MODELING AND CONTROL OF A PLANAR BOUNDING QUADRUPEDAL ROBOT

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Mechanical Engineering

by

Patrick Ward

June 2022

COMMITTEE MEMBERSHIP

TITLE: Modeling and Control of a Planar Bounding Quadrupedal Robot

AUTHOR: Patrick Ward

DATE SUBMITTED: June 2022

COMMITTEE CHAIR: Siyuan Xing, Ph.D.
Assistant Professor of Mechanical Engineering

COMMITTEE MEMBER: Charlie Refvem, M.S.
Lecturer of Mechanical Engineering

COMMITTEE MEMBER: William Murray, Ph.D.
Professor of Mechanical Engineering

ABSTRACT

Modeling and Control of a Planar Bounding Quadrupedal Robot

Patrick Ward

Legged robots have the potential to be a valuable technology that provides agile and adaptive locomotion over complex terrain. To realize legged locomotion's full abilities a control design must consider the nonlinear piecewise dynamics of the systems. This paper aims to develop a controller for the planar bounding of a quadrupedal robot.

The bounding of the quadruped robot is characterized by a simplified hybrid model that consists of two subsystems for stance and flight phases and the switching laws between the two states. An additional model, the Multibody model, with fewer simplifications, is used concurrently to best approximate real-world behavior. The bounding gait (periodic orbit) of the robot is predicted by an optimization method based on the numerical integration of the differential equations of subsystems. To stabilize the gait, a switching controller is applied which can be split into two separate phases: stance-phase and swing-phase control. The stance phase implements reaction force control utilizing a body state feedback controller and a gait stabilizer, while the swing phase deploys position control in conjunction with a trajectory planning algorithm to ensure proper footfall. Numerical simulations are carried out for the system with/without control. The control strategy is further validated by simulations of the Simscape multibody model. The overall simulated controller results are promising and demonstrate stable bounding for four system cycles.

ACKNOWLEDGMENTS

Thanks to:

- Dr. Siyuan Xing for leading me through this project. Thank you for your patience and support. It has been a pleasure working with you and learning from you.
- Dr. Paul Choboter for helping me with numerical analysis and other mathematical techniques.
- My committee members, Charlie Refvem and Dr. William Murray, for the general guidance throughout the thesis project and taking the time to review my work.
- Lili LeBaron for giving me all of the mental support I needed to tackle the project.
- My parents for believing in me and always supporting my educational goals.

TABLE OF CONTENTS

Page

APPENDICES

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Wheeled locomotion is power efficient and easy to implement, benefits that have lead to its importance in the field today. These benefits allow cars and buses to remain dominant in transit worldwide. Even within the robotics field, companies like Nuro or Starship Technologies have implemented wheeled robots [1]. Despite wheeled locomotion's successes, there is one major caveat to these vehicles and robots. Wheeled locomotion has the inability to navigate or traverse complex environments. Fortunately, nature provides a thoroughly tested solution. Animals such as sheep, canines, or big cats all have the ability to move quickly and with high agility through treacherous environments. All of these examples rely on legged locomotion for their fast and dexterous movements. While legged locomotion exhibits this superior ability to deal with severe terrain, it is far more difficult to design robots and control systems that can utilize this behavior.

1.1 Background

There are many potential benefits of legged locomotion. Subsection 1.1.1 covers some of these potential advantages and where they can be applied. Researchers have been trying to gain access to these abilities through different solution methods for hundreds of years. Subsection 1.1.2 covers the general trajectory of these legged locomotion attempts up to current products.

1.1.1 Legged Locomotion Applications and Advantages

One sector that is in need and has countless applications for quadrupedal robotics is mining. Inspection robots would be great for viewing and checking infrastructure to support some stationary equipment and the miners themselves. Miners' work hours are limited to 6-hour shifts and are projected to continue to shorten. Since these mines can often be as large as 3000 [km], there becomes a necessity to have the ability to send quadrupeds down when and where workers are low. In addition to inspection, quadrupeds could be useful for search and rescue, main technology processes, or auxiliary technology processes. Search and rescue could be highly useful in the case of tunnel collapse. Robots could provide main technology processes such as drilling, autonomous transport, or fragmentation of oversized lumps especially as mines become deeper and branch into dangerous locations. For the same reasons just mentioned mapping and exploration within the auxiliary technology processes could prove just as fruitful [2].

Generally, the main areas of application are search-and-rescue or last-mile delivery. For example, a legged robot could be used to traverse the harsh natural terrain of the mountains to find a fallen hiker or a lost snow skier while wheeled locomotion would be unable to get past large boulders or thick vegetation. Another great application would be to use a legged robot for police activity. Instead of putting a person's life at risk, a legged robot could be used to search an apartment. Wheeled locomotives would be unable to get over blocked passageways or furniture while the legged robot would be able to jump or step over whatever obstructions were in the way.

1.1.2 Legged Locomotion's Past and Present

Researchers, engineers, and inventors have been investigating the dynamic and robust motion of legged locomotion dating back to the late fifteenth century when Leonardo Da Vinci first designed and potentially created an armored knight [3]. The next step in legged locomotion was a patent in 1893. This was purely mechanically driven just as Leonardo Da Vinci's was. It was a mechanical horse that relied on gear power transmission from the user's feet to kinematic linkages. These initial attempts and those developed up until the first world war were focused on harnessing a single power source to be transmitted by linkages and/or gears. This limitation required all of the designs to maintain a fixed gait [3].

Space General Corporation and General Electric are among the first to implement variable gait legged locomotion during the 1960s. Space General Corporation's inspiration was rooted in the need to traverse the moon's complex terrain. Both General Electric's and Space General Corporation's mechanisms were human-controlled. General Electric's design proved to be adept at dealing with difficult terrain but the control was too demanding for most people to operate the system [3]. General Electric's findings left the field of legged locomotion with the clear realization that a control system would be necessary to help people realize legged locomotion's full potential.

Another project during this time, the Phoney Poney, provided a solution to the human control dilemma. It successfully demonstrated computer control and electrical actuation for the first time [3]. There was a large loss of complexity and ability to handle terrain but this project lead the way for computerized control.

Today, researchers at MIT have achieved high-level designs of quadrupedal robots. MIT has conducted research on a series of robots called the Cheetah. Cheetah 2 has accomplished robust bounding up to speeds of 6.4 [m/s] [4]. While Cheetah 3 has built upon this model allowing for motion outside of the sagittal plane. A notable achievement in this design is the Cheetah 3's ability to move over complex and abrupt obstacles such as stairs without any external sensors. The robot does not have any external sensors such as a visual system or force transducers. The robot adjusts to obstacles purely based on proprioception [5]. Much of the work done in this paper has been inspired by the controller design in the Cheetah 2.

Boston Dynamics has taken one of the giant steps in quadrupedal robotics and has released a robot for commercial use. The robot is advertised to be capable of dynamic balancing, 360-degree terrain mapping, mobility across difficult terrain, and capable of both remote and autonomous control [6].

1.2 Legged locomotion Control Solutions and Research

There is a large amount of research going into developing legged locomotion control solutions. To gain some perspective on previous ideas proposed, a series of papers are reviewed below.

Hodgins focuses on how step length acts as a solution for legged locomotion over rough terrain. The paper examines three methods for controlling step length: modulating the duration of flight, the duration of stance, and the forward running speed. Modifying the duration of flight and modifying the forward running speed are found to be the most useful for controlling step length. These methods were utilized to enable a biped to traverse a staircase [7].

One quadruped control solution posed by Raibert in "Trotting, Pacing, and Bounding by a Quadruped Robot" considers a quadruped with linearly actuated legs. This control solution tackles the three, leg pair gaits: trotting, pacing, and bounding. The gaits are transformed into a common underlying gait which can be produced by a single set of control algorithms. The high-level algorithms are inspired by control schemes commonly used for the one-legged hopping machine. The high-level control regulates the vertical motion, stabilizes the running speed, and keeps the body level. The low-level control modulates leg behavior through foot placement, variable force transduction, and hip torque manipulation. Gait transitions were possible by considering a special sequence for each specific change of gait [8].

Another quadrupedal control approach that takes a very modern perspective is proposed by Wang, Hu, and Zhu. This controller utilizes a low-level central pattern generator for establishing a baseline gait but utilizes machine learning to develop task-oriented abilities. This system was able to learn a variety of skills demonstrated through experimentation such as velocity tracking, path following, and navigating to a target [9].

Kalakrishnan et al. present a control architecture for fast quadrupedal locomotion over rough terrain. This controller utilizes machine learning for foothold selection based on a terrain template. Additionally, a body trajectory optimizer and a floating-base inverse dynamics control scheme are implemented. Their overall control design allows for robust locomotion over unperceived obstacles [10].

1.3 Objectives

While legged robots provide many advantages over traditional locomotive robots, they require advanced control methods to gain these desired capabilities. While some early attempts have utilized human control or relied on gear and linkage trains to accomplish legged locomotion, it is clear from the research being done today and current products that computer control is currently the most effective way to achieve optimal performance with a quadrupedal robot. With this assumption in mind, the primary objective of this paper is to investigate a computer control algorithm that can support stable planar bounding.

The controller design process almost always requires a system model for analysis, control derivation, and control testing. The closer a model is to the real-world the more likely a control designer is to create a successful control algorithm. When the system being controlled is highly complex it can make developing an accurate model a significant challenge. Additionally, generating model simulations can require a strong background in computer science and a large amount of time. One potential option is to actually build the system and run experiments. This option can require a large financial commitment and experiments can be risky. For this reason, a secondary objective of this paper is to demonstrate Matlab's Simcape Multibody as an effective way to model a mechanical system without requiring the same investment or incurring the same risks.

In summary the objectives include:

- Stable planar bounding control algorithm
- Simscape Multibody Evaluation

1.4 Significance

This thesis utilizes one of MATLAB's ad-ons, Simscape Multibody, to model the quadrupedal robot. This environment provides a user-friendly modeling environment for simulating mechanical systems with limited assumptions and simplifications. This program is effective because solid-body elements can be created or imported from computer-aided design (CAD) software. These solid-body blocks can contain complex geometries and accurately represent the associated inertial properties. The solid-body blocks can be connected through various joint types or frame references allowing for large mechanical systems to be assembled and analyzed. In addition, the environment has sophisticated ways for implementing torques or forces on the system including contact between two bodies. Normally testing a controller would require expert-level programming knowledge for simulation or require real-world experimentation.

The successful use of this modeling environment has major implications for control designers with limited programming experience. Many controls researchers and developers come from a mechanical engineering background. It is very common for students in this field to be exposed to very limited programming experience. This can make model simulation an extreme barrier to controller design. With a user-friendly application like Simscape Multibody, engineers across all fields have a better opportunity to analyze and control dynamic systems. By making controller design more accessible, new and creative approaches can more commonly be investigated.

Another benefit to using Simscape Multibody is that more accurate models can be used for controller testing before experimentation. If the control designer is unable to design their own high-level simulation from scratch, they may move directly to real-world testing. This can be very dangerous and expensive. Any critical failures during real-world experimentation can often hurt the researcher or at minimum damage equipment. Simscape Multibody modeling helps minimize these risks.

1.5 Outline

The thesis is structured into 6 main chapters. In Chapter 1, legged locomotion's applications and history are discussed in Section 1.1. In addition, the chapter covers the structure and importance of the paper in sections 1.3, 1.4, and 1.5.

Chapter 2 breaks down all of the system modeling. The chapter is split into two main topics: the hybrid model and the solid-body model. The hybrid modeling is heavily inspired by the work done by MIT with the Cheetah 2 [4]. The primary goal of the hybrid model is to simplify the robotic system without losing its fundamental behavior. These simplifications make it possible to establish the foundational concepts for the controller. Section 2.1 covers the hybrid model in detail. The solid-body modeling stage utilizes MATLAB's Simulink toolbox Simscape Multibody to model solid-bodies and their interactions. This modeling stage's purpose is to reduce simplifications and introduce behaviors found in real-world implementation. The solid-body model provides a way to test and add additional controller layers for improved application performance. Section 2.2 details the construction of the solid-body model.

Chapter 3 investigates the existence of periodic orbits, their stability, and the necessary system input to sustain them. Section 3.1 defines periodic orbits in the context of the hybrid model and analyses the stability of these orbits. Section 3.2 details the nominal reaction force profiles and gait design to achieve the periodic orbits. The needed torques to generate the desired reaction force profiles are discussed in Section 3.3. This torque-reaction force mapping is heavily based on the work done in *Modeling and Control of a Vertical Hopping Robot* and *Controlling Locomotion of a Robotic Leg* [11] [12]. The last section of Chapter 3 is Section 3.4 where the nominal foot trajectory during the swing phase is detailed. This trajectory is critical for maintaining the periodic orbits during solid-body or real-world implementation.

The following chapter, Chapter 4, details the overall controller design and its individual components. The design is heavily inspired by MIT's Cheetah 2. The controller is broken down into two main modes: the stance phase and the swing phase. Each of these phases has feedforward profiles based on the work done in Chapter 3. On top of the feedforward profile, a body state feedback controller is implemented during stance phases to stabilize the hybrid model around the periodic orbits. The body state feedback controller design is found in Subsection 4.1.2. Using the solid-body model, additional control layers are designed to improve potential real-world implementation. The first additional control is the creation of a gate pattern stabilizer described in Subsection 4.1.3. Subsection 4.2.2 describes another control layer designed around the solid-body model, a Proportional, Integral, and Derivative (PID) controller. This PID controller allows foot position control and is necessary to follow the desired foot trajectory. The last control layer is a swing phase trajectory planner that accurately predicts the time until the next impact. This air time estimate will be critical for ensuring the complete foot trajectory can occur before impact. See Subsection 4.2.3 for further details.

Chapter 5 examines some of the results from varying models and varying control schemes. Finally, Chapter 6 gives my final thoughts on the research and future steps.

Chapter 2

SYSTEM MODELING

Before any controller design process begins there needs to be a sufficiently accurate mathematical model of the system being controlled. This system is a quadrupedal robot in the planar bounding locomotion mode. The following chapter cover multiple modeling scenarios and methods. These sections present insight into the system as a whole while providing the foundation for controller design. The first section, 2.1, details the modeling process, assumptions, and simplifications for the hybrid bounding model. This model provides the basis for designing both the feed-forward and state feedback control during the stance phase. The other section, 2.2, details a solid-body model. This model best approximates the real-world behavior of a quadrupedal system providing a way to test the hybrid model controller and improve it.

2.1 Hybrid Bounding Model

The descriptor planar in the context of bounding refers to the system's constrained motion to one plane. This plane of motion can be referred to as the sagittal plane, splitting the quadrupedal robot in half. The robot can move in the horizontal or vertical direction within this plane. Therefore, all following analyses are done with the assumption that there are no out of plane forces or non-normal torques. The type of motion characterized by bounding includes four phases of a periodic cycle: front stance, flight, back stance, and flight again. During front stance the two front legs are in contact with the ground. During both flight phases all legs are off the ground. During back stance the two back legs are in contact with the ground. The hybrid model assumes phase transitions on a pre-defined timing schedule. Each state has a certain dwell time, $T_j \in \mathbb{R}_+$, that must occur before each transition, $j \in \{0, 1, 2, 3, 4\}$. $j = 0$ denotes a special transition that initializes the system with no preceding dwell time. $T = \sum_j T_j$, is the total time to complete a full bounding cycle. Figure 2.1 shows a road map for this phase cycle.

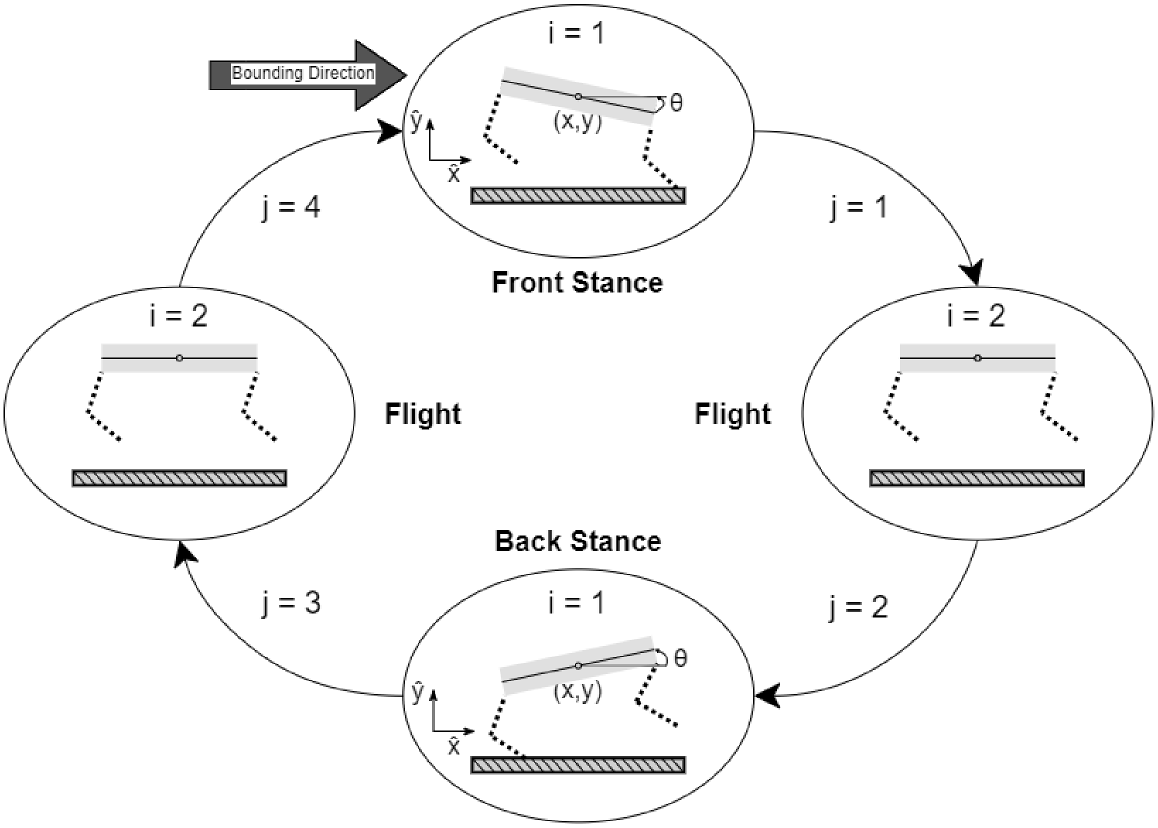

Figure 2.1: The road map for periodic phase switching.

Equation 2.1 illustrates how the system is governed by two sets of ordinary differential equations (ODEs). There is one set of ODEs for the stance phases and one set of ODEs for the flight phases denoted by i.

$$\dot{\mathbf{q}} = \mathbf{f_i}(\mathbf{q}, \mathbf{t}), \mathbf{i} \in \{\mathbf{1}, \mathbf{2}\} \tag{2.1}$$

The state variable, $\mathbf{q} \in \mathbb{R}^6$, includes x, y, θ, and their first derivative. x represents the mass center's horizontal position. y represents the mass center's vertical position. Lastly, θ represents the angular position of the mass center. $t \in \mathbb{R}_+$, represents the continuous time of the system.

The inertia of the system is assumed to be symmetrically distributed along the length, ℓ, of the hybrid model. The sum of this distributed mass is denoted as m. The centroidal mass moment of inertia is denoted as I. For this model the legs are assumed to be massless. In addition, system damping due to joint friction or air resistance is neglected. The feet of the legs are assumed to exhibit no slip while in contact with the ground. The final assumption is perfect footfall. This assumption is that foot location is prescribed by a predetermined hip offset, γ_x, occurring at impact. These two assumptions make it possible to neglect the non-linearities of the leg kinematics and body angle. Equation 2.2 displays how the horizontal foot location, x_{foot}, is reset after each flight phase on transitions $j = 2$ and $j = 4$.

$$x_{foot} = \begin{cases} \gamma_x + x + \frac{\ell}{2}cos(\theta) \;\; j = 2 \\ \gamma_x + x - \frac{\ell}{2}cos(\theta) \;\; j = 4 \end{cases} \tag{2.2}$$

The last two terms of each x_{foot} equation place the horizontal hip location relative to the global reference frame. Therefore, hip offset determines the foot position relative to the global reference frame. Figure 2.2 displays the simplified model utilized for finding the governing equations. x_{hip} and y_{hip}, in the figure, denote the Cartesian coordinates for the hip of the leg that is in contact with the ground.

Both sets of equations of motion were generated using Newtonian mechanics. The equations of motion modeling a stance phase, $i = 1$, are shown below in Equations 2.3, 2.4, and 2.5. These same equations can be applied to both stance phases. This is ac-

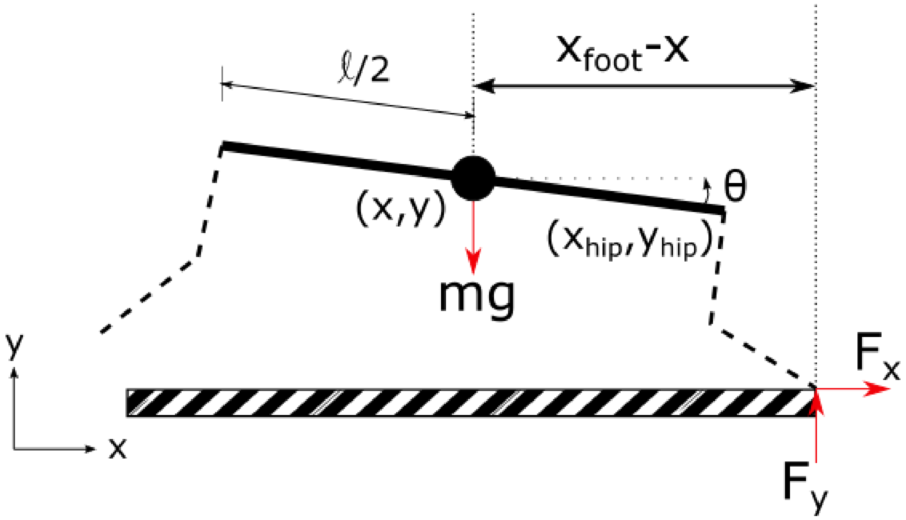

Figure 2.2: The free body diagram of the simplified sagittal plane quadruped model.

complished by considering the correct foot for reaction force production, information determined by the preceding j transition.

$$m\ddot{x} = F_x \tag{2.3}$$

$$m\ddot{y} = -mg + F_y \tag{2.4}$$

$$I\ddot{\theta} = F_y(x_{foot} - x) + F_x(y) \tag{2.5}$$

F_x is the horizontal reaction force produced by whichever foot is in contact with the ground. The vertical reaction force produced by this foot is denoted as F_y. g represents the acceleration due to gravity observed on the surface of earth.

The equations of motion for both flight phases, $i = 2$, can be found in Equations 2.6, 2.7, and 2.8. These equations are only governed by gravity making the system very simple during flight.

$$m\ddot{x} = 0 \tag{2.6}$$

$$m\ddot{y} = -mg \tag{2.7}$$

$$I\ddot{\theta} = 0 \tag{2.8}$$

2.2 Simscape Multibody

The hybrid model, as discussed in Section 2.1, has phase switching events tied purely to periodic timing. A more accurate way to model the system is to incorporate ground contact triggered phase switching. Another modeling improvement is to incorporate leg inertia and leg joint friction. To achieve this, the system is modeled using MATLAB's add-on, Simscape Multibody. The Multibody model better simulates real world bounding, however the system is still restricted to the sagittal plane and the force production by each set of legs is still assumed to be symmetric. The following subsection 2.2.1 details the overall system modeling construction and process. The additional subsection 2.2.2 gives a more in depth look at the Spatial Contact Force block since it acts as the focal point for reaction force creation and phase transitions.

2.2.1 Multibody Model Overview

The primary modeling difference between the hybrid model and the Multibody model is the incorporation of leg dynamics. The legs are two member bodies that are linked together through a knee joint. The overall leg is connected to the robot through a hip joint. Each individual leg member is modeled as a distributed mass. Torques at the knee and the hip act as the system input, opposed to the hybrid model where reaction forces could be directly applied to the system. Additionally the leg joints are modeled to contain some internal friction giving the system damping. By incorporating the leg dynamics, footfall is no longer idealized and bounding phase switching is now based on impact detection. Contact between the ground and the feet is modeled using the Spacial Contact Force block further covered in Subsection 2.2.2. The main assumption this modeling technique eliminates is the no-slip condition.

The Simscape Multibody environment relies on a signal diagram format. The blocks that are connected by the signal diagram include joint blocks, inertia/geometry blocks, coordinate transformation blocks, and some unique modeling blocks like the spatial contact force block. Table 2.1 lists all of the blocks used and their general purpose. Table 2.2 lists all of the constants and selected parameters for the blocks found in Table 2.1. It is important to note that the values of the robot mass, moment of inertia, and body length, are all borrowed from MIT [4].

Table 2.1: Blocks used in modeling a quadrupedal robot in MATLAB's Simscape Multibody

Block Name	Function	Quantity
Solver Configuration	Solver settings	1
World Frame	Establishes a reference frame	1
Mechanism Configuration	Non-linearities and gravity	1
Rigid Transform	Relate and orient reference frames	13
Solid	Contains model geometry and inertia	8
Planar Joint	Allows motion in a single plane	1
Revolute Joint	Allows rotation about a single axis	4
Spatial Contact Force	Models reaction forces between bodies	1

The Multibody system I designed can be broken down into three body group types: the ground plane, the quadruped body, and a lumped leg system. The ground plane is the simplest of the three, requiring only two blocks: the Rigid Transform block and the Solid block. The Solid block generates the geometry and the inertial properties of the ground plane. The Rigid Transform block allows me the ability to fix the ground plane at a specific location relative to the world frame.The Multibody ground plane construction can be seen below in Figure 2.3.The block titled Ground Plane is the Solid block. The rest of the blocks displayed are labeled with their respective block types. The three additional blocks displayed in the left are necessary for any

Table 2.2: Model constant parameters.

Symbol	**Parameter Name**	**Value**	**Units**
m	Quadruped Body Mass	33	$[kg]$
I	Quadruped Moment of Inertia	2.9	$[kg * m^2]$
r_1	Thigh Length	0.4	$[m]$
r_2	Shank Length	0.3	$[m]$
ℓ	Quadruped Body Length	0.7	$[m]$
g	Gravity	9.80665	$[m/s^2]$
k_{stiff}	Spatial Contact Force Stiffness	4x10^5	$[N/m]$
k_{damp}	Spatial Contact Force Damping	4x10^4	$[N * s/m]$
μ_s	Static Friction Coefficient	0.7	$[-]$
μ_k	Dynamic Friction Coefficient	0.7	$[-]$
μ_{vth}	Friction Velocity Threshold	0.1	$[-]$
d_1	Hip Joint Damping Coefficient	0.5	$[N * m * s/rad]$
d_2	Knee Joint Damping Coefficient	0.2	$[N * m * s/rad]$

Multibody model. The Solver Configuration block allows for selections to be made so that MATLAB utilizes the correct solver for the particular Multibody model. The World Frame block creates the world frame that all modeled bodies are related to. The Mechanism Configuration block allows for the selection of gravity and linearization parameters.

The quadruped body is the next most simple component consisting of 4 blocks. A Planar Joint block and two Rigid Transform blocks are used to relate the quadruped's Solid block to the world frame. The Planar Joint block is a type of joint block that constricts motion to a single plane. Therefore, I aligned the joint's plane of motion with the model robot's sagittal plane. Figure 2.4 shows a snapshot of the block configuration for modeling the quadruped body. The block titled Quadruped Body is the Solid block. The block titled Planar Joint is the Planar Joint block.

Since the force production by each set of legs in the front and back is assumed to have symmetric force production, a single lumped leg system is constructed for each set of legs, back and front. Each lumped leg is co-planar with the center of mass forming

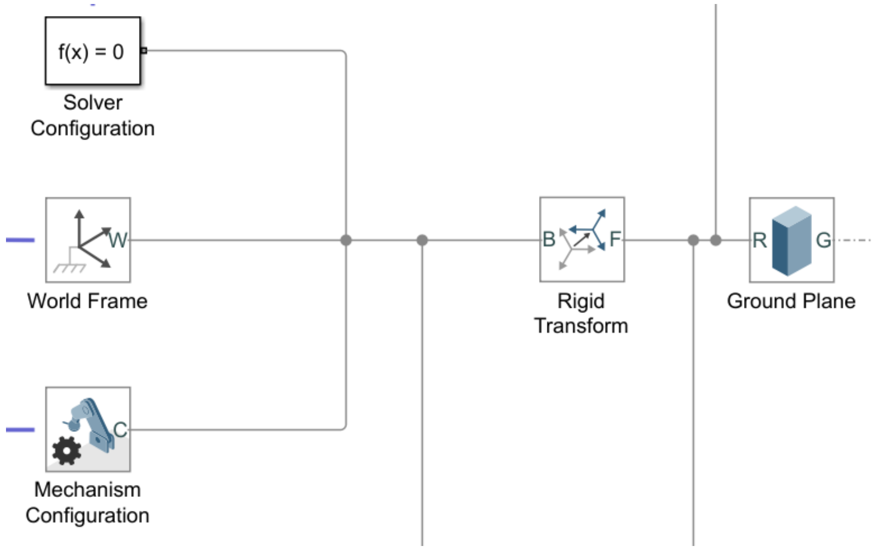

Figure 2.3: Multibody environment snapshot of the ground plane construction.

the sagittal plane. Torques and rotations at the knee and hip joints act normal to the sagittal plane.

A single lumped leg system is constructed with ten blocks. Five of the ten blocks are Rigid Transform Blocks that orient the other five blocks, Solid blocks and Revolute Joint blocks. There are three Solid blocks: the foot, the shank, and the thigh. The Revolute Joint blocks model the knee and the hip allowing rotation about one axis. Figure 2.5 displays the full block diagram for a leg body system. Each Solid block and Revolute Joint block is titled as the leg component it represents. The rest of the blocks displayed are the Rigid Transform blocks.

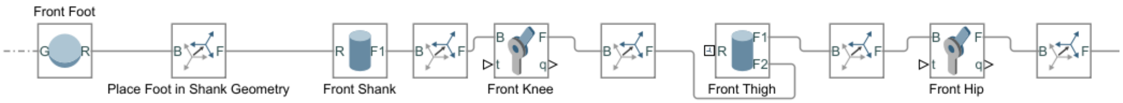

Figure 2.5: Multibody environment snapshot of the leg body system.

Figure 2.6 displays how this whole system integrates together. The Spatial Contact Force block shown connecting the robot legs to the ground plane is explained in sub-

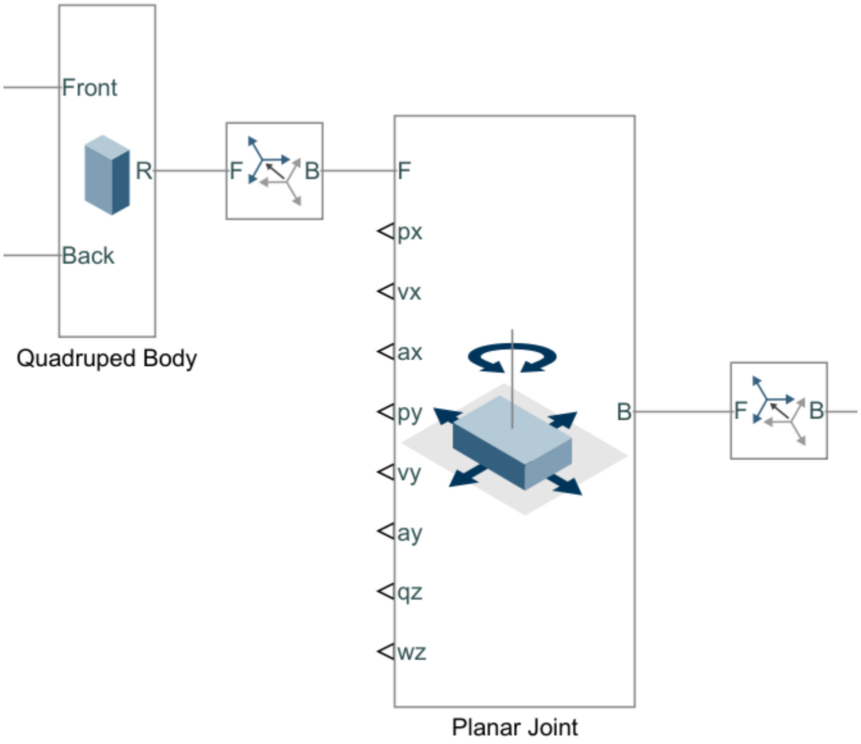

Figure 2.4: Multibody environment snapshot of the quadruped body design.

section 2.2.2. Each grey subsection block represents a type of block system discussed in the previous paragraphs.

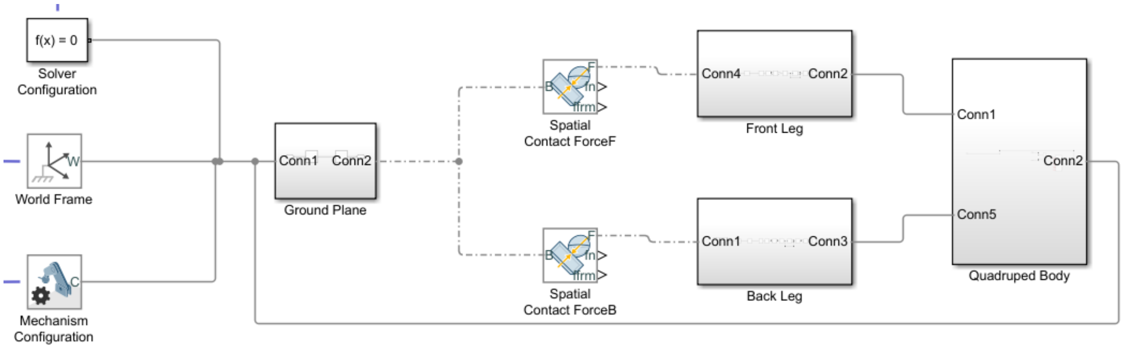

Figure 2.6: Multibody environment snapshot of the complete model.

When the Simscape Multibody model is simulated it automatically generates an animation helping to visualize the the system response. Figure 2.7 shows a snapshot of one of these 3 dimensional animations.

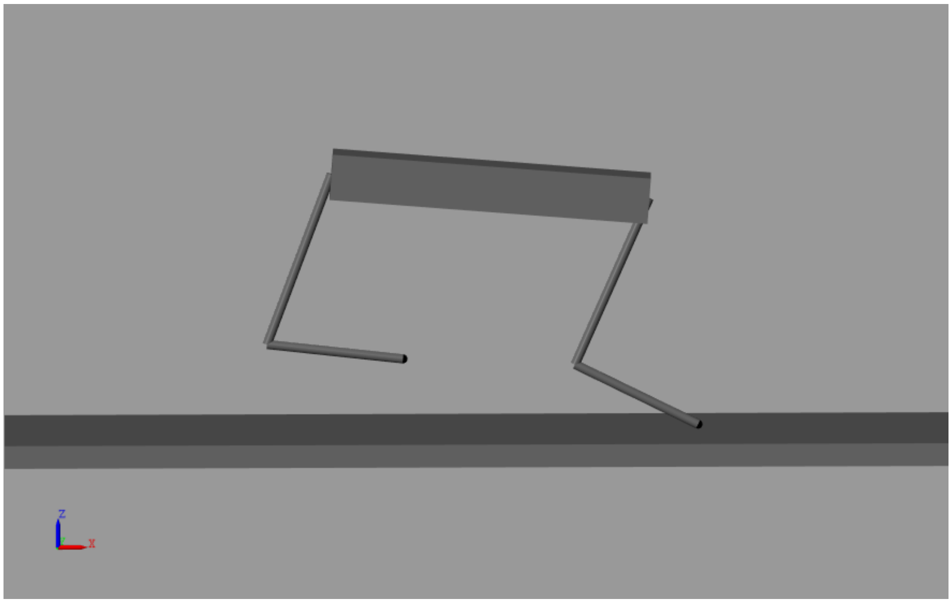

Figure 2.7: A 3-dimensional rendering of the solid-body model

2.2.2 Simscape Spatial Contact Force Block

One key block that was not detailed in subsection 2.2.1 is the Spatial Contact Force Block. This block is used to facilitate all of the reaction forces between the foot and the ground plane. Figure 2.8 displays the Spatial Contact Force Block. The foot geometry is passed to the block through port F and the ground plane geometry is passed to the block through port B. The other two ports shown are for reaction force sensing.

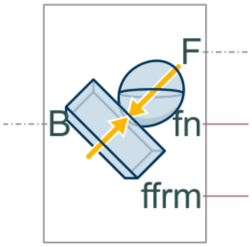

Figure 2.8: The Spacial Contact Force Block displayed as it would be in a Simulink model

In order for the Spatial Contact Force Block to model contact between two bodies, the block utilizes the penalty method. The penalty method is implemented in this case by allowing the two geometries to have a small region of penetration. The amount of penetration directly corresponds to the reaction forces produced between the two surfaces. The block utilizes a spring-damper system to model the normal force. This makes the normal force proportional to the penetration distance and velocity. A user specified transition region is also used by the block to avoid discontinuities in the solver. This region allows for a gradual ramping of the spring and damping constants. Once the penetration matches or exceeds the transition region the full user specified spring and damping constants are applied to the penetration position and velocity.

Friction force generated from the Spatial Contact Force Block is calculated from the current normal force and friction coefficient. The block uses what the Simscape Multibody Documentation refers to as ”Smooth Stick-Slip” to determine the current friction coefficient. This method has the fricition coefficient as a continuous function of the tangential velocity near the penetration region. The function mimics a second order response where the selected static friction coefficient acts as the peak value, the selected dynamic friciton coefficient acts as the steady-state value, and velocity replaces time. The peak time in this analogy is the user specified critical velocity value. With the normal force calculated and the friction coefficient function the full reaction forces are generated by the Spatial Contact Force Block.

Chapter 3

IMPULSE BASED GAIT DESIGN AND PERIODIC ORBITS

The primary desire for the control algorithm established in this paper is to maintain stable bounding. This means that this controller is only active once the system enters bounding locomotion. Therefore, the initial states of the system are desired as an end result of each complete bounding cycle. With this conclusion, the feed-forward control should support net-zero impulse in the horizontal, vertical, and pitch directions over the course of a bounding cycle. Section 3.1 details the periodic orbits created by the aforementioned feed-forward control. Section 3.2 details the formulation and selection of the feed-forward reaction force profiles. Lastly, Section 3.4 details the necessary foot trajectory to maintain the periodic orbits for solid body implementation.

3.1 Periodic Orbits

A periodic orbit describes the behavior of a special dynamic system. The system must generate state trajectories that end at the starting state on a repeating time basis. It is desired for the quadrupedal system to exhibit this behavior in order to sustain bounding. The idealized quadruped system follows four distinct state mapping trajectories that make up the full periodic orbit. Figure 3.1 shows a phase plot of one of these idealized orbits.

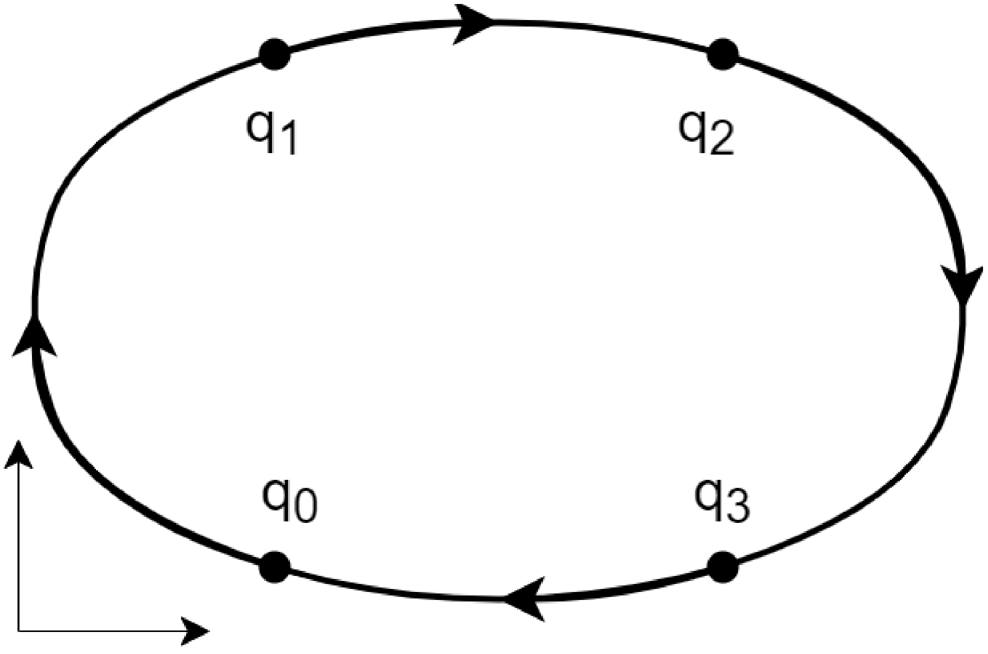

Figure 3.1: The Idealized system phase plot forming a periodic orbit.

The system governing Equation 2.1 is linear time variant because the reaction forces are known time dependent functions. Therefore, the equation can be rewritten in state space as Equation 3.1 to reorganize the system for determining a closed form solution.

$$\dot{\mathbf{q}}(t) = \mathbf{A}_i(t)\mathbf{q}(t) + \mathbf{B}_i(t) \tag{3.1}$$

where $\mathbf{A}_i$ and $\mathbf{B}_i$ are matrices relating the state variables to their time derivatives. The contents of $\mathbf{A}_i$ and $\mathbf{B}_i$ are dependent upon which phase the system is in. See Section 2.1 for further details. Since the $\mathbf{A}_i$ and $\mathbf{B}_i$ matrices are time variant for both stance phases, Equations 3.2 and 3.3 are used to determine the closed form solution for each phase state trajectory, $\mathbf{g}$. These equations are realized from Duhamel's principal but are extended to time variant coefficients [13].

$$\mathbf{g}_j(\mathbf{q}_{j-1}) = \mathbf{H}_i(T_j, 0)\mathbf{q}_{j-1} + \int_0^{T_j} \mathbf{H}_i(T_j, \tau)\mathbf{B}_i(\tau)\, d\tau \tag{3.2}$$

where T_j is the dwell time for each system phase and

$$\mathbf{H}_i(t, t_o) = e^{\int_{t_o}^{t} \mathbf{A}_i(\eta)\, d\eta} \tag{3.3}$$

Each following state in the orbit can be mapped by the previous one until the orbit is complete and reset. The final state, $\boldsymbol{q}_4$, is reset to the initial state, $\boldsymbol{q}_0$, for the next periodic orbit state mapping. Equation set 3.4, shows the state to state relationships through the state trajectory mapping.

$$\begin{aligned} &\boldsymbol{q}_j = \boldsymbol{g}_j(\boldsymbol{q}_{j-1}),\ \forall\, j \in \{1, 2, 3, 4\} \\ &\boldsymbol{q_0} = \boldsymbol{q_4} \end{aligned} \tag{3.4}$$

In order for the feed-forward system to operate on a periodic orbit the initial state needs to fall on a periodic orbit. If you let P in Equation 3.5 represent the successive execution of g over the range of j then Equation 3.6 can be used to approximately find a periodic orbit. However, periodic orbits found in simulation are unstable and diverge due to rounding error.

$$\mathbf{P}(\mathbf{q_0}) = \mathbf{q_4} \tag{3.5}$$

$$q_0 = \arg\min_{\mathbf{q}} ||\mathbf{q} - \mathbf{P}(\mathbf{q})||^2 \tag{3.6}$$

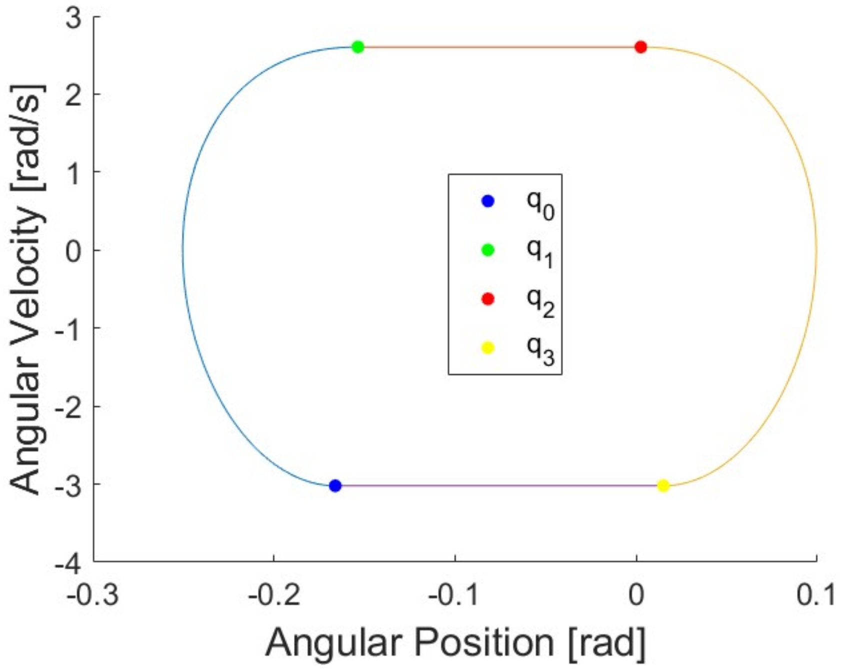

Figure 3.2: A phase plot of the open-loop hybrid model in the angular domain. Demonstrates the existence of periodic orbits.

