Passive Dynamic Walking with Knees: A Point Foot Model

by

Vanessa F. Hsu Chen

B.S., Electrical Science and Engineering (2005)
Massachusetts Institute of Technology

B.A., Physics (2005)
Wellesley College

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2007

Author . . . . . .
Department of Electrical Engineering and Computer Science
February 2, 2007

Certified by . . . . . .
Russ Tedrake
Assistant Professor
Thesis Supervisor

Accepted by . . . . . .
Arthur C. Smith
Professor of Electrical Engineering
Chairman, Department Committee on Graduate Theses

Passive Dynamic Walking with Knees: A Point Foot Model

by

Vanessa F. Hsu Chen

Submitted to the Department of Electrical Engineering and Computer Science
on February 2, 2007, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

In this thesis, a hybrid model for a passive 2D walker with knees and point feet is presented. The step cycle of the model has two phases of continuous dynamics: one with an unlocked knee configuration and a second one with a locked knee configuration. These stages are modeled as three-link and two-link pendulums correspondingly. The model switches between the two stages at knee-strike and heel-strike, which are both discrete events modeled as instantaneous inelastic collisions.

The dynamics of this model were fully derived analytically. Furthermore, a stable gait was found given a set of physical parameters and initial conditions. A basic stability analysis of this stable limit cycle was performed around the fixed point of the Poincaré return map examined right after heel-strike.

This thesis also presents the design and construction of a planar robot based on this kneed walker model, as well as a careful examination of its correspondence to the motion predicted by the model simulation. The goal is to be able to study the nonlinear dynamics of simplified dynamic models which are also physically realizable, in order to build robots based on them in a more rigorous and reproducible manner.

The work presented here aims to bridge the gap between existing theoretical models and successful experiments in passive dynamic walking.

Thesis Supervisor: Russ Tedrake
Title: Assistant Professor

Acknowledgments

Thanks to my advisor Russ Tedrake for offering his guidance and support throughout the project and in many other ways beyond the project itself, for sitting down with me to carefully go through specific concrete problems on many occasions, as well as for the many great abstract, ‘big-picture’ research conversations we had. I am most grateful for his passion and enthusiasm for robotics, which has inspired me and taught me to grow as a researcher and an engineer.

I’d like to thank Stephen Proulx for designing and building the robot and for his excellent intuition for mechanical design which proved to be indispensable while we debugged the robot. Thanks for the countless hours of work on the robot, for helping me fix the little quirks and for fixing it the many times we crashed the robot entirely. Thanks also to Katie Byl, for the compass gait results and her insights and intuition with the project in general, and to Rick Cory for working through two classes together and the helpful comments about my research. To the entire lab, thanks for all the fun group meetings and jokes at the lab, and for being a source of energy and learning for me.

I want to extend my thanks to my academic advisor, George Verghese, for his kindness and support and for giving me the liberty to pursue all my interests since I arrived at MIT, academic or non-academic. Thanks to my Physics advisor at Wellesley, Robbie Berg, for first introducing me to electronics and robotics. Also, thanks to Paula Echeverri, for persuading me to take that robotics class with her that first got me into robotics research.

Most importantly, I am immeasurably grateful to family: to my parents for all their love and support which has helped me get to where I am today, and to my sisters, Mimian and Wen, for always being there for me.

Contents

List of Figures

Chapter 1

Introduction

This thesis presents a mathematical model for a simple two-dimensional planar kneed walker, its stability analysis and comparison to data collected from a robot built after the model. The model is based on passive dynamic walking principles. I aim to bridge an existing gap between successful theoretical models and experimental results by proposing a model that is both simple enough to study its dynamics and stability without any linearizing, and complex enough to be able to build a robot faithful to the model's dynamics.

1.1 Motivation

As robots become more ubiquitous in our daily lives, enabling them to move efficiently through their environments is particularly important. Most of today's robots use simpler locomotion solutions using wheels or more than two legs, as these achieve stability more easily than two-legged solutions. However, in order for robots to be fully incorporated into our world, they must be able to move comfortably in a world built by humans, that is one built for bipedal locomotion.

1.1.1 Conservative Walking

Walking is inherently an underactuated problem, which means walking systems possess less actuators than degrees of freedom. Due to their free-moving feet, walkers cannot always produce arbitrary ground reaction forces at their ankles. Since they are not bolted down, this might cause the robots' feet to roll over and cause the robot to fall.

In maintaining static stability, a robot bypasses the problem of underactuation by simply avoiding positions where the ankles cannot produce the adequate ground reaction forces. If a robot has many legs, this constraint is not too restrictive, because the robot can lift one or more legs and still maintain a reasonable range of motion that is statically stable.

However, in the case of two-legged walking, it is particularly difficult to achieve stability on one hand

and efficiency on the other. Given a smaller polygon of support, bipeds need to constrain their motion considerably to achieve static stability at all times during their walking cycle. This results in high energy-losses during walking and very slow and awkward movement overall. They are also not capable of walking over any sort of rough terrain, as keeping a foot flat on the ground is necessary to exert the necessary torques at the ankles.

1.1.2 Dynamic Stability

Conversely, *dynamic stability* allows a system to have the ground projection of its center of mass leave the support polygon as long as it re-stabilizes again before falling over. By allowing this, dynamic gaits can allow more motion, but we then need other effective measures to ensure stability.

Many of today's most successful bipeds are controlled with position-control but use the system's center of pressure instead of the center of mass constraint. The center of pressure point is where all the ground reaction forces sum up to give a force but no moment. By keeping the center of pressure close to the center of the foot, these robots can avoid rolling over on their feet.

Honda's ASIMO robot [13], shown in Figure 1-1 walking down a set of stairs, is a very successful and well known example of such robots. It can kick a soccer ball as well as walk up and down stairs. However, these precise and impressive results are obtained with a high cost. ASIMO is considered to be about 20 times less efficient than humans are in walking [6]. This high rate of energy consumption propagates to decreased performance in many other ways. ASIMO's battery pack which makes up the majority of the robot's torso and mass, only lasts for 26 minutes. The maximum walking speed for ASIMO is 0.56 m/s. Moreover, ASIMO has only been shown to work on completely flat ground, given that its close-to-static-stability approach breaks down in rough terrain.

In the past few decades, in trying to find more dynamic gaits, *passive dynamic walking* (PDW) has been developed and has emerged as an important discipline of biped locomotion research.

Passive dynamic walkers exhibit a stable gait when placed on a downward slope with no actuation. The kinetic energy gained from going down an incline balances out exactly with the energy lost in the various collisions during the walk and other energy losses. These systems demonstrate how the inherent dynamics of walkers can (and should) be exploited to achieve natural and energy-efficient gaits.

Instead of explicitly controlling the trajectory of a system, we can let them simply 'fall' and have the natural dynamics of the system solve the control problem. PDW results in very natural-looking walks. This makes sense if we think of how we humans walk. As we walk, especially downhill, we do no stop ourselves at every step. On the contrary, we let our body fall forward, only stopping ourselves at each footstep.

While PDW is a powerful idea, its main drawback is stability. The natural passive limit cycles of the systems are highly susceptible to their specific construction and any slight perturbations during the gait. Currently, manual fine-tuning and 'a good hand' are necessary to start the robot in the right initial conditions for a successful trial. Simple control systems for these walkers have been designed in order achieve more

Figure 1-1: Honda Asimo

stability and also flat-terrain walking.

However, in order to achieve more rigorous and consistent results with PDW we need to fully understand the nonlinear dynamics of these systems and their stability and try to approximate the theoretical models of PDW to real physical systems.

1.2 Contributions and Organization

Results from passive dynamics walking have fallen mainly into one of two categories: on one hand, there are a number of simple dynamic models of walking [5, 11, 9], and on the other, a number of impressive machines have successfully been built [15, 16, 7]. Surprisingly, these results have emerged somewhat independently. Most experimental successes have resulted from the designers' intuition and relied more on mechanical tweaking than on a close correspondence with an analytical model.

The power of the dynamic models is their simplicity. The rimless wheel [5, 26] is analytically tractable. There exists a closed-form solution for the step-to-step return map and a complete analytical expression for the basins of attraction and convergence to the fixed points. Both the compass gait [11] and the simplest walking model [9] (which has the limiting dynamics of the compass gait with the leg masses approaching zero) are low-dimensional models and amenable to exhaustive computational analysis.

These models have point feet and point masses and assume inelastic collisions which cause an instantaneous transfer of support. These straight-legged models also ignore any premature ground contacts (foot scuffing). In contrast, experimental passive walking machines typically have curved feet, which increase their basins of attraction [25] and have some mechanism for achieving foot clearance (frontal plane oscillations

and/or knees). Collisions with the ground have some elasticity by design and some gaits include a short double-support phase.

My goal was to take some of the artistry out of successful passive dynamic design by understanding their dynamics. In this thesis, I describe a mathematical model developed for a two-legged planar biped with knees, using the concepts of passive dynamic walking. I also present the stability analysis of the model, as well as the experimental results of a physical robot matching this model. This thesis work tackles the dichotomy between the simplified models and existing experimental walking machines. Specifically, the contributions of this thesis are the following:

- A simple model of a planar passive dynamic walker with point feet and knees, making it both a logical extension of the compass gait model and a physically realizable model
- A basic dynamic systems analysis of kneed walker model, including local stability analysis and an approximation to the basin of attraction, as well as a comparison with those of the compass gait model
- Processing and parameter estimation of capture data from the motion of a robot built to match the kneed walker model

In terms of organization, Chapter 2 gives an overview of the most important passive dynamic walking and other relevant results in the field, both in theory and experiment. Chapter 3 is a detailed description of the proposed kneed walker model. The model's stability analysis is also presented here. In chapter 4, I discuss the physical robot built based on the kneed walker model, as well as the various difficulties which arose during its construction. I also discuss the data collection and processing from the test runs of this robot. Finally, Chapter 5 summarizes the conclusions drawn from the project and proposes directions for future work.

Chapter 2

Passive Dynamic Walking

In 1990, McGeer published a seminal paper [15] introducing the concept of passive dynamic walking (PDW). The basic idea behind passive dynamic walkers is that the energy lost in the inelastic collisions is balanced exactly by the kinetic energy gained going down a ramp. This allows statically unstable walkers to achieve dynamically stable limit cycles, resulting in a very natural and anthropomorphic motion without any external energy input.

McGeer demonstrated these ideas by building a real robot that could walk stably down a ramp using stepping stones to avoid foot scuffing. He also published a second paper describing a passive dynamic walker with knees[16].

2.1 Theoretical models

To understand how a stable gait is achieved, simplified models of such systems have been studied in detail.

2.1.1 The Rimless Wheel

The simplest mathematical of passive dynamic walking is the rimless wheel, as shown in Figure 2-1. The rimless wheel has rigid, massless spokes and a single point mass at the hip. The foot collisions are modeled to be perfectly inelastic while conserving angular momentum. Furthermore, it is assumed that there is no slip in the stance foot and that the stance leg is switched instantaneously. The system exhibits periodic motions at different velocities, given a certain leg length and slope angle.

With the assumptions made, the eigenvalues of the limit cycle can be solved analytically, and stability of the system is proven. The nonlinear dynamics and stability of the rimless wheel were studied extensively by Coleman and Ruina[5, 4].

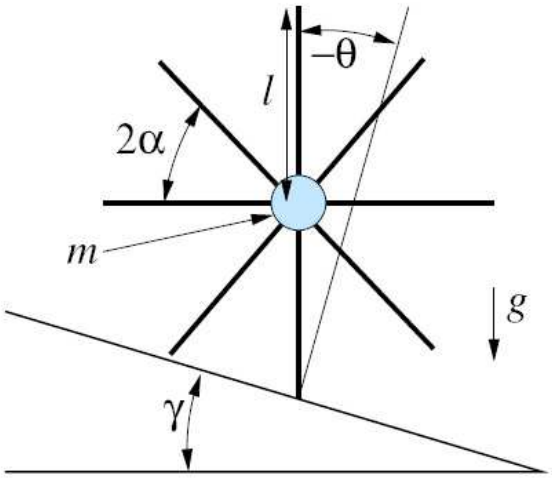

Figure 2-1: The Rimless Wheel model

2.1.2 The Compass Gait

If we take only two legs from the rimless wheel and place a pin joint between them, allowing the legs to swing freely, we arrive at the next model shown in Figure 2-2, the compass gait. For this case, the legs are modeled as point masses in the position of the center of mass, which are responsible for swinging the leg forward. The heel-strikes are modeled as inelastic collisions conserving angular momentum. The stance and swing legs switch instantaneously during the collision as well to go into the next step.

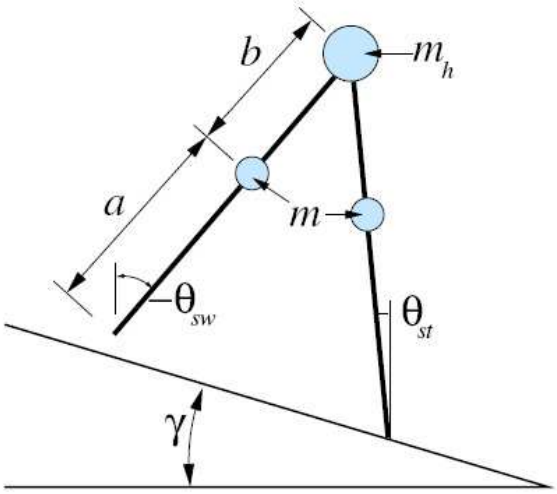

Figure 2-2: The Compass Gait model

This model, despite being extensively studied [10, 11], is not solved analytically. In numerical integration methods, the simulation of this model shows a stable limit cycle. Although it is hard to predict what gait it will end up in, it is able to hold these indefinitely.

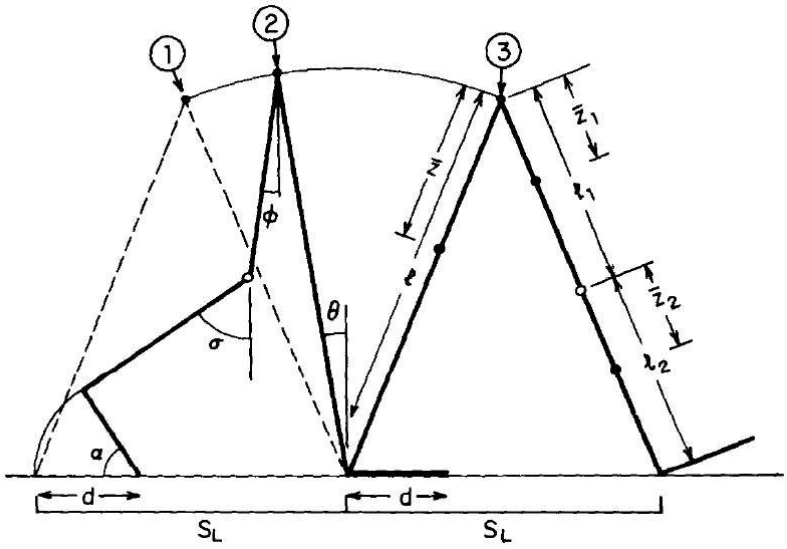

Figure 2-3: The Ballistic walker

A special case of the compass gait, called the simplest walker, places the masses at the legs and makes them infinitesimally small[9, 20, 14, 27]. These further simplifications allow the system to be characterized with only one parameter, the ramp slope. It is also important to point out that during the swing phase, the compass gait possesses the exact same dynamics as the Acrobot [23, 8, 22, 2], an underactuated two-link manipulator with only one actuator at the second joint (at the hip, for this case). The study and control of such system then helps us describe the underactuated problem common to all forms of robotic walking.

2.1.3 Kneed Walker Models

The next natural step as we build up the complexity of our walking modes, is to look at passive kneed walking. Adding knees has other benefits as well, such as achieving foot clearance and a more human-like gait. Moreover, the knees could potentially not just avoid foot scuffing but also help make bipeds that are capable of clearing rough terrain.

Mochon and McMahon presented the ballistic walker[17, 18], which is shown in Figure 2-3. The ballistic walker is a mathematical model of the swing phase of walking. The entire stance leg is represented as a pendulum fixed at its origin, assuming that the knee does not bend as the other leg is swinging forward. The swing leg is modeled as two shorter links connected to this first one, which start out straight with respect to each other and bend as the leg swings forward. This motion occurs completely passively.

Given that it is started in an appropriate configuration, the forward and bending motion of the swing leg are due entirely to gravity. This model only studies one step for the walker and does not model the impacts of the knee or foot at all. Therefore, the ballistic walker does not breach the issue of stability at all. Nevertheless, it characterizes the one-step cycle of a planar walker with knees.

Another kneed biped model [29, 19, 1] uses a knee-constraint force to lock the knee joint after it swings forward and a velocity field controller to find an active cycle which exploits the natural gait. However, there is no stability analysis of the system nor an experimental test for the specific model. Yamakita and Asano did build a kneed walker robot, but this one had curved feet and a control system to help it walk.

This thesis extends these studies by performing a complete dynamics analysis comparable to the studies of the rimless wheel and compass gait models for an equivalent model with an additional degree of freedom.

2.2 Experimental Results

A number of stable walkers [15, 16, 6, 28] have been built. For the machines without knees, stepping stones are used to avoid foot scuffing which requires special purpose terrain. Similarly, most of these walkers possess curved feet which help them achieve a stable gait. While the feet aid in balance and stability, they further distance these machines from existing theoretical models.

In order to make energy-efficient robots based on the idea of passive dynamic walking, which are also able to walk on level-ground, Collins, Wisse and Tedrake have all made actuated versions based on the McGeer walkers.

Collins and Wisse [6, 28] have each built their version of an actuated kneed walker. Both robots have careful mechanical designs for the foot shape, actuating muscles and knee locking mechanisms in order to achieve a stable gait. In the case of the Collins robot, the walker is highly sensitive to change in their physical parameters and initial conditions and would very easily fall into an unstable region. The Delft robot built by Wisse has a greater stable region.

The controls strategies implemented on these robots were designed to enable level-ground walking, but do not necessarily make them more robust. Wisse's robot, MIKE, for example, simply uses the actuators to pull the swing leg forward when a heel-strike is detected. This pumps energy into the system as gravity would in a downward ramp. Both MIKE and Collins' Cornell walker are shown in Figure 2-4. The stable gaits exhibited in these models with very simple control strategies imply that much of human walking depends on the morphology and not necessarily only on motor control. This highlights the importance of optimizing certain physical parameters to give the robots an adequate morphology for a stable gait.

Tedrake [24], on the other hand, designed Toddler, a 3D biped which used an online learning controller that would readjust to find a stable gait under changing conditions by using reinforcement learning methods. The adaptive nature of the controller enabled the robot to be robust to disturbances in the environments as it quickly re-learned an approximation of the optimal policy for the new condition.

Using an actor-critic algorithm, Toddler approximates the optimal control policy by climbing a gradient descent to improve performance. The intuitive notion behind this idea is that there is a cost function which rewards the desired behavior, which is a stable motion in the frontal plane (sagittal plane stability is taken care of with velocity control). Given this cost function and an estimate of the value function using a function

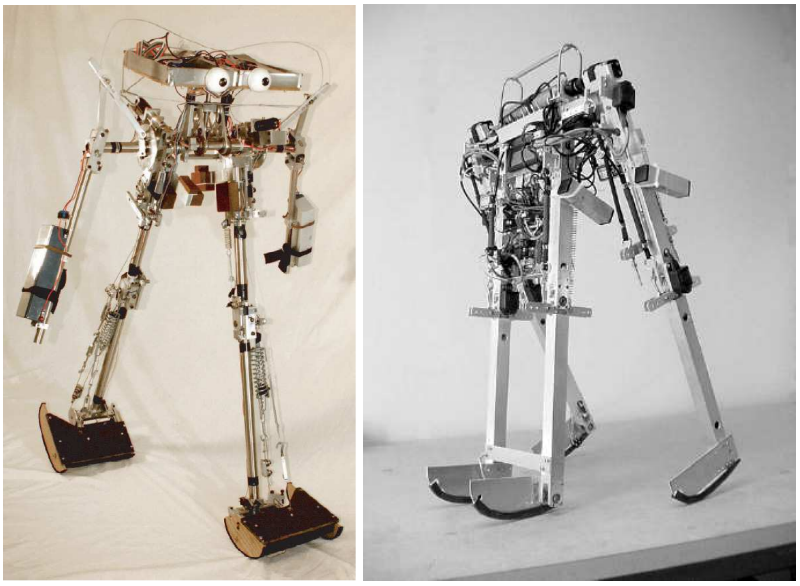

Figure 2-4: The Cornell walker built by Collins, Delft University's MIKE built by Wisse

approximator, he compared the actual cost with the one that was predicted and readjust the weights in the estimate accordingly. This readjustment of the weights eventually makes our function approximator converge to a controller that is very close to the optimal control.

The key advantage to this controller, however, is that it readjusts to different situations and it does it fast enough to be implemented online. Toddler successfully walked on carpets after having done the learning on a tiled floor, performing better than the feedback controller. This learning controller is also a better approximation of how animals adapt their behavior to find satisfactory walking gaits under different circumstances. From the way we move ourselves to how we see animals cross different terrains, we can conclude that a similar adaptation of a control policy is necessary for successful animal locomotion, even when given a well-designed but still fixed anatomy.

The kneed walker robot we built is unactuated in terms of its walking cycle (it possesses motors but only to trigger the knees at the right times), so that we can match it with the theoretical model. In this way, we can see how varying the different physical parameters and starting it at different initial conditions affects a point-foot walker with a small basin of attraction. Eventually, the goal would be to use a similar form of reinforcement learning to achieve a larger stable region for this walker.

Figure 2-5: Toddler, a 3D biped with optimal control

Chapter 3

Kneed Walker Model

The kneed walker model is an unactuated hybrid system placed on a downward incline with a fixed angle γ from the horizontal plane. A diagram of the model is shown in Figure 3-1 with its physical parameters.

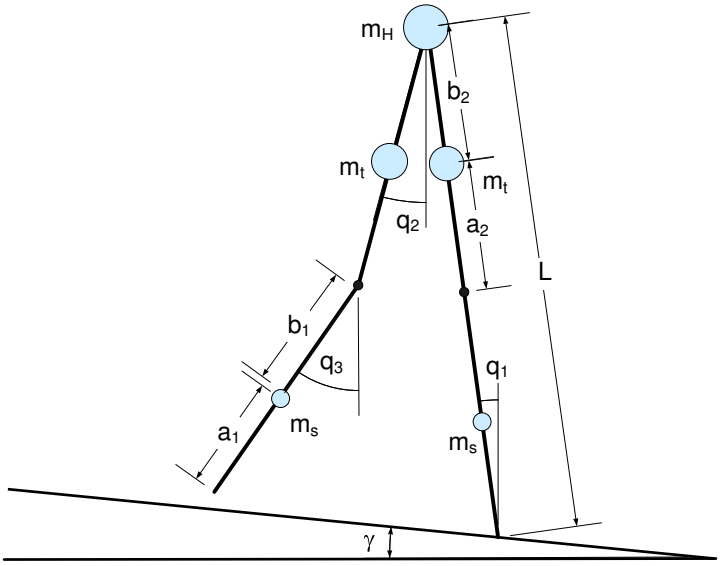

Figure 3-1: The Kneed Walker

From the figure, we note that all angles are defined globally from the vertical axis. Each leg has two point masses,m_t for the upper leg (thigh) and m_s for the lower leg (shank). There is also a mass at the hip, m_H. Also, the link lengths are composed as follows: length $L = l_t + l_s$, $l_s = a_1 + b_1$ and $l_t = a_2 + b_2$.

At the start of each step, the stance leg is modeled as a single link of length L, while the swing leg is modeled as two links connected by a frictionless joint. The system is governed by its unlocked knee dynamics until the swing legs comes forward and straightens out. When the leg is fully extended, kneestrike occurs. At this point, the velocities change instantly due to the collision, and immediately afterwards, we

switch to a two-link system in its locked knee dynamics phase.

The system remains in its locked-knee phase until the swing foot hits the ground. We model a heelstrike event here with the appropriate velocity changes. After this collision, the system returns to its initial unlocked swing phase. A diagram of the entire step cycle is shown in Figure 3-2.

It is also important to note that for the case when the heelstrike occurs before kneestrike (i.e. when the foot hits the ground with a bent leg), the robot will not be able to recover passively. Therefore, this is considered a falling state.

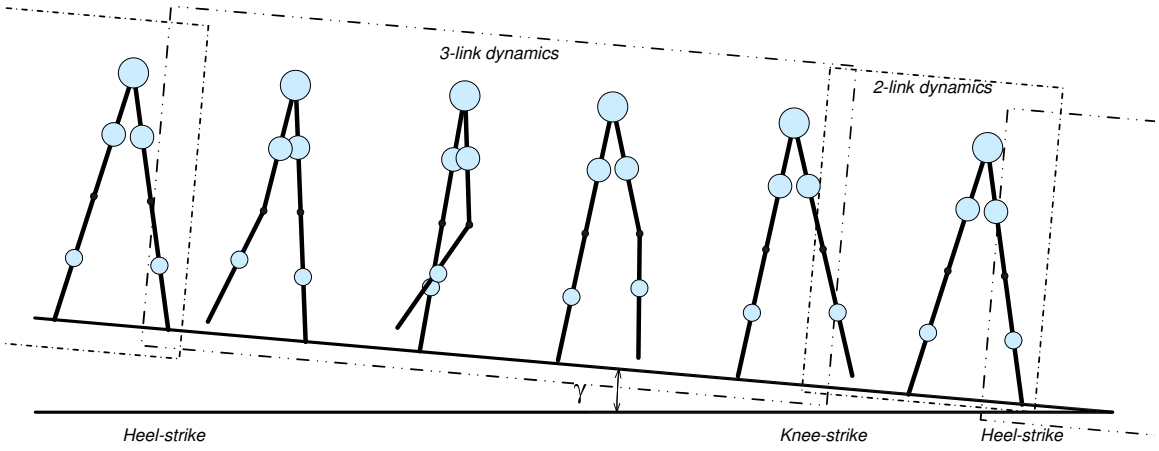

Figure 3-2: Stages in the step cycle for kneed walker

3.1 Continuous Dynamics

3.1.1 Unlocked Knee Dynamics

During the unlocked swing phase, the system is a three-link pendulum, like the ballistic walker [17, 18]. The full equations of motion for such a system are derived using Lagrangian formulation, which is clearly described in [21]. The dynamics are shown in the standard form of planar manipulator dynamics in Equation 3.1. The specific inertia, velocity-dependent and gravitational matrices for the three-link pendulum are given in Equation 3.2.

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{B}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = 0 \tag{3.1}$$

$$\mathbf{H} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{12} & H_{22} & H_{23} \\ H_{13} & H_{23} & H_{33} \end{bmatrix} \tag{3.2a}$$

$$\mathbf{B} = \begin{bmatrix} 0 & h_{122}\dot{q}_2 & h_{133}\dot{q}_3 \\ h_{211}\dot{q}_1 & 0 & h_{233}\dot{q}_3 \\ h_{311}\dot{q}_1 & h_{322}\dot{q}_2 & 0 \end{bmatrix} \tag{3.2b}$$

$$\mathbf{G} = \begin{bmatrix} -(m_s a_1 + m_t(l_s + a_2) + (m_h + m_s + m_t)L)g\sin(q_1) \\ (m_t b_2 + m_s l_t)g\sin(q_2) \\ m_s b_1 g \sin(q_3) \end{bmatrix} \tag{3.2c}$$

where,

$$H_{11} = m_s a_1^2 + m_t(l_s + a_2)^2 + (m_h + m_s + m_t)L^2 \qquad h_{122} = -(m_t b_2 + m_s l_t)L\sin(q_1 - q_2)$$
$$H_{12} = -(m_t b_2 + m_s l_t)L\cos(q_2 - q_1) \qquad h_{133} = -m_s b_1 L \sin(q_1 - q_3)$$
$$H_{13} = -m_s b_1 L\cos(q_3 - q_1) \qquad h_{211} = -h_{122}$$
$$H_{22} = m_t b_2^2 + m_s l_t^2 \qquad h_{233} = m_s l_t b_1 \sin(q_3 - q_2)$$
$$H_{23} = m_s l_t b_1 \cos(q_3 - q_2) \qquad h_{311} = -h_{133}$$
$$H_{33} = m_s b_1^2 \qquad h_{322} = -h_{233}.$$

Given appropriate mass distributions and initial conditions, the swing leg bends the knee as it swings forward. I found in simulation that a ratio of at least 1:5 between the masses of the lower and upper leg links is needed for this to happen, since the heavier upper leg gives the relative moments of inertia in the leg links necessary for a stable gait. At the instant the lower leg straightens out and aligns with the upper leg, a kneestrike collision is modeled.

3.1.2 Locked Knee Dynamics

After the kneestrike, the knee remains locked and we switch to double-link pendulum dynamics. The remainder of the swing phase occurs with straight legs. The dynamics for the newly-locked system are exactly those of the compass gait dynamics but with a different mass configuration. They are derived using Lagrangian formulation and shown in Equation 3.3 for completeness.

$$\mathbf{H} = \begin{bmatrix} H_{11} & H_{12} \\ H_{12} & H_{22} \end{bmatrix} \tag{3.3a}$$

$$\mathbf{B} = \begin{bmatrix} 0 & h\dot{q}_2 \\ -h\dot{q}_1 & 0 \end{bmatrix} \tag{3.3b}$$

$$\mathbf{G} = \begin{bmatrix} -(m_s a_1 + m_t(l_s + a_2) + (m_h + m_t + m_s)L)g\sin(q_1) \\ (m_t b_2 + m_s(l_t + b_1))g\sin(q_2) \end{bmatrix} \tag{3.3c}$$

where,

$$\begin{aligned} H_{11} &= m_s a_1^2 + m_t(l_s + a_2)^2 + (m_h + m_s + m_t)L^2 \\ H_{12} &= -(m_t b_2 + m_s(l_t + b_1))L\cos(q_2 - q_1) \\ H_{22} &= m_t b_2^2 + m_s(l_t + b_1)^2 \\ h &= -(m_t b_2 + m_s(l_t + b_1))L\sin(q_1 - q_2). \end{aligned}$$

When the swing foot touches the ground, we model another discrete event, the heelstrike collision. After this, we switch the stance and swing legs. This completes a full step, and we begin a new step using the three-link unlocked dynamics.

3.2 Discrete Collision Events

The collisions mentioned are the kneestrike and the heelstrike: these are modeled as instantaneous collisions. The post-collision velocities are calculated using conservation of angular momentum about the appropriate joints, given the pre-collision velocities. A more detailed explanation of how these conservation equations are derived is explained in Appendix A.

Furthermore, we assume there is infinite friction with any external surface, such that the system's configuration remains the same during the instantaneous collision.

3.2.1 Kneestrike

Instead of using a constraint force [29], we model the kneestrike as a discrete collision event in a three-link chain and switch to the compass gait model afterwards. Since the only external force on this system is at the stance foot, angular momentum is preserved for the entire system about the stance foot and for the swing leg about the hip. When looking at the lower link of the swing leg, however, the kneestrike acts as an external impulse. Therefore, angular momentum is not conserved about the knee.

Using these conservation equations, we obtain the post-collision velocities for the first two joint angles.

The third joint angle corresponding to the knee is locked after the collision. Therefore, its post-collision velocity will be that of the second link. We express the change in velocities as:

$$\mathbf{Q}^{+}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix}^{+} = \mathbf{Q}^{-}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\end{bmatrix}^{-} \tag{3.4}$$
$$\dot{q}_3^{+} = \dot{q}_2^{+}$$

The matrices, $\mathbf{Q}^{-}$ and $\mathbf{Q}^{+}$, are derived from the two conditions of angular momentum conservation mentioned above. The angular momentum before the knee collision is expressed as a function of all three joint angles, making $\mathbf{Q}^{-}$ a 2x3 matrix. After kneestrike, the system only has two joint angles as the knee is locked, and therefore, $\mathbf{Q}^{+}$ is a 2x2 matrix. The elements of each matrix are specified below.

$$\alpha = cos(q_1 - q_2)$$
$$\beta = cos(q_1 - q_3)$$
$$\gamma = cos(q_2 - q_3)$$
$$Q_{11}^{-} = -(m_s l_t + m_t b_2)L\cos\alpha - m_s b_1 L\cos\beta + (m_t + m_s + m_h)L^2 + m_s a_1^2 + m_t(l_s + a_2)^2$$
$$Q_{12}^{-} = -(m_s l_t + m_t b_2)L\cos\alpha + m_s b_1 l_t\cos\gamma + m_t b_2^2 + m_s l_t^2$$
$$Q_{13}^{-} = -m_s b_1 L\cos\beta + m_s b_1 l_t\cos\gamma + m_s b_1^2$$
$$Q_{21}^{-} = -(m_s l_t + m_t b_2)L\cos\alpha - m_s b_1 L\cos\beta$$
$$Q_{22}^{-} = m_s b_1 l_t\cos\gamma + m_s l_t^2 + m_t b_2^2$$
$$Q_{23}^{-} = m_s b_1 l_t\cos\gamma + m_s b_1^2$$
$$Q_{11}^{+} = Q_{21}^{+} + m_t(l_s + a_2)^2 + (m_h + m_t + m_s)L^2 + m_s a_1^2$$
$$Q_{12}^{+} = Q_{21}^{+} + m_s(l_t + b_1)^2 + m_t b_2^2$$
$$Q_{21}^{+} = -(m_s(b_1 + l_t) + m_t b_2)L\cos\alpha$$
$$Q_{22}^{+} = m_s(l_t + b_1)^2 + m_t b_2^2$$

3.2.2 Heelstrike

The heelstrike is modeled as an inelastic collision about the colliding foot. This heelstrike event is, again, identical to the heelstrike for the compass gait. Since the only external force occurs at the point of impact, there are no moments created around this point and therefore, no external torques act on the system. Angular momentum is then conserved for the entire system about the colliding foot and for the swing leg after impact about the hip.

Right after the event, the model switches both legs and the impact foot becomes the new stance foot. The model also switches back to the unlocked three-link dynamics to start a new step cycle. The third joint angle starts with the same angular position and velocity as the second one. This collision event is expressed in Equation 3.5.

$$\begin{aligned} \mathbf{q}^+ &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \mathbf{q}^- \\ \mathbf{Q}^+\dot{\mathbf{q}}^+ &= \mathbf{Q}^-\dot{\mathbf{q}}^- \\ \dot{q}_3^+ &= \dot{q}_2^+ \end{aligned} \tag{3.5}$$

Since heelstrike is assumed to only occur during the locked knee stage, the velocity transformation occurs for two joint angles only. Therefore, $\mathbf{Q}^-$ and $\mathbf{Q}^+$ are both 2x2 matrices. The individual elements of the velocity change matrices are shown below.

$$\begin{aligned} \alpha &= cos(q_1 - q_2) \\ Q_{11}^- &= Q_{12}^- + (m_h L + 2m_t(a_2 + l_s) + m_s a_1) L \cos\alpha \\ Q_{12}^- &= -m_s a_1(l_t + b_1) + m_t b_2(l_s + a_2) \\ Q_{21}^- &= Q_{12}^- \\ Q_{22}^- &= 0 \\ Q_{11}^+ &= Q_{21}^+ + (m_s + m_t + m_h)L^2 + m_s a_1^2 + m_t(a_2 + l_s)^2 \\ Q_{12}^+ &= Q_{21}^+ + m_s(b_1 + l_t)^2 + m_t b_2^2 \\ Q_{21}^+ &= -(m_s(b_1 + l_t) + m_t b_2) L \cos\alpha \\ Q_{22}^+ &= m_s(l_t + b_1)^2 + m_t b_2^2 \end{aligned}$$

3.3 Model Simulation

By switching between the dynamics of the continuous three-link and two-link pendulums with the two discrete collision events, we characterize a full cycle for the point-feet kneed walker.

Using the physical parameters shown in Table 3.1, the model is set on a ramp with a downward angle of $\gamma = 0.0504$ rad, and simulated to find a stable limit cycle. In general, we use a time step of 1^{-3} s. However, in order to find precise pre- and post- collision velocities, the collision events are detected with an accuracy of 1^{-20} s. Note that the upper leg mass is 10 times the mass of the lower leg. In general, we found that the greater the ratio between the upper and lower links, the more stable the walker became.

Table 3.1: Parameters for Kneed Walker Simulation

Parameter	Symbol	Value
leg length	L	1
shank length (below point mass)	a_1	0.375
shank length (above point mass)	b_1	0.125
thigh length (below point mass)	a_2	0.175
thigh length (above point mass)	b_2	0.325
hip mass	m_H	0.5
thigh mass	m_t	0.5
shank mass	m_s	0.05

In order to find the fixed point, we had initially run some local optimization methods unsuccessfully. Eventually, we tried examining the simulation's behavior in the different limiting cases. By setting the two point masses on the legs to be the same, we observed that a greater upper link mass was in fact necessary for the knee to bend. In order to see how much the motion compared to that of the compass gait, we ran the simulation with an infinitely small mass at the lower links. We initially tested a simulation with a 1:50 mass ratio in each leg, and started it close to the fixed point of the compass gait using the same mass on each leg as those on the upper link. This made the system walk stably.

After finding the fixed point for those parameters, we searched close to that point when reducing the mass ratio to reflect reasonable physical parameters. Eventually, using the parameters in table 3.1, we found the fixed point right after the heelstrike collision at,

$$q_1 = 0.1877$$
$$q_2 = q_3 = -0.2884$$
$$\dot{q_1} = -1.1014$$
$$\dot{q_2} = \dot{q_3} = -0.0399.$$

A limit cycle for the upper link of the right leg starting from this fixed point is shown in Figure 3-3. The instantaneous velocity changes from the heelstrike and kneestrike events can be observed in this limit cycle as straight lines where the cycle jumps with the instantaneous velocity changes while the positions remain the same. An energy plot, showing kinetic and potential energy, as well as total mechanical energy is also shown in Figure 3-4. In this plot, there are step increases in the potential energy on each foot transfer, since we write the energy of the system relative to the stance point. These increases are shown to exactly balance out the kinetic energy lost throughout one step. We can see that the final mechanical energy is constant throughout.

We see that the limit cycle closely resembles that of the compass gait. There is a swing phase (top half of the curve) and a stance phase (bottom half of curve) for each leg. In contrast with the compass gait, however, in addition to the two heel-strikes, there are two more instantaneous velocity changes produced by the knee-strikes. This limit cycle is traversed clockwise. If the walker starts anywhere slightly away from the fixed

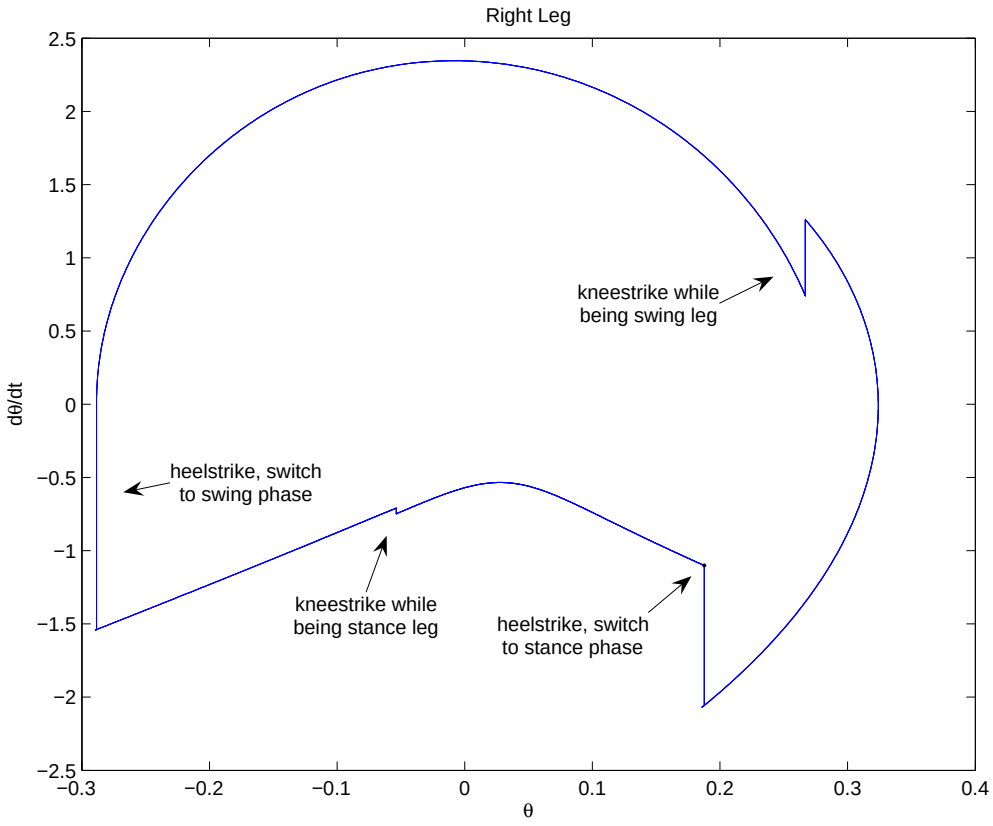

Figure 3-3: Limit cycle trajectory for the upper link of the right leg

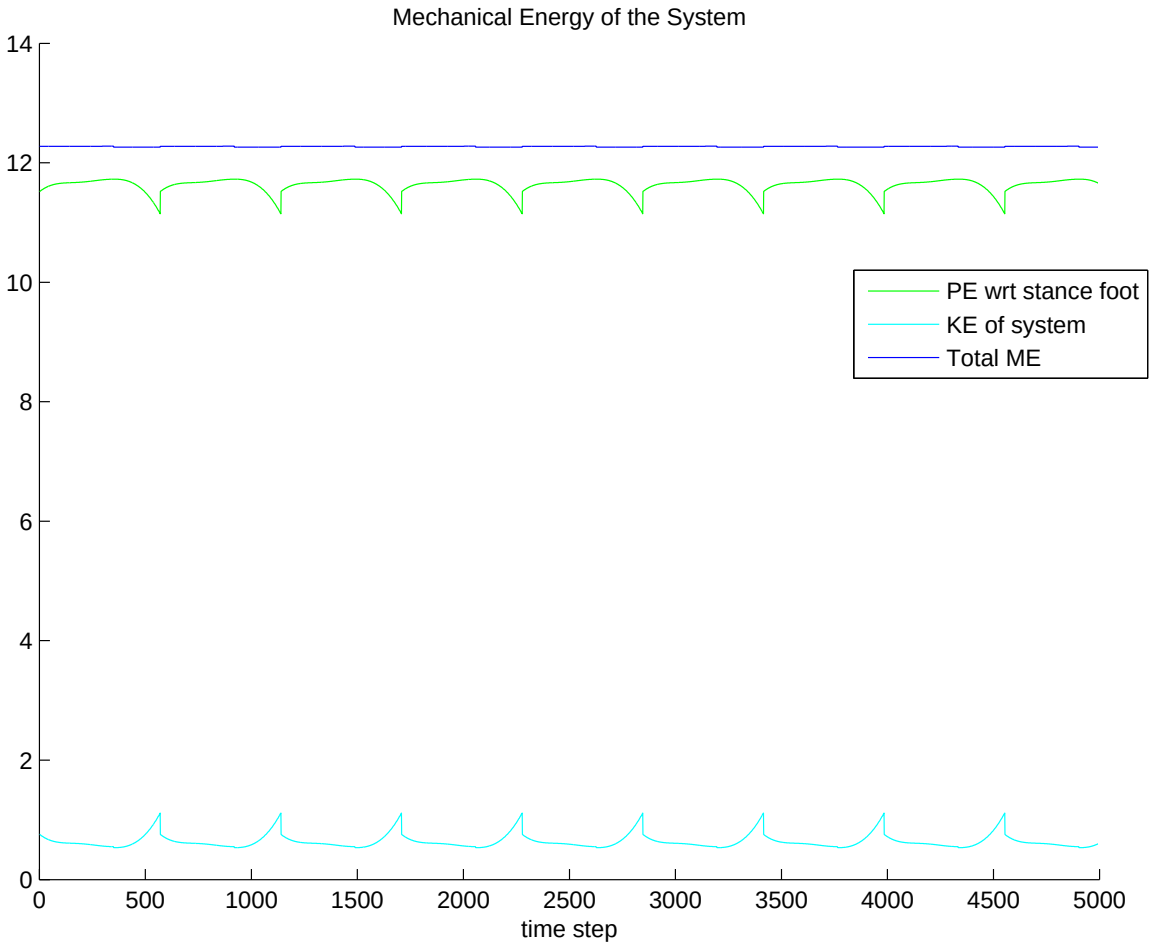

Figure 3-4: Mechanical energy of the system during a 5s simulation

point, it will converge to this limit cycle within a few steps.

The simulations of the model predict a stable gait when the mass in the upper leg is roughly 10 times the mass in the lower leg. In this limit, the return map and stability properties closely match that of the compass gait. This stability is accomplished without any assumptions about foot clearance since passive bending at the knees retracts the leg to avoid foot scuffing.

We also performed a few tests for the simulation in rough terrain. One of the main motivations for having knees in a walker is that it achieves foot clearance and could potentially help obstacle avoidance as well. The simulation showed that the model has no trouble clearing little perturbations in the terrain. Due to both its knees as well as the point feet, it does not require a flat surface to find a stance spot. Although large enough perturbations would cause the kneed walker to fall, many times the walker could sustain a series of uneven steps for at least 10 steps. This result is very promising, especially given that our model is completely passive. An example of the results from the rough terrain simulations are shown in Figure 3-5.

3.3.1 Local Stability Analysis

To study the stability of the robot, we first reduce the state space from six to three variables. Given a fixed ramp angle, γ and an interleg angle, α, we define the initial angular positions for the walker when it has straight legs (so, $q_3 = q_2$) and both feet are touching the ground. We also define the lower swing leg to start at the same velocity as the upper leg. Thus, the system is fully determined by the interleg angle α, and the stance leg and swing leg velocities, $\dot{q}_1$ and $\dot{q}_2 = \dot{q}_3$. Furthermore, for comparison purposes, this is convenient since there is an exact equivalent for the compass gait model in this configuration.

Given this dimension reduction, we can take the Poincaré section right after heelstrike to observe its first return map. We relate two successive heelstrike states with the step-to-step function F,

$$x_{i+1} = F(x_i).$$

We also define a fixed point, x^*, which is the point for which $x^* = F(x^*)$. To analyze the local stability, we look at the eigenvalues of the first return map which are calculated numerically. The kneed walker is started at random initial conditions perturbed slightly from the fixed point. The perturbations are smaller than 0.005 for each variable.

The first return map can then be expressed as follows,

$$(x_{i+1} - x^*) = \mathbf{A}(x_i - x^*).$$

In order to approximate **A**, data is taken for 25 runs 10 s each. A run is considered as converging if the last two states immediately after heelstrike have a mean distance between the last two of less than 0.001.

