Thesis Proposal

LEARNING OPTIMAL POLICIES FOR COMPLIANT GAITS AND THEIR IMPLEMENTATION ON ROBOT HARDWARE

William C. Martin
January 19, 2017

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Hartmut Geyer, Chair
Chris Atkeson
Stelian Coros
Jan Peters, TU Darmstadt

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Copyright © 2017 William C. Martin
Abstract

Bipedal animals exhibit a diverse range of gaits and gait transitions, which can robustly travel over terrains of varying grade, roughness, and compliance. Bipedal robots should be capable of the same. Despite these clear goals, state-of-the-art humanoid robots have not yet demonstrated locomotion behaviors that are as robust or varied as those of humans and animals. Current model-based controllers for bipedal locomotion target individual gaits rather than realizing walking and running behaviors within a single framework. Recently, researchers have proposed using the spring mass model (SMM) as a compliant locomotion paradigm to create a unified controller for multiple gaits. Initial studies have revealed that policies exist for the SMM that exhibit diverse behaviors including walking, running, and transitions between them. However, many of these control laws are designed empirically and do not necessarily maximize robustness. Furthermore, the vast majority of these controllers have not yet been demonstrated on physical hardware, so their utility for real-world machines remains unclear.

This thesis will investigate gait transition policies for the SMM that maximize an objective measure of robustness. We hypothesize that these control policies exist within the SMM framework and can be numerically calculated with guaranteed optimality and convergence. Specifically, we aim to investigate the following two claims. 1) All proposed SMM gait transition policies can be computed using reinforcement learning techniques with linear function approximators. 2) This method can generate new policies which maximize the basin of attraction between walking and running states. Initial results show that these reinforcement learning methods can indeed learn existing SMM policies previously found through Poincaré analysis. If these algorithms are successful in finding globally optimal policies, they may lead to bipedal locomotion controllers with both diverse behaviors and largely improved robustness.

We will experimentally evaluate the utility of these control policies for human-scale bipedal robots. This thesis will extend our analysis of SMM policies on the ATRIAS robot platform to include multiple gaits and gait transitions. Our initial hardware implementation of SMM running has revealed two technical challenges we will address. 1) Modeling errors for both the simplified model and higher-order robot lead to performance degradation from simulation. We will investigate improving this with online methods of parameter estimation and learning. 2) Our experiments have only evaluated planar running and must be extended to include 3D locomotion. If these two challenges are overcome we will have experimentally evaluated SMM running, walking, and transitions between on a physical bipedal robot.
# Contents

1 Introduction .......................................................... 1
   1.1 Barriers to progress ............................................. 1
   1.2 Summary of Goals and Approach .............................. 2
   1.3 Expected Contributions ......................................... 4

2 Background ............................................................ 7
   2.1 The Spring Mass Model .......................................... 7
   2.2 The Promise of Reinforcement Learning ....................... 14
   2.3 Realization on Hardware ....................................... 19

3 Deriving Optimal Gait Transition Policies ..................... 23
   3.1 Completed Work ................................................ 23
   3.2 Proposed Work .................................................. 24

4 Implementation and Demonstration on Robot Hardware ....... 27
   4.1 Completed Work ................................................ 27
   4.2 Proposed Work .................................................. 49

5 Proposed Timeline ..................................................... 51

Bibliography ............................................................ 53
1.1 Barriers to progress

Legged animals are capable of accomplishing an extremely diverse range of dynamic behaviors across a myriad of different environments. Although wheeled vehicles exhibit remarkable agility across continuous flat surfaces, they cannot efficiently traverse naturally rough terrains, such as mountains, or incredibly soft surfaces, such as sand dunes. Furthermore, legged systems are able to turn in exceptionally tight spaces cluttered with obstacles, such as a dense forest. These feats of locomotion are accomplished using various gaits that exhibit distinct dynamic motions yet require only a single biological system.

We expect bipedal robots to be capable of the same level of performance. However, state-of-the-art humanoid robots match neither the robustness nor diversity of human and animal locomotion. Robots that have demonstrated walking or running in the presence of disturbances have not utilized highly varied gaits. Modern model-based control strategies for bipedal locomotion typically target individual walking or running gaits. In order to match the performance of biological systems, bipedal robots must be able to walk, run, and naturally transition between these gaits.

The spring mass model (SMM), commonly referred to as the spring loaded inverted
pendulum (SLIP), offers a potential framework for studying both walking and running using a single simplified model. This compliant locomotion paradigm has been used abundantly over the last 30 years to design robust running and walking controllers. Studies have revealed that the SMM can exhibit an exceedingly wide range of gaits akin to biological systems. Thus, researchers have recently proposed unified controllers for the SMM that can generate walking, running, and gait transition behaviors.

Gait transition policies that have been proposed for the SMM rely on heuristics and empirical design to simplify the problem. As a result, it is unclear to what extent these current controllers maximize robustness or meet other optimality criteria. Additionally, these gait transition policies have not progressed beyond simulation and have not yet been evaluated on physical hardware. The utility of the SMM for transitioning between gaits on real-world machines is currently uncertain.

1.2 Summary of Goals and Approach

This proposal document will detail a research plan including pertinent background information, necessary work, and an approximate schedule for the next 17 months. The goal of this thesis is to study the theoretical limits and practical utility of the spring mass model framework for walking and running. Specifically, we aim to answer the following questions.

1. What is the maximum robustness attainable for gait transitions within the bipedal spring mass model framework?

2. How does the corresponding theoretical performance compare to existing heuristic policies in simulation?

3. Does the simulated controller performance translate usefully to real-world bipedal robot hardware?

By answering these questions we expect to improve the current understanding of gait tran-
sitions within the bipedal spring mass model framework and the extent of their practical utility.

Studying the theoretical limitations of the spring mass model requires us to agree upon a metric for assessing robustness. Ideally, the metric should be easily quantified and readily applicable to any control policy. Example metrics include maximum tolerated control error and maximum tolerated external disturbance. For the purposes of this thesis we will focus on using the size of the basin of attraction shared between walking and running gaits. This can best be explained by the conceptual diagram in figure 1.1. For a given control policy, the left circle represents the set of all states which can be attracted into a stable walking gait, while the right circle represents the set of all states which can be attracted into a stable running gait. Thus, under this metric, robustness is directly proportional to the size of the overlap between the right and left circles. The larger the cardinality of the intersection between stabilizable walking states and stabilizable running states, the more robust we consider the policy.

Despite the simplicity of this robustness metric, the process of finding the policy which
maximizes it is nontrivial. The spring mass model has non-integrable stance dynamics, which has forced researchers studying its behavior to rely on either numerical calculations or analytical approximations. This thesis will use numerical tools to find robust policies using mathematical optimization. In particular, we will utilize reinforcement learning algorithms with guaranteed convergence and optimality within a specified linear basis. Our aim is to determine not only the robustness of optimal spring mass model control, but how this robustness varies with different control parameterizations.

The long term goal of this work is to understand the practical utility of the bipedal spring mass model for physical robots. Optimization of a conceptual model only investigates how control policies will behave within an ideal simulated environment. Real robots are significantly more complex systems whose dynamics cannot be so easily predicted. Thus, an extensive portion of this thesis will focus on experimentally evaluating these simplified model control policies on a bipedal robot. This will entail realizing embedded SMM walking, running, and gait transition behaviors on an ATRIAS biped. Although the focus of this work is on planar control of the spring mass model, we intend to also demonstrate these controls in 3D.

Once complete, this thesis will show both the theoretical and practical levels of robustness that can be expected from a unified spring mