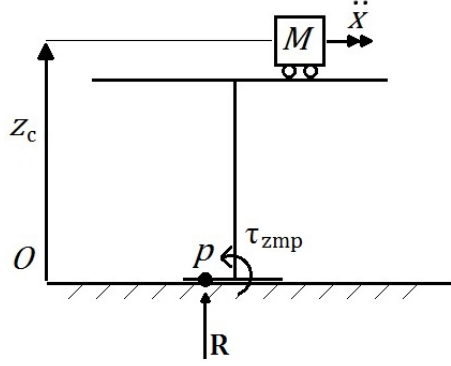
Figure 1.2. Cart-Table Model

Looking closer at Eq. 1.5 , two interesting observations emerge about the computed ZMP from this definition.

1. The ZMP becomes the CoM when there is no acceleration.

2. The ZMP can be located outside the support polygon.

If the ZMP leaves the support polygon during the gait, the motion is determined to be unstable because it can cause the robot to tip over about an edge of the support polygon. This, however, does not necessarily mean that the robot will fall. For example, if a bipedal robot has feet glued to the ground, the ZMP could be outside the support polygon without the robot falling. The cohesive forces of the glue adds an extra force to balance the torque around the ZMP so that the robot does not fall until the glue fails. If falling, the robot can recover stable walking either through its motion (compensating body movement) or increasing its support polygon (placing the other foot on the ground).

The ZMP can also be calculated for the 3D case using Newton's and Euler's laws for the change in linear and angular momentum. More general expressions, similar to Eq. 1.5, were presented in [7] and summarized in Appendix A.2. This was used in modeling the full 3D dynamics of DARwIn-OP to calculate the ZMP. However, this

3D model was never formally used in the controller, but was developed using Kane's method.

1.3   THE HUMAN GAIT AND A SIMPLIFIED MODEL

One of the goals of implementing this humanoid walking controller is to achieve a walking gait that is more human-like. In order to achieve this, the human gait is presented and then simplified to match the needs of the controller. During walking, at least one foot remains in contact with the ground at all times. Each step consists of two main phases, stance and swing, consisting of about 62% and 38% of the gait cycle, respectively. Each of these phases can be subdivided into 8 more categories [4]: initial contact, loading response, midstance, terminal stance, preswing, initial swing, midswing, and terminal swing as shown in Fig. 1.3.

Figure 1.3. Human Gait Cycle [2]

When the leading foot touches the ground, initial contact begins, starting the
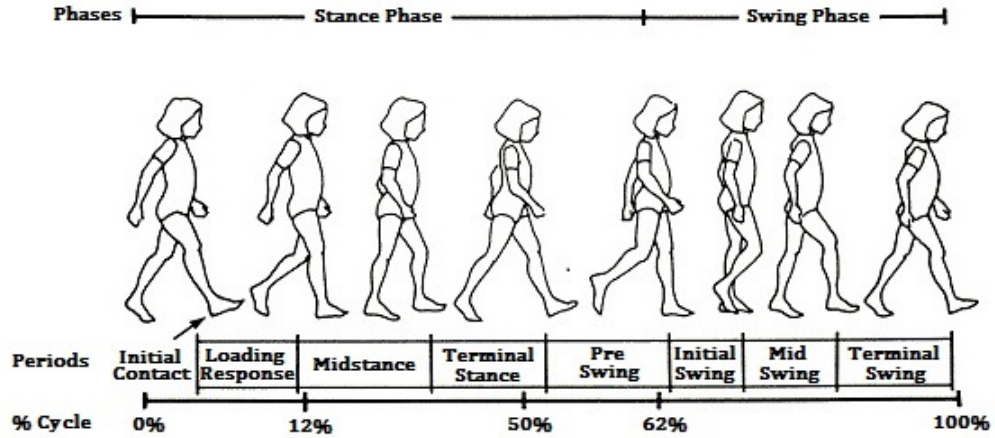
stance phase. Usually, this involves a simple heel strike to the ground. The main implusive force from ground contact is experienced in the next phase, the loading response. In this phase, the leading foot flattens, coming in full contact with the ground. The leading foot then absorbs the contact force from the ground. After this, the first single support phase, called midstance, begins when the trailing foot leaves the ground and ends when the body weight is aligned over the front of the current stance foot. During midstance, the body has a maximum potential energy as the body's center of gravity reaches its maximum height, rising over the stance leg. The terminal stance phase begins as soon as the body's weight shifts past the current stance foot. Heel off occurs during this phase when the stance foot's heel leaves the ground. The final stance phase is preswing, in which the previous stance foot prepares to lift off, but is still in contact with the ground. This is called toe off as the previous stance foot's toe leaves the ground and can no longer provide active forward propulsion.

Now, the previous stance leg during single support becomes the swing foot for the next three phases. Initial swing begins immediately after the swing foot leaves the ground and lasts until maximum knee flexion, where midswing starts. During midswing, the swing foot is brought past the stance foot, making sure that the foot has enough ground clearance. The phase ends when the tibia is perpendicular to the ground. The final phase, terminal swing, prepares the body for contact with the ground and fully extends the knee. This completes one gait cycle [4].

In order to simplify the human gait cycle, the proposed controller is broken into four distinct phases: right leg double support, right leg single support, left leg double support, and left leg single support. Relating these to the aforementioned phases, right leg double support tries to capture initial contact to midstance. Right leg single support refers to the period of time during which the left leg is swinging through the air, namely midstance and most of terminal stance. As soon as the left swing leg

touches the ground, left leg double support begins, which includes the last part of terminal stance and preswing. The final part of the gait, left leg single support, includes the entire swing phase from initial swing to terminal swing. The controller develops schemes to handle these four phases throughout the walking gait cycle.

1.4 DARWIN-OP

For testing purposes, the controller was implemented on DARwIn-OP, a humanoid developed at the Robotics and Mechanisms Laboratory (RoMeLa) at Virginia Tech [3]. DARwIn-OP stands for dynamic anthropomorphic robot with intelligence - open platform and is used for research and educational purposes. DARwIn-OP is a twenty-degree-of-freedom mechanism with twenty MX-28 Dynamixel servo motors [1]. Each of these servos has a 12 bit encoder with a resolution of $0.088°$, or values ranging from 0-4095 for $360°$ which are position controlled. Each of the legs contains six motors, and each of the arms has three motors. The remaining two motors actuate the pitch and yaw motions for the head.

Each of these twenty motors has a single axis of rotation, so coordinate systems must be defined to describe the relative orientation between links. In Figure 1.4, the axes of rotations for the joints are shown as red arrows. These directions were chosen based on the positive rotational directions in hardware to ease the transformation from model to implementation. A front view of DARwIn-OP is shown on the left, and a side view is shown on the right. The global inertial coordinate frame is named the N frame and is also shown in the diagram. When referring to a direction for walking throughout this paper, the N frame is the coordinate system being referenced unless otherwise stated. For example, walking forward is walking in the positive x direction. Also, the N frame is located on the ground, directly between the feet in the y direction, and in the middle of the feet in the x direction. The N frame location is denoted by a ♦ for each view in Fig. 1.4. The position of the N frame is important in defining

7
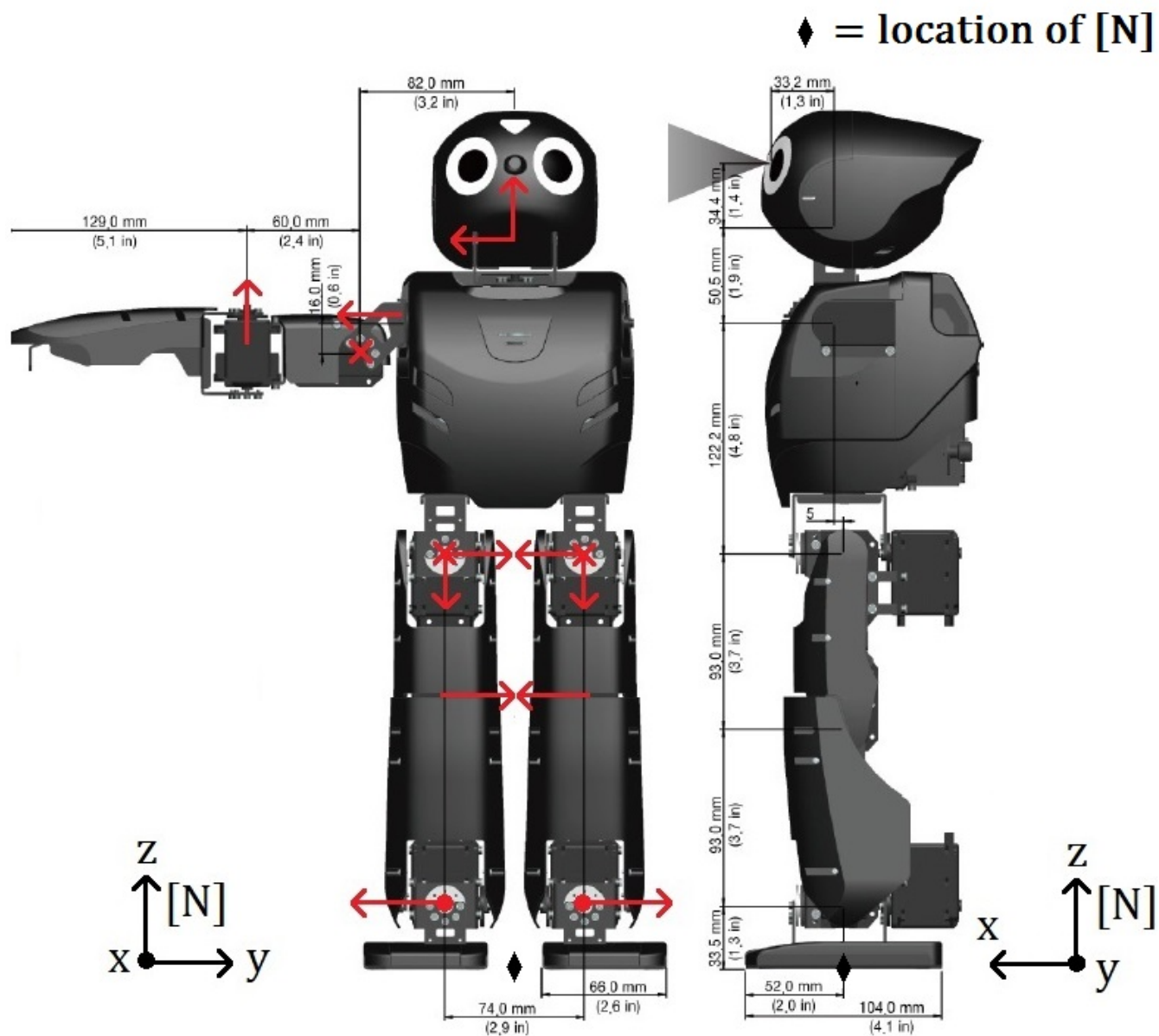
Figure 1.4. DARwIn-OP with Coordinate Systems

the reference trajectories for the CoM and the swing foot. This is the zero position for DARwIn-OP so all of the joint coordinate systems are aligned with the N frame except the shoulder roll is rotated 45° and the elbow is rotated -90°. Arm swing was not included in the controller, so this is not critical, but the zero position was used for every joint that was uncontrolled. Therefore, all coordinate systems besides the ones previously mentioned are aligned with the N frame and joint rotations can be described in terms of roll (x-axis), pitch (y-axis), and yaw (z-axis) rotations. For example, the right ankle has a joint with pitch in the negative y direction and a joint with roll in the positive x direction.

Currently, DARwIn-OP has a walking controller programmed to locomote while playing soccer. This walking controller creates a sinusoid for lateral body motion as well as a step period to constantly be moving forward in time. Step parameters such as stride length and stride width are used to achieve a desired goal location for the swing foot. The swing foot also uses a sinusoid for ground clearance during the single support phase. With this current controller, DARwIn-OP appears to take short, choppy steps with a stride length usually less than DARwIn-OP's foot length (104 mm). Going above this stride length usually leads to instability, or falling over. Using the ZMP method, longer stride lengths are able to be obtained in a more predictable manner that can be adjusted based on the step length, width, and period.

CHAPTER 2

METHODS

This chapter presents the scheme for the controller in order to accomplish a walking gait. A control law is developed using preview terms in order to track a reference value. Trajectory generation for the main body's CoM is accomplished through a simple cart-table model, and the swing foot trajectory is determined through parametrized equations.

2.1 CONTROLLER AND TRAJECTORY GENERATION SCHEME

In order to produce one gait cycle, the controller must account for the four different phases: right leg double support, right leg single support, left leg double support, and left leg single support. The double support and single support phases for each leg are treated similarly due to the body symmetry in the xz plane. The two main parts of the controller consist of tracking a CoM trajectory with the stance leg and following a swing trajectory with the swing leg. From these two trajectories and additional constraints, inverse kinematics can be used to calculate the desired angles. All of these constraints are holonomic and enforced through geometry in all phases.