3.1.1 Periodic Orbit Stability

To take a closer look at the actual stability of the periodic orbits the state transition matrix is utilized. Equation 3.8 displays how each state transition matrix, Φ_j, is calculated. The state transition matrix for the entire bounding cycle, Ψ, can be found by pre-multiplying each phase state transition matrix as seen in Equation 3.7. This combined total state transition matrix's eigenvalues can be used to find the system stability. By using the associated equations and calculating the state transition matrix eigenvalues numerically it is possible to verify that the constructed system actually exhibits periodic behavior.

$$\boldsymbol{\Psi} = \boldsymbol{\Phi_4 \Phi_3 \Phi_2 \Phi_1}, \tag{3.7}$$

where

$$\begin{aligned}\mathbf{\Phi_j} &= \frac{d\mathbf{g_j}}{d\mathbf{q_{j-1}}} \\ &= \mathbf{H}_i(T_j, 0)\end{aligned} \tag{3.8}$$

with $j \in \{1, 2, 3, 4\}$. The eigenvalues of $\mathbf{\Psi}$ are all equal to one. Since the transition matrix is associated with a system discretized in time, the eigenvalues of 1 tell us that the system, at best, is marginally stable. Perturbations to the system even as small as rounding error during simulation can accumulate over consecutive bounding cycles causing the orbits to destabilize. Therefore, from simulation and stability analysis it is necessary for this system to utilize a feedback controller on top of the feedforward design.

3.2 Reaction Force Profiles and Gait Timing

As discussed earlier in this chapter, the ideal reaction force profiles will cause a net zero change on the momentum of the hybrid model over the course of one full bounding cycle. Using the equations of motion 2.3 and 2.4 the linear momentum equations can be calculated by integration. The linear momentum equations are shown in equations 3.9 and 3.10.

$$m\left[\dot{x}(T) - \dot{x}(0)\right] = \int_0^T F_x dt \tag{3.9}$$

$$m\left[\dot{y}(T) - \dot{y}(0)\right] = \int_0^T (F_y - mg) dt \tag{3.10}$$

It is apparent that if linear momentum is to be conserved the left sides of equations 3.9 and 3.10 need to be set to zero. Therefore,

$$\int_0^T F_x dt = 0 \tag{3.11}$$

$$\int_0^T (F_y - mg)dt = 0 \tag{3.12}$$

When utilizing these two equations gait timing must be taken into account. For both Equation 3.11 and Equation 3.12 the F terms are only non-zero for the duration of $T_{st} \in \{T_1, T_3\}$, the time to complete a stance phase. For a desired forward bounding velocity of the robot, v_d, Equation 3.13 can be used to determine a nominal stance time.

$$T_{st} = L/v_d \tag{3.13}$$

Where L functions as the stroke length of the quadruped leg. The L value selected for any v_d in this paper is $0.4[m]$, directly taken from MIT's work done with the cheetah 2 [4]. This length can be thought of as the horizontal distance the foot travels relative to the robot body during a stance phase.

By using Equation 3.13 to determine stance time and selecting the hip offset, γ_x, to be half the stroke length, symmetric force production about the robot mass center is achieved. This symmetric force production over the course of a bounding cycle satisfies the net-zero impulse in the pitch direction. Keep in mind, the hip offset γ_x, is only guaranteed for the hybrid model.

With the reaction force timing found, the last step is to select a scalable reaction force function. For this paper, half sine wave functions are chosen as the base force profiles since they start at zero and end at zero with smooth symmetric ramping in between. These characteristics help to ensure smooth phase transitions by minimizing discontinuities at impact and foot-lift. Using equations 3.11 and 3.12 the amplitude coefficient can be selected for each reaction force profile. The amplitude coefficient for the reaction force in the x-direction is trivial. As long as the front stance and the back stance exhibit equal and opposite force profiles linear momentum in the x-direction will be conserved. In theory, the reaction forces in the x-direction can be set to zero. However, robot pitch oscillation is at its most extreme with zero reaction forces in the x-direction [14]. To find the reaction force amplitude coefficient in the y-direction, A_y, Equation 3.12 can be modified to the form of Equation 3.14.

$$mgT = 2\int_0^{T_{st}} A_y sin(\pi t/T_{st})dt \tag{3.14}$$

Solving equation 3.14 for A_y gives equation 3.15

$$A_y = \frac{1}{4}mg\pi T/T_{st}; \tag{3.15}$$

With the vertical reaction force amplitude coefficient derived a full set of reaction force profiles can be selected for any given bounding parameters. Figures 3.3 and 3.4 display an example set of reaction force profiles for one full bounding cycle.

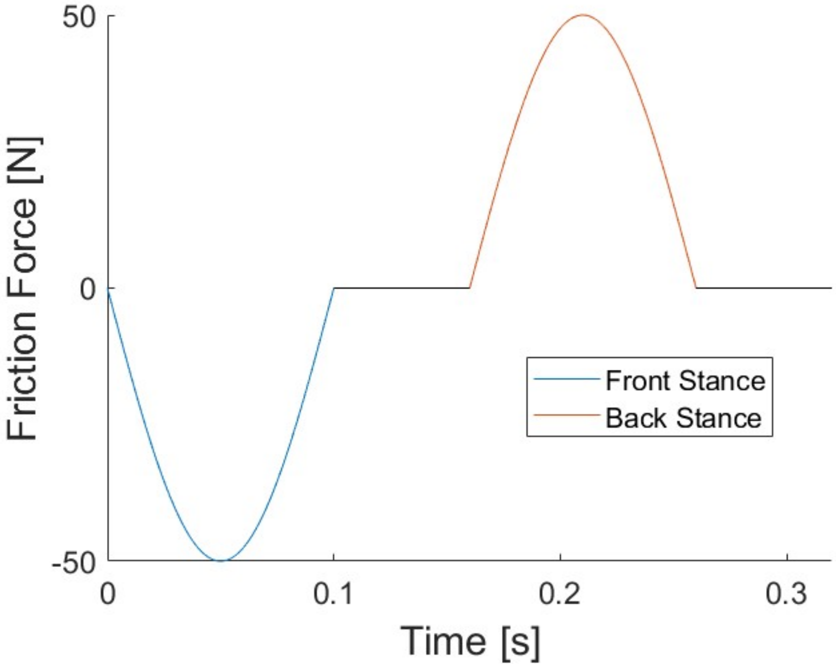

Figure 3.3: A plot of the friction force as a function of time over the course of one full bounding cycle

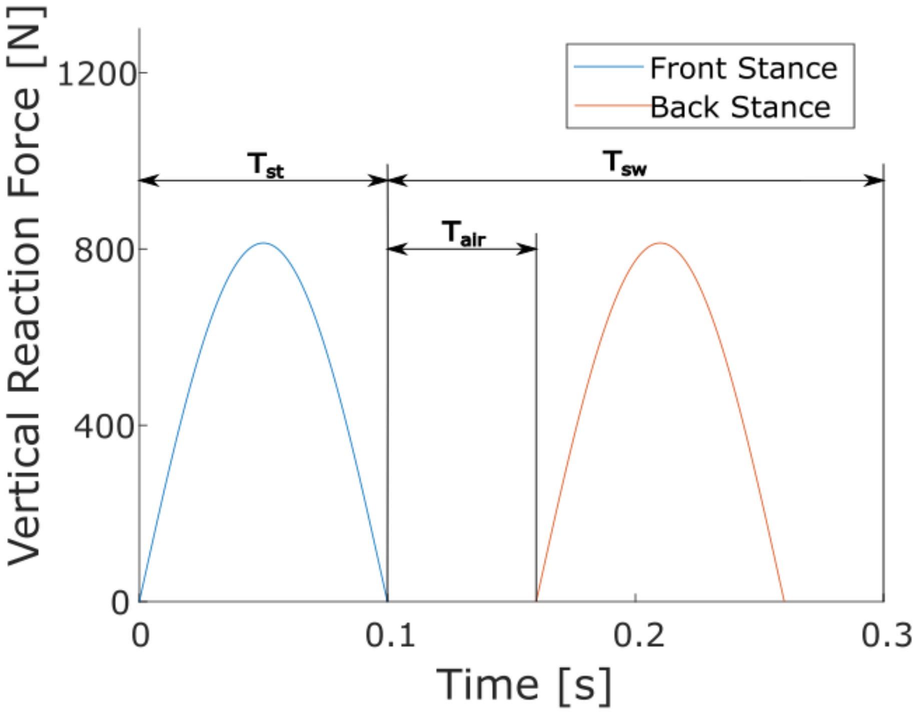

Figure 3.4: A plot of the normal force as a function of time over the course of one full bounding cycle

T_{sw}, the time a specific leg is off the ground, is defined by Equation 3.16. This time is held constant for any v_d at $0.22[s]$, again utilizing MIT's work with the Cheetah 2 [4].

$$T_{sw} := T_{st} + 2T_{air} \tag{3.16}$$

$T_{air} \in \{T_2, T_4\}$ represents the time the quadruped has both feet off the ground. With the known value of T_{st} from Equation 3.13 and the constant value of T_{sw}, Equation 3.16 can be manipulated to solve for T_{air} in Equation 3.17.

$$T_{air} = \frac{1}{2}(T_{sw} - T_{st}) \tag{3.17}$$

The new gait timing terms from this chapter are shown in Figure 3.4 for visual reference.

3.3 Torque Reaction Force Mapping

This section finds the equations necessary to link the knee and hip input torques, M_A and M_O, to the foot horizontal and vertical reaction profiles. The leg system being analyzed is made up of two mass-less limbs linked by a knee joint. The top limb, r_1, is called the thigh which creates a rigid body connection between the lumped mass at the hip and the knee joint. The bottom limb, r_2, is called the shank which creates a rigid-body connection between the knee joint and the mass less foot. The foot in this leg analysis is assumed to exhibit no slip and remain in contact with the ground. For a visual of the leg look to the free body diagram in Figure 3.5.

Since the legs are assumed to be mass-less, the power input at at the hip and knee is equal to the power used to move the hip. Because of this relationship the input torques can be set equal and opposite to the moments created by the reaction forces. This can be written as Equation 3.18 where **M** is a column vector containing M_A and M_O, and **F** is a column vector containing F_x and F_y [4].

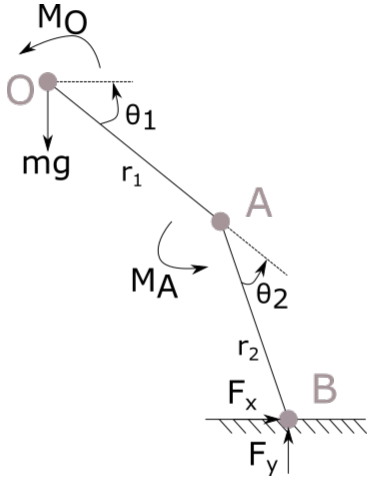

Figure 3.5: The free body diagram of a single leg

$$\mathbf{M} = -\mathbf{J}^T\mathbf{F} \tag{3.18}$$

The Jacobian matrix, **J**, is made up of the partial derivatives of each system function with respect to each function parameter. The system functions being considered are the x and y positions. Using the forward kinematics, the two functions can be generated, shown as Equation 3.19 and Equation 3.20 .

$$x(\theta_1, \theta_2) = r_1 cos(\theta_1) + r_2 cos(\theta_1 + \theta_2) \tag{3.19}$$

$$y(\theta_1, \theta_2) = r_1 sin(\theta_1) + r_2 sin(\theta_1 + \theta_2) \tag{3.20}$$

The Jacobian matrix is made by taking the partial derivatives of x and y with respect to θ_1 and θ_2. The Jacobian matrix is displayed as Equation 3.21.

$$\mathbf{J} = \begin{bmatrix} -r_1 sin(\theta_1) - r_2 sin(\theta_1 + \theta_2) & -r_2 sin(\theta_1 + \theta_2) \\ r_1 cos(\theta_1) + r_2 cos(\theta_1 + \theta_2) & r_2 cos(\theta_1 + \theta_2) \end{bmatrix} \tag{3.21}$$

Knowing the above information Equation 3.18 can be solved for **F** by premultiplying Equation 3.22 to both sides.This gives the reaction forces at the foot as a function of the input torques.

$$\mathbf{J}^{-T} = \begin{bmatrix} \frac{cos(\theta_1+\theta_2)}{r_1 sin(\theta_2)} & -\frac{r_2 cos(\theta_1+\theta_2)+r_1 cos(\theta_1)}{r_1 r_2 sin(\theta_2)} \\ \frac{sin(\theta_1+\theta_2)}{r_1 sin(\theta_2)} & -\frac{r_2 sin(\theta_1+\theta_2)+r_1 sin(\theta_1)}{r_1 r_2 sin(\theta_2)} \end{bmatrix} \tag{3.22}$$

where the exponent of $-T$ is the transpose of the inverse.

3.4 Foot Trajectory

The foot position for the hybrid model is enforced by j transitions allowing for equal and opposite torques to be generated about the mass center over a bounding cycle. This is a critical component to periodic orbit behavior within this system. In general the hybrid model assumes perfect footfall independent of the system states, however for solid-body and real world implementation the transitions out of flight phase are caused by leg contact with the ground. To begin to handle this increase in system complexity, a foot trajectory must be utilized for leg swing to satisfy the j transitions from flight to stance phases.