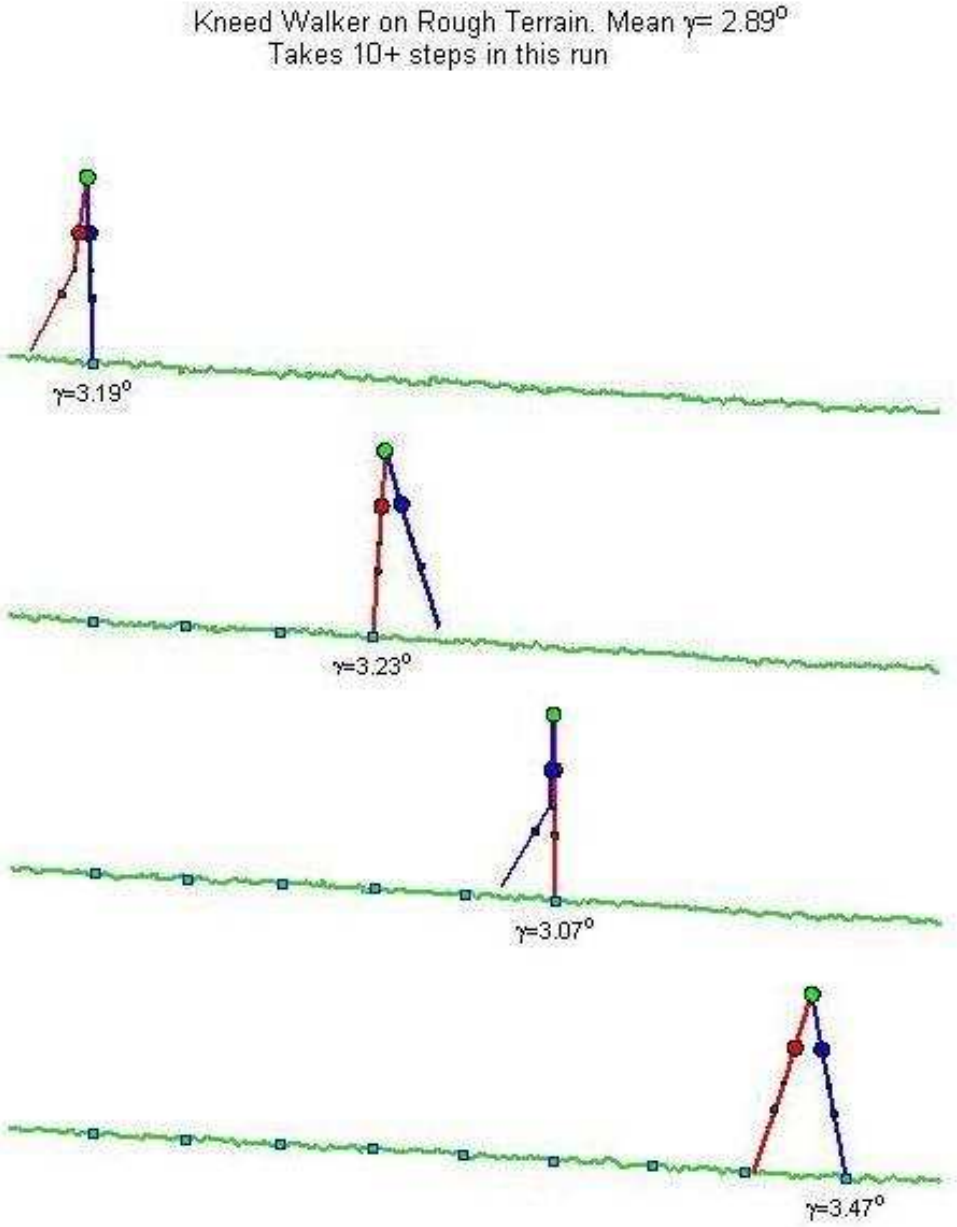

Figure 3-5: Screenshots from rough terrain simulation

With the data from the runs, we define two matrices,

$$\mathbf{X} = [(x_1 - x^*), (x_2 - x^*), ...]$$
$$\mathbf{Y} = [(x_2 - x^*), (x_3 - x^*), ...]$$

X and **Y** are extended after each trial by appending the data collected from each trial. We compute a least squares fit of A and find the eigenvalues of **A**. The eigenvalues found are 0.4053, and a complex conjugate pair, -0.2129 $\pm$ 0.3454i. Since all the magnitudes are within the unit circle, the system is locally stable.

Similarly, we compare these to the eigenvalues of a compass gait with an equivalent inertia to the kneed model but concentrated in 3 point masses. The eigenvalues found were 0.15 and 0.13 $\pm$ 0.57i. Although these values do not match, since they are all smaller than 0.5 in magnitude, any disturbance rejection occurs within half a limit cycle, so the impact of the initial disturbance on the steady state is negligible.

Moreover, if we take the same matrix without reducing the state space, we find the same eigenvalues and three zero eigenvalues. These correspond to the redundancy of the state space when described by all six variables.

3.3.2 Global Stability Analysis: Basin of Attraction

To gain more understanding about the system in general, we want to look at the global stability analysis. We compute the slice of the basin of attraction which crosses the fixed point. We simulated the dynamics for different initial velocities, using the fixed point's interleg angle (0.4761 rad with the stance leg in front). A state is considered stable if its last two steps are within a threshold of 0.001 m. We also assume that if a large number of steps (500) are taken, the model will walk forever. The resulting section of the basin of attraction is shown in Figure 3-6, with the fixed point marked with a star.

For comparison, Figure 3-7 shows the basin of attraction for the equivalent compass gait. Also, note that in Figure 3-6 there are two regions (colored differently). The larger region is computed by ignoring any foot scuffing, like the compass gait. In this case, the two basins of attraction are almost identical. This is not surprising, since the lower leg mass is much smaller than the upper leg, and the equivalent inertias of the two models result in essentially the same dynamics. However, when we take only the physically possible states by assuming that foot scuffing leads to a fall, the actual basin of attraction is shown to be a smaller region within the first overestimate.

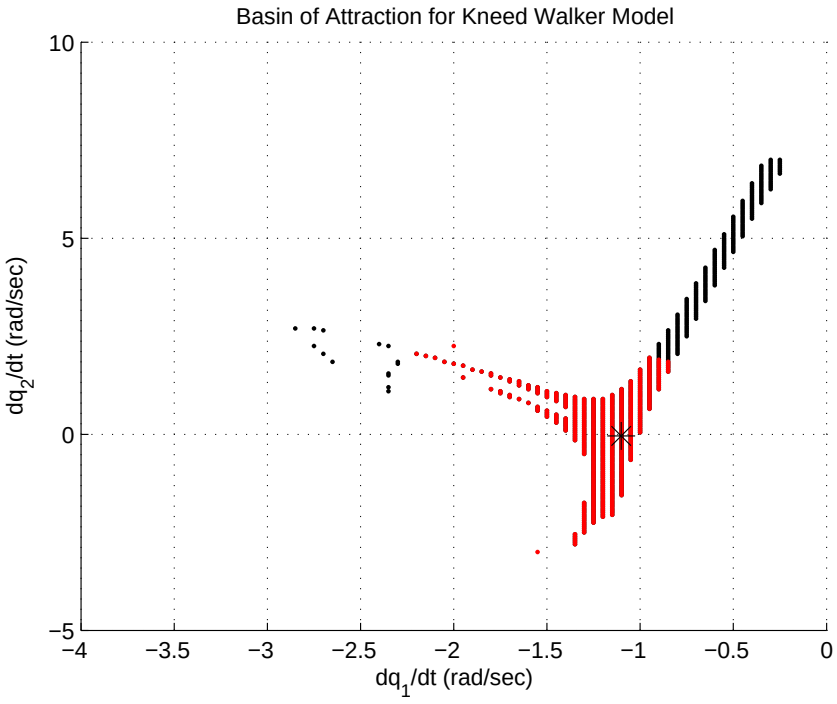

Figure 3-6: Basin of attraction of the kneed walker

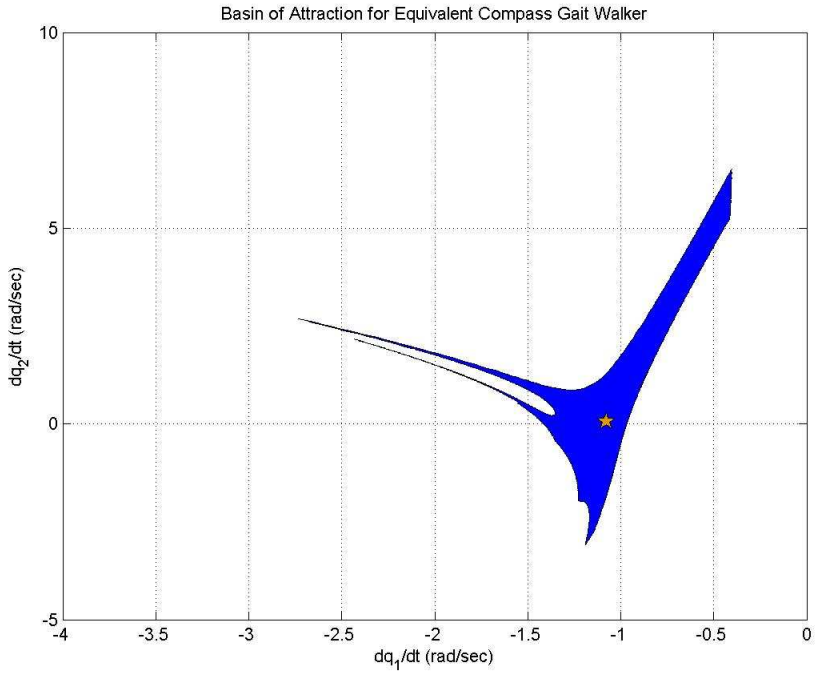

Figure 3-7: Basin of attraction of the equivalent compass gait

Chapter 4

Kneed Walker Implementation

Using the general guidelines from the simulation, we have built a planar robot with knees and point feet, which is shown in Figure 4-1.

4.1 Robot Design and Construction

The robot has two pairs of legs to ensure lateral stability. Each of the four upper legs has a mass of 0.491kg each. The upper legs are made out of 1.25” square tube aluminum, with a 1/8” wall thickness. The lower legs are made of 1/16” thick Garolite, that, along with a perforated construction make the lower legs about 1/4 of the mass of the upper legs. There is a horizontal bar connecting the two outer legs at the top, and a smaller leg underneath connecting the inner legs. In this way, we couple the motion of each pair of legs, with both sets rotating about the same axis. These bars give the robot’s ‘hip’ a mass of roughly one upper leg.

Although the ratio between the upper and lower masses is not 1:10, we could potentially add mass to the legs. We decided to avoid making the robot too heavy in favor of maneuverability, important particularly when starting the robot. Moreover, each link has uniform mass, so by adding the weights closer to the knees, we can concentrate the mass near the knee matching our simulation parameters.

4.1.1 Knee Design

The biggest challenge in the construction of the real robot was to come up with an adequate knee design. Other kneed walkers have successfully used suction cups together with curved feet. The suction cups make the legs lock when they swing forward, and then through a carefully tuned air valve, eventually release at the right time for the next step’s swing phase. This method not only required very careful tuning of the release valve, it was also greatly aided by these robots’ curved feet, as that would naturally exert force on the cup to release. In our case, we wanted to avoid such issues and iterated through a few different designs for the knee locking mechanism.

Figure 4-1: Kneed walker robot

Mechanical Latch

This was the initial design used to the knee, where a simple mechanical latch locked the lower link when it swung forward slightly past the straight-legged position. Small servos at each knee were used to unlock the latch by simply moving it from its central position. The servos were directed to release, via the foot switch, when the swing legs contacted the ground, causing the knees of the stance legs to release. After a short pause, the latch returned to its original position ready to lock the next time the lower leg swung forward. This would keep the knee locked for as long as we wanted without any further energy input.

Although this was a clean design that minimized actuation in the overall robot, its main issue was that the design required the lower leg to go past the straight-legged position. Even if the effect of this is neglected at the moment of kneestrike, the design allowed for an inherently 'loose' knee that would bend a few degrees forwards and backwards at all times after being locked. This changed our dynamics substantially. We had originally modeled our stance leg as a single longer link. The knee movement while locked made the leg behave more like two jittery links.

Another consideration was that the design of the knee itself was fairly complicated, requiring many small parts to work properly. A few diagrams and a photograph of the design are shown in Figure 4-2.

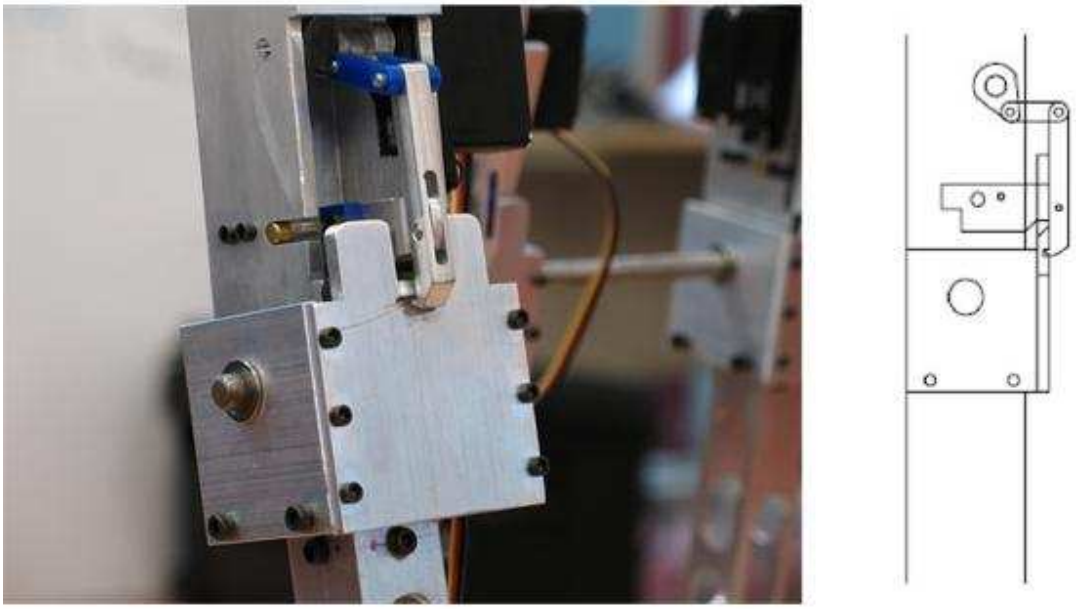

Figure 4-2: Mechanical latch for knee

Electromagnets

In order to prevent the legs from hyper-extending, we added knee caps that would stop the lower link when the lower leg came forward straightened out. These knee caps would then be kept in place by turning on electromagnets placed on the upper link. This change in our robot involved adding switch sensors at the knees that would trigger the electromagnets to turn on right at kneestrike.

The main problem with this option was the magnetic strength of the electromagnets. The space in our knees allowed for magnets only as big as 1" in depth and 0.75" in diameter. These magnets did not exert enough force to keep the leg straight even in a horizontal position.

Attempting to increase the magnetic force of the electromagnets, we added permanent magnets to the knee cap and also tried increasing the input voltage. Neither of these options, however, improved performance. When trying to run the robot, the weaker knees made it more difficult to start the robot without it collapsing before we released it.

Electromagnetic Clutches

We tried using electromagnetic clutches to exert more torque than the electromagnets at the knees. This also gave us the property of a rigid locked knee once the leg was fully extended. The clutches were placed on the side of each knee, as shown in Figure 4-3.