1. The stance leg and body behave like an inverted pendulum with varying length.

2. The body link is aligned with the N frame, or parallel to the ground.

3. The swing foot is also aligned with the N frame, or parallel to the ground.

10

Figure 2.1 shows a flow diagram for the individual components of the controller. There are also knee flexion and body pitch controllers to directly control the stance knee pitch and the body pitch, respectively. These controllers are crucial when transitioning between steps to maintain a cyclic pattern.   Neglecting the body pitch
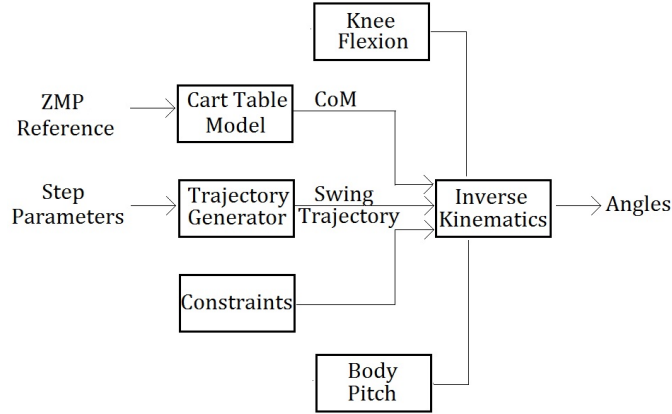
Figure 2.1. Control Scheme for the ZMP Walking Controller

and knee flexion controllers, the CoM trajectory and constraints 1 and 2 define all of the stance angles for both the double support and single support phases. With these angles, the swing foot trajectory, and constraint 3, the swing leg angles can now be fully defined using inverse kinematics. This includes the assumption that ten of the twelve leg joints were used in the controller, neglecting the two hip yaw joints. The next sections define the details for each part of the controller.

2.1.1 DEFINING ANGLES

Before developing more details, there are two sets of angles that must be defined to simplify the calculations for the inverse kinematics. One set is the local angles and the other the global angles, both referencing the same axes of rotations for the joints in Fig. 1.4. The local angles are defined by the stance and swing leg which switch as the stance and swing legs change. A local angle is denoted simply by θ. However, the global angles do not change and they identify a specific joint. These global angles are simply denoted by θ'. Table 2.1 shows all the definitions for local and global angles. For implementation, local angles are calculated and then transformed to global angles depending on the phase.

TABLE 2.1: Local and Global Angle Definitions

θ	Definition	θ'	Definition
θ_1	Stance Ankle Roll	θ'_1	Right Ankle Roll
θ_2	Stance Ankle Pitch	θ'_2	Right Ankle Pitch
θ_3	Stance Knee Pitch	θ'_3	Right Knee Pitch
θ_4	Stance Hip Pitch	θ'_4	Right Hip Pitch
θ_5	Stance Hip Roll	θ'_5	Right Hip Roll
θ_6	Support Hip Roll	θ'_6	Left Hip Roll
θ_7	Support Hip Pitch	θ'_7	Left Hip Pitch
θ_8	Support Knee Pitch	θ'_8	Left Knee Pitch
θ_9	Support Ankle Pitch	θ'_9	Left Ankle Roll
θ_{10}	Support Ankle Roll	θ'_{10}	Left Ankle Pitch

2.2 CENTER OF MASS TRAJECTORY

Expanding the cart-table model to a 3D system, a CoM trajectory can be calculated in the xy plane using the linearized system. The linearization of the 3D

cart-table model can be found in Appendix A.3, and the result shown in Fig. 2.2 is a cart that can move along a flat table at a set height z_c. Let the location of the center of mass be $\mathbf{c} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ and the position of the ZMP be $\mathbf{p} = \begin{bmatrix} p_x & p_y \end{bmatrix}^T$. The z-component of the position of the ZMP is zero because it is assumed to be on the ground.
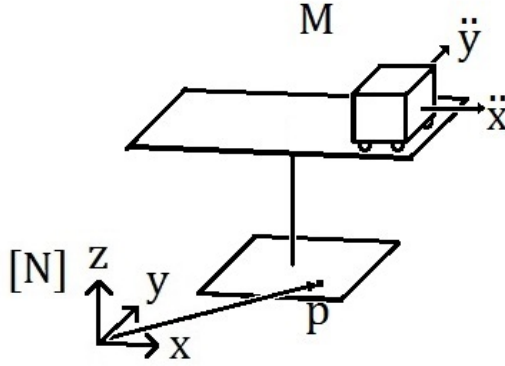

Figure 2.2. 3D Cart-Table Model

This system can be put into state-space form with the position of the ZMP in the xy plane as the outputs. In order to control the ZMP, the jerk of the CoM was chosen as the input because the acceleration terms must appear as a state to define

the position of the ZMP. With these inputs and outputs, the state space system is

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ y \\ \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ y \\ \dot{y} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{2.1}$$

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\frac{z_c}{g} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -\frac{z_c}{g} \end{bmatrix} \begin{bmatrix} x & \dot{x} & \ddot{x} & y & \dot{y} & \ddot{y} \end{bmatrix}^T. \tag{2.2}$$

The dynamics in the two dimensions are completely decoupled from each other and identical, so that means the same control law can be developed for u_x and u_y. The goal of this controller is to force the ZMP to follow an input reference value which directly relates to the stance foot's position. Without having some feedforward term in the control law that accounts for future reference values, the actual ZMP does not reach the reference value fast enough. In terms of walking, the robot is already on the next step before its body can catch up.

Illustrating this point, consider a type 1 servo system described by Ogata [6] with a control law

$$u = K_I e - \mathbf{K}\mathbf{x}, \tag{2.3}$$

where $u_x = u$ for $\mathbf{x} = \begin{bmatrix} x & \dot{x} & \ddot{x} \end{bmatrix}^T$ and $e = e_x$, and $u_y = u$ for $\mathbf{x} = \begin{bmatrix} y & \dot{y} & \ddot{y} \end{bmatrix}^T$ and $e = e_y$. The gain matrix $\mathbf{K}$ is a 1x3 matrix multiplying the states. The error in the actual ZMP relative to the reference ZMP value is defined as

$$\mathbf{e} = \mathbf{p}^{ref} - \mathbf{p}. \tag{2.4}$$

14

Gains of $K_I = 40$ and $K_x = \begin{bmatrix} 1 & 50 & 4 \end{bmatrix}^T$ were chosen by trial and error in simulation and reference values p_x^{ref} and p_y^{ref} were generated. Section 3.1 goes into further detail about how to generate the reference values. Simulation results using the cart-table model and the controller in Eq. 2.3 are shown in Figure 2.3. In the simulation, the ZMP value tries to track the reference value with the integral gain and state feedback, but it cannot react fast enough to the changing reference. For the y direction, the actual ZMP reaches the reference as soon as the reference changes. Therefore, this is not a sufficient controller to force the ZMP to folllow a certain reference due to this phase delay. The control law must have some terms that include future reference values.
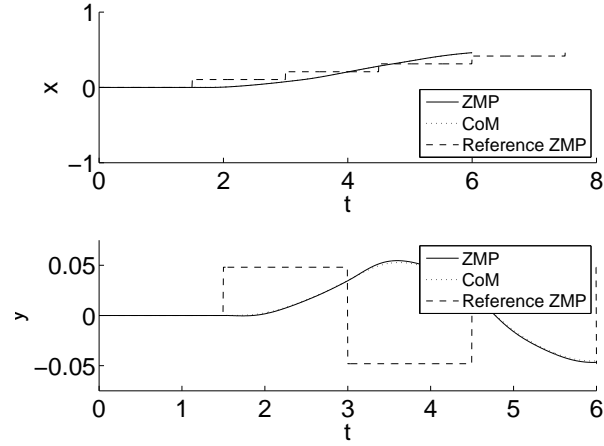
Figure 2.3. Response of the Cart-Table Model Using No Preview Controller

2.2.1 PREVIEW CONTROLLER

A preview controller has a control law whose input is a function of future reference values. Using these preview terms is similar to looking down the road while driving. A reference value farther down the road, and not the car's current position, is used to steer the car. In considering a preview controller, the system in 2.1 must be discretized because the preview terms need to use discrete reference values. Using a sampling time of T_s, the system is discretized as

$$\mathbf{x}\left(k+1\right) = A\mathbf{x}\left(k\right) + Bu\left(k\right), \tag{2.5}$$

$$p\left(k\right) = C\mathbf{x}\left(k\right), \tag{2.6}$$

where

$$\mathbf{x}\left(k\right) = \begin{bmatrix} x\left(kT_s\right) & \dot{x}\left(kT_s\right) & \ddot{x}\left(kT_s\right) & y\left(kT_s\right) & \dot{y}\left(kT_s\right) & \ddot{y}\left(kT_s\right) \end{bmatrix}^T, \tag{2.7}$$

$$u\left(k\right) = \begin{bmatrix} u_x\left(kT_s\right) & u_y\left(kT_s\right) \end{bmatrix}^T, \tag{2.8}$$

$$p\left(k\right) = \begin{bmatrix} p_x\left(kT_s\right) & p_y\left(kT_s\right) \end{bmatrix}^T, \tag{2.9}$$

$$A = \begin{bmatrix} 1 & T_s & T_s^2/2 & 0 & 0 & 0 \\ 0 & 1 & T_s & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T_s & T_s^2/2 \\ 0 & 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} T_s^3/6 & 0 \\ T_s^2/2 & 0 \\ T_s & 0 \\ 0 & T_s^3/6 \\ 0 & T_s^2/2 \\ 0 & T_s \end{bmatrix} \tag{2.10}$$

$$C = \begin{bmatrix} 1 & 0 & \frac{z_c}{g} \\ 1 & 0 & \frac{z_c}{g} \end{bmatrix}. \tag{2.11}$$

16

Assume that the actual ZMP needs to track the reference ZMP, p^{ref}. By previewing N_L time steps into the future, the discrete control law proposed by Katayama [5] can be used.

$$u\left(k\right) = -G_I e\left(k\right) - G_x x\left(x\right) - \sum_{j=1}^{N_L} G_p\left(j\right) p^{ref}\left(k+j\right), \tag{2.12}$$

where $e\left(k\right) = p\left(k\right) - p^{ref}\left(k\right)$. Now, the actual ZMP in the cart-table model can sufficiently track the reference with this extra feedforward term. The simple block diagram in Fig. 2.4 broadly illustrates the control signals for the preview control. The gains G_I, G_x, and G_d are associated with the integral, state feedback, and preview
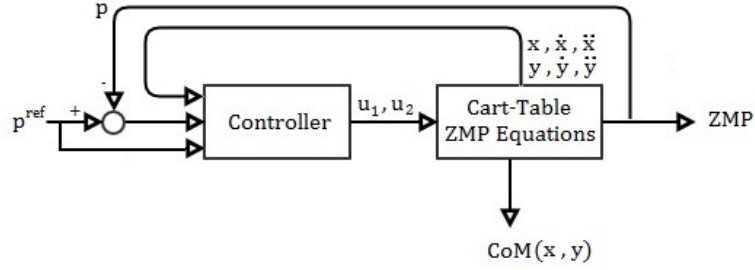

Figure 2.4. Block Diagram of Preview Controller

terms, respectively. These gains must be chosen to achieve proper tracking of the ZMP without much phase delay. Katayama [5] describes a method for designing an optimal preview controller for a discrete-time system, and the method is summarized in Appendix B. Given a system of n states, m outputs, and r inputs, this method

tries to minimize the performance index

$$J = \sum_{i=k}^{\infty} \left[e^T(i) Q_e e(i) + \Delta x^T(i) Q_x \Delta x(i) + \Delta u^T(i) R \Delta u(i) \right] \qquad (2.13)$$

at each time k, where Q_e and R are $m \times m$ and $r \times r$ symmetric positive definite matrices. The matrix Q_x is an $n \times n$ symmetric non-negative definite matrix associated with the incremental state vector. Decoupling the system into separate x and y components, the following values were chosen for the controller,

$$Q_e = 0.1 \ , \ Q_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \ , \ R = 1 \times 10^{-6}, \qquad (2.14)$$

with a sampling time of $T_s = 8$ ms. Using the discrete-time system in Eqs. 2.7 to 2.11, the gains could be calculated using the linear quadratic regulator design for discrete time systems described in Appendix B. Using the values in Eq. 2.14, the gains were calculated to be $G_I = 257.7$, $G_x = \begin{bmatrix} 10285.4 & 1683.3 & 1683.3 \end{bmatrix}^T$, and the preview gains are shown in Fig. 2.5. The preview gains become negligible towards the end of the previewing period. With these gains, the cart-table model can be simulated, and Fig. 2.6 shows that the actual ZMP tracks the reference ZMP much better without much phase delay.