A new time reference is defined in Equation 3.23; s_n is the normalized time where $t^n_{sw} \in \mathbb{R}_+$ is the current time duration of a particular leg swing, front or back. t^n_{sw} for a particular swing is reset to zero when that swing starts again.

$$s_n = \frac{t^n_{sw}}{T_{sw}}, \; \forall \; n = f, b \tag{3.23}$$

where f is for the front leg swing and b is for the back leg swing. The trajectory used in this paper is generated using a set of Bézier curves defined in equations 3.24 and 3.25. The coefficients $\beta_x \in \mathbb{R}^{10}$ and $\beta_y \in \mathbb{R}^{10}$, listed in Equation 3.26 and Equation 3.27, were recommended by MIT [15].

$$\gamma_x := \gamma_x(\beta_x, s_n) \tag{3.24}$$

$$\gamma_y := \gamma_y(\beta_y, s_n) \tag{3.25}$$

$$\beta_x = [-0.2, -0.259, -0.275, -0.384, 0.261, -0.017, 0.248, 0.267, 0.259, 0.2] \tag{3.26}$$

$$\beta_y = -[0.5, 0.45, 0.406, 0.065, 1.031, -0.095, 0.545, 0.374, 0.45, 0.5] \tag{3.27}$$

Figure 3.6 displays the foot trajectory over the course of one leg swing relative to the hip position. This trajectory satisfies the location constraint of the flight to stance transition, however it still relies on the no-slip assumption. To help ensure the no-slip condition during solid-body and real-world implementation, foot-to-ground velocity

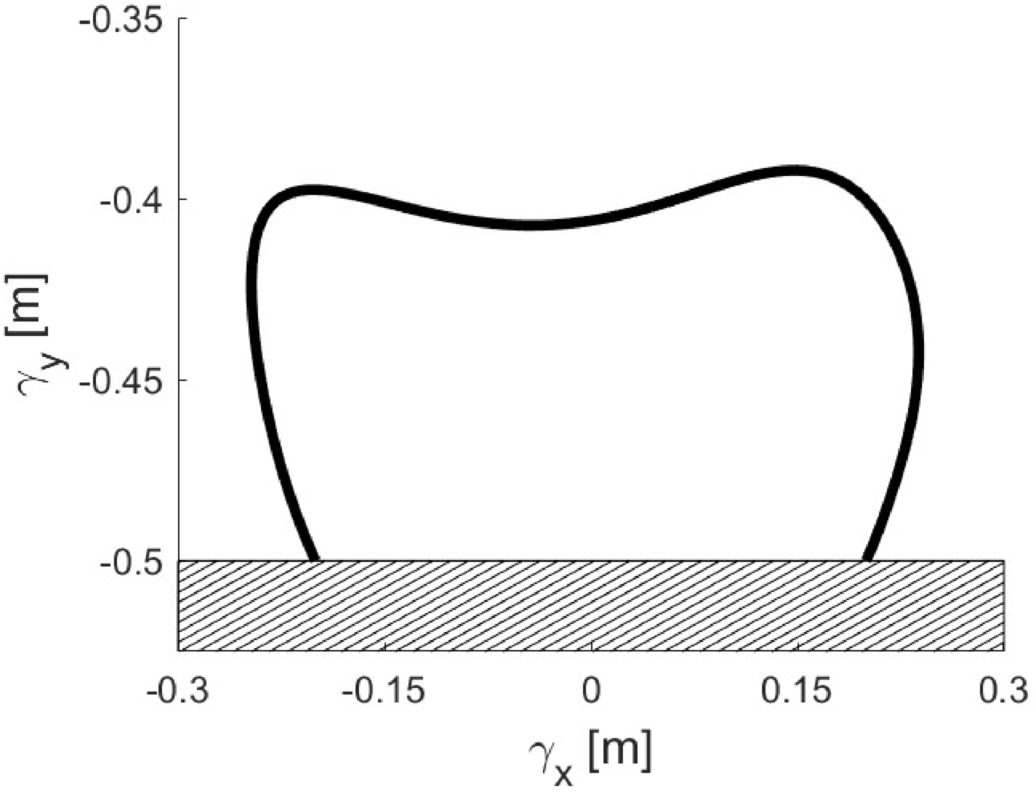

Figure 3.6: Foot position relative to the hip over the course of a foot swing.

matching needs to be added to the foot trajectory by redefining γ_x in Equation 3.28.

$$\gamma_x := \gamma_x(\beta_x, s_{swing}(\beta_{swing}, s_n)) \tag{3.28}$$

where s_{swing} is a new Bézier curve with coefficients $\beta_{swing} \in \mathbb{R}^4$. It is important to note that s_n is still used as a parameter for s_{swing} and that β_x is still the set of Bézier coefficients for γ_x. The goal is to modify s_n so that the Bézier curve satisfies equations, 3.29 and 3.30.

$$\dot{\gamma}_x(s_{swing}(0)) = \frac{\partial \gamma_x}{\partial s_{swing}} \dot{s}_{swing}\bigg|_{s=0} = \dot{\gamma}_x(\tau) \tag{3.29}$$

$$\dot{\gamma}_x(s_{swing}(1)) = \frac{\partial \gamma_x}{\partial s_{swing}} \dot{s}_{swing}\bigg|_{s=1} = v_d \tag{3.30}$$

The β_{swing} values are calculated by Equation 3.31. τ represents the time at the end of a stance phase and v_d represents the nominal horizontal velocity desired.

$$\beta_{swing} = \begin{bmatrix} 0, & \frac{1}{3}(\dot{\gamma}_x(\beta_x, 0))^{-1}\dot{\gamma}_x(\tau), & 1 - \frac{1}{3}(\dot{\gamma}_x(\beta_x, 1))^{-1}v_d, & 1 \end{bmatrix} \tag{3.31}$$

With the modified horizontal foot trajectory in place the overall foot trajectory satisfies both the position and velocity constraints. Despite meeting theses constraints, perfect footfall in real-world or solid-body implementation is still not guaranteed. Since the normalized time relies on accurate swing duration, disturbances to gait timing can still interfere with perfect footfall. This issue is handled later in Subsection 4.2.3.

Chapter 4

CONTROLLER DESIGN

With the two models from Chapter 2 and the gait analysis conducted in Chapter 3, controller design and implementation is possible. The complete controller is made up of two separate leg phases. One control scheme is implemented for a leg in the stance phase and a different control scheme is implemented for the duration of T_{sw}. A leg finite state machine coupled with impact detection handles the controller scheme switching, shown in Figure 4.1. Both of these separate control states contain a feed-forward component and a feedback component. The following sections will discuss both components for each leg phase controller.

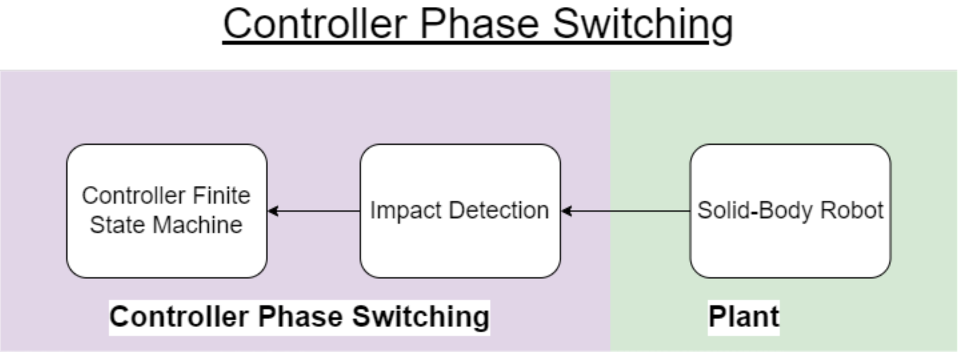
Controller Phase Switching
Controller Finite State Machine
Impact Detection
Solid-Body Robot
Controller Phase Switching
Plant

Figure 4.1: A signal diagram of the controller switching

4.1 Stance Phase Controller

While a leg is in the stance phase it has the ability to provide input to the overall system by generating reaction forces with the ground. This characteristic is unique to the stance phase. Since the ground reaction forces are the only way to control the quadruped's gait and body state variables, all of the control schemes regarding those aspects are implemented during the stance phase. Figure 4.2 shows the signal control diagram for the Stance Phase Controller within the controller finite state machine.

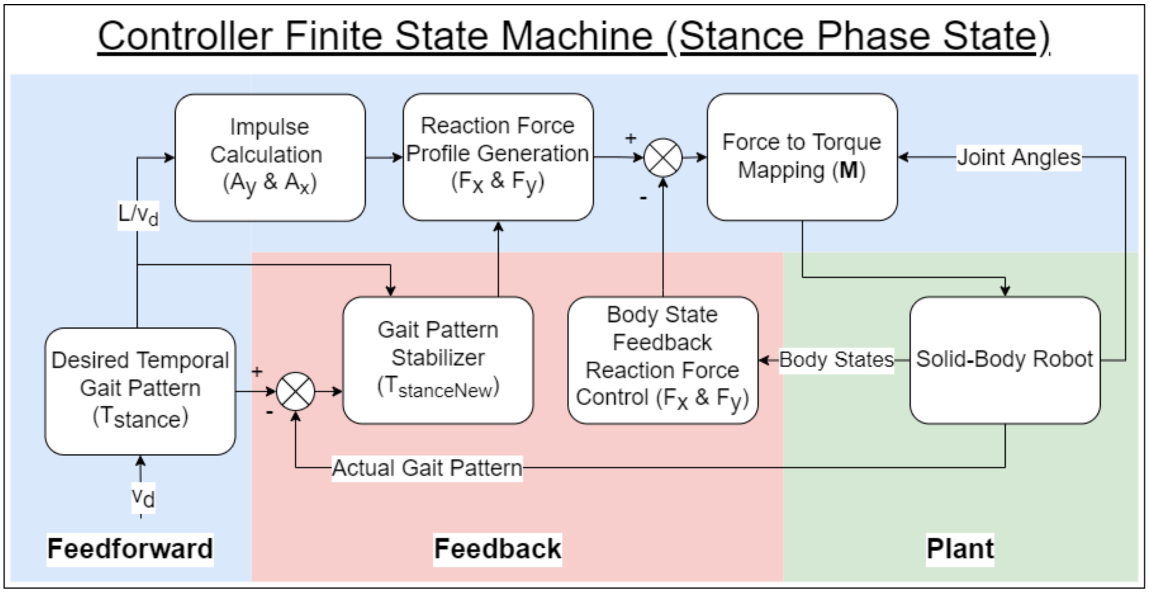
Controller Finite State Machine (Stance Phase State)
Impulse Calculation (A_y & A_x)
Reaction Force Profile Generation (F_x & F_y)
+
-
Force to Torque Mapping (**M**)
Joint Angles
L/v_d
Gait Pattern Stabilizer ($T_{stanceNew}$)
Body State Feedback Reaction Force Control (F_x & F_y)
Body States
Solid-Body Robot
Desired Temporal Gait Pattern (T_{stance})
+
-
Actual Gait Pattern
v_d
Feedforward
Feedback
Plant

Figure 4.2: Block diagram for the stance phase state internal control implementation.

4.1.1 Stance Feedforward

To give the system a trajectory to stabilize around, feedforward reaction force profiles and gait timing is employed. This is the first layer of the Stance Phase controller and sets the stage for the rest of the design.

The first parameter selected for the feedforward control is the desired horizontal bounding speed, v_d. Using the selected v_d with Equation 3.13 gives the stance time, T_{st}. Using Equation 3.17 and the experimental swing time value, T_{sw}, the full nominal gait timing is determined.

With the Gait timing determined, Equation 3.15 can be used to find the normal reaction force amplitude, A_y. With A_y found and a user selected friction reaction force amplitude, A_x, the reaction force profiles can be generated as half sine waves.

Equation 4.1 and Equation 4.2 give the general forms of the reaction force profiles. Examples of these two profiles can be seen in Figure 3.4 and Figure 3.3.

$$F_y = A_y sin(\pi t/T_{st}) \tag{4.1}$$

$$F_x = \pm A_x sin(\pi t/T_{st}) \tag{4.2}$$

The last aspect of the feedforward control design is the utilization of Equation 3.18 to generate torque inputs that will cause the reaction forces desired at the foot.

4.1.2 Body State Feedback Controller

The first layer of control to add on top of the feedforward control is a body-state feedback controller. This layer of the overall controller is purely focused on adding stability around the periodic orbits found through the hybrid model in Section 2.1. The body-state feedback controller utilizes impedance based control law to generate modified reaction force profiles during the stance phase. The total reaction force profile generated is broken down into components in Equation 4.3.

$$\mathbf{F} = \mathbf{F}^* + \mathbf{F}_{hip} + \mathbf{F}_v + \mathbf{F}_\theta \tag{4.3}$$

$\mathbf{F}^*$ is the nominal reaction force profile discussed in sections 3.2 and 4.1. $\mathbf{F}_{hip}$ is the feedback force profile focused on stabilizing quadruped hip height, $y_{hip} \in \mathbb{R}$, shown in Equation 4.4.

$$\mathbf{F}_{hip} = \Big[0, \quad g_{fb}(s_{st})(-k_{Py}(y_{hip} - y_{des}) - k_{Dy}(\dot{y}_{hip}))\Big]^T \tag{4.4}$$

$$s_{st} = \frac{\bar{t}}{T_{st}} \tag{4.5}$$

k_{Py} is the proportional gain selected for biasing the system towards the desired hip height, y_{des}. k_{Dy} is the damping gain selected for minimizing the hip vertical velocity, $\dot{y}_{hip} \in \mathbb{R}$. The g_{fb} function acts as a way to ensure the overall reaction force profile does not become negative while the nominal reaction force profile is small. The function behavior can be seen in Figure 4.3.

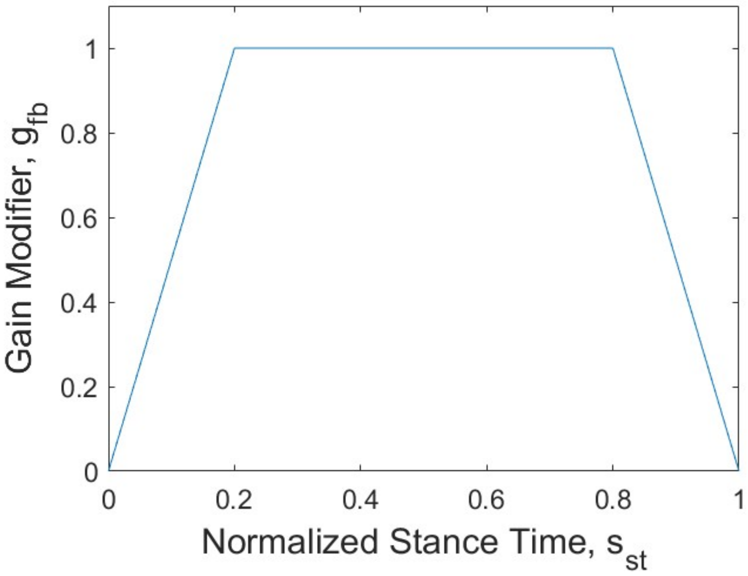

Figure 4.3: Gain modifier as a function of normalized stance time.

$\mathbf{F}_v$ is the feedback force profile designed to stabilize the system around the desired bounding velocity, shown in Equation 4.6.

$$\mathbf{F}_v = \begin{bmatrix} -k_{Dx}(\dot{x} - v_d), & 0 \end{bmatrix}^T \tag{4.6}$$

k_{Dx} is a damping coefficient selected to minimize horizontal velocity deviations from the desired speed.

$\mathbf{F}_\theta$ is the feedback force profile that biases the system toward zero pitch, shown in Equation 4.7.

$$\mathbf{F}_\theta = \frac{1}{x - x_{foot}} \begin{bmatrix} 0, & k_{P\theta}\theta + k_{D\theta}(\dot{\theta} - \dot{\theta}_{des}) \end{bmatrix}^T \tag{4.7}$$

$k_{P\theta}$ is the proportional gain for minimizing quadruped body angle. $k_{D\theta}$ is the damping gain associated with biasing the angular velocity toward a desired angular velocity, $\dot{\theta}_{des}$. The fraction in front functions as a way to scale the feedback force profile with the actual moment arm available to the system.

4.1.3 Gait Pattern Stabilizer

The feedforward reaction force profile combined with the state feedback controller is sufficient for stabilizing the hybrid model but remains lacking for solid-body and real world implementation. This is primarily due to imperfect foot fall and gait timing. To help alleviate the gait timing issues, Equation 4.8 is utilized to make the stance time adaptive.

$$T_{st}^{+} = \frac{L}{v_d} - \kappa(T_{st}^{-} + T_{air}^{-} - \frac{T}{2}) \tag{4.8}$$

In this equation, $^{+}$ denotes the new variable value while $^{-}$ denotes the previous variable value. In the equation above, the old T_{st} is added to the most recent T_{air} to be compared against half a bounding cycle, $\frac{T}{2}$. The disparity between the two half cycles is modified using the gait gain, κ, before being subtracted from the nominal stance time. The T_{st}^{+} value is calculated during transitions to a new stance phase.