An initial problem was there was a delay in the lock signal, causing the knee to lock slightly after the knee switch was triggered, after the leg had bounced back from the fully extended position. In order to fix this, we added suction cups to the knee caps. These suction cups would stop the leg when it fully extended,

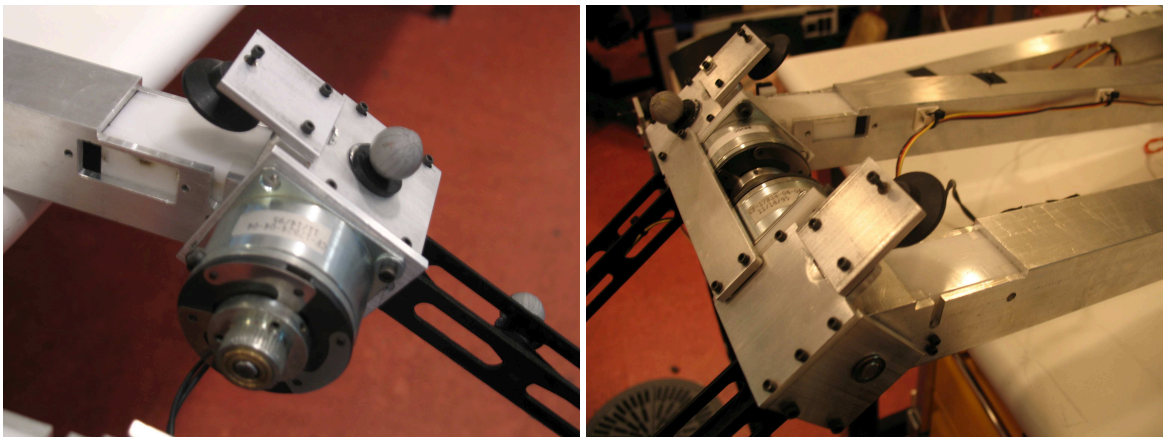

Figure 4-3: Electromagnetic clutch for knee

keeping it in a straight position, and then slowly releasing the air through small holes to detach from the knee cap. By the time the suction cups released, the clutches would have locked the knee and the leg then remains locked until the next heelstrike event.

4.1.2 Other construction issues

Other important issues for the robot included the choice and placement of the foot and knee switches, as well as an adequate foot design to avoid foot slip.

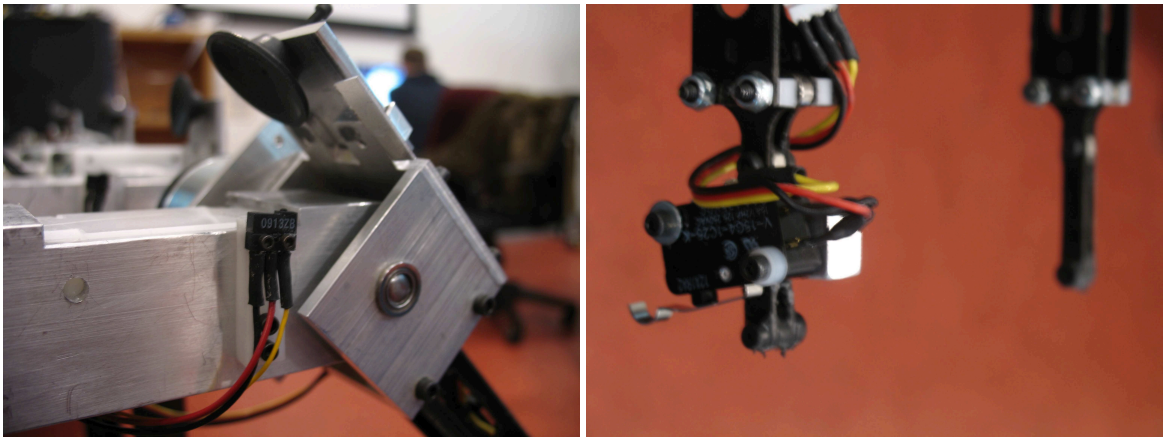

Figure 4-4: Details on knee and foot switches

We needed to provide switches that would trigger in parallel with the heelstrike and kneestrike events. For kneestrike, the switch should not obtrude the knee cap swinging forward in order to avoid being triggered before the leg is straight, causing too early of a switch signal. The foot switch, on the other hand, had to be placed at an angle such that its extension arm could make contact as the foot hit the ground. However, we had to be careful that the arm did not make the robot slip on that foot. Figure 4-4 shows a detail on both of these switches.

The foot switch detail also shows what the end of the legs looked like. The robot uses point feet, as specified by the model. In order to avoid foot slippage, the feet were dipped in PlastiDip multipurpose rubber coating, which claims to be flexible, durable, and to provide grip.

Lastly, we setup a cable that hung from above, sliding on a horizontal cable rail. This provided a safety catch for the robot for the falling trials, allowing us to repeat more trials without damaging the robot.

4.1.3 Control Sequence

Like the model, our robot has an entirely passive gait, which means that there is no energy introduced during its step cycle. However, the real robot needs a simple controller that powers the knees at the appropriate times of the step cycle to ensure that the knees lock and unlock at the right moments.

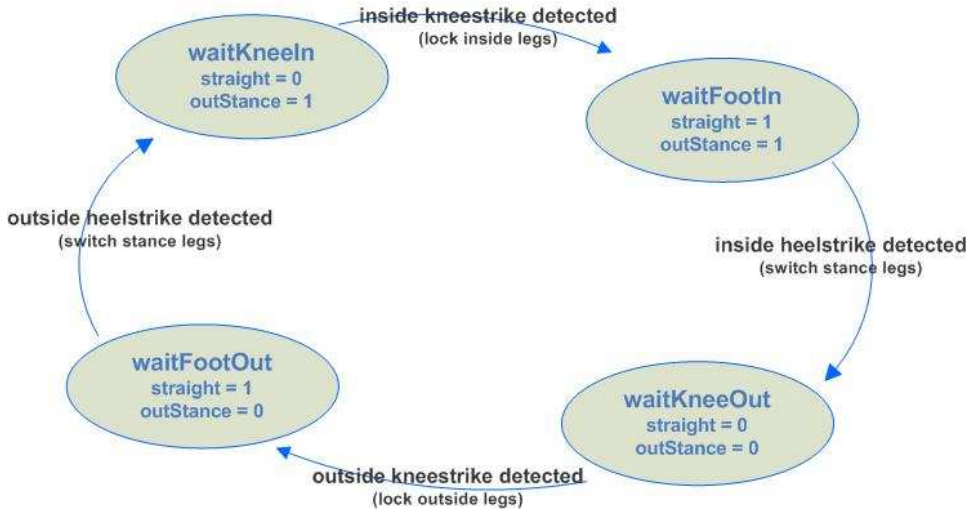

Figure 4-5: Finite State Machine for system's controller

To do this, we used a programmable servo motor controller board, SVBAS203B/C. The control sequence is illustrated in the finite state machine shown in Figure 4-5. Basically, we cycle through 4 states in which the robot waits for the next switch in consecutive order: inside kneestrike, inside heelstrike, outside kneestrike, outside heelstrike. At each heelstrike, we unlock the other pair of legs; whereas at each kneestrike, we lock the pair of legs which had the strike. This sequence will emulate a passive bipedal walking assuming kneestrike always happens before heelstrike, as our simulation does.

The controller board has 4 input sensors, 2 switches at the feet and 2 at the knee, which detect heelstrike and kneestrike for each pair of legs. The board can also output a PWM signal to servo motors or use these as digital outputs to other components, depending on which mechanism is being used to lock and unlock the knees.

The basic block diagram for the final design of the walker is shown in Figure 4-6. In this configuration, the controller board uses its output port as digital output signals that go through a 5V regulator and then to an H-bridge to provide the necessary power to the knees. However, in the initial design where the knees latched

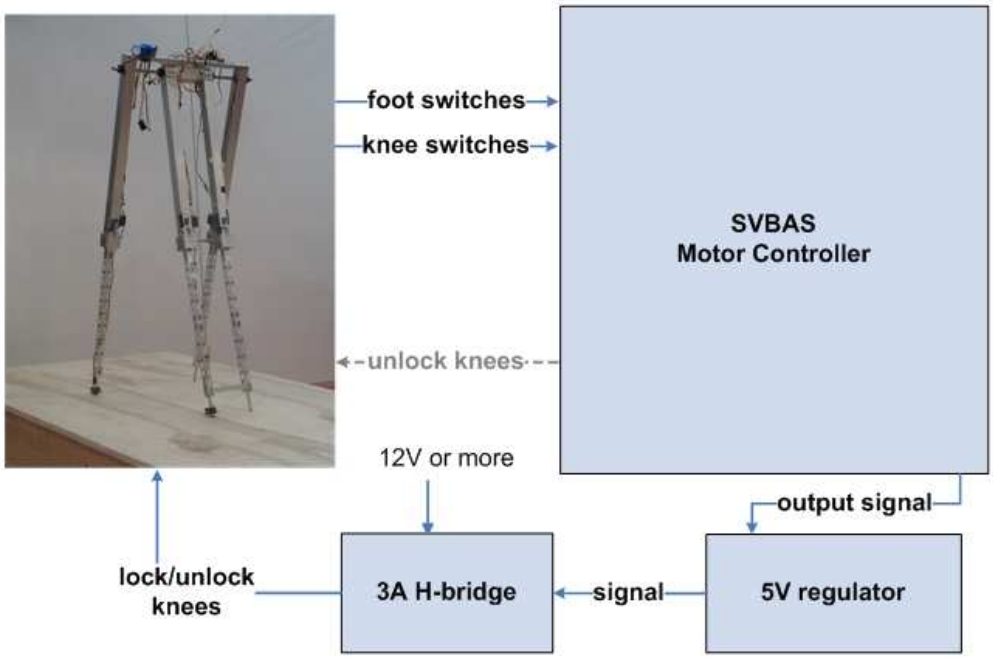

Figure 4-6: Block Diagram for kneed walker robot

mechanically, we did not need to put any power to keep the knees locked. For this configuration, we simply connected servo motors directly to the board, skipping the regulator and H-bridge. This simpler configuration is shown in grey in the diagram.

4.2 Data Collection

We ran numerous trials using the various different designs, while trying to fix the causes for mechanical failures each time. We placed our robot hanging from a cable, on a 12' long ramp built on plywood. The elevation angle of the ramp is $\gamma = 0.0504$. For all of the trials, the robot successfully took a few steps before falling over.

In order to be able to quantitatively describe the actual robot's gait, and match the robot's motion to the theoretical model, we use a motion capture system to collect and analyze data. The system consists of 16 infrared cameras placed around a rectangular structure hung above the subject. The subject has markers with reflective material placed on multiple points, which are used to track its motion within the capture volume. We used a Vicon motion capture system and software, which has sub-millimeter resolution.

In any given trial, the marker positions are captured and then labeled and calibrated to a specific model. In this way, we remove the need for online sensors, which typically provide noisy data. Using Vicon's capture, modeling and calibration software, we can collect real data from the walking runs. Motion capture is often

used on humans who wear a suit with the markers. This naturally allows for more error than using a robot with solid legs on which we can fix the markers.

4.2.1 Skeleton and Calibration

The first step needed to use the motion capture system was to create a 'skeleton' for the robot. To do so, we define a model following our robot's dimensions and place markers on different locations. By tracking the exact position and motion of the markers, we are able to define the robot's links and their corresponding features (i.e. length, rotation and relative motions). Instead of measuring and manually specifying precise distances between the markers, we can parameterize them in the model and allow the motion capture system to calibrate them with real data. The model built for the walker is shown in Figure 4-7. The markers are shown by red and green points that stem off from their parent segment. Also, the pair of inner legs are modeled as one leg only, since their motion is kinematically coupled. The markers on the two inner legs are marked on either side of this link in the model.

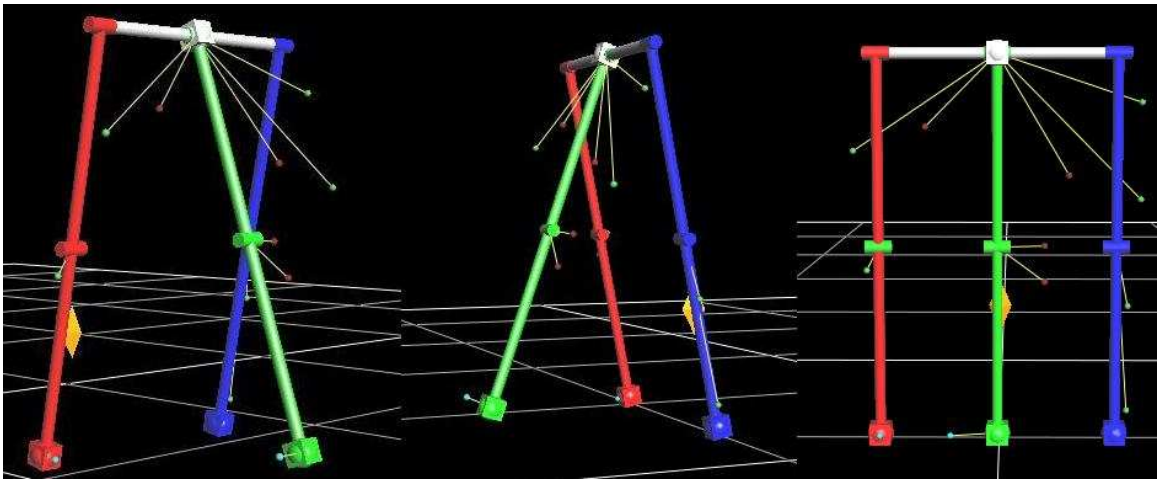

Figure 4-7: Skeleton template for Kneed Walker

Before using this skeleton, we had to capture a range of motion trial to calibrate the template with our specific robot. This allowed us to obtain accurate data in each trial with the calibrated subject. Our initial testing in trying to calibrate the template were giving us a large error (¿25mm). As recommended by the Vicon software manuals, we wanted to find an error smaller than 10-15mm in the calibration. Part of the problems was that by manually moving the robot and swinging its legs back and forth, we were also occluding some of the markers, giving less accurate data. In order to fix this, we hung the robot from above and moved each leg individually. This also allowed the other links to move freely while we moved one of them. After doing so, our calibration error diminished considerably and was within the desired range.

4.2.2 Data Capture

After obtaining a calibrated subject, we placed the robot on the ramp for the trials. On most trials, the robot would take 3-4 steps before falling forward. These trials were recorded with both the motion capture software and on digital videotape.