In conclusion, the cart-table model is a simplified way to model the humanoid as one large body mass accelerating in a plane at a constant height. The ZMP can be controlled to follow a reference trajectory, and a CoM trajectory can be obtained through simulation. The CoM trajectory is used in implementation with inverse kinematics to calculate the stance leg angles.
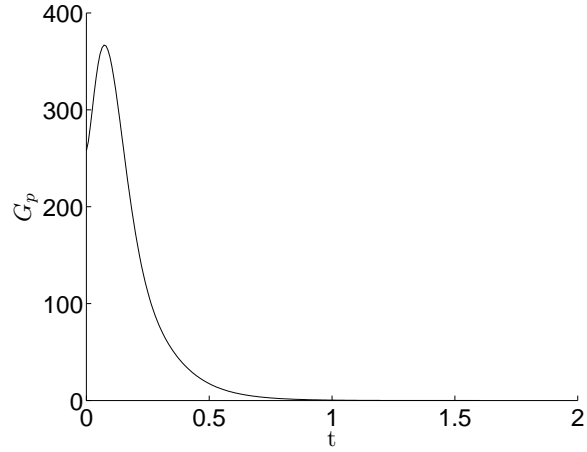
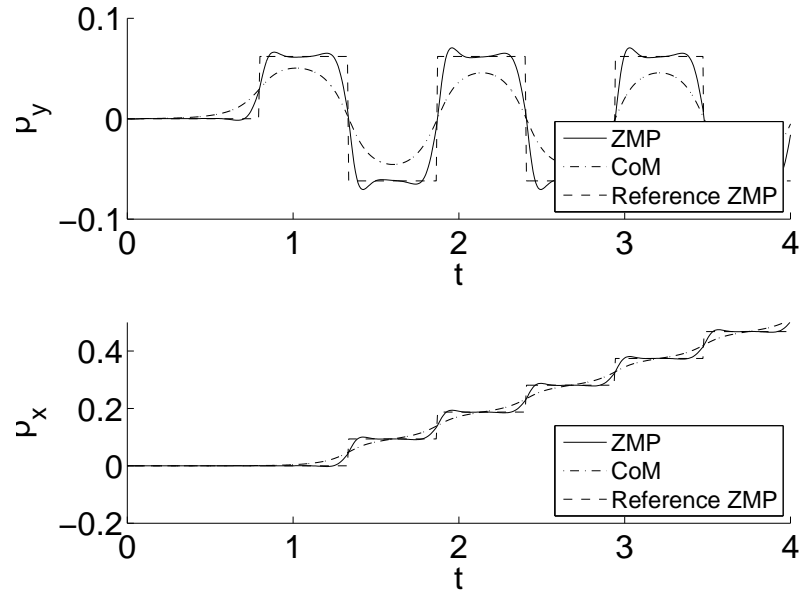Figure 2.5. Preview Gains Using the Optimal Controller Design

Figure 2.6. Response of the Cart-Table Model Using the Preview Controller

2.3   SWING FOOT TRAJECTORY

The stance leg uses the CoM trajectory and other constraints to define joint angles, but the swing leg needs to establish a trajectory to define its angles.   This

trajectory $\mathbf{p}^{swing}$ tracks the location of the swing foot in the local foot frame $\mathcal{F}$. The $\mathcal{F}$ frame is aligned with the N frame, but it is located at the ankle joints of the stance foot as shown in Fig. 2.7. The subscripts L and R denote the left and right foot frames, respectively. The $\mathcal{F}$ frame switches between $\mathcal{F}_L$ and $\mathcal{F}_R$ depending on the stance leg. Under the conditions that the humanoid is not tipping and the foot is not slipping, the $\mathcal{F}$ frame is an inertial reference frame.

The swing foot trajectory $\mathbf{p}^{swing}$ is defined relative to the current $\mathcal{F}$ frame. Constantly updating the location of the $\mathcal{F}$ frame after each step requires the definition of two more vectors. The stride vector for ith step, $\mathbf{S}_i$, locates the swing foot from its position right before initial swing to right after terminal swing. The x and y components of $\mathbf{S}_i$ are the stride length and stride width, respectively. The foot reference vector for the ith step, $\mathbf{f}_i^{ref}$, locates the swing foot before the initial swing phase. Intermediate points along the swing foot trajectory are needed, so the following definition is given.

$$\mathbf{p}^{swing} = \mathbf{s}_i, \tag{2.15}$$

where $\mathbf{s}_i$ is called the foot path vector which takes some path that starts at $\mathbf{f}_i^{ref}$ and ends at $\mathbf{f}_i^{ref} + \mathbf{S}_i$. A foot path function must be defined that calculates the value for $\mathbf{s}_i$ at different times. The foot reference value is just a translational shift so the foot path function has a similar shape to the plot in Fig. 2.8.

The swing leg trajectory needs to accomplish two distinct tasks:

1. Remain on the ground during the double support phase before swinging.

2. Establish a trajectory for the swing foot that reaches a specified stride length S_x and width S_y within a certain single support period T_{SS}.

With these tasks, the swing leg trajectory is split into the double support and single support phases. The double support phase imposes that the swing foot remain on
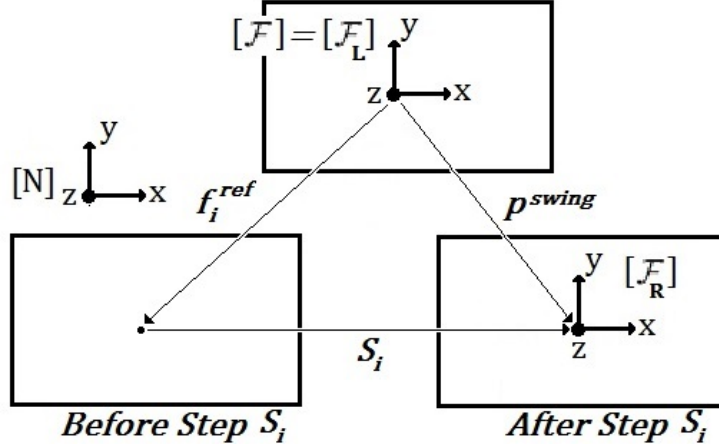
Figure 2.7. Humanoid Feet with the $\mathcal{F}$ Frame and N Frame

the ground where it had landed, implying that $\mathbf{p}^{swing} = \mathbf{f}_i^{ref}$. The single support phase is divided further into 3 different parts: initial swing, midswing, and terminal swing. The time spent in initial swing and terminal swing is a percentage of the step period, αT_{SS}, and the rest of the time is spent in midswing. The final constraint on the swing foot trajectory is that the swing foot must reach a certain height above the ground called ground clearance c_g. This prevents toe drag so that the toe does not get too close to the ground. One possible swing foot trajectory is shown in Fig. 2.8 to illustrate the desired shape of the curve. The upper graph shows a side view, while the lower graph shows a top view of the swing foot trajectory during the single support phase given a stride vector $\mathbf{S} = \begin{bmatrix} 0.104 & 0.01 \end{bmatrix}^T$.

2.3.1 INITIAL SWING

Initial swing occurs immediately after the double support phase ends and lasts for αT_{SS} seconds. The foot path function for the initial swing is marked by a parametrization to create an ellipse-type shape. Given a parametrized variable x_t, the following

21
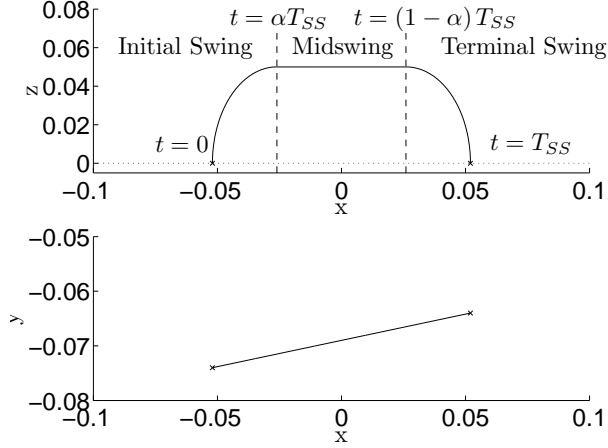
Figure 2.8. Possible Swing Foot Trajectory, $\mathbf{p}^{swing}$

foot path function can be generated.

$$s_x\left(x_t\right) = -r\cos\left(x_t\right) + f_{i,x}^{ref} \tag{2.16}$$

$$s_y\left(s_x\right) = \eta\left(s_x - f_{i,x}^{ref}\right) + f_{i,y}^{ref} \tag{2.17}$$

$$s_z\left(x_t\right) = c_g\sin\left(x_t\right), \tag{2.18}$$

where $0 \le x_t < \frac{\pi}{2}$ and

$$r = \alpha S_{i,x}. \tag{2.19}$$

The r value is the length of one of the axes of the ellipse. This creates an ellipse that is shifted by the foot reference value in the xz plane and linearly approaches the stride width in the xy plane. The value for η can be calculated by a simple slope formula,

$$\eta = \frac{S_{i,y}}{S_{i,x}}. \tag{2.20}$$

22

Using time as the parametrized variable gives an easier result to implement in the xz plane,

$$s_x(t) = -r \cos\left(\frac{\pi}{2\alpha T_{SS}} t\right) + f_{i,x}^{ref} \tag{2.21}$$

$$s_y(s_x) = \eta\left(s_x - f_{i,x}^{ref}\right) + f_{i,y}^{ref} \tag{2.22}$$

$$s_z(t) = c_g \sin\left(\frac{\pi}{2\alpha T_{SS}} t\right), \tag{2.23}$$

where $0 \leq t < \alpha T_{SS}$. This defines the initial swing phase which lifts the foot off the ground until it reaches the ground clearance c_g.

2.3.2   MIDSWING

The next swing period, midswing, starts after initial swing and lasts until $(1 - \alpha) T_{SS}$ seconds. During this phase, the foot stays at the constant ground clearance height while moving forward in time. Laterally, the step width is accounted for in the same manner through linear interpolation. To clearly show the relationship between the time t and the current step length s_x, a conversion constant is defined as $b = \frac{S_{i,x}}{T_{SS}}$.

For midswing, the foot path function is defined as

$$s_x(t) = bt + f_{i,x}^{ref} \tag{2.24}$$

$$s_y(s_x) = \eta\left(s_x - f_{i,x}^{ref}\right) + f_{i,y}^{ref} \tag{2.25}$$

$$s_z(t) = c_g, \tag{2.26}$$

where $\alpha T_{SS} \leq t < (1 - \alpha) T_{SS}$.

2.3.3   TERMINAL SWING

The final swing phase is terminal swing, which occurs after midswing and prepares the foot for ground contact. This has a very similar shape to initial swing except

23

that the function starts with the foot near ground clearance and brings the foot to the ground. Again, the desired curve shape in the xz plane is an ellipse. One key difference is the starting point for the step length s_x must be shifted by a constant since the terminal swing starts at $s_x(0) = f_{i,x}^{ref} + (1-\alpha) S_{i,x}$. Starting with the same parametrized variable x_t, the foot path function for terminal swing is

$$s_x(x_t) = r \sin(x_t) + f_{i,x}^{ref} + (1-\alpha) S_{i,x} \tag{2.27}$$

$$s_y(s_x) = \eta \left(s_x - f_{i,x}^{ref} \right) + f_{i,y}^{ref} \tag{2.28}$$

$$s_z(x_t) = c_g \cos(x_t), \tag{2.29}$$

where $0 \le x_t < \frac{\pi}{2}$ and $r = \alpha \left(S_{i,x} - f_{i,x}^{ref} \right)$. At the end of the period $x_t = \frac{\pi}{2}$, the foot path function must return final values of

$$s_x = S_{i,x} + f_{i,x}^{ref} \tag{2.30}$$

$$s_y = S_{i,y} + f_{i,y}^{ref} \tag{2.31}$$

$$s_z = 0. \tag{2.32}$$

Looking back at the previous equations describing the foot path function during terminal swing, these all give the appropriate final values. Again, converting to time gives

$$s_x(t) = r \sin\left(\frac{\pi}{2\alpha T} (t - \phi) \right) + f_{i,x}^{ref} + (1-\alpha) bT_{SS} \tag{2.33}$$

$$s_y(s_x) = \eta \left(s_x - f_{i,x}^{ref} \right) + f_{i,y}^{ref} \tag{2.34}$$

$$s_z(t) = c_g \cos\left(\frac{\pi}{2\alpha T} (t - \phi) \right), \tag{2.35}$$

where $\phi = (1-\alpha) T_{SS}$ and $\phi \le t < T_{SS}$. With the foot path functions from initial swing, midswing, and terminal swing, the swing foot trajectory is now defined and can

be used to calculate the swing leg angles. For implementation, the inverse kinematics must be developed to actually calculate these angles.

CHAPTER 3

IMPLEMENTATION

This chapter develops an easy method to generate reference values for $\mathbf{f}^{ref}$ and $\mathbf{p}^{ref}$. From the previously calculated trajectories, equations for the inverse kinematics of the humanoid are used to calculate the joint angles. Constraints are used to define all of the angles, and the body pitch and knee controllers are presented. Lastly, an overview of the program used to implement the walking controller is explained.

3.1   REFERENCE GENERATION

To generate the two trajectories in the previous section, there were two references that needed to be known. The foot reference $\mathbf{f}^{ref}$ and the ZMP reference $\mathbf{p}^{ref}$ are similar, but have subtle differences to provide more flexibility with the controller. First, the foot reference defines the locations of the ankle joints and the $\mathcal{F}$ frame. The support polygon can be determined from the foot reference. The ZMP reference is just the desired location of the ZMP. The ZMP reference does not determine the support polygon other than that the ZMP reference should be generated with the understanding that the ZMP needs to be inside the support polygon.