4.2 Swing Phase Controller

The Stance Phase Controller does the heavy lifting when it comes to stabilizing the system. However, it relies upon perfect footfall in order to do that. The Swing Phase Controller does its best to provide this footfall. In order to achieve this goal the controller utilizes a feedforward scheme to generate angle trajectories from a nominal foot trajectory. The Swing Phase Controller also utilizes a feedback control scheme to adjust foot trajectories based upon air time and ground velocity matching. In addition, the feedback control employs Proportional, Integral, and Derivative (PID) control to minimize angle error during swing. Figure 4.4 displays the Swing Phase Control signal diagram.

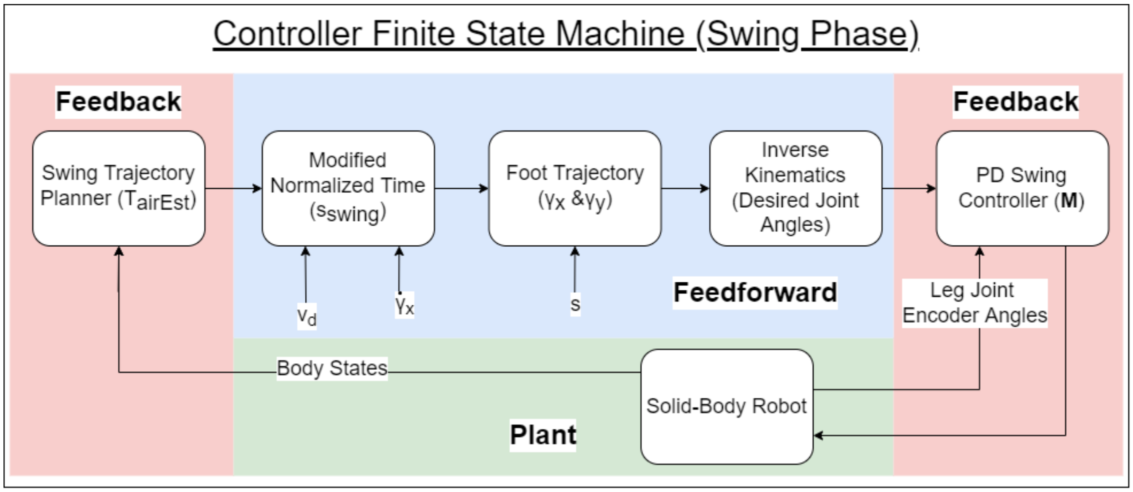
Controller Finite State Machine (Swing Phase)
Feedback
Swing Trajectory Planner (T_{airEst})
Modified Normalized Time (s_{swing})
Foot Trajectory (γ_x & γ_y)
Inverse Kinematics (Desired Joint Angles)
Feedback
PD Swing Controller (**M**)
v_d
γ_x
s
Feedforward
Leg Joint Encoder Angles
Body States
Solid-Body Robot
Plant

Figure 4.4: Block diagram for the swing phase state internal control implementation.

4.2.1 Swing Feedforward

The feedforward part of the swing phase control scheme is concerned with generating a desired nominal foot trajectory and the corresponding leg angle trajectories. The foot trajectory design aims to place the foot at the correct horizontal location for footfall while matching the horizontal ground velocity. In addition, the foot trajectory should provide enough clearance with the ground so that an early stance phase is not triggered. For in depth details regarding the formulation of the trajectory see Section 3.4.

The first step in the feedforward control is to generate a normalized time, s_n, from Equation 3.23 that can be passed into the s_{swing} Bézier curve. Now with both s_n and s_{swing} calculated, Equation 3.25 and Equation 3.28 can be used to find the current desired foot location. Even though the desired foot location is determined, that still leaves the leg joint angles as unknowns.

To find the foot position from the current angles is relatively straight forward. Equations 3.19 and 3.20 provide a simple solution. However, if the unknown values are the angles and the know values are the Cartesian coordinates of the foot, the solution process is a lot more difficult. The set of equations becomes a coupled nonlinear system. To solve this system the Newton-Raphson method is employed. The basic idea behind the Newton-Raphson method is to find a function's roots by approximating the function with its tangent line. It is necessary to provide an initial guess to the solution method. The initial guess needs to be close to the desired solution so that it doesn't converge to another valid solution. This potential issue is avoided because in both simulation and real-world implementation the current joint angular position is known. The current joint angles can be used as the guess to find the closest desired joint angles. This solution technique both minimizes simulation and system response time. Equation 4.9 shows the general form of the Newton-Raphson method for a single function. However, Inverse Kinematics for a two member leg requires a system of equations. Equation 4.10 displays the Newton-Raphson method applied to the leg system in a matrix form.

$$w_{d+1} = w_d - \frac{f(w_d)}{f'(w_d)} \tag{4.9}$$

$$\chi_{\mathbf{d+1}} = \chi_{\mathbf{d}} - \mathbf{J^{-1}f}(\chi_{\mathbf{d}}) \tag{4.10}$$

where $\chi = [\theta_1, \theta_2]^T$ and J can be found from Equation 3.21. $d \in Z^+$ represents the current iteration of the solution. $\mathbf{f}(\chi_{\mathbf{d}})$, shown in Equation 4.11, is the rearranged form of Equation 3.19 and Equation 3.20 so that the desired foot coordinates, $x_{footDes}$ and $y_{footDes}$, are included.

$$\mathbf{f}(\chi_{\mathbf{d}}) = \begin{bmatrix} r_1 cos(\theta_1) + r_2 cos(\theta_1 + \theta_2) - x_{footDes} \\ r_1 sin(\theta_1) + r_2 sin(\theta_1 + \theta_2) - y_{footDes} \end{bmatrix} \tag{4.11}$$

Using Equation 4.10 and Equation 4.11 iteratively until $|\chi_d - \chi_{d-1}| < \epsilon$ gives a sufficiently accurate desired angular trajectory to match the desired foot trajectory. ϵ is a small user specified error margin.

4.2.2 Proportional and Derivative Control

Selecting and generating a leg trajectory is critical to maintaining stable foot fall. However, any trajectory is useless if the robot does not have a way to control its leg position. This section details the method used for creating a position controller for the robot leg during the swing phase.

For details of the feedforward components, foot trajectory and the inverse kinematics, see Section 3.4 and Subsection 4.2.1. The feedback portion of the control scheme utilizes proportional-derivative (PD) style control. The controller is designed to position the knee and hip motors at specific angles in order to follow the foot trajectory processed in the feedforward control. The way the controller works is by minimizing the joint angle error. Equation 4.12 shows the formula for calculating the knee or hip error, $e_\theta(t)$.

$$e_\theta(t) = \theta_{des}(t) - \theta_{cur}(t) \tag{4.12}$$

θ_{des} is the desired hip angle and θ_{cur} is the current hip angle. After the error is calculated its derivative is taken. These two different error values are used to determine the torque input to a motor, $M(t)$. The control algorithm, shown in Equation 4.13, is best framed in the context of the S-Domain where $E_\theta(s) = \mathcal{L}\{e_\theta(t)\}$ and $M(s) = \mathcal{L}\{M(t)\}$.

$$M(s) = E_\theta(s)(k_P + \frac{k_D k_N}{1 + k_N/s}) \tag{4.13}$$

k_P is the proportional gain, k_D is the derivative gain, and k_N is the filter gain. These gains can be tuned and selected for a desired leg system response. Variables can be selected from a Simulink model to optimize for a given set of response requirements.

4.2.3 Swing Trajectory Planner

A limiting factor of the foot trajectory designed so far is that it relies on a constant T_{sw} value. As stated earlier in Section 3.2 the nominal expected T_{sw} value is 0.22[s]. However, this value fails to take into account the variations in T_{st} from the Gait Pattern Stabalizer and the variations in T_{air} from any disturbances in the vertical or rotational directions. To solve this problem, T_{sw} in Equation 3.23 is replaced with T_{sw}^{+}. T_{sw}^{+} is calculated by Equation 4.14 at each transition $j = \{1, 3\}$ when the system switches from a stance phase to a flight phase.

$$T_{sw}^{+} = T_{st}^{+} + T_{air}^{-} + T_{airEst} \tag{4.14}$$

T_{airEst} is a value that estimates the duration of the following flight phase. This value is calculated using the robot body state variables at the end of the previous stance phase. Equation 4.15 provides the nonlinear equation that dictates the value T_{airEst}. Equation 4.15, like the inverse kinematics, is solved numerically using the Newton-Raphson method. Just as the inverse kinematics have multiple solutions so does this equation. To ensure the correct T_{airEst} is found, the nominal T_{air} is used as the initial guess.

$$y_{des} = -\frac{1}{2}gT_{airEst}^2 + \dot{y}T_{airEst} + y \pm \frac{1}{2}\ell sin(\dot{\theta}T_{airEst} + \theta) \tag{4.15}$$

The first three terms describe the projectile motion of the robot mass center in the vertical domain. The last term is positive for transition $j = 3$ and is negative for transition $j = 1$. This final term is the vertical distance between the mass center and the hip. Therefore, all four terms on the right estimate the hip location at T_{airEst}. The idea is to find a T_{airEst} value that places the hip at the desired location for impact, y_{des}. With Equation 4.14 and Equation 4.15 in place the foot trajectory can be more reliable when the system is experiencing gait perturbations.

Chapter 5

RESULTS

With the system modeled and controller design complete, simulation results can be reviewed for modeling accuracy and controller performance insight. This chapter will take a look at 4 different simulation configurations and their results. Section 5.1 reviews the hybrid model performance while only utilizing the nominal reaction force profiles for input. Section 5.2 examines the results of implementing the body state feedback controller on the hybrid model. Section 5.3 takes a look at the solid-body performance with limited control during both stance and swing phases. Lastly, Section 5.4 reviews the results from the full control scheme implemented on the solid-body model.

5.1 Open-loop Hybrid Model

The hybrid model open-loop simulations utilize zero feedback control and rely upon the nominal feedforward reaction force profiles to dictate system behavior. In addition, the hybrid model assumes perfect foot fall. Therefore, the swing phase feedforward and feedback control are not implemented. The simulation initial conditions were found through the use of Equation 3.6. These initial conditions should theoretically place the system on a periodic orbit. Figure 5.1 and Figure 5.2 show these periodic orbits over the duration of $n = 4$ bounding cycles. Table 5.1 lists all the needed parameters to replicate this simulation. As was the case in Table 2.2 some of the values for the robot are borrowed from MIT's work with the Cheetah 2. These values include the robot body mass, mass moment of inertia, body length, stride length, and swing time [4]. Additionally, the value for the friction force amplitude was inspired by a recommended value from MIT [14].

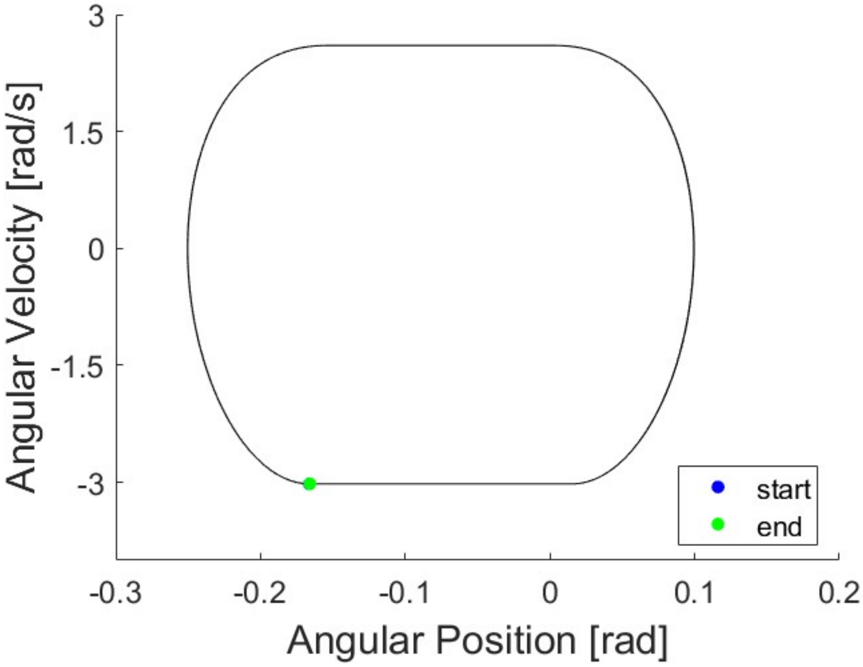

Figure 5.1: A phase plot for the open-loop hybrid model in the angular domain. $n = 4$

Figures 5.1 and 5.2 suggest that the system is completely stable on its own. However, once the number of bounding cycles increases simulations suggest some instability. Figure 5.3 shows the angular phase plane after 15 bounding cycles. The instability shown in the figure may only be due to rounding error found in any simulation. No matter where the instability comes from, it is important to realize the system could easily be destabilized when less model simplifications are in place. In order to design a robust system, feedback control needs to be implemented.

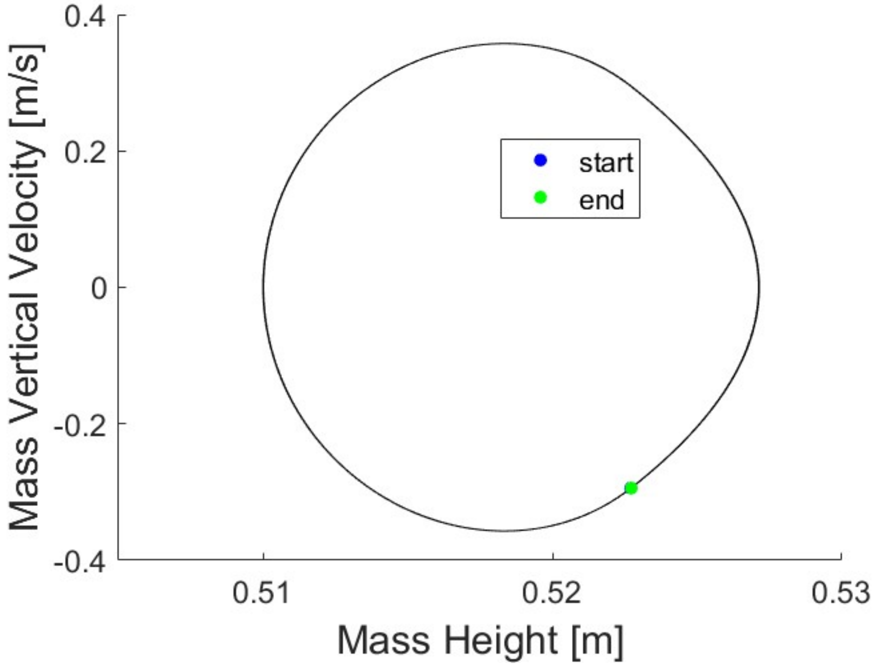

Figure 5.2: A phase plot for the open-loop hybrid model in the vertical domain. $n = 4$

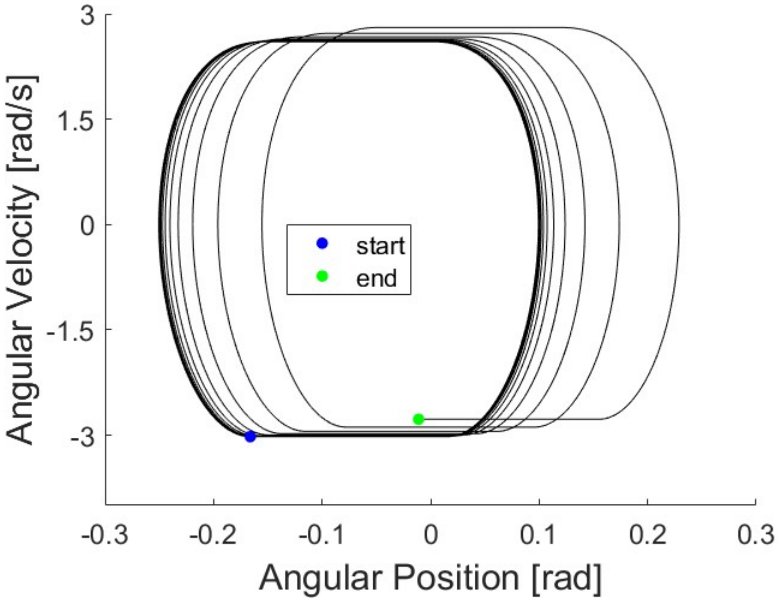

Figure 5.3: Open-loop hybrid model phase plot in the angular domain. $n = 15$

Table 5.1: Open-loop hybrid model simulation parameters.