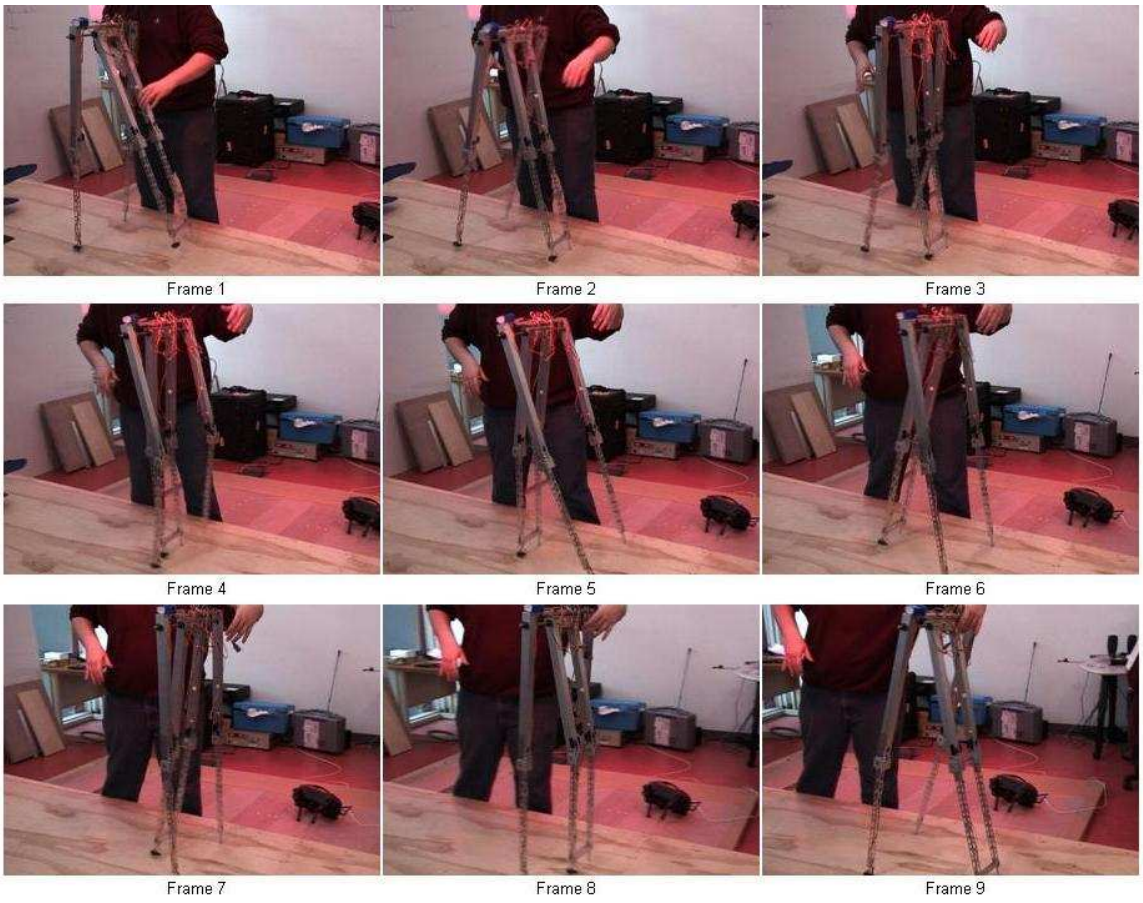

Figure 4-8: Nine progressive screenshots of video footage of the trials

Figure 4-8 shows screenshots of the digital video taped from a trial using the first legs. The frame rate shown is 5 frames per second and we show the progression between the start of a trial and right before the robot falls over after taking a few steps.

After review of the different trials on video, we could quantitatively describe the main issues in the mechanical robot. Firstly, since the outer legs are not physically coupled except for at the top bar, the outer feet would not necessarily be lined up, making the robot steer in one direction or another. This was minimized considerably after we switched the knee design to make it stiffer. However, the foot misalignment was still present.

Also, having the outer legs uncoupled below the knee also made the knee lock timing extremely sensitive. When the legs swung forward, they would not necessarily lock at the same time. Likewise, if the suction cups had not entirely released the leg at the time the clutches released (since they were still close to the plastic

plate), the two legs might also start bending asynchronously. There does not seem to be a simple way to couple these outer legs without significantly changing the dynamics.

The difference in bending also causes foot scuffing on occasions and uneven landings. These, in turn, sometimes cause the feet to slip or bounce after heelstrike. This last problem could potentially be reduced by placing a mat on the wooden ramp to increase damping.

4.3 Data Processing

To quantify and analyze these different issues and have a more rigorous comparison between the model and the robot, we reconstructed and processed the collected data with the motion capture software. After calibrating and fitting the data, we could playback the exact motion recorded at 60 frames per second (fps).

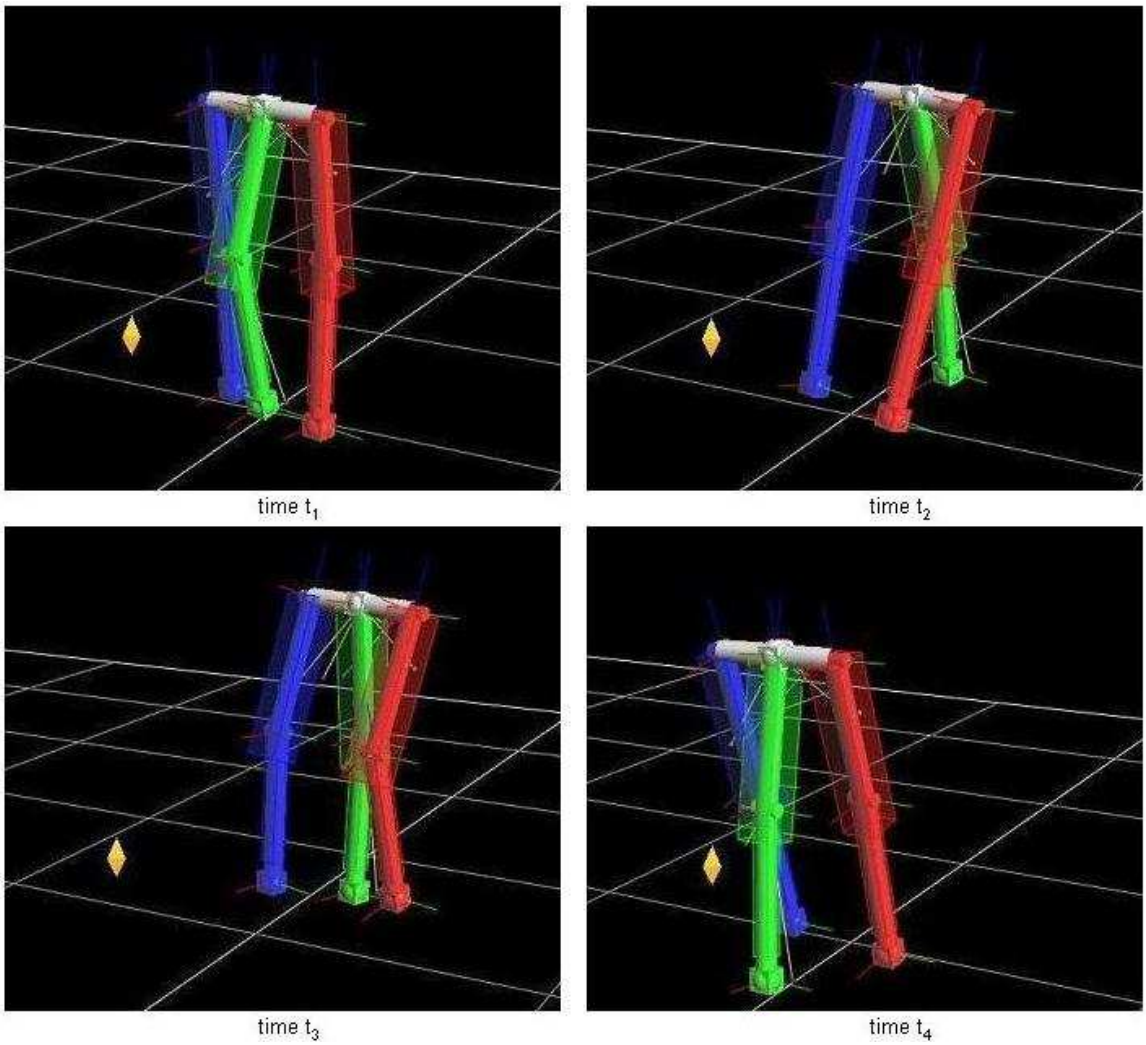

Figure 4-9: Four progressive screenshots of motion captured data

An example of the how the reproduced data is shown in the Vicon software is shown in Figure 4-9. These

are progressive shots of a specific trial taken at times $t_1 < t_2 < t_3 < t_4$, although the time between each shot is not the necessarily same. All data shown hereinafter refers to data from this same trial.

We exported the Vicon data into a comma-separated file containing the positions and rotations of each segment defined in the model skeleton for each frame. This data was then loaded and parsed so that we could reproduce the motion in a Matlab simulation, and also derive the velocities and accelerations.

The first step in the data processing needed to compare it with our planar model is to convert the obtained data into 2D as well. To do so, we take rotation between the three segments at the hip (one for each of the robot's leg) and rotate our entire data set so that they are aligned in the y-axis. After doing so, we can playback the data using the calibrated positions, as shown in Figure 4-10.

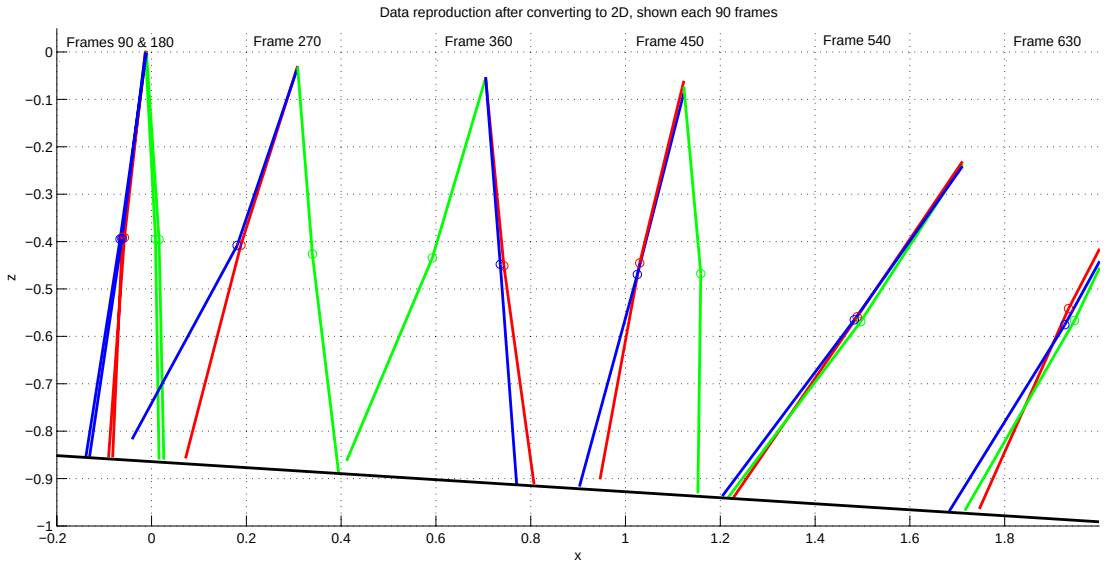

Figure 4-10: Screenshots of parsed data converted to 2D

In Figure 4-10, the blue lines represent the outer right leg, the red lines are the outer left leg, and the green shows the position of the inner legs. For this trial, the data was rotated 0.2822 in the y-axis to obtain this planar data. To show the progression of the trial (originally 689 frames in length), we show a frame for every 90 frames recorded. It is worth noting that the outer legs do not exactly match up on the plane and move in a significantly disjointed manner. This trial was taken while the robot used the mechanical latch. Better coupled motion was seen in the later trials.

Secondly, we need to derive the angular velocities and accelerations from the position data. To obtain velocities, we simply take our position data and take the difference in each frame over the time elapsed to obtain velocities. We do this again to obtain accelerations. On both occasions, to reduce the noise from this rough derivation, we filter the resulting data with a low-pass filter at a sampling rate of 120 Hz with a cutoff frequency of 10Hz.

Finally, we resample the data at twice the rate such that all three vectors correspond to the same time

instants. The resulting angular positions, velocities and accelerations for a single swing phase of the trial above is shown in Figure 4-11. Using the start of this swing phase as the initial conditions, we can also run our simulation to compare. These results are shown in Figure 4-12.

In the graphs, the blue line corresponds to the angle of the first link (the stance leg) whereas the green corresponds to the second link (the upper link of the swing leg) and the red to the third link (the lower link of the swing leg). For the real data plots, we notice how for the joint angles, the stance leg starts in a positive angle (going back from the vertical position) and smoothly moves to a mirroring position while the leg swings forward. At the same time, both links on the swing leg start from a negative angle. As the angle of the upper link smoothly increases, the lower link first decreases as the knee bends, and then increases when the swing leg starts straightening out. Both links end the swing step at a roughly the same angle as well, as the knee locks and the robot switches to two-link pendulum dynamics.

Roughly, the simulation shows very similar behavior, except it is going much faster overall. The swing phase occurs in a total of 0.25 s, while it takes the real robot about 0.4 s. Furthermore, since the two links of the swing leg match up exactly, these positions and velocities are the same in the beginning. The two joint angles also end up in the same point after the leg swings forward and extends. The velocity and acceleration curves for the swing leg also seem more uniform in the simulation data. These differences are mostly likely consequences of a difference in the mass distribution of the simulation and that of the real robot.

4.3.1 Parameter Estimation

To do a careful comparison between our model and the robot's motion, we can use parameter estimation to reduce the parameter uncertainty [21, 12]. This means inferring the values of our parameters from the robot's recorded motion.

To do this, we can re-write the dynamics of the system as shown in Equation 4.1. After doing so, the parameter estimation problem turns into a linear estimation problem.

$$\mathbf{W}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \cdot \mathbf{x} = \mathbf{y}, \tag{4.1}$$

In Equation 4.1, **a** represents a vector containing the unknown parameters of the system; **W** is the matrix with the remaining terms containing only terms combined from different measurable state variables (i.e. angular positions, velocities and accelerations for each link); and **y** usually contains the external torques applied to the system. In this specific case, **y** is just the zero vector since we are dealing with unactuated dynamics.

Now, by parameterizing our unlocked dynamics, we obtain the matrix and vectors shown in Equations 4.2.

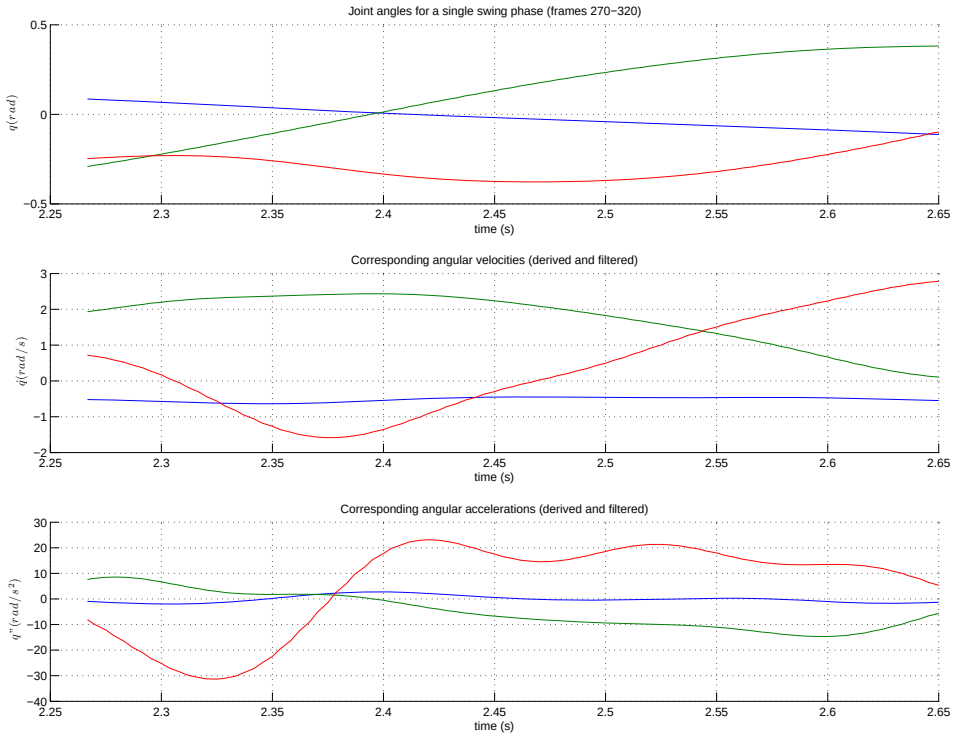

Figure 4-11: Swing phase of collected data

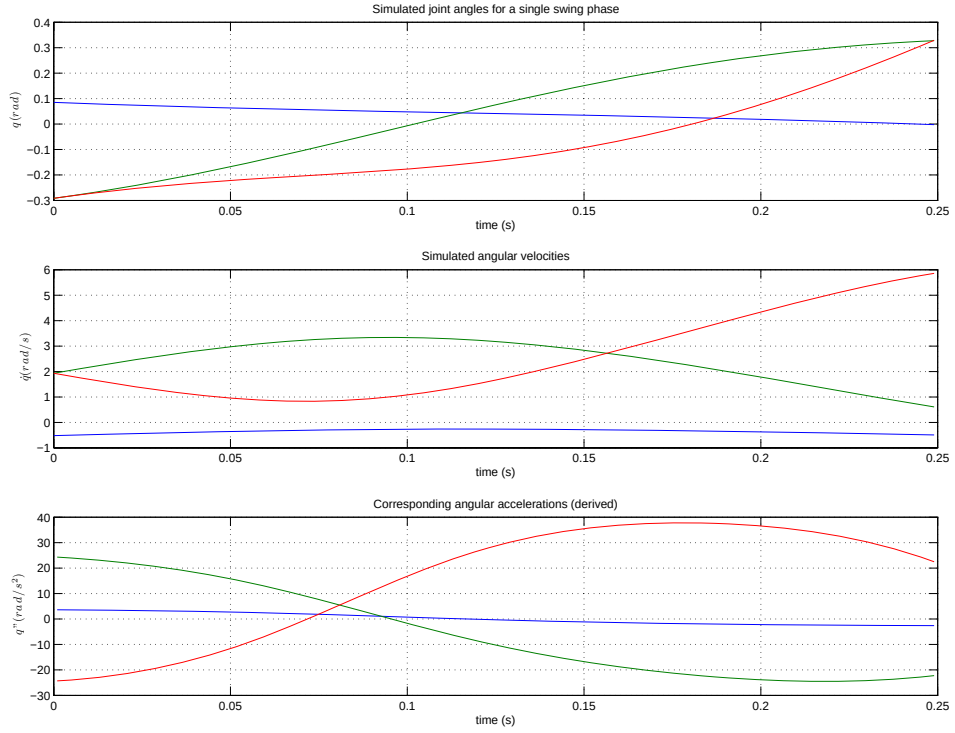

Figure 4-12: Simulated swing phase using first frame of collected data as initial conditions