Define a set of m steps $\mathcal{S} = [S_0, S_1, \ldots, S_{m-1}, S_m]$, where S_0 is an initialization period, S_1 is the first step, and S_m is the last step. S_0 is a double support phase that shifts the body mass toward the first stance foot, chosen to be the left foot, and $S_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$. Also, a ZMP reference offset δ is going to be added to the ZMP reference equation in order to be able to shift the ZMP reference towards the outer side of the foot or inside the foot. Adding this offset to the ZMP reference requires

knowledge of the stance foot, so a function that takes a step as an input and returns a -1 or 1 based on the stance foot must be defined. For the ith step, this stance foot function is

$$stance\left(\mathbf{S}_i\right) = \begin{cases} -1, & \text{if stance} == \text{RIGHT} \\ 1, & \text{if stance} == \text{LEFT}. \end{cases} \tag{3.1}$$

Figure 3.1 shows a visual of the reference vectors to more clearly formulate a mathematical method to update these reference values. Looking at Fig. 3.1 and using simple vector addition, the following update formulas can be used to define the reference values after the initialization period.

$$\mathbf{f}_i^{ref} = -\left(\mathbf{f}_{i-1}^{ref} + \mathbf{S}_{i-1}^{ref}\right) \tag{3.2}$$

$$\mathbf{p}_i^{ref} = \mathbf{p}_{i-1}^{ref} - \mathbf{f}_i^{ref} \tag{3.3}$$

if $i > 1$. Then, the initial values must be set for the references. These are $\mathbf{f}_1^{ref} = \begin{bmatrix} 0 & -s_w & 0 \end{bmatrix}^T$ and $\mathbf{p}_0^{ref} = \begin{bmatrix} 0 & -\frac{1}{2}s_w & 0 \end{bmatrix}^T$, where the stance width s_w is 0.074 m. It is important to note that these are not the reference values during the initialization period and are only used to update the values. Including the reference values for the initialization terms and adding the offset to the ZMP reference, the two reference trajectories become

$$\mathbf{f}^{ref} = \begin{cases} \mathbf{f}_1^{ref}, & \text{i} = 0 \\ \mathbf{f}_i^{ref}, & \text{i} > 0. \end{cases} \tag{3.4}$$

$$\mathbf{p}^{ref} = \begin{cases} \mathbf{0}, & \text{i} = 0 \\ \mathbf{p}_i^{ref} + \left(stance\left(\mathbf{S}_i\right)\delta\right)\hat{\mathbf{j}}, & \text{i} > 0 \end{cases}. \tag{3.5}$$

These initial values have a logical basis to them. Because the initialization period

27

Figure 3.1. Vectors used to Identify the Location of the Foot and ZMP at Each Step

does not have a single support phase, the foot reference is the same as the foot reference of the first step. In other words, the stance foot has not changed. For the ZMP reference, the updating equation in Eq. 3.3 allows for an easy method to alternate between the left and right stance leg, but the ZMP needs to start in the middle of the two feet if the robot is standing up straight with both feet on the ground. Since the N frame is defined as the middle of the two feet initially, the ZMP reference is the origin during the initialization period.

With only a set of steps and a predefined initialization period, foot and ZMP reference values can be calculated from Eqs. 3.4 and 3.5. These reference values are used with the formulations in the previous sections to calculate a CoM trajectory and swing foot trajectory. With these trajectories, the inverse kinematics can be developed in the local foot frame $\mathcal{F}$ to determine the angles.

3.2 INVERSE KINEMATICS

The following section develops angles for the local angle set defined in Table 2.1. The relative angle of rotation is defined as the angle of rotation from the outer link to the inner link of a joint. For global angles, the initial link is the body, but for the local angles, the initial link is either foot depending on the stance leg. This means that local swing angles are equivalent to their global angle counterparts, but local stance angles are the negatives of their global angle counterparts. Global angles are only used in implementing the software, and only local angles are discussed in future sections.

The following assumptions are made while trying to track the CoM trajectory and the swing leg trajectory.

1. All of the mass is located in the body along the stance hip pitch axis with $M = 2.93$ kg.

2. The robot is symmetrical around the xz plane of the N frame.

3. The stance leg and body mass act like an inverted pendulum that can vary in length.

These assumptions are used throughout the inverse kinematics formulation to simplify some of calculations and the dimensions shown in Fig. 3.2. Another vector p^{hip} is defined to locate the swing hip relative to the $\mathcal{F}$ frame. With these assumptions, the leg lengths are the same, with $a = 0.093$ m, and $l = 0.037$ m, which is half of the stance width. These dimensions are used frequently to locate the CoM and the swing foot trajectories.

3.2.1 STANCE LEG

The strategy for the stance leg is to use the CoM trajectory to determine the two ankle joint angles, the knee to determine the height of the body, and the two
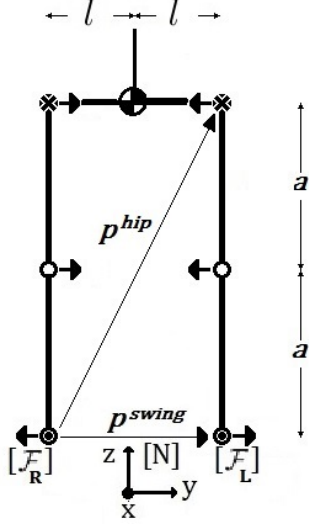
Figure 3.2. Local Coordinate Systems for DARwIn-OP

hip joint angles to orient the body. First, the assumption about the behavior of the stance leg and body as an inverted pendulum must be fully developed. The knee angle determines the height of the inverted pendulum through the law of cosines. Looking at the stance leg in Fig. 3.3, the bent knee forms a triangle with a height h_p being the height of the pendulum. This can be calculated using the law of cosines,

$$h_p = \sqrt{2a^2 \left(1 - \cos\left(\pi - \theta_3\right)\right)}. \tag{3.6}$$

The angle constraints to maintain this inverted pendulum requires that

$$\theta_2 = \frac{1}{2}\theta_3 + \text{more terms} \tag{3.7}$$

$$\theta_4 = \frac{1}{2}\theta_3 + \text{more terms}. \tag{3.8}$$

Without more terms, the body would bend down, adjusting the height of the pendulum based on the knee bend. More terms are included in θ_2 to track the CoM, while more terms are included in θ_4 to orient the body using the body pitch controller.
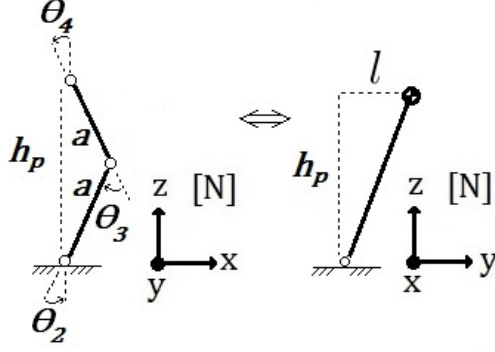
Figure 3.3. Stance Leg Pitch Angles Forming the Height of the Pendulum

With this pendulum height, inverse kinematics for a 3D pendulum with a roll then pitch joint at the base can be used to calculate the angles. However, the location of the body mass must be identified relative to the $\mathcal{F}$ frame, and currently, the CoM trajectory is relative to the N frame. This transformation is simply a translation because the two coordinate systems are aligned. This translation $\mathbf{f}$ is the location of the $\mathcal{F}$ frame relative to the N frame. For the ith step S_i, this vector can be calculated (see Fig. 3.1),

$$\mathbf{f}_i = \mathbf{p}_0^{ref} - \sum_{j=1}^{i} \mathbf{f}_j^{ref}. \tag{3.9}$$

where $\mathbf{p}_0^{ref} = \begin{bmatrix} 0 & -\frac{1}{2}s_w & 0 \end{bmatrix}^T$. Now, the local CoM trajectory $\mathbf{c}_{N/\mathcal{F}}$ can be calculated from the CoM trajectory $\mathbf{c}$,

$$\mathbf{c}_{N/\mathcal{F}} = \mathbf{c} - \mathbf{f}. \tag{3.10}$$

Figure 3.4 shows the location of the foot, CoM trajectory, and local CoM trajectory for the given gait of $S_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$, $S_1 = \begin{bmatrix} \frac{1}{2}f_l & 0 & 0 \end{bmatrix}^T$, and $S_2 = S_3 = S_4 = S_5 =$

31

$S_6 = S_7 = \begin{bmatrix} f_l & 0 & 0 \end{bmatrix}^T$ with an initialization period of $T_0 = 0.8$ seconds and step periods of $T = 0.536$ seconds. The local CoM $\mathbf{c}_{N/\mathcal{F}}$ has a very repeatable pattern, but is discontinuous in time when the reference $\mathcal{F}$ frame changes. Returning to
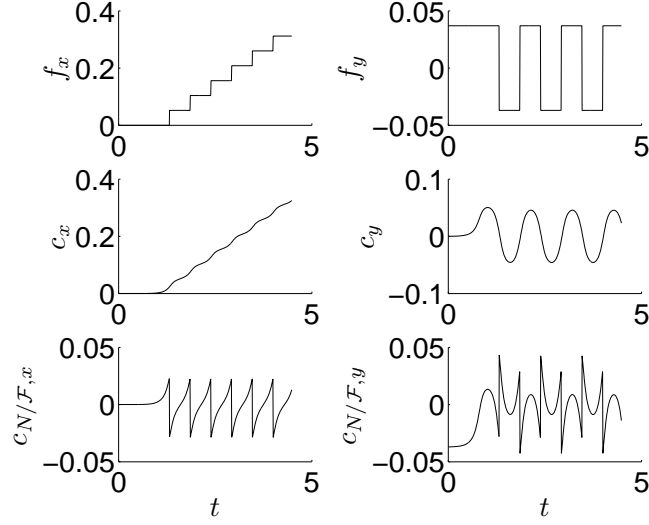
Figure 3.4. Graphs for the $\mathbf{f}$, $\mathbf{c}$, and $\mathbf{c}_{N/\mathcal{F}}$ as Functions of Time for a Possible Gait

Fig. 3.3, the inverse kinematics for the 3D pendulum are slightly different based on a left foot or right foot stance leg, so these are treated separately. If $\mathcal{F} = \mathcal{F}_R$, the location of the center of the mass relative to the $\mathcal{F}$ frame is

$$\mathbf{c}_{N/\mathcal{F}} = {}^N C^2 \left(l\hat{\mathbf{j}} + h_p\hat{\mathbf{k}} \right), \tag{3.11}$$

with the rotation matrices

$$
{}^{N}C^{1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{\theta_1} & -s_{\theta_1} \\ 0 & s_{\theta_1} & c_{\theta_1} \end{bmatrix} \quad , \quad {}^{1}C^{2} = \begin{bmatrix} c_{\theta_{2p}} & 0 & s_{\theta_{2p}} \\ 0 & 1 & 0 \\ -s_{\theta_{2p}} & 0 & c_{\theta_{2p}} \end{bmatrix} \tag{3.12}
$$

and ${}^{N}C^{2} = {}^{N}C^{1}{}^{1}C^{2}$. The subscript p denotes an angle that is not the entire stance ankle angle, but only the angle from the inverse kinematics of the 3D pendulum. Separating these equations into their components yields

$$
c_{N/\mathcal{F},x} = -h_p s_{\theta_{2p}} \tag{3.13}
$$

$$
c_{N/\mathcal{F},y} = l c_{\theta_1} - h_p c_{\theta_{2p}} s_{\theta_1} \tag{3.14}
$$

$$
c_{N/\mathcal{F},z} = l s_{\theta_1} + h_p c_{\theta_{2p}} c_{\theta_1}. \tag{3.15}
$$

Solving 3.13 for θ_{2p} gives

$$
\theta_{2p} = sin^{-1}\left(\frac{-c_{N/\mathcal{F},x}}{h_p}\right). \tag{3.16}
$$

From 3.14 and 3.15, θ_1 can be calculated as

$$
\theta_1 = sin^{-1}\left(\frac{l c_{N/\mathcal{F},z} - c_{N/\mathcal{F},y} h_p c_{\theta_{2p}}}{l^2 + h_p^2 c_{\theta_{2p}}^2}\right) \quad \text{where} \tag{3.17}
$$

$$
c_{N/\mathcal{F},z} = \sqrt{h_p^2 + l^2 - c_{N/\mathcal{F},x}^2 - c_{N/\mathcal{F},y}^2}. \tag{3.18}
$$

Equation 3.18 is derived from equating the lengths of the pendulum calculated using the center of mass coordinates $\mathbf{c}_{N/\mathcal{F}}$ and the link lengths h_p and l. When $\mathcal{F} = \mathcal{F}_L$

and $l < 0$, these angles are

$$\theta_{2_p} = sin^{-1}\left(\frac{c_{N/\mathcal{F},x}}{h_p}\right) \tag{3.19}$$

$$\theta_1 = sin^{-1}\left(\frac{lc_{N/\mathcal{F},z} - c_{N/\mathcal{F},y}h_p c_{\theta_{2_p}}}{l^2 + h_p^2 c_{\theta_{2_p}}^2}\right). \tag{3.20}$$

A more complete derivation for these angles can be found in Appendix C. The only difference between the left and right stance legs is in θ_{2_p} because the axis of rotation of the ankle pitch is reversed for the left stance leg. Combining this result with Eq. 3.7 gives the stance ankle pitch angle

$$\theta_2 = \frac{1}{2}\theta_3 + \theta_{2_p}. \tag{3.21}$$

3.2.1.1 PARALLEL BODY CONSTRAINT

The only stance leg angles that need to be calculated in order to stand the body upright are the stance hip roll and stance hip pitch. This relationship can be calculated geometrically through interior angles of the triangle formed by the knee bend. The total roll rotation and total pitch rotation must be zero. Looking at Fig. 3.3 and accounting for the direction of rotation in Fig. 3.2, the relationship to keep the body parallel to the ground for the pitch direction is

$$\frac{\pi}{2} = -\theta_4 + \frac{\pi}{2} - \theta_3 + \theta_2, \tag{3.22}$$

$$\theta_4 = \theta_2 - \theta_3. \tag{3.23}$$

The constraint for the body roll is straightforward,

$$\theta_5 = \theta_1, \tag{3.24}$$

34

although this does add error for the ankle joints that are attempting to track the CoM because the stance leg is no longer perpendicular to the body, causing Eq. 3.11 to be slightly incorrect. This error is neglected due to the fact that the stance hip and ankle joints should have very small rotations, and therefore, this error is very small. The extra rotation for the stance hip pitch does not add any error to this model because of the assumption that the CoM lies along the axis of rotation of the stance hip pitch. Now, all the stance leg joint angles have been defined and can track a CoM trajectory.