Symbol	Parameter Name	Value	Units
m	Body Mass	33	$[kg]$
I	Mass Moment of Inertia	2.9	$[kg * m^2]$
ℓ	Body Length	0.7	$[m]$
g	Gravity	9.80665	$[m/s^2]$
L	Stride Length	0.4	$[m]$
T_{sw}	Swing Time	0.22	$[s]$
θ_0	Initial Angular Position	-0.1662	$[rad]$
$\dot{\theta}_0$	Initial Angular Velocity	-3.0207	$[rad/s]$
y_0	Initial Vertical Position	0.5227	$[m]$
$\dot{y}_0$	Initial Vertical Velocity	-0.2942	$[m/s]$
x_0	Initial Horizontal Position	-0.5734	$[m]$
$\dot{x}_0$	Initial Horizontal Velocity	4	$[m/s]$
v_{des}	Desired Horizontal Velocity	4	$[m/s]$
A_y	Normal Reaction Force Amplitude	813.6222	$[N]$
A_x	Friction Reaction Force Amplitude	50	$[N]$

5.2 Closed-loop Hybrid Model

The hybrid model closed-loop simulations still do not use the full stance phase controller. These simulations only utilize the reaction force profiles generated by combining the nominal reaction force profiles with the reaction force profiles created from the body state feedback control. As stated in the previous section, the hybrid model does not need any swing phase control implemented because of its perfect footfall assumption. In addition, the footfall assumption makes the gait pattern stabilizer irrelevant for the hybrid model.

The closed-loop hybrid model simulations were done starting at a set of arbitrary initial conditions that do not fall on a periodic orbit. All new or changed parameter values from the open-loop simulations are displayed in Table 5.2. The full set of open-loop parameters can be found in Table 5.1. The gains selected in Table 5.2 are suggested values from MIT [4].

Table 5.2: Closed-loop hybrid model simulation parameters.

Symbol	Parameter Name	Value	Units
θ_0	Initial Angular Position	-0.16	$[rad]$
$\dot{\theta}_0$	Initial Angular Velocity	-3.02	$[rad/s]$
$\dot{y}_0$	Initial Vertical Velocity	0	$[m/s]$
x_0	Initial Horizontal Position	0	$[m]$
k_{Py}	Vertical Proportional Gain	1000	$[N/m]$
k_{Dy}	Vertical Damping Gain	120	$[N * s/m]$
k_{Dx}	Horizontal Damping Gain	60	$[N * s/m]$
$k_{P\theta}$	Angular Proportional Gain	30	$[N/rad]$
$k_{D\theta}$	Angular Damping Gain	15	$[N * s/rad]$
y_{des}	Vertical Hip Bias	0.48	$[m]$
$\dot{\theta}_{des}$	Angular Velocity Bias	0	$[rad/s]$

Figure 5.4 and Figure 5.5 display the system response of the closed-loop hybrid model.

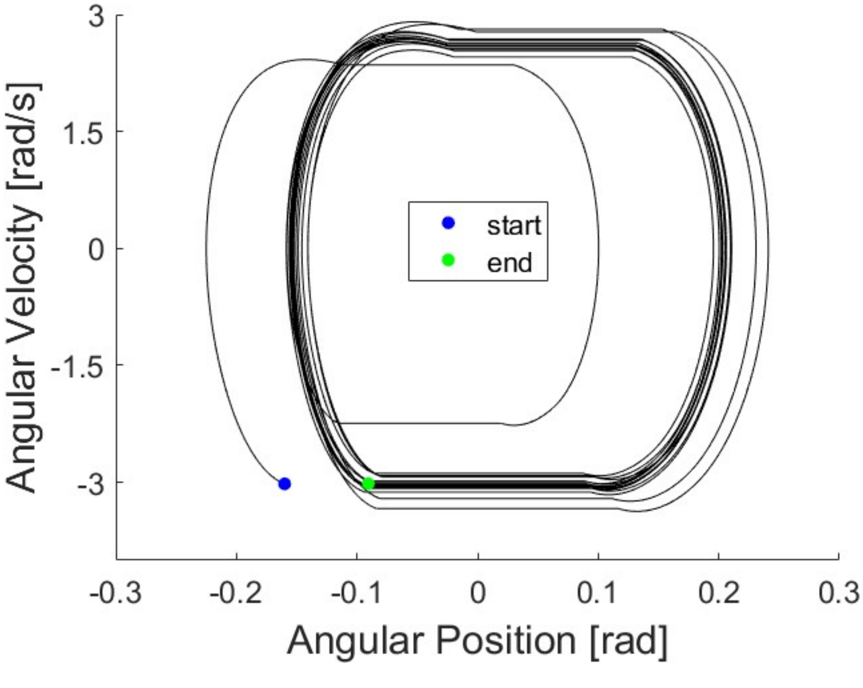

Figure 5.4: Closed-loop hybrid model phase plot in the angular domain. $n = 25$

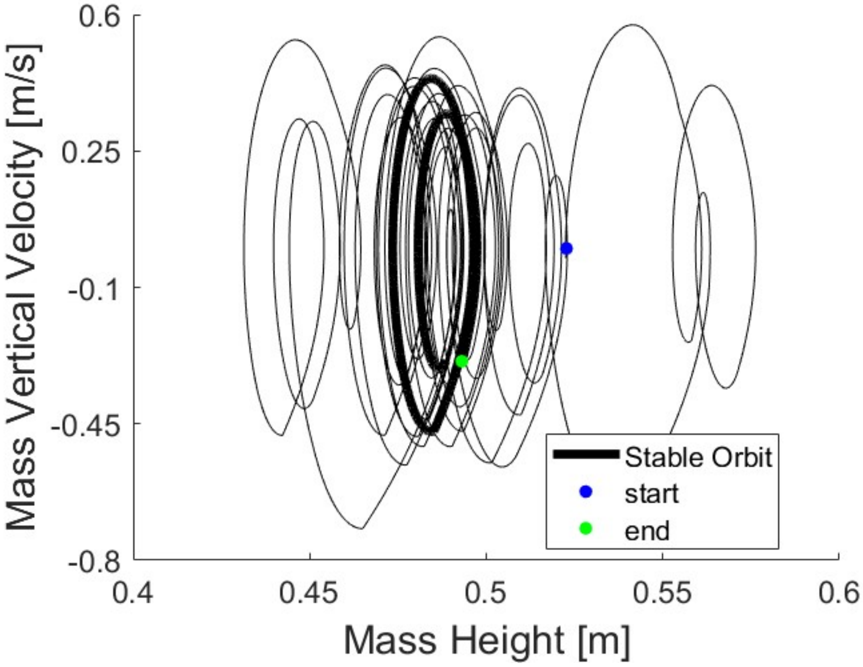

Figure 5.5: Closed-loop hybrid model phase plot in the vertical domain. The black thick orbit highlights where the system finds stability. $n = 25$

Figure 5.4 makes it easy to see that the system stabilizes toward a periodic orbit in the angular domain. However, the response in the vertical domain takes longer to stabilize. Figure 5.6 shows bounding cycles 21-25 where the system has found its stable periodic cycle in the vertical domain. Each distinct loop is a flight phase and a stance phase.

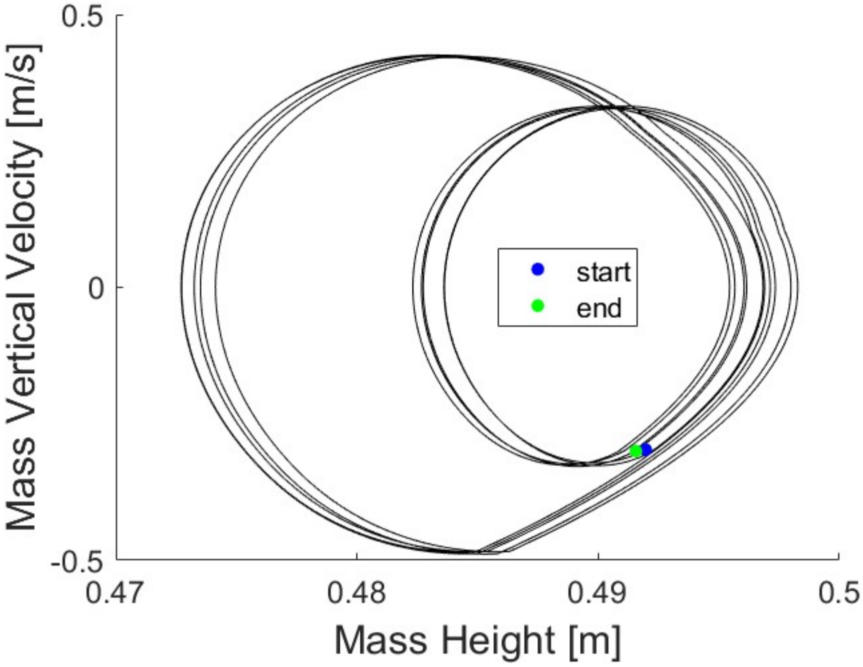

Figure 5.6: A phase plot for the closed-loop hybrid model in the vertical domain. Cycles 1-20 have been removed from the plot. See Figure 5.5 for the transient response. $n = 25$

5.3 Open-loop Solid-Body Model

The open-loop solid-body simulations utilize the full feedforward control for both the Swing Phase and the Stance Phase. Most of the feedback control is neglected except the swing phase PD controller. The PD controller is necessary for any potential bounding motion of the solid-body. The solid-body model also uses the impact detection and controller state switching to go back and forth between the swing phase and the stance phase. Figure 5.7, Figure 5.8, and Figure 5.9 show the system response over 4 bounding cycles. Table 5.3 displays the open-loop solid-body simulation parameters. As noted with Table 5.1, the stride length, swing time, and friction reaction force amplitude are from MIT [4][14]. For general solid-body parameters, see Table 2.2.

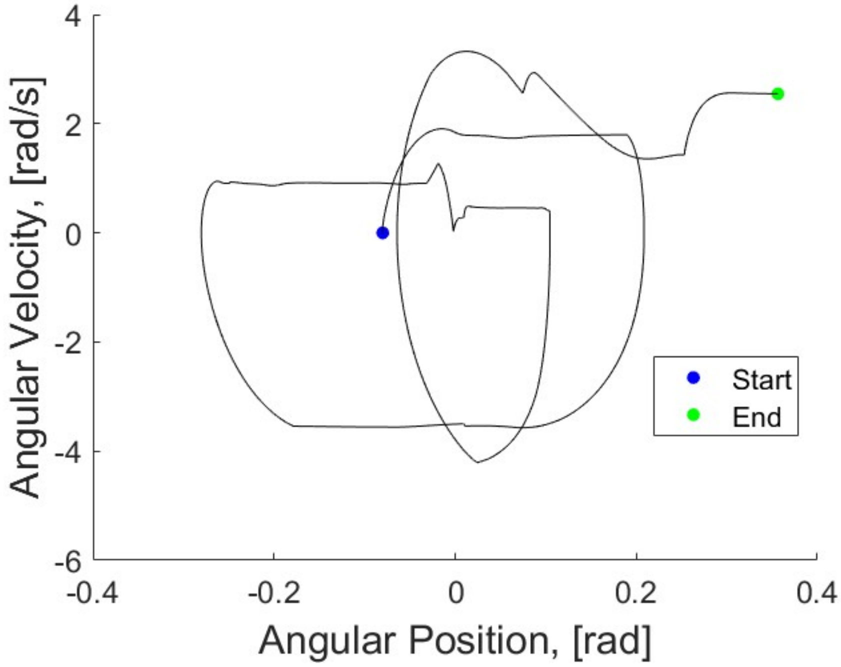

Figure 5.7: Open-loop solid-body phase plot in the angular domain. $n = 4$

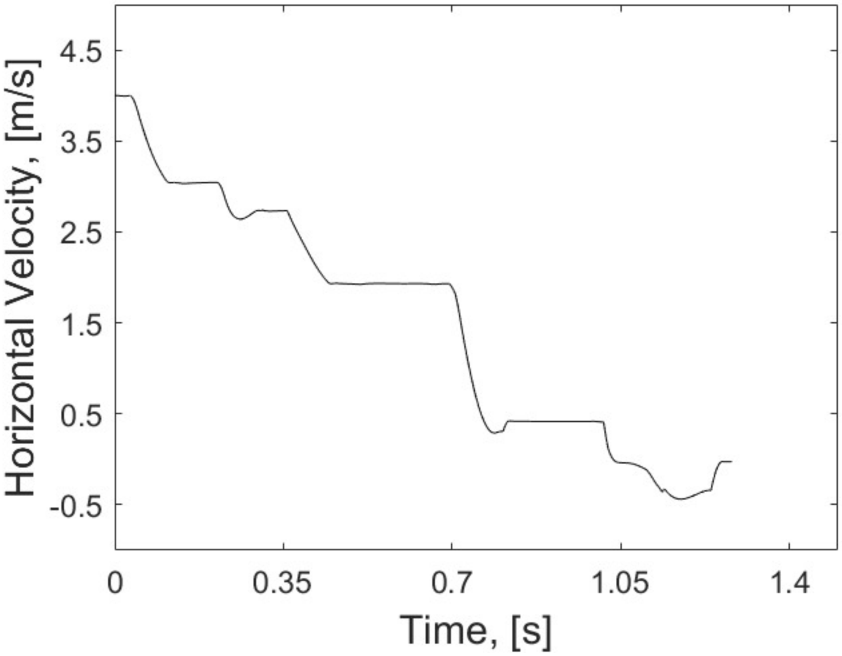

Figure 5.8: Open-loop solid-body horizontal velocity response. $n = 4$

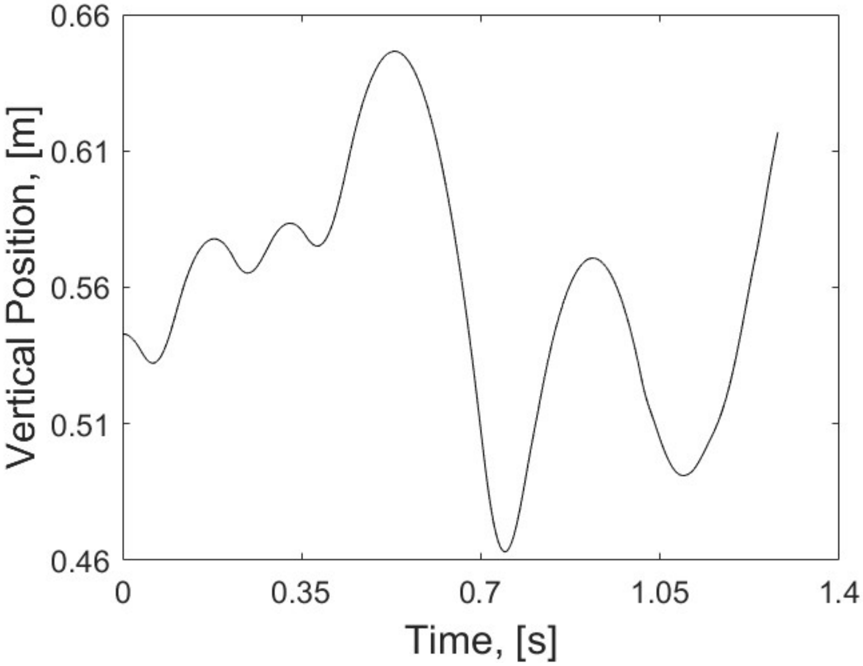

Figure 5.9: Open-loop solid-body vertical position response. $n = 4$

The figures demonstrate how the open-loop solid-body system is immediately unstable and begins loosing all horizontal velocity. Within only 4 bounding cycles the system reaches a horizontal velocity of 0. Despite improvement in the closed-loop simulation results, the horizontal velocity issue is persistent among all solid body controller configurations tested. The nature of the contact modeling detailed in Subsection 2.2.2 seems to be more conducive for resisting overall system motion rather than adding to it. Overall, these results show the need for the full controller design.