$$\mathbf{W} = \begin{bmatrix} \ddot{q}_1 & c_{12}\ddot{q}_2 + s_{12}\dot{q}_2{}^2 & c_{13}\ddot{q}_3 + s_{13}\dot{q}_3{}^2 & s_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_{12}\ddot{q}_1 - s_{12}\dot{q}_1{}^2 & 0 & 0 & \ddot{q}_2 & c_{23}\ddot{q}_3 + s_{23}\dot{q}_3{}^2 & s_2 & 0 & 0 \\ 0 & 0 & c_{13}\ddot{q}_1 - s_{13}\dot{q}_1{}^2 & 0 & 0 & c_{23}\ddot{q}_2 - s_{23}\dot{q}_2{}^2 & 0 & \ddot{q}_3 & s_3 \end{bmatrix} \tag{4.2a}$$

$$\mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 \end{bmatrix}^{\mathbf{T}} \tag{4.2b}$$

$$\mathbf{y} = \mathbf{0} \tag{4.2c}$$

where,

$$\begin{aligned} a_1 &= m_s a_1^2 + m_t(l_s + a_2)^2 + (m_h + m_s + m_t)L^2 & c_{12} &= \cos(q_2 - q_1) \\ a_2 &= -(m_t b_2 + m_s l_t)l & s_{12} &= \sin(q_2 - q_1) \\ a_3 &= -m_s b_1 L & c_{13} &= \cos(q_3 - q_1) \\ a_4 &= -(m_s a_1 + m_t(l_s + a_2) + (m_h + m_s + m_t)L)g & s_{13} &= \sin(q_3 - q_1) \\ a_5 &= m_t b_2^2 + m_s l_t^2 & c_{23} &= \cos(q_3 - q_2) \\ a_6 &= m_s l_t b_1 & s_{23} &= \sin(q_3 - q_2) \\ a_7 &= (m_t b_2 + m_s l_t)g & s_1 &= \sin q_1 \\ a_8 &= m_s b_1^2 & s_2 &= \sin q_2 \\ a_9 &= m_s b_1 g & s_3 &= \sin q_3. \end{aligned}$$

To find estimates for the parameter vector, **a**, we use a linear least-squares estimator (LLSE). Like the name suggests, a LLSE reduces the squared error, defined as $\|\mathbf{W}\mathbf{a} - \mathbf{y}\|^2$. By taking the derivative of this with respect to **a**, we can explicitly derive the value for $\hat{\mathbf{a}}$ that gives the smallest error using an Euclidean norm (Equation 4.3).

$$\hat{\mathbf{a}} = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{y} \tag{4.3}$$

However, in this case, the **y** vector is zero because our system is completely unactuated. To use the estimator, we artificially 'lock down' at least one of the parameter and estimate the other parameters relative to this one. This solution works precisely because we have zero torques, so the scale of the system is entirely irrelevant to our dynamics outputs. We need a combination of parameters which will give rise to the same dynamics of our robot, but basically, these could be in either microscopic or huge dimensions. For our estimation problem, scale is not just irrelevant but also undetermined unless we apply external torques to the system or manually specify one of the parameters.

To do this, we found the physical measurements that would most easily define one or more of the parameters in **a**. We defined $m_s = 0.1\text{kg}$ and $b_1 = 0.25\text{m}$. By doing so, a_8 and a_9 are determined and used as known quantities that give us a non-zero **y** vector. This results in a new set of parametrization matrices, shown in Equation 4.4.

$$\mathbf{W} = \begin{bmatrix} \ddot{q_1} & c_{12}\ddot{q_2} + s_{12}\dot{q_2}^2 & c_{13}\ddot{q_3} + s_{13}\dot{q_3}^2 & s_1 & 0 & 0 & 0 \\ 0 & c_{12}\ddot{q_1} - s_{12}\dot{q_1}^2 & 0 & 0 & \ddot{q_2} & c_{23}\ddot{q_3} + s_{23}\dot{q_3}^2 & s_2 \\ 0 & 0 & c_{13}\ddot{q_1} - s_{13}\dot{q_1}^2 & 0 & 0 & c_{23}\ddot{q_2} - s_{23}\dot{q_2}^2 & 0 \end{bmatrix} \tag{4.4a}$$

$$\mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \end{bmatrix}^{\mathbf{T}} \tag{4.4b}$$

$$\mathbf{y} = \begin{bmatrix} 0 & 0 & -m_s b_1^2 \ddot{q_3} - m_s b_1 g s_3 \end{bmatrix} \tag{4.4c}$$

where,

$$a_1 = m_s a_1^2 + m_t(l_s + a_2)^2 + (m_h + m_s + m_t)L^2 \qquad c_{12} = \cos(q_2 - q_1)$$
$$a_2 = -(m_t b_2 + m_s l_t)l \qquad s_{12} = \sin(q_2 - q_1)$$
$$a_3 = -m_s b_1 L \qquad c_{13} = \cos(q_3 - q_1)$$
$$a_4 = -(m_s a_1 + m_t(l_s + a_2) + (m_h + m_s + m_t)L)g \qquad s_{13} = \sin(q_3 - q_1)$$
$$a_5 = m_t b_2^2 + m_s l_t^2 \qquad c_{23} = \cos(q_3 - q_2)$$
$$a_6 = m_s l_t b_1 \qquad s_{23} = \sin(q_3 - q_2)$$
$$a_7 = (m_t b_2 + m_s l_t)g \qquad s_1 = \sin q_1$$
$$s_2 = \sin q_2 \qquad s_3 = \sin q_3.$$

Parameter Estimation on Simulated Data

To test the LLSE with a fixed parameter method, we ran the estimator with a set of 100 points of simulated data first. To do so, we generated the data set of inputs using the system's dynamics. After picking arbitrary initial conditions, we simulate forward the original dynamics equations, using hand-picked link lengths and masses within a reasonable range. The parameter vector **a** is in turn defined with these physical parameters, and is shown below,

$$\mathbf{a_{guess}} = \begin{bmatrix} 1.3875 & -0.1750 & -0.0250 & -1.5000 & 0.0563 & 0.0125 & 0.1750 \end{bmatrix}^T.$$

By using the generated data, we estimate the parameters after fixing m_s and b_1, and as expected, obtain $\hat{\mathbf{a}} = \mathbf{a}$.

We can do the same set of experiments by adding noise either to the positions or velocities of the system. We noticed that the velocities were much more insensitive to noise than the positions. To quantify this, we define a Euclidean error between the estimated and the real parameters, as shown in Equation 4.5.

$$e = (\hat{\mathbf{a}} - \mathbf{a})'(\hat{\mathbf{a}} - \mathbf{a}) \tag{4.5}$$

We add random noise drawn from a normal distribution of mean 0.5 to the velocity and position sets separately. After running the experiment 10 times for each set, we see that e ¡ 0.1 for the tests with noisy velocities, and e 4.2 when we add noise to the positions. For reference, the norm of our real parameter vector, $\|\mathbf{a}\| = 2.0592$.

This initial set of experiments indicate the error in the position data is most important. Given that we can directly collect the angular position data from the motion capture system but have to derive the velocities and accelerations, this is a good result because we can more effectively correct for errors in the position data.

Parameter Estimation on Collected Data

We also tested the LLSE with the same fixed parameters (m_s and b_1) on our collected data. After parsing the data for two swing phases, we concatenated the data and ran the parameter estimation on 303 data points.

The estimated parameter vector is shown below,

$$\hat{\mathbf{a}} = \begin{bmatrix} -0.0016 & -0.0007 & 0.0005 & 0.0139 & 0.0006 & 0.0023 & -0.0026 \end{bmatrix}^T.$$

We can also calculate the squared error for the estimated parameters, where the error is defined as in Equation 4.6. Comparing the squared error with the estimated parameters and those we had arbitrarily assigned in the previous section, we see that the estimation does the norm of the error significantly, $E_{guess} = 280.3992, \quad E_{LSE} = 0.0857.$

$$E = (\mathbf{W}\hat{\mathbf{a}} - \mathbf{y})'(\mathbf{W}\hat{\mathbf{a}} - \mathbf{y}) \tag{4.6}$$

However, if we look at the signs in the estimated parameters and look at the expressions they are supposed to correspond, we see that only a_2, a_3 and a_4 can and should be negative, since the masses and link lengths of these terms can only have positive values. This is a clear indicator that our estimation is not getting the right parameters. Furthermore, playing back the robot's motion using these parameters give wrong trajectories that do not match those of a swing phase.

4.4 Future improvements

Errors in the calibration of our captured raw data might be causing our parameter estimation on the real data to break down. As noted in the simulated data tests though, the position data is the most important. This observation is promising as improvements on the calibration and collection of the motion capture data could drastically improve our parameter estimation.

Currently, our motion capture trials are calibrated with an error of typically 15mm. Exactly how this error translates into the random noise used in the simulated data runs, however, is not clear. Improving on more precise measures of the motion capture data, though, could improve our parameter estimation and help us compare quantitatively the robot with our model, to see how they match up.

Chapter 5

Conclusions

The kneed walker model and its corresponding robot have been developed as a more rigorous tool to study passive dynamic walking. This work was an effort to make passive dynamic walking results more quantifiable problem by reducing the amount of artistry needed to make these sensitive systems work. Before adding actuation to the system, we wanted to study the system's stability region and try to find the actual one in the robot. Building the robot would also allow us to see how well the assumptions our model made held up in experiment.

From the simulation results, we learned general guidelines for finding a stable gait. In general, we found in simulation that a mass ratio of at least 1:10 not only helps the knee bend but also helps achieve a stable gait. Also, the hip mass should be roughly the size of the upper leg masses. Another important point is to keep the point masses close to the knee joint. These specifications would make the kneed walker approximate the compass gait dynamics in the limiting case.

Finally, the objective was to develop a simple enough model so that we could characterize its full step cycle, but also one that was physically realizable. We found a planar walker with point feet and knees to be such model. The point feet allowed straight-forward pendumlum dynamics (as opposed to the friction model needed for curved feet) and the knees allowed natural foot clearance. In using the passive dynamic ideas, we showed that our model had a basin of attraction. The system can be started in this region and remain stable, converging eventually to the system's fixed point. After finding a point in the basin of attraction, we subsequently found the fixed point. A local and global stability analysis was then performed on the model.

Due to the point feet and knees, we had expected this walker to be able to walk in rough terrain effectively. The simulation did in fact take multiple steps under rough terrain conditions successfully. Future work in the stability analysis includes studying how stable this model is by calculating the mean-first passage time [3] both for the deterministic flat-terrain system and for stochastic rough terrain conditions.

After developing and studying the model, our next step was to build a machine that would match our model and test our theoretical results. The machine we built takes a few steps after which it falls forward.

Various designs were tried for the knee to fix the different mechanical problems that were identified, such as hyper-extension of the leg, mismatch in the position of the outer feet and delayed knee lock. The results seen with the robot are very promising and we needed a more detailed analysis tool to study what was causing it to fail.

To do so, we took digital video footage and motion capture data of the robot and carefully quantified the motion of the robot. This allowed us to compare the machine's dynamics directly to those of the model. After reproducing the collected data in simulation, we could compare the joint angles of the robot during a swing phase and the corresponding velocities and accelerations with those predicted in simulation. The robot did move roughly as the model predicts. However, in order to match the two more closely and fit the length of the stride as well as the small characteristics in the curves, we needed to first perform a careful parameter estimation of the robot to find the correct physical parameters with which to run the simulation.

Although the parameter estimation was shown to work with simulated data with a controlled amount of noise, it did not result in reasonable values when we ran it with our collected data. An improvement on this could be calibrating the robot to the motion capture 'skeleton' more carefully. We can define the template markers exactly using fixed and carefully measured distances, instead of relying on the system to calibrate parameters. To take full advantage of the resolution of the motion capture system, we should make sure our captured model fit the robot as accurately as possible. Doing this would allow us to exactly playback the corresponding simulation, and make sure we are starting it within the basin of attraction.

We believe that by doing so, we can perform a satisfactory parameter estimation. This would help us catch the specific problems either with how we are starting it or how its feet are slipping, in order to achieve a successful gait on the robot, for example. Furthermore, the motion capture data will also allow us to compare the heelstrike and kneestrike events with the inelastic collisions we modeled.

Appendix A

Collision Models for Kinematic Chains

We can simplify any planar robot into a kinematic chain with certain link lengths with point masses either at the joints or in the middle of the links. To understand the dynamics during robot collisions then, let us study what happens when collisions occur in kinematic chains.

Consider the kinematic chain shown in Figure A-1. The continuous dynamics of such a system are derived easily using Lagrange's equations of motion. The collisions in this chain, however, either internal or external are modeled as discrete events because we assume instant velocity changes due to rigid body collisions.

For the collisions, we make two basic assumptions:

- No slip occurs due to infinite friction with the external surface
- Collisions are inelastic, losing kinetic energy in the process

As long as the impact forces do not create a moment about a certain axis, we can say that angular momentum is conserved about that point.

If an impulsive collision occurs at point A, an instantaneous force is exerted. The force creates zero torque at this point. Also, because this is the only external force in *system a* (as marked in the diagram) we can say that the angular momentum for *system a* is conserved about the impact point, A. This collision is equivalent to our heelstrike event, where the impact point is the foot that is hitting the ground.

Furthermore, the effects of the collision ripple through the rigid body. If we take *system b* for example, that is, if we isolate everything beyond point B, we notice that the only force that can be created from the impulse at A is a net force at point B. Since that is the only external force from the collision, angular momentum of *system b*, then, is conserved about B. Similarly, angular momentum about point C of *system c* is conserved. In this way, we propagate up the chain, obtaining n equations for n joints.

To derive the equations for our specific example, we name the different segments by denoting the two endpoints in the subscript (e.g. $\overrightarrow{Aa}$ corresponds to the straight line going from point A and the position of the point mass m_a). We also denote the linear velocities from a certain point using v and the corresponding subscript. The subscripts, - and +, denote pre- and post- collision velocities.