3.2.2 SWING LEG

The swing leg joint angles have a similar strategy. The inward joints angles for the swing hip roll (θ_6), swing hip pitch (θ_7), and the swing knee pitch (θ_8) are used to track the swing foot trajectory $\mathbf{p}^{swing}$, while the two ankle joint angles attempt to keep the foot parallel with the ground. Again, the formulations must be treated differently for a left swing leg compared to a right swing leg, but for the concepts are the same.

The first part in calculating the vector from $\mathcal{F}$ to the swing hip $\mathbf{p}^{hip}$ is exactly the same for each swing leg under the assumption that $l < 0$ when $\mathcal{F} = \mathcal{F}_L$. Using forward kinematics, the vector loop equation is

$$
{}^{N}C^2 \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} + {}^{N}C^3 \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} + {}^{N}C^5 \begin{bmatrix} 0 \\ 2l \\ 0 \end{bmatrix} + {}^{N}C^7 \begin{bmatrix} 0 \\ 0 \\ -a \end{bmatrix} + {}^{N}C^8 \begin{bmatrix} 0 \\ 0 \\ -a \end{bmatrix} = {}^{N}\mathbf{p}^{swing}. \quad (3.25)
$$

Converting Eq. 3.25 to the stance hip roll frame (frame 5), the equation becomes

$$
\left({}^{5}C^{2} \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} + {}^{5}C^{3} \left(\begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} + \begin{bmatrix} 0 \\ 2l \\ 0 \end{bmatrix} \right) \right) + {}^{5}C^{7} \begin{bmatrix} 0 \\ 0 \\ -a \end{bmatrix} + {}^{5}C^{8} \begin{bmatrix} 0 \\ 0 \\ -a \end{bmatrix} = {}^{5}\mathbf{p}^{swing},
\tag{3.26}
$$

with the terms in parentheses equal to ${}^{5}\mathbf{p}^{hip}$. Substituting this value into Eq. 3.26 and rearranging gives

$$
{}^{5}C^{7} \begin{bmatrix} 0 \\ 0 \\ -a \end{bmatrix} + {}^{5}C^{8} \begin{bmatrix} 0 \\ 0 \\ -a \end{bmatrix} = {}^{5}\mathbf{p}^{swing} - {}^{5}\mathbf{p}^{hip} = {}^{5}\Delta\mathbf{p}.
\tag{3.27}
$$

This is where the analysis must split into two different formulations depending on the stance leg because the rotation matrices are not the same in Eq. 3.27. For notational clarity, assume that $\Delta\mathbf{p} = {}^{5}\Delta\mathbf{p}$. If $\mathcal{F} = \mathcal{F}_R$, the angles for the swing hip and knee are

$$
\theta_6 = ATAN2\left(-\Delta p^2{}_y, -\Delta p^2{}_z\right)
\tag{3.28}
$$

$$
\theta_8 = -\cos^{-1}\left(\frac{\Delta p^2{}_x + \Delta p^2{}_y + \Delta p^2{}_z - 2a^2}{2a^2}\right)
\tag{3.29}
$$

$$
\theta_7 = ATAN2\left(\frac{\Delta p^2{}_x\left(1 + c_{\theta_8}\right)c_{\theta_6} + \Delta p^2{}_z s_{\theta_8}}{2ac_{\theta_6}\left(1 + c_{\theta_8}\right)}, \frac{\Delta p^2{}_x s_{\theta_8} c_{\theta_6} - \Delta p^2{}_z\left(1 + c_{\theta_8}\right)}{2ac_{\theta_6}\left(1 + c_{\theta_8}\right)}\right).
\tag{3.30}
$$

A more complete derivation for these angles can be found in Appendix C. When $\mathcal{F} = \mathcal{F}_L$, $\Delta\mathbf{p}$ must be slightly adjusted by changing $\Delta p_{x,\mathcal{F}_L} = -\Delta p_{x,\mathcal{F}_R}$. The swing angles become very similar to those for the right leg as the swing leg except for a

sign change in the hip roll joint angle,

$$\theta_6 = ATAN2\left(-\Delta p^2{}_y, -\Delta p^2{}_z\right) \tag{3.31}$$

$$\theta_8 = cos^{-1}\left(\frac{\Delta p^2{}_x + \Delta p^2{}_y + \Delta p^2{}_z - 2a^2}{2a^2}\right) \tag{3.32}$$

$$\theta_7 = ATAN2\left(\frac{\Delta p^2{}_x\left(1 + c_{\theta_8}\right)c_{\theta_6} + \Delta p^2{}_z s_{\theta_8}}{2ac_{\theta_6}\left(1 + c_{\theta_8}\right)}, \frac{\Delta p^2{}_x s_{\theta_8}c_{\theta_6} - \Delta p^2{}_z\left(1 + c_{\theta_8}\right)}{2ac_{\theta_6}\left(1 + c_{\theta_8}\right)}\right). \tag{3.33}$$

3.2.2.1   PARALLEL FOOT CONSTRAINT

The parallel foot constraint is similar to the parallel body constraint, except it is not intuitive to establish the relationship geometrically. With the foot parallel to the ground at all times, the humanoid can land flat during terminal swing and smoothly transition into double support phase. With the foot being flat on the ground, the end of the double support phase causes problems for the swing knee. In order to keep the foot flat on the ground, the swing knee has to extend, and with large stride lengths, the knee may not be able to satisfy this constraint.

The method for calculating the swing ankle angles uses rotational matrices,

$$^{N}C^{10} = I_{3\times3} \tag{3.34}$$

$$^{8}C^{N}\,^{N}C^{10} = \,^{8}C^{N} \tag{3.35}$$

$$^{8}C^{10} = \,^{8}C^{N}, \tag{3.36}$$

because $^{8}C^{N}$ can be calculated from the previous joint angles. Therefore, looking at $^{8}C^{10}$ for $\mathcal{F} = \mathcal{F}_R$,

$$^{8}C^{10} = \begin{bmatrix} c_{\theta_9} & 0 & -s_{\theta_9} \\ s_{\theta_{10}}s_{\theta_9} & c_{\theta_{10}} & c_{\theta_9}s_{\theta_{10}} \\ c_{\theta_{10}}s_{\theta_9} & -s_{\theta_{10}} & c_{\theta_9}c_{\theta_{10}} \end{bmatrix}, \tag{3.37}$$

the swing ankle joint angles are

$$\theta_9 = ATAN2\left(-m_{13}, m_{11}\right) \tag{3.38}$$

$$\theta_{10} = ATAN2\left(-m_{32}, m_{22}\right), \tag{3.39}$$

where m_{ij} represents the element in the ith row and jth column of the matrix ${}^8C^N$. When $\mathcal{F} = \mathcal{F}_L$, these swing ankle joint angles become

$$\theta_9 = ATAN2\left(m_{13}, m_{11}\right) \tag{3.40}$$

$$\theta_{10} = ATAN2\left(-m_{32}, m_{22}\right). \tag{3.41}$$

Finally, all of the trajectories and constraints for this controller have been satisfied. The stance leg behaves like an inverted pendulum trying to track the CoM trajectory. The stance leg also has the constraint that the body must be parallel to the ground, defining the stance hip joint angles. The swing leg defines the swing hip and knee joint angles by tracking the swing foot trajectory, and the constraint of a flat foot defines the swing ankle joint angles.

3.3 CONTROLLER ADDITIONS

The controller is not complete yet though. There are a few additional parts to add in order to easily adjust the control and provide a stable walking pattern. Most of these were added after experimentation with the idea of compensating for the simplified model. The first of these controllers is a body pitch controller, which allows the main body to maintain a desired goal pitch instead of being parallel to the ground. All that is needed to achieve this desired body pitch is an additional term

in Eq. 3.23 for the desired body pitch,

$$\theta_4 = \theta_2 - \theta_3 + \theta_{bp}, \tag{3.42}$$

where θ_{bp} is the pitch of the main body. Since the body pitch may be different when the stance leg changes, a controller is necessary to force the body pitch to remain at the desired body pitch angle θ_{bp_d}. This controller uses a proportional control law which acts like a servo system,

$$\theta_{bp} = \theta_{bp} + K_{bp} \left(\theta_{bp_d} - \theta_{bp} \right). \tag{3.43}$$

This forces the body pitch angle to approach the desired value because the term with the proportional gain on the right-hand side of the equation will be zero near the desired value. The value for K_{bp} used during simulation and testing was 0.05. During implementation, this body pitch angle must be recalculated by rearranging Eq. 3.42 for θ_{bp} whenever the stance leg changes. This body controller compensates for forward and backward tipping.

The next controller addition was a knee flexion controller. This knee flexion controller is important to a stable, smooth, and repeatable walking gait. After the stance leg changes, there is no guarantee that the knee flexion is still at the same constant angle, and in fact, it is not going to be unless the stride length happens to be exactly the proper length. Therefore, a similar proportional controller can be used to control the knee flexion to a desired value,

$$\theta_3 = \theta_3 + K_{kf} \left(\theta_{kf} - \theta_3 \right), \tag{3.44}$$

where θ_{kf} is the desired knee flexion. The value for K_{kf} used during simulation and testing was 0.10. The graphs in Fig. 3.5 clearly show the effect of the knee flexion
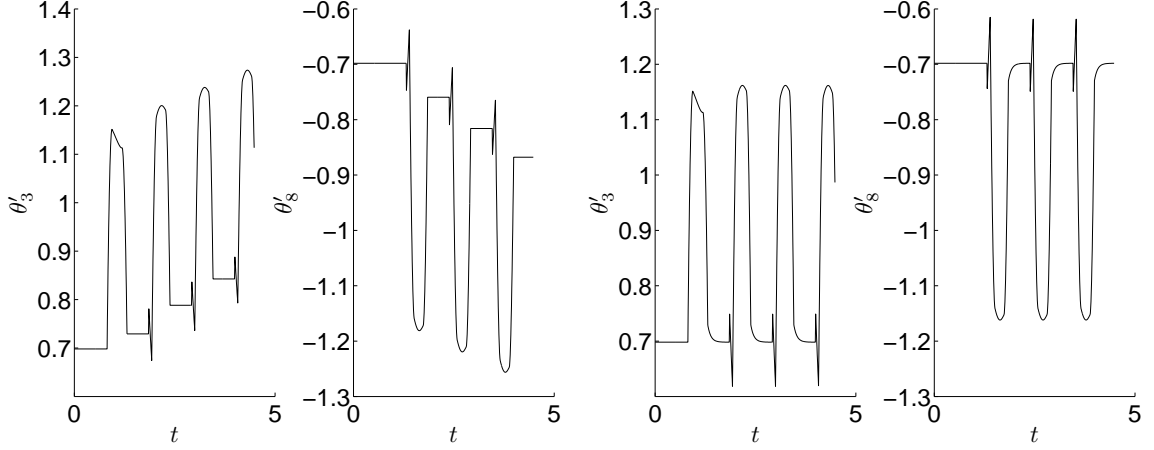
Figure 3.5. Global Knee Angles without (left) and with (right) Knee
Flexion Controller

controller. Without the knee flexion controller, the knee angles stay constant during double support phase, and the knee bends more and more with each double support phase. The knee flexion controller forces the knee flexion to maintain a cyclic and sustainable pattern during walking.

The final addition to the walking controller seems unjustified and random at first glance. This part of the controller adds a constant value to offset the CoM trajectory in the y direction so that

$$\mathbf{c}' = \mathbf{c} + \nu\hat{\mathbf{j}}, \tag{3.45}$$

where $\mathbf{c}'$ is the adjusted CoM trajectory with this offset, which is used as the actual CoM trajectory. The reason this was added to the controller is to compensate for the error in the assumption of symmetry in the xz plane. According to the multibody mass model, this is not true, and the CoM is in fact shifted toward the right foot by about 3 mm. This approach had a noticeable effect in experiments when setting $\nu = 0.005$, or 5 mm. When the actual CoM is toward the right leg, $\nu > 0$ to properly

shift the CoM. All of these additional controllers try to compensate for errors in the model and provide easy methods for controlling crucial angles such as the body pitch and knee flexion.