Table 5.3: Open-loop solid-body model simulation parameters.

Symbol	Parameter Name	Value	Units
L	Stride Length	0.4	$[m]$
T_{sw}	Swing Time	0.22	$[s]$
θ_{1F}	Front Hip Initial Angular Position	-2.6266	$[rad]$
θ_{2F}	Front Knee Initial Angular Position	1.6646	$[rad]$
θ_{1H}	Hind Hip Initial Angular Position	-1.7259	$[rad]$
θ_{2H}	Hind Knee Initial Angular Position	1.7179	$[rad]$
θ_0	Initial Angular Position	-0.08	$[rad]$
$\dot{\theta}_0$	Initial Angular Velocity	0	$[rad/s]$
y_0	Initial Vertical Position	0.543	$[m]$
$\dot{y}_0$	Initial Vertical Velocity	0	$[m/s]$
x_0	Initial Horizontal Position	-0.5489	$[m]$
$\dot{x}_0$	Initial Horizontal Velocity	4	$[m/s]$
v_{des}	Desired Horizontal Velocity	4	$[m/s]$
A_y	Normal Reaction Force Amplitude	813.6222	$[N]$
A_x	Friction Reaction Force Amplitude	50	$[N]$
k_{Phip}	Hip Proportional Gain	1731.4	$[N*m/rad]$
k_{Dhip}	Hip Derivative Gain	224.26	$[N*m*s/rad]$
k_{Nhip}	Hip Filter Gain	4.32	$[rad/s]$
k_{Pknee}	Knee Proportional Gain	1.15e3	$[N*m/rad]$
k_{Dknee}	Knee Derivative Gain	0	$[N*m*s/rad]$
k_{Nknee}	Knee Filter Gain	0	$[rad/s]$

5.4 Closed-loop Solid-Body Model

The closed-loop solid-body simulations have the entire control design implemented. These simulations were tweaked and tuned to attempt to find the best possible configuration. Figure 5.10 and Figure 5.11 display the results from the best control configuration so far. Since there is an infinite number of gain and parameter values, there certainly exists a more perfect configuration. Table 5.4 lists all of the new or differing parameters from Table 5.3. All of the parameters selected in Table 5.4 except the friction reaction force amplitude and the angular velocity bias are inspired by the work done with the Cheetah 2 [4].

Table 5.4: Closed-loop solid-body model simulation parameters.

Symbol	Parameter Name	Value	Units
A_x	Friction Reaction Force Amplitude	180	$[N]$
k_{Py}	Vertical Proportional Gain	800	$[N/m]$
k_{Dy}	Vertical Damping Gain	120	$[N*s/m]$
k_{Dx}	Horizontal Damping Gain	60	$[N*s/m]$
$k_{P\theta}$	Angular Proportional Gain	25	$[N/rad]$
$k_{D\theta}$	Angular Damping Gain	15	$[N*s/rad]$
κ	Gait Gain	0.3	$[-]$
y_{des}	Vertical Hip Bias	0.48	$[m]$
$\dot{\theta}_{des}$	Angular Velocity Bias	0.25	$[rad/s]$

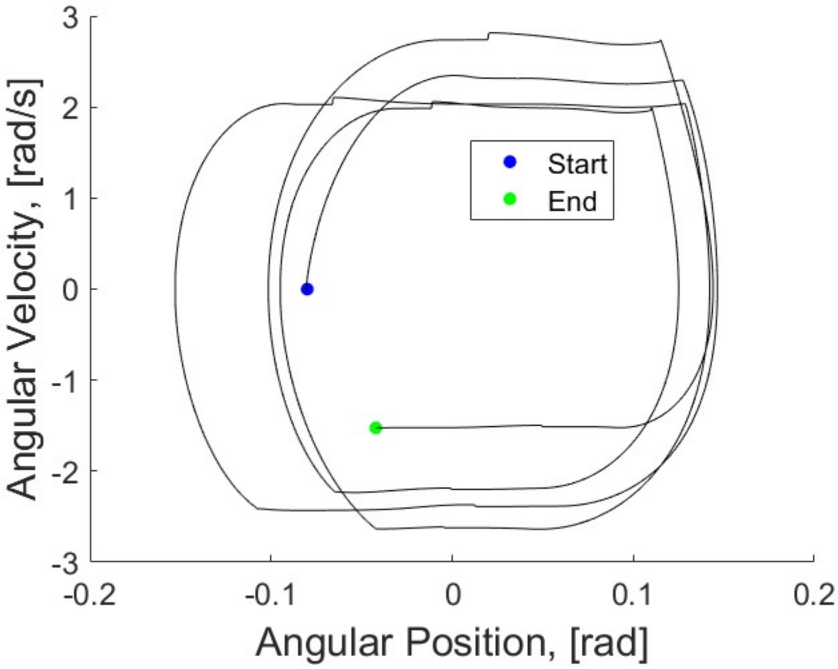

Figure 5.10: A phase plot for the closed-loop solid-body model in the angular domain. $n = 4$

Figure 5.10 suggests stability in the angular domain for the first four bounding cycles. The outlying orbit at the end of the cycle is due to the system approaching a double stance phase. Double stance is where both the front leg and back leg are in contact with ground. This system mode, despite being stable, is not desirable since it does

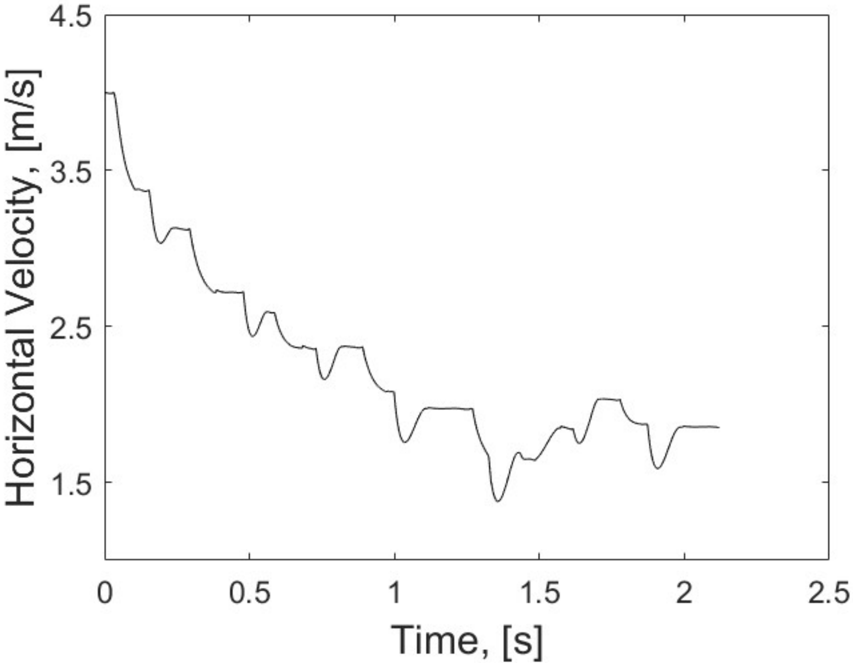

Figure 5.11: Horizontal velocity response of the closed-loop solid-body model. $n = 7$

not fit the characteristics of bounding. Simulation data shows that the system is able to recover from this single double stance and return to an acceptable bounding gait.

The horizontal velocity unfortunately does not stabilize around the desired value but instead approaches a value between 1.5 and 2 $[m/s]$. This is still an improvement upon the open-loop configuration and demonstrates stable bounding despite the decrease in speed. The first reason for improvement, besides the body state feedback control, is that foot to ground speed matching was implemented. Theoretically entirely positive friction profiles should be possible with foot to ground speed matching. Another major reason for improvement is a change to the nominal friction reaction force profile. The nominal profile commanded from the system is three times that of the open-loop configuration. In addition, both front and back profiles are selected as positive. It is important to note that these selections do not accurately represent the friction forces experienced during simulation. Even with these drastic changes to the friction profile the system still sees a predominantly negative friction force during front stance and some negative friction during back stance.

These results show the controller configuration supports stable bounding over 4 bounding cycles. This conclusion is very promising for continued research and work regarding quadrupedal bounding using the techniques detailed in this paper.

Chapter 6

CONCLUSION AND FUTURE STEPS

It is important to continue investigation into quadrupedal control design in order to take full advantage of the beneficial characteristics of legged locomotion. There are countless potential control schemes and configurations that have the potential for optimized performance.

The hybrid modeling done in this paper displays the importance of understanding the fundamental behavior of a system in order to create the frame work for intelligent control design. The solid-body modeling done in MATLAB's Simscape environment demonstrates the ability to create a model that may be more realistic than a control designer's mathematical model. This is of further importance because it provides a way to test and modify a control scheme for implementation. This type of controller design reduces financial and safety risks associated with real world experimentation. Any critical failures during experimentation may put a person's health at risk or damage equipment. Finding these critical failures in simulation provide a much more effective controller design procedure.

The controller demonstrates promising improvements to open-loop behavior and provides robust bounding for 4 consecutive cycles. This validates the fundamental logic behind the control scheme and compels further investigation in tuning or additional control layers. The primary next steps for improving the controller design would be focused on producing more desirable friction reaction force profiles. If this single issue is fixed I believe the controller performance will reach its full potential. It could be useful to analyze system simulation response when stabilizing around lower desired speeds. It is possible that by reducing bounding speed, friction force production may be easier. Another potential way to improve the controller is from the perspective of optimal control. This could be useful for minimizing energy consumption.

Now that the control design has been verified, the system can be tested experimentally. This stage of the research will require a quadruped and a test stand for restricting the system to the sagittal plane. Once successful implementation in the sagittal plane is complete, full quadruped function can be achieved by implementing out of plane stability and transient start up control.

BIBLIOGRAPHY

[1] Ueland, S., “10 Autonomous Robots for Last-Mile Deliveries,” PracticalEcommerce, last modified Jun. 28, 2021, accessed Oct. 9, 2021, `https://www.practicalecommerce.com/10-autonomous-robots-for-last-mile-deliveries`

[2] Zimroz, R., Hutter, M., Mistry, M., Stefaniak, P., Walas, K., and Wodecki, J., 2019, “Why Should Inspection Robots be used in Deep Underground Mines?” *Proceedings of the 27th International Symposium on Mine Planning and Equipment Selection - MPES 2018*, E. Widzyk-Capehart, A. Hekmat, and R. Singhal, eds., Springer International Publishing, Cham, pp. 497–507.

[3] Silva, M. F. and Machado, J. A. T., 2007-09, “A Historical Perspective of Legged Robots,” Journal of Vibration and Control, **13**(9-10), pp. 1447–1486.

[4] Park, H.-W., Wensing, P. M., and Kim, S., 2017, “High-speed Bounding with the MIT Cheetah 2: Control design and experiments,” The International Journal of Robotics Research, **36**(2), pp. 167–192.

[5] Bledt, G., Powell, M. J., Katz, B., Di Carlo, J., Wensing, P. M., and Kim, S., 2018, “MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot,” *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, October 1–5, pp. 2245–2252.

[6] "Spot," BostonDynamics, accessed Oct. 10, 2021, `https://www.bostondynamics.com/spot`

[7] Hodgins, J., 1988, "Legged Robots on Rough Terrain: Experiments in Adjusting Step Length," *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pp. 824–826 vol.2.

[8] Raibert, M. H., 1990, "Trotting, Pacing and Bounding by a Quadruped Robot," Journal of Biomechanics, **23**, pp. 79–98.

[9] Wang, J., Hu, C., and Zhu, Y., 2021, "CPG-Based Hierarchical Locomotion Control for Modular Quadrupedal Robots Using Deep Reinforcement Learning," IEEE Robotics and Automation Letters, **6**(4), pp. 7193–7200.

[10] Kalakrishnan, M., Buchli, J., Pastor, P., Mistry, M., and Schaal, S., 2011, "Learning, Planning, and Control for Quadruped Locomotion Over Challenging Terrain," The International Journal of Robotics Research, **30**(2), pp. 236–258.

[11] Kwan, B., 2021, "Modeling and Control of a Vertical Hopping Robot and Controlling Locomotion of a Robotic Leg," Master's thesis, California Polytechnic State University, 1 Grand San Luis Obispo, CA 93407.

[12] Ward, P., Mitchell, T., Spin, B., and Bush, C., 2021, "Controlling Locomotion of a Robotic Leg," `https://digitalcommons.calpoly.edu/mesp/600`

[13] LeVeque, R. J., 2007, *Finite Difference Methods for Ordinary and Partial Differential Equations : Steady-State and Time-Dependent Problems*, Society for Industrial and Applied Mathematics, Philadelphia, PA.

[14] Park, H.-W., Park, S., and Kim, S., 2015, "Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3D Running of

MIT Cheetah 2," *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5163–5170.

[15] Park, H.-W. and Kim, S., 2015, "Quadrupedal Galloping Control for a Wide Range of Speed via Vertical Impulse Scaling," Bioinspiration & Biomimetics, **10, no. 2**, p. 025003.

APPENDICES

Appendix A

GITHUB REPOSITORY

All necessary files or calculations for recreating the work done in this thesis can be found in the GitHub repository `https://github.com/paward661/Sagittal_Bounding`