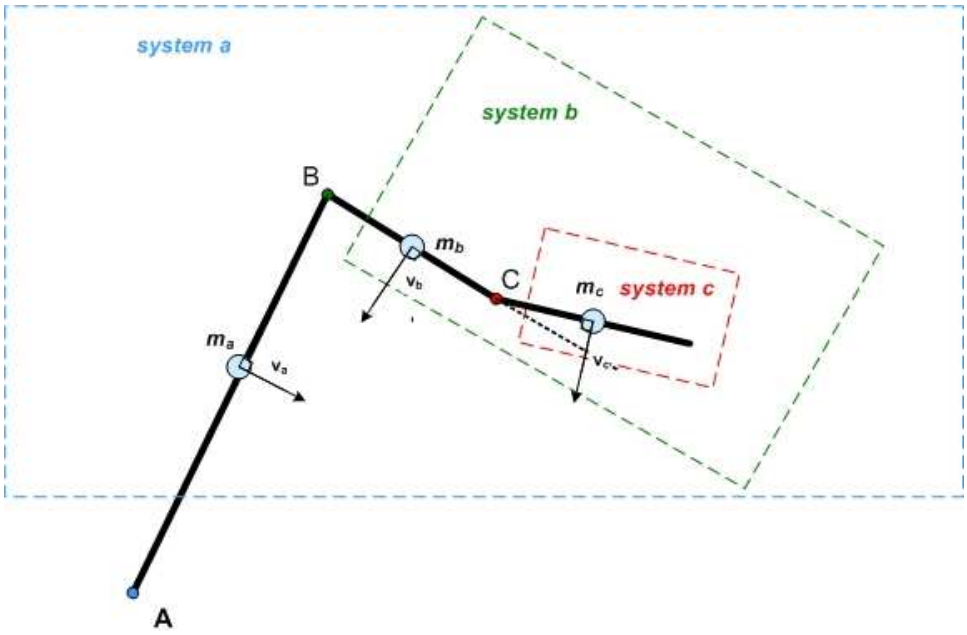

Figure A-1: A 3 degree of freedom planar kinematic chain

Using this nomenclature, we can write the equations of conservation of angular momentum as follows,

$$m_a\overrightarrow{Aa}\times v_a^+ + m_b\overrightarrow{Ab}\times v_b^+ + m_c\overrightarrow{Ac}\times v_c^+ = m_a\overrightarrow{Aa}\times v_a^- + m_b\overrightarrow{Ab}\times v_b^- + m_c\overrightarrow{Ac}\times v_c^- \quad \text{(A.1)}$$

$$m_b\overrightarrow{Bb}\times v_b^+ + m_c\overrightarrow{Bc}\times v_c^+ = m_b\overrightarrow{Bb}\times v_b^- + m_c\overrightarrow{Ac}\times v_c^- \quad \text{(A.2)}$$

$$m_c\overrightarrow{Cc}\times v_c^+ = m_c\overrightarrow{Cc}\times v_c^- . \quad \text{(A.3)}$$

Conversely, consider an internal collision, such as an impulse torque at point C (from locking link c, for example, when it aligns with link b), as is the case of our kneed walker's kneestrike event. In this case, the internal force is the impact force at the knee when the lower swing leg swings forward and locks in the straight leg position.

This torque again, will be the only force acting on the entire system. However, since this is an internal torque for *system a* and *system b*, we can say that both linear and angular momentum are conserved for these systems. However, the angular momentum of *system c* is changed by the torque at C.

In this case, only equations A.1 and A.2 hold true. However, if it is the case that link c is locked after aligning with link b as it is for our model, then there is a new kinematic constraint: link c possesses the same angular position and velocity as link c after the collision.

A.1 Collisions in the Kneed Walker model

Below are the specific conservation equations used to derive the transformation matrices Q^- and Q^+ for the heelstrike and kneestrike events of the kneed walker model.

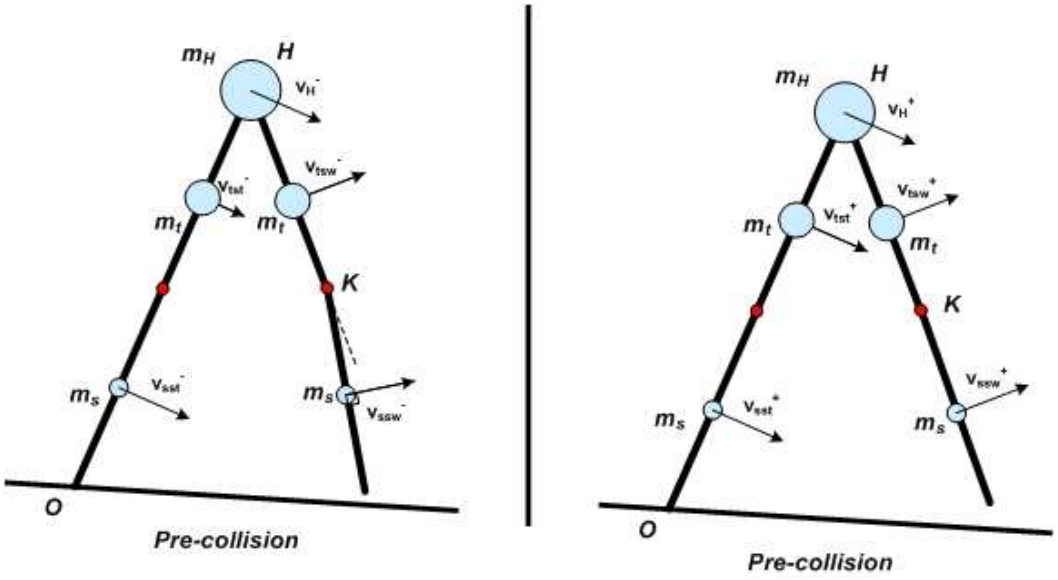

Figure A-2: Kneestrike: pre- and post- collision configurations

In Figures A-3 and A-2, the segment names are specified as above, using the names of both endpoints. O corresponds to the stance foot before collision, which is also the swing foot after collision; H corresponds to the point at the hip, and K to the point at the knee where the impact occurs. For the points at the masses, we use t for the upper links, s for the lower links, and the subscripts are used to denote the stance (st) or swing (sw) leg.

In the kneestrike event (shown in Figure A-2), we have an internal collision in this 3 degree of freedom kinematic chain. This means that angular momentum is conserved for the whole systems about the stance foot, and for the swing leg about the hip. Angular momentum is not conserved, however, about the impacting knee joint.

For the heelstrike case (shown in Figure A-3), the conservation equations are those of a 2 degree of freedom kinematic chain with 5 point masses that experiences an external torque exerted at the impact foot. In this case, angular momentum is also conserved for the whole system about the back pre-collision stance foot, as it is conserved for the front pre-collision swing leg with respect to the hip.

Therefore, the conservation of angular momentum equations are the same for both events, and written as follows,

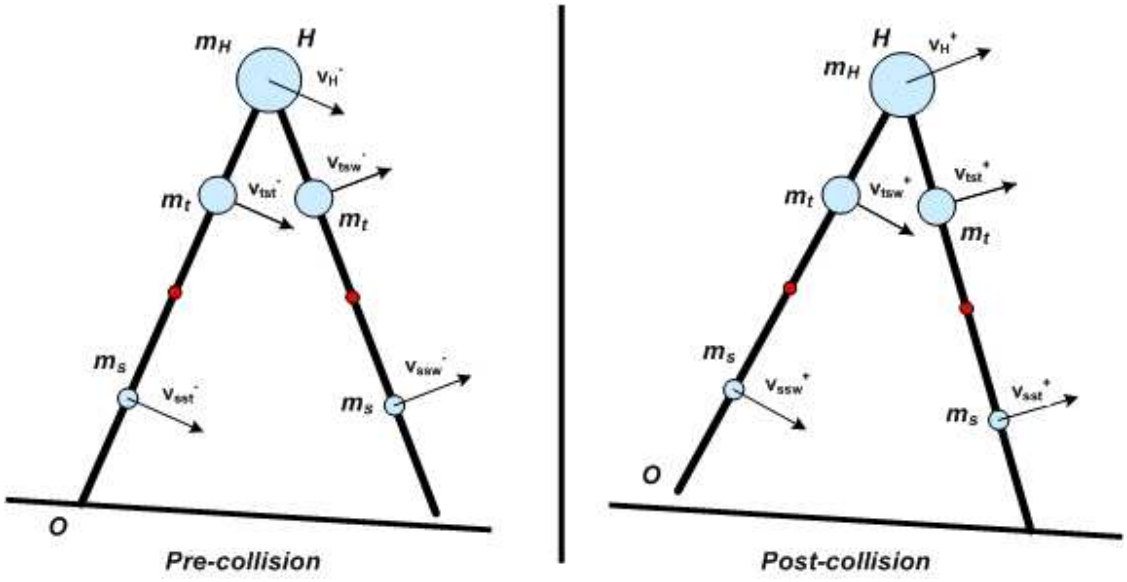

Figure A-3: Heelstrike: pre- and post- collision configurations

$$m_s\overrightarrow{Os_{st}^-} \times v_{s_{st}}^- + m_t\overrightarrow{Ot_{st}^-} \times v_{t_{st}}^- + m_H\overrightarrow{OH} \times v_H^- + m_t\overrightarrow{Ot_{sw}^-} \times v_{t_{sw}}^- + m_s\overrightarrow{Os_{sw}^-} \times v_{s_{sw}}^- = \ldots \tag{A.4}$$

$$m_s\overrightarrow{Os_{sw}^-} \times v_{s_{sw}}^- + m_t\overrightarrow{Ot_{sw}^-} \times v_{t_{st}}^- + m_H\overrightarrow{OH} \times v_H^- + m_t\overrightarrow{Ot_{st}^-} \times v_{t_{st}}^- + m_s\overrightarrow{Os_{st}^-} \times v_{s_{st}}^-$$

$$m_t\overrightarrow{Ht_{sw}^-} \times v_{t_{sw}}^- + m_s\overrightarrow{Hs_{sw}^-} \times v_{s_{sw}}^- = m_t\overrightarrow{Ht_{st}^-} \times v_{t_{st}}^- + m_s\overrightarrow{Hs_{st}^-} \times v_{s_{st}}^- \tag{A.5}$$

These equations yield different results for the two events. The heelstrike event occurs when the robot has straight locked-knee legs, resulting in a 2 degree of freedom system. This means we will have 2 equations with 2 state variables, q_1 and q_2. The resulting matrices will therefore be 2x2 in dimension. The velocity of the third link is assumed to be the same as the second because the legs remain straight during this event. This kinematic constraint can be expressed as follows, $\dot{q}_3 = \dot{q}_2$.

In the case of kneestrike, this straight-legged constraint is only true after the knee hits and locks. For the pre-collision configuration, our system has 3 degrees of freedom, which also means that the velocity of the swing shank does not equal that of the corresponding thigh (i.e., $\dot{q}_3 \neq \dot{q}_2$. Therefore, we obtain a 2x3 matrix on the left-hand side of the conservation equations and a 2x2 matrix on the right hand side, as seen in Section 3.2.1.

Bibliography

[1] F. Asano, M. Yamakita, N. Kamamichi, and Luo Zhi-Wei. A novel gait generation for biped walking robots based on mechanical energy constraint. *IEEE Transactions of Robotics and Automation*, 20(3):565–573, June 2004.

[2] Gary Boone. Minimum-time control of the acrobot. volume 4, pages 3281–3287. IEEE International Conference on Robotics and Automation (ICRA), 1997.

[3] Katie Byl and Russ Tedrake. Stability of passive dynamic walking on uneven terrain. In Art Kuo, editor, *Proceedings of Dynamic Walking 2006*, May 2006.

[4] Michael J. Coleman, Anindya Chatterjee, and Andy Ruina. Motions of a rimless spoked wheel: a simple 3d system with impacts. *Dynamics and Stability of Systems*, 12(3):139–160, 1997.

[5] Michael J. Coleman and Andy Ruina. Motions of a rimless spoked wheel: a simple 2d systems with impacts. August 2002.

[6] Steven H. Collins, Andy Ruina, Russ Tedrake, and Martijn Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307:1082–1085, February 18 2005.

[7] Steven H. Collins, Martijn Wisse, and Andy Ruina. A three-dimensional passive-dynamic walking robot with two legs and knees. *International Journal of Robotics Research*, 20(7):607–615, July 2001.

[8] G. DeJong and Mark W. Spong. Swinging up the acrobot: An example of intelligent control. volume 2, pages 2158–2162. American Control Conference, 1994.

[9] M Garcia, A Chatterjee, A Ruina, and M Coleman. The simplest walking model: Stability, complexity, and scaling. *Journal of Biomechanical Engineering – Transactions of the ASME*, 120(2):281–288, Apr 1998.

[10] A. Goswami, B. Espiau, and A. Keramane. Limit cycles and their stability in a passive bipedal gait. pages 246 – 251. IEEE International Conference on Robotics and Automation (ICRA), 1996.

[11] Ambarish Goswami, Benoit Thuilot, and Bernard Espiau. Compass-like biped robot part I : Stability and bifurcation of passive gaits. Technical Report RR-2996, INRIA, October 1996.

[12] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, August 2001.

[13] Honda Motor Co. http://world.honda.com/ASIMO/, July 2004.

[14] Arthur D. Kuo. Energetics of actively powered locomotion using the simplest walking model. *Journal of Biomechanical Engineering*, 124:113–120, 2002.

[15] Tad McGeer. Passive dynamic walking. *International Journal of Robotics Research*, 9(2):62–82, April 1990.

[16] Tad McGeer. Passive walking with knees. pages 1640 – 1645. IEEE International Conference on Robotics and Automation (ICRA), 1990.

[17] Simon Mochon and Thomas A. McMahon. Ballistic walking. *Journal of Biomechanics*, 13:49–57, 1980.

[18] Simon Mochon and Thomas A. McMahon. Ballistic walking: An improved model. *Mathematical Biosciences*, 52(3-4):241–260, December 1980.

[19] Hiroki Ohta, Masaki Yamakita, and Katsuhisa Furuta. From passive to active dynamic walking. *International Journal of Robust and Nonlinear Control*, 11(3):287–303, Mar 2001.

[20] A. L. Schwab and M. Wisse. Basin of attraction of the simplest walking model. *Proceedings of the ASME Design Engineering Technical Conference*, 6:531–539, Sep 2001.

[21] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice Hall, October 1990.

[22] Mark Spong. The swingup control problem for the acrobot. *IEEE Control Systems Magazine*, 15(1):49–55, February 1995.

[23] Mark W. Spong. Swing up control of the acrobot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2356–2361, 1994.

[24] Russ Tedrake, Teresa Weirui Zhang, Ming-fai Fong, and H. Sebastian Seung. Actuating a simple 3D passive dynamic walker. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 5, pages 4656–4661, New Orleans, LA, April 2004.

[25] Russ Tedrake, Teresa Weirui Zhang, and H. Sebastian Seung. Stochastic policy gradient reinforcement learning on a simple 3D biped. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2849–2854, Sendai, Japan, September 2004.

[26] Russell L Tedrake. *Applied Optimal Control for Dynamically Stable Legged Locomotion*. PhD thesis, Massachusetts Institute of Technology, 2004.

[27] M Wisse, A L Schwab, R Q van der Linde, and F C T van der Helm. How to keep from falling forward: elementary swing leg action for passive dynamic walkers. *IEEE Transactions of Robotics*, 21(3):393–401, Jun 2005.

[28] Martijn Wisse and Jan van Frankenhuyzen. Design and construction of Mike; a 2D autonomous biped based on passive dynamic walking. In *Proceedings of the International Symposium on Adaptive Motion of Animals and Machines*, 2003.

[29] M. Yamakita and F. Asano. Extended passive velocity field control with variable velocity fields for a kneed biped. *Advanced Robotics*, 15(2):139–168, June 2001.