3.4 PROGRAMMING SYNOPSIS

A brief summary of the program algorithm is provided for a full understanding and example of the implementation of the controller. The algorithm is divided into 3 different sections: a ZMP walking controller, a step manager, and the cart-table dynamics. Each of these divisions includes important tasks that were previously described. The algorithm for the controller is shown in the flow chart shown in Fig. 3.6.

The ZMP walking controller's main task is to complete the current step movement. The main data associated with a step are its stride length and width, the CoM trajectory, the swing foot trajectory, and its step period (double support and single support period). These steps are held in a data structure that acts like a queue. The ZMP walking controller executes the current step by computing the angles through inverse kinematics until the step has finished. When the step is complete, the ZMP walking controller requests the next step from the step manager.

The step manager facilitates the transfer of data between the ZMP walking controller and the cart-table dynamics. This is where all the step data is stored and where the queue of steps is implemented. It is important to note that DARwIn-OP could not allocate more than 500 bytes of data dynamically, which means that only data for 6 steps could be stored at once, with a maximum step period of $T = 800ms$. This is the maximum because the data is stored as one big chunk at the beginning to avoid dynamically storing data. The reason to avoid dynamic memory allocation is for one time step of $\Delta t = 8ms$, just storing the CoM trajectory and swing foot trajectory is 48 bytes. Therefore, the maximum chunk of dynamic memory that can
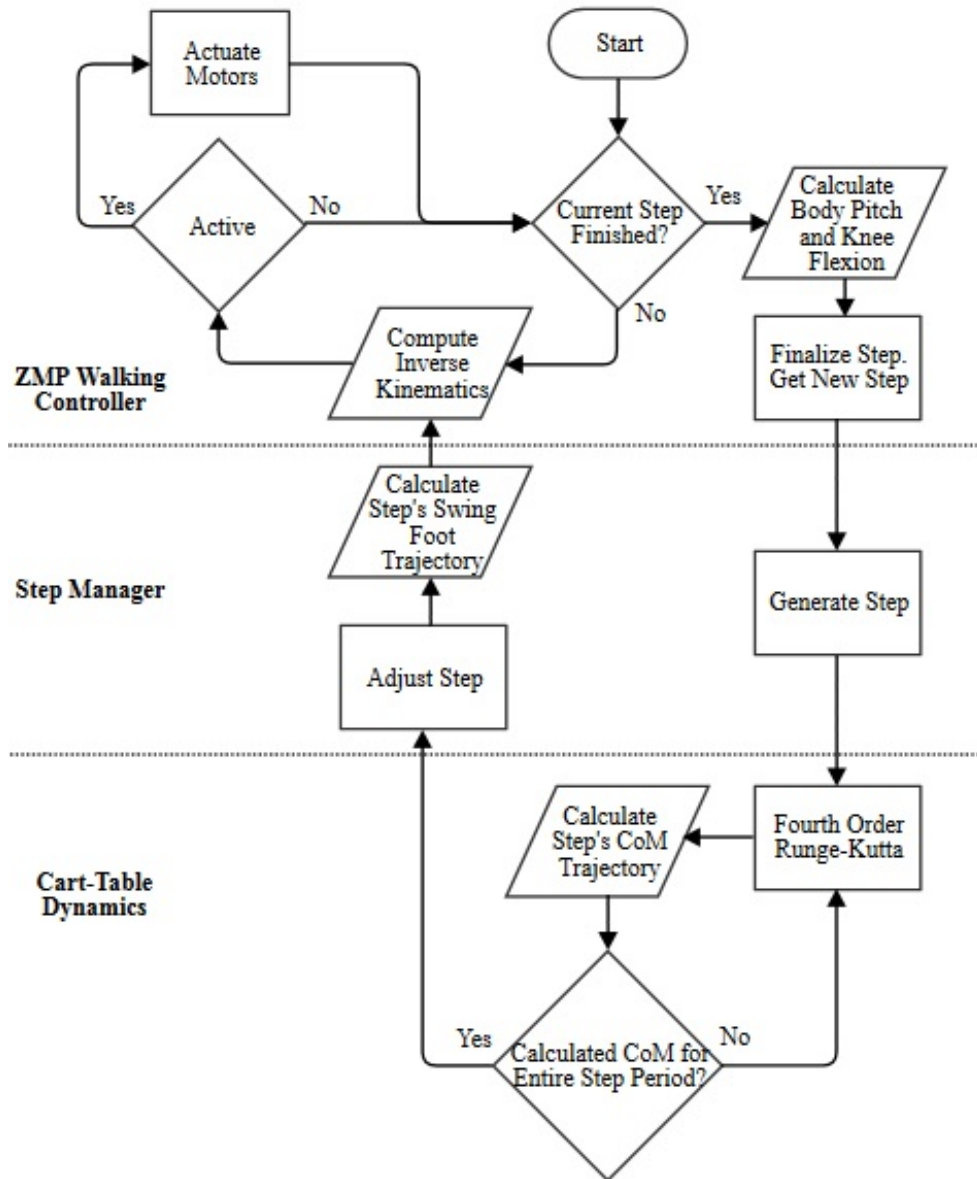
Figure 3.6. Flow Chart for the Implementation of the Controller

be allocated at once (500 bytes) can only store 80ms of CoM and swing foot positions.

After the current step is complete, the step manager creates a new step through a set of step parameters, such as stride length or step period. This step gets added onto the end of the queue, but the CoM trajectory cannot be calculated for this step because there are no steps ahead of it to preview. The queue has three pointers that point to a set of data for one step, pictured as a row in Fig. 3.7. The first pointer, called 'current', stores a pointer to the current step data. The second pointer, 'calculated', stores a pointer to the next step that needs the CoM trajectory calculated. The final pointer,'last', points to the last step in the queue. These are all incremented at the appropriate times to achieve the queue data structure.   The cart-table
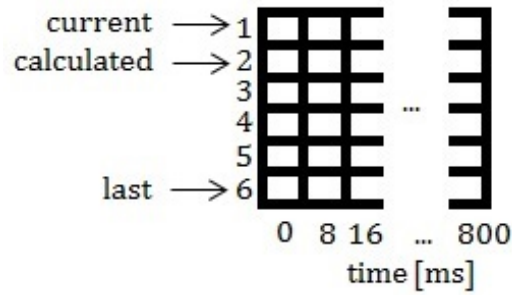
Figure 3.7. Representation for the Queue of Steps in the Step Manager

dynamics division of the controller calculates the CoM trajectory given the next step to be calculated. This uses fourth order Runge-Kutta as a numerical integrator and previews into the next steps for 1.6 seconds. After this is calculated, the step period must be adjusted by the step manager to determine how to divide the step period into a double support phase period T_{DS} and a single support phase period T_{SS}. The idea is that when the ZMP enters the support foot polygon, the humanoid can leave

the double support phase, and when the ZMP leaves the support foot polygon, the double support phase begins and single support phase ends. This way, the difference between the locations of the $\mathcal{F}$ frame and the ZMP can be used to determine where the ZMP is relative to the support foot polygon,

$$\mathbf{d} = |\mathbf{f} - \mathbf{p}|. \tag{3.46}$$

The location of the inner front corner of the foot is 0.022 m in the y direction and 0.052 m in the x direction. Starting at the beginning of the step in double support, the end of the double support phase is found when the ZMP crosses into the support foot's polygon. Mathematically, this relationship is

$$d_x < 0.052 \;\; \&\& \;\; d_y < 0.022 \tag{3.47}$$

due to the absolute value in Eq. 3.46 and the assumption that the ZMP always enters the support foot's polygon through the interior side or bottom edge and leaves through the interior side or front edge. The end of single support can then be found by starting at the end of the step and backstepping until Eq. 3.47 holds. Finally, all the tools have been presented to implement this walking controller in DARwIn-OP.

CHAPTER 4

RESULTS AND DISCUSSION

The main goal of this thesis is to properly implement a walking controller in hardware and software. Therefore, many tests were performed and analyzed after the controller had been implemented in simulation. Various step lengths and periods were used, and certain problems arose during testing, such as swing knee extension before single support phase and slip on certain surfaces.

4.1 TRIALS

Experiments were performed on two different surfaces; surface 1 had less traction, and surface 2 had more traction. Surface 1 could not produce consistent results due to slipping. Each trial was set up with the same $S_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ and $S_1 = \begin{bmatrix} \frac{1}{2}f_l & 0 & 0 \end{bmatrix}^T$, both with periods of 800 ms. After that period, different stride lengths and step periods were used. Other parameters that were changed included the ZMP offset (δ), the desired body pitch (θ_{bp_d}), and the CoM offset (ν). The knee flexion was set to $\theta_{kf} = 40°$, and the preview period was $T_{prev} = 1.6s$ with a time step of $\Delta t = 0.008s$.

The specific trials are shown in Table 4.1 for surface 1 and Table 4.2 for surface 2. These trials are given a rating of 1-5 based on how well they did. In order to get a 4 or 5, the robot had to walk to the end of the track and then be manually placed again and allowed to walk again. A rating of 5 means that the robot completed two tracks after being manually set down. The most successful trial, test #15 on surface

2, had a step period of $T = 0.536$ seconds with a step length of the foot length f_l.

TABLE 4.1: Tests on Surface 1

Test #	$S\,[m]$	$T\,[s]$	$\delta\,[m]$	$\theta_{bp_d}\,[°]$	$\nu\,[m]$	Rating
0	f_l	0.8	0.011	-	-	1
1	f_l	0.8	0.011	5	-	3
2	f_l	0.8	0.011	10	-	2
3	f_l	0.6	0.011	10	-	4
4	f_l	0.8	0.011	10	-	2

TABLE 4.2: Tests on Surface 2

Test #	$S\,[m]$	$T\,[s]$	$\delta\,[m]$	$\theta_{bp_d}\,[°]$	$\nu\,[m]$	Rating
0	f_l	0.8	0.011	15	-	2
1	f_l	0.8	0.011	15	-	4
2	f_l	0.8	0.011	15	-	4
3	f_l	0.6	0.015	15	-	2
4	f_l	0.6	0.020	15	-	2
5	f_l	0.6	0.025	15	-	1
6	f_l	0.536	0.011	10	-	2
7	f_l	0.536	0.02	10	0.003	2
8	f_l	0.536	0.011	10	0.005	3
9	f_l	0.536	0.011	10	0.005	5
10	f_l	0.536	0.011	10	0.005	5
11	$\frac{5}{4}f_l$	0.536	0.011	10	0.005	5
12	$\frac{3}{2}f_l$	0.536	0.011	10	0.005	2
13	f_l	0.8	0.011	15	0.005	2
14	f_l	0.536	0.011	10	0.005	4
15	f_l	0.536	0.011	10	0.005	5

4.2 DISCUSSION

Most of the tests were successful in starting off stable after the initialization period when the body pitch controller as well as the CoM offset were added to the controller. Some of the main problems causing instability included slipping of the support foot and swing foot interference with the stance foot. Slipping occurred frequently during the double support phase before initial swing, while the knee has a quick extension period. This quick extension is caused by the constraint to keep the stance foot on the ground as the body moves forward quickly to shift the ZMP into the stance foot's support polygon. This knee extension problem was exaggerated with larger stride lengths in test #13 for surface 2. This problem could be alleviated by implementing toe off to allow the heel of the stance foot to lift off the ground during double support phase.

Interference of the swing foot and the stance foot occurred occassionally. The result was the swing foot never reached its proper goal and did not land completely with a flat foot. This interference was not caused by an improper swing foot trajectory. In fact, the robot's own weight was causing the swing leg to move closer to the stance foot and slightly hit the stance foot. This was tested by lifting the robot off the ground to check for interference, and no interference was observed. This problem can be alleviated with toe off as well because the higher CoM from less knee flexion can cause less ankle roll.

These tests, especially the first few, demonstrated the need for the body pitch controller as well as the CoM offset. In test # 0 on surface 1, the robot kept falling backwards after the first step. By supporting its back, the robot was able to walk and was laterally stable. The body pitch controller stabilized this backwards falling motion by pitching the body forward. The need for the CoM offset became evident after a few trials with the robot. These trials showed that after one or two gait cycles, the robot was repeatedly falling over the inner part of the left stance foot. By shifting

47

the modeled CoM trajectory to the left (+y direction) due to the fact that the actual CoM was shifted toward the right in the model, the walking gait no longer fell over the inner part of the left stance foot.

The knee flexion controller was critical to maintaining the assumptions made and a cyclic walking pattern. If the stance knee joint angle kept decreasing throughout time, the robot got closer and closer to the ground, which is not a realistic gait. The knee flexion controller and body pitch controller isolated key parameters to adjust during different trials. When different stride lengths or step periods were used, adjusting the body pitch stabilized the walking gait in many trials. A body roll controller should be considered in order to control the full orientation of the body.

4.3 FUTURE WORK

Possible future work includes improving some of the missing parts in modeling human walking. More phases of the human walking gait could be modeled instead of the four currently implemented. The most important improvment would be to add toe off to the controller during the double support phase. This toe off would allow the knee flexion to decrease and simulate human walking more closely. Toe off could also be used to push the swing foot forward to begin the initial swing phase.

Another addition to the controller is a better estimate of the error in the ZMP from the cart to the real system. This can be accomplished by using the multibody 3D model to calculate the ZMP. An error can be found between the 3D model and the cart-table model. This error could be used to adjust the CoM trajectory in some manner to create a better estimate of the actual ZMP.

DARwIn-OP has an accelerometer and gyroscope, which has the ability to provide some sensor feedback. This sensor feedback can be added to the walking controller in a variety of different ways. For example, the inputs from the gyroscope and accelerometer could be used to approximate the ZMP and then adjust the CoM

trajectory based on the values that are being sensed.

4.4   CONCLUSION

This thesis establishes a simple and elegant walking controller that achieves a stable and adjustable walking gait in DARwIn-OP. Although simplified, using the cart-table model produced a steady and sufficient CoM trajectory to track using the stance leg. This walking controller produced a steady gait for a range of step periods, $0.536s < T < 0.8s$, and step lengths $0.104m < S_x < 0.130m$. The body pitch and knee flexion controllers greatly assisted in stabilizing the gait, and the stride length can be increased by adding toe off to the controller. Further work must be accomplished to improve the controller in order to have feedback while walking and improve the walking gait pattern itself to better model a human gait.

APPENDIX A

ZERO-MOMENT POINT

A.1 TORQUE ABOUT ZMP

The derivation for the torque around the ZMP is given in chapter 16 of the Springer Handbook for Robotics [7]. Recall that

$$\mathbf{p} = \frac{\sum_{i=1}^{N} \mathbf{p}_i f_{iz}}{\sum_{i=1}^{N} f_{iz}}. \tag{A.1}$$

Given a set of contacting points $\mathbf{p}_i$ for $i = 1, \ldots, N$ with force vectors $\mathbf{f}_i$ fixed in the ground with a line of action through $\mathbf{p}_i$, the net torque about the ZMP is

$$\boldsymbol{\tau} = \sum_{i=1}^{N} (\mathbf{p}_i - \mathbf{p}) \times \mathbf{f}_i. \tag{A.2}$$

Writing this vector equation in its different components yields

$$\tau_x = \sum_{i=1}^{N} (p_{iy} - p_y) f_{iz} - \sum_{i=1}^{N} (p_{iz} - p_z) f_{iy} \tag{A.3}$$

$$\tau_y = \sum_{i=1}^{N} (p_{iz} - p_z) f_{ix} - \sum_{i=1}^{N} (p_{ix} - p_x) f_{iz} \tag{A.4}$$

$$\tau_z = \sum_{i=1}^{N} (p_{ix} - p_x) f_{iy} - \sum_{i=1}^{N} (p_{iy} - p_y) f_{ix}. \tag{A.5}$$

Mathematically, a horizontal floor means that $p_{iz} = p_z$ because the ZMP must remain on the floor. Substituting Eq. A.1 into Eqs. A.3 and A.4 simplifies the equations to

$$\tau_x = \tau_y = 0 \qquad\qquad (A.6)$$

at the Zero-Moment Point. This is how the Zero-Moment Point gets its name.

A.2 ZMP IN 3D DYNAMICS

The computed ZMP can also be calculated for a 3D mass model given the mass, inertia tensors, and dynamics of a system. The following formulation for the ZMP in 3D summarizes material outlined in chapter 16 of the Springer Handbook for Robotics [7]. In Fig. A.1, an example of a mutltibody system in 3D is shown, and each body is given a mass m_j and an inertia tensor $\mathbf{I}_j$, where the inertia tensor is defined in the N frame. The N frame is an inertial reference frame. Also, there is a single resultant external force $\mathbf{R}$ acting on the bodies.   For j bodies, the total mass M and the
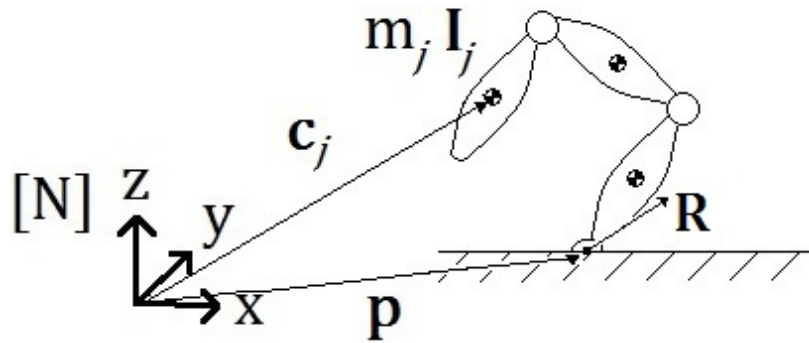

Figure A.1. Example of a Multi-Body System in 3 Dimensions

center of mass **c** of all the bodies are

$$M = \sum_{j=1}^{N} m_j \tag{A.7}$$

$$\mathbf{c} = \frac{1}{M} \sum_{j=1}^{N} m_j \mathbf{c}_j. \tag{A.8}$$

The total linear momentum $\mathcal{P}$ and total angular momentum $\mathcal{L}$ with respect to the N frame can be defined from the time derivatives of the centers of mass of the bodies as

$$\mathcal{P} = \sum_{j=1}^{N} m_j \dot{\mathbf{c}}_j \tag{A.9}$$

$$\mathcal{L} = \sum_{j=1}^{N} \left[\mathbf{c}_j \times (m_j \dot{\mathbf{c}}_j) + \mathbf{I}_j \boldsymbol{\omega}_j \right], \tag{A.10}$$

where $\boldsymbol{\omega}_j$ is the angular velocity of the jth body. Calculating the external torque about the origin gives

$$\boldsymbol{\tau} = \mathbf{p} \times \mathbf{R} + \boldsymbol{\tau}_{zmp}, \tag{A.11}$$

where $\boldsymbol{\tau}_{zmp}$ is the moment at p whose x and y components are zero according to Eq. A.6. Defining **g** as the gravitational acceleration vector, $\begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T$, and substituting Newton's and Euler's laws,

$$\mathbf{R} = \dot{\mathcal{P}} - M\mathbf{g} \tag{A.12}$$

$$\boldsymbol{\tau} = \dot{\mathcal{L}} - \mathbf{c} \times M\mathbf{g}, \tag{A.13}$$

into Eq. A.11 yields

$$\boldsymbol{\tau}_{zmp} = \dot{\mathcal{L}} - \mathbf{c} \times M\mathbf{g} + \left(\dot{\mathcal{P}} - M\mathbf{g} \right) \times \mathbf{p}. \tag{A.14}$$

Dividing the vector equation into its individual components, the equations in the x and y direction are

$$\tau_{zmp,x} = \dot{\mathcal{L}}_x + Mgy + \dot{\mathcal{P}}_y p_z - \left(\dot{\mathcal{P}}_z + Mg \right) p_y \tag{A.15}$$

$$\tau_{zmp,y} = \dot{\mathcal{L}}_y - Mgx - \dot{\mathcal{P}}_x p_z + \left(\dot{\mathcal{P}}_z + Mg \right) p_x. \tag{A.16}$$

Using the definition of the ZMP in Eq. A.6 and solving for the ZMP position vector, Eqs. A.15 and A.16 become

$$p_x = \frac{Mgx + p_z \dot{\mathcal{P}}_x - \dot{\mathcal{L}}_y}{mg + \dot{\mathcal{P}}_z} \tag{A.17}$$

$$p_y = \frac{Mgy + p_z \dot{\mathcal{P}}_y - \dot{\mathcal{L}}_x}{mg + \dot{\mathcal{P}}_z}. \tag{A.18}$$

These equations can then be used to calculated the location of the ZMP with any multibody 3D system.

A.3   3D LINEARIZED CART-TABLE MODEL

Given the 3D cart-table model in Fig. A.2, the cart on top of the table can move along the top of the table in any direction without slipping. The cart always remains in contact with the top of the table as well, and there is no friction force between the base of the cart and the ground. The position of the cart is given by $\mathbf{c} = \begin{bmatrix} x & y & z \end{bmatrix}^T$, and the cart has a mass of M.   Calculating the the linear momentum $\mathcal{P}$ and its
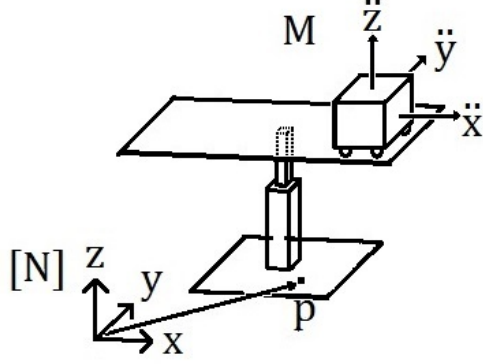
Figure A.2. 3D Cart-Table Model

time derivative gives,

$$\mathcal{P} = M \left(\dot{x}\hat{\mathbf{i}} + \dot{y}\hat{\mathbf{j}} + \dot{z}\hat{\mathbf{k}} \right) \tag{A.19}$$

$$\dot{\mathcal{P}} = M \left(\ddot{x}\hat{\mathbf{i}} + \ddot{y}\hat{\mathbf{j}} + \ddot{z}\hat{\mathbf{k}} \right). \tag{A.20}$$

Calculating the angular momentum $\mathcal{L}$ about the origin of the N frame and calculating its time derivative yields

$$\mathcal{L} = \mathbf{c} \times M\dot{\mathbf{c}} = M \left[(y\dot{z} - z\dot{y})\,\hat{\mathbf{i}} + (z\dot{x} - x\dot{z})\,\hat{\mathbf{j}} + (x\dot{y} - y\dot{x})\,\hat{\mathbf{k}} \right] \tag{A.21}$$

$$\dot{\mathcal{L}} = M \left[(y\ddot{z} - z\ddot{y})\,\hat{\mathbf{i}} + (z\ddot{x} - x\ddot{z})\,\hat{\mathbf{j}} + (x\ddot{y} - y\ddot{x})\,\hat{\mathbf{k}} \right]. \tag{A.22}$$

Now substituting values into Eqs. A.17 and A.18 gives

$$p_x \left(x, z, \ddot{x}, \ddot{z} \right) = \frac{gx - z\ddot{x} + x\ddot{z}}{g + \ddot{z}} \tag{A.23}$$

$$p_y \left(y, z, \ddot{y}, \ddot{z} \right) = \frac{gy + y\ddot{z} - z\ddot{y}}{g + \ddot{z}}. \tag{A.24}$$

54

Now, both of these equations must be linearized about the point $x = y = 0$, $\ddot{x} = \ddot{y} = \ddot{z} = 0$, and $z = z_c$. The linearized system takes the form of

$$\begin{Bmatrix} p_x \\ p_y \end{Bmatrix} = J \left\{ x \quad \dot{x} \quad \ddot{x} \quad y \quad \dot{y} \quad \ddot{y} \quad z \quad \dot{z} \quad \ddot{z} \right\}^T, \quad \text{where} \tag{A.25}$$

$$J = \begin{bmatrix} \frac{\partial p_x}{\partial x} & \frac{\partial p_x}{\partial \dot{x}} & \frac{\partial p_x}{\partial \ddot{x}} & \frac{\partial p_x}{\partial y} & \frac{\partial p_x}{\partial \dot{y}} & \frac{\partial p_x}{\partial \ddot{y}} & \frac{\partial p_x}{\partial z} & \frac{\partial p_x}{\partial \dot{z}} & \frac{\partial p_x}{\partial \ddot{z}} \\ \frac{\partial p_y}{\partial x} & \frac{\partial p_y}{\partial \dot{x}} & \frac{\partial p_y}{\partial \ddot{x}} & \frac{\partial p_y}{\partial y} & \frac{\partial p_y}{\partial \dot{y}} & \frac{\partial p_y}{\partial \ddot{y}} & \frac{\partial p_y}{\partial z} & \frac{\partial p_y}{\partial \dot{z}} & \frac{\partial p_y}{\partial \ddot{z}} \end{bmatrix} \tag{A.26}$$

and J is the Jacobian matrix. Taking the partial derivatives with respect to each variable for Eqs. A.23 and A.24 yields

$$\frac{\partial p_x}{\partial x} = 1 \tag{A.27}$$

$$\frac{\partial p_x}{\partial z} = -\frac{\ddot{x}}{g + \ddot{z}} \tag{A.28}$$

$$\frac{\partial p_x}{\partial \ddot{x}} = -\frac{z}{g + \ddot{z}} \tag{A.29}$$

$$\frac{\partial p_x}{\partial \ddot{z}} = \frac{\ddot{x} z}{(g + \ddot{z})^2} \tag{A.30}$$

$$\frac{\partial p_y}{\partial y} = 1 \tag{A.31}$$

$$\frac{\partial p_y}{\partial z} = -\frac{\ddot{y}}{g + \ddot{z}} \tag{A.32}$$

$$\frac{\partial p_y}{\partial \ddot{y}} = -\frac{z}{g + \ddot{z}} \tag{A.33}$$

$$\frac{\partial p_y}{\partial \ddot{z}} = \frac{\ddot{y} z}{(g + \ddot{z})^2}. \tag{A.34}$$

Substituting these into the Jacobian and evaluating at the linearization point gives the linearized system

$$\begin{Bmatrix} p_x \\ p_y \end{Bmatrix} = \begin{bmatrix} 1 & 0 & \frac{-z_c}{g} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \frac{-z_c}{g} & 0 & 0 & 0 \end{bmatrix} \left\{ x \quad \dot{x} \quad \ddot{x} \quad y \quad \dot{y} \quad \ddot{y} \quad z \quad \dot{z} \quad \ddot{z} \right\}^T.$$

$$\tag{A.35}$$

55

Notice that the linearized system is no longer a function of z and its time derivatives, so this state can be removed from the system. Essentially this removes the translational joint on the table so that the cart cannot move up or down.

APPENDIX B

OPTIMAL CONTROL DESIGN FOR A PREVIEW CONTROLLER

This appendix provides a short description of how to implement the preview controller described by Katayama [5]. Consider a time-invariant linear discrete system,

$$x(k+1) = Ax(k) + Bu(k) + Ew(k) \tag{B.1}$$

$$y(k) = Cx(k), \tag{B.2}$$

where $x(k)$ is the $n \times 1$ state vector, $u(k)$ is the $r \times 1$ control vector, $y(k)$ is the $m \times 1$ output vector to be cotnrolled, and $w(k)$ is the $q \times 1$ inaccessible constant disturbance. The incremental state vector is

$$\Delta x(i+1) = A\Delta x(i) + B\Delta u(i), \quad i = k, k+1, \ldots, \tag{B.3}$$

removing the disturbance $w(k)$ because it is a step function. The tracking error is

$$e(i+1) = e(i) + CA\Delta x(i) + CB\Delta u(i) - \Delta y_d(i+1), \quad i = k, k+1, \ldots, \tag{B.4}$$

where $\Delta y_d(i+1)$ is the difference between the output from the next time step and the current time step. With N_L future demands $y_d(i), i = k+1, \ldots, k+N_L$ available at time k, the incremental demand can be placed in a $pN_L \times 1$ vector

$$x_d(k) = \left[\Delta y_d^T(k+1), \ldots, \Delta y_d^T(k+N_L) \right]^T \tag{B.5}$$

which satisfies

$$x_d(i+1) = A_d x_d(i) \quad i = k, k+1, \ldots,$$ (B.6)

where

$$A_d = \begin{bmatrix} 0 & I_p & & & \mathbf{0} \\ & 0 & \ddots & & \\ & & & \ddots & I_p \\ \mathbf{0} & & & & 0 \end{bmatrix}.$$ (B.7)

The incremental state vector, the tracking error, and the incremental demand vector can create an augmented state vector,

$$\bar{x}(i) = \begin{bmatrix} e^T(i) & \Delta x^T(i) & x_d^t(i) \end{bmatrix}^T.$$ (B.8)

Then, Eqs. B.3, B.4, and B.6 can be combined into

$$\bar{x}(i+1) = \begin{bmatrix} I_p & CA & -I_p & 0 & \ldots & 0 \\ 0 & A & 0 & 0 & \ldots & 0 \\ 0 & 0 & & A_d & & \end{bmatrix} \bar{x}(i) + \begin{bmatrix} CB \\ B \\ 0 \end{bmatrix} \Delta u(i), \quad i = k, k+1, \ldots$$

(B.9)

Now, Q_e, Q_x, and R must be chosen, which are $m \times m$, $r \times r$, and $n \times n$, respectively. For example, the 1D cart-table system would have a Q_e with dimensions 1×1, a Q_x with dimensions 3×3, and an R with dimensions of 1×1. The 'dlqr' command in MATLAB can be used with the augmented A and B matrices found in Eq. B.9 and

the Q and R matrices, where

$$Q = \begin{bmatrix} Q_e & 0 & 0 \\ 0 & Q_x & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

(B.10)

APPENDIX C

INVERSE KINEMATICS

In this section, the inverse kinematics for the CoM tracking and the swing foot tracking are fully developed. First, the specific rotational matrices must be defined for both the $\mathcal{F}_L$ frame and $\mathcal{F}_R$ frame. For the $\mathcal{F}_R$ frame, the rotations based on local angles are

$$
{}^N C^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{\theta_1} & -s_{\theta_1} \\ 0 & s_{\theta_1} & c_{\theta_1} \end{bmatrix} \quad
{}^1 C^2 = \begin{bmatrix} c_{\theta_2} & 0 & -s_{\theta_2} \\ 0 & 1 & 0 \\ s_{\theta_2} & 0 & c_{\theta_2} \end{bmatrix} \quad
{}^2 C^3 = \begin{bmatrix} c_{\theta_3} & 0 & s_{\theta_3} \\ 0 & 1 & 0 \\ -s_{\theta_3} & 0 & c_{\theta_3} \end{bmatrix}
$$

$$
{}^3 C^4 = \begin{bmatrix} c_{\theta_4} & 0 & s_{\theta_4} \\ 0 & 1 & 0 \\ -s_{\theta_4} & 0 & c_{\theta_4} \end{bmatrix} \quad
{}^4 C^5 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{\theta_5} & s_{\theta_5} \\ 0 & -s_{\theta_5} & c_{\theta_5} \end{bmatrix} \quad
{}^5 C^6 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{\theta_6} & s_{\theta_6} \\ 0 & -s_{\theta_6} & c_{\theta_6} \end{bmatrix}
$$

$$
{}^6 C^7 = \begin{bmatrix} c_{\theta_7} & 0 & -s_{\theta_7} \\ 0 & 1 & 0 \\ s_{\theta_7} & 0 & c_{\theta_7} \end{bmatrix} \quad
{}^7 C^8 = \begin{bmatrix} c_{\theta_8} & 0 & -s_{\theta_8} \\ 0 & 1 & 0 \\ s_{\theta_8} & 0 & c_{\theta_8} \end{bmatrix} \quad
{}^8 C^9 = \begin{bmatrix} c_{\theta_9} & 0 & s_{\theta_9} \\ 0 & 1 & 0 \\ -s_{\theta_9} & 0 & c_{\theta_9} \end{bmatrix}
$$

$$
{}^9 C^{10} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_{\theta_{10}} & -s_{\theta_{10}} \\ 0 & s_{\theta_{10}} & c_{\theta_{10}} \end{bmatrix} .
$$

For the $\mathcal{F}_L$ frame, the stance leg and swing leg switch, so θ_6 switches with θ_5, θ_7 switches with θ_4, and so on. For notational simplicity, let $\mathbf{c}_{N/\mathcal{F}} = x\hat{\mathbf{i}} + y\hat{\mathbf{j}} + z\hat{\mathbf{k}}$.

Calculating the forward kinematics for the CoM trajectory and referencing Fig 3.2,

$$\mathbf{c}_{N/\mathcal{F}} = {}^N C^2 \left(l\hat{\mathbf{j}} + h_p \hat{\mathbf{k}} \right), \tag{C.1}$$

and separating Eq. C.1 into separate components yields

$$x = -h_p s_{\theta_2} \tag{C.2}$$

$$y = l c_{\theta_1} - h_p c_{\theta_2} s_{\theta_1} \tag{C.3}$$

$$z = l s_{\theta_1} + h_p c_{\theta_2} c_{\theta_1}. \tag{C.4}$$

θ_2 can be directly solved from Eq. C.2,

$$\theta_2 = \sin^{-1} \left(\frac{x}{h_p} \right). \tag{C.5}$$

Solving Eq. C.4 for c_{θ_1},

$$c_{\theta_1} = \frac{z - l s_{\theta_1}}{h_p c_{\theta_2}}, \tag{C.6}$$

and substituting the result into Eq. C.3

$$y h c_{\theta_2} = lz - l^2 s_{\theta_1} - h_p^2 c_{\theta_2}^2 s_{\theta_1} \tag{C.7}$$

$$s_{\theta_1} \left(l^2 + h_p^2 c_{\theta_2}^2 \right) = zl - y h_p c_{\theta_2} \tag{C.8}$$

$$\theta_1 = \sin^{-1} \left(\frac{lz - y h_p c_{\theta_2}}{l^2 + h_p^2 c_{\theta_1}} \right). \tag{C.9}$$

However, z is not explicitly formulated in the CoM trajectory. Therefore, z must be calculated by calculating the length of the pendulum with two different methods and equating them. The first way uses the link lengths h_p and l, and the second method

uses the distance formula for the center of mass, $\mathbf{c}_{N/\mathcal{F}}$. This equation is

$$h_p^2 + l^2 = x^2 + y^2 + z^2, \tag{C.10}$$

and solving for z gives

$$z = \sqrt{h_p^2 + l^2 - x^2 - y^2}. \tag{C.11}$$

Also note that when $\mathcal{F} = \mathcal{F}_R$, $l = 0.037$, but when $\mathcal{F} = \mathcal{F}_L$, $l = $ -0.037.

Expanding the forward kinematics for the swing leg in Eq. 3.27 into individual components yields

$$\Delta p_x = a \left(s_{\theta_7 + \theta_8} + s_{\theta_7} \right) \tag{C.12}$$

$$\Delta p_y = -a s_{\theta_6} \left(c_{\theta_7 + \theta_8} + c_{\theta_7} \right) \tag{C.13}$$

$$\Delta p_z = -a c_{\theta_6} \left(c_{\theta_7 + \theta_8} + c_{\theta_7} \right). \tag{C.14}$$

By squaring and adding all three of these equations together, the result is

$$\Delta p_x^2 + \Delta p_y^2 + \Delta p_z^2 = a^2 \left[\left(s_{\theta_7 + \theta_8} + s_{\theta_7} \right)^2 + \left(c_{\theta_7 + \theta_8} + c_{\theta_7} \right)^2 \right]$$

$$= 2a^2 \left(s_{\theta_7 + \theta_8} s_{\theta_7} + c_{\theta_7 + \theta_8} c_{\theta_7} + 1 \right)$$

$$= 2a^2 \left(1 + c_{\theta_8} \right) \tag{C.15}$$

$$\theta_8 = -cos^{-1} \left(\frac{\Delta p_x^2 + \Delta p_y^2 + \Delta p_z^2 - 2a^2}{2a^2} \right). \tag{C.16}$$

This negative sign is added to Eq. C.16 because $\theta_8 > 0$ is not within the joint limits of the left knee pitch joint angle. Using Eqs. C.13 and C.14, θ_6 can be determined.

$$s_{\theta_6} = \frac{-a\Delta p_y}{c_{\theta_7+\theta_8} + c_{\theta_7}} \tag{C.17}$$

$$c_{\theta_6} = \frac{-a\Delta p_z}{c_{\theta_7+\theta_8} + c_{\theta_7}} \tag{C.18}$$

$$\theta_6 = ATAN2\left(-\Delta p_y, -\Delta p_z\right). \tag{C.19}$$

Solving for the knee pitch θ_7 is more difficult. This requires Eq. C.12 to be solved twice, once for s_{θ_7} and once for c_{θ_7}, after expanding the equation through trigonometric identities.

$$s_{\theta_7} = \frac{\Delta p_z s_{\theta_8} + \Delta p_x c_{\theta_6}\left(1+c_{\theta_8}\right)}{2ac_{\theta_6}\left(1+c_{\theta_8}\right)} \tag{C.20}$$

$$c_{\theta_7} = \frac{\Delta p_x s_{\theta_8} c_{\theta_6} - \Delta p_z\left(1+c_{\theta_8}\right)}{2ac_{\theta_6}\left(1+c_{\theta_8}\right)} \tag{C.21}$$

$$\theta_7 = ATAN2\left(s_{\theta_7}, c_{\theta_7}\right). \tag{C.22}$$

The forward kinematics can be calculated when $\mathcal{F} = \mathcal{F}_L$, and the equations are very similar to the previous forward kinematic equations,

$$\Delta p_x = -a\left(s_{\theta_7+\theta_8} + s_{\theta_7}\right) \tag{C.23}$$

$$\Delta p_y = -as_{\theta_6}\left(c_{\theta_7+\theta_8} + c_{\theta_7}\right) \tag{C.24}$$

$$\Delta p_z = -ac_{\theta_6}\left(c_{\theta_7+\theta_8} + c_{\theta_7}\right), \tag{C.25}$$

except Eq. C.23 is multiplied by -1. If Δp_x is multiplied by -1, the same Eqs. C.19, C.22, and C.16, can be used to solve for θ_6, θ_7, and θ_8, respectively when $\mathcal{F} = \mathcal{F}_L$.

BIBLIOGRAPHY

1. Mx-28. Website, 2010. `http://support.robotis.com/en/product/dynamixel/rx_series/mx-28.htm`.

2. History of the study of locomotion. Website. `http://www.clinicalgaitanalysis.com/history/modern.html`.

3. Darwin op: Open platform humanoid robot for research and education. Website, 2011. `http://www.romela.org/main/DARwIn_OP:_Open_Platform_Humanoid_Robot_for_Research_and_Education`.

4. E. Ayyappa. Normal human locomotion, part 1: Basic concepts and terminology. *Journal of Prosthetics & Orthotics*, 9(1):10–17, 1997.

5. T. Katayama. Design of an optimal controller for a discrete-time system subject to previewable demand. *International Journal of Control, Automation and Systems*, 41(3):677–699, 1985.

6. K. Ogata. *Modern Control Engineering*. Prentice Hall, Indianapolis, fifth edition, 2010.

7. B. Siciliano and O. Khatib. *Springer Handbook of Robotics*. Springer, New York, illustrated edition, 2008.

8. M. Vukobratovic and J. Stepanenko. On the stability of anthropomorphic systems, mathematical biosciences. 15:1–37, 1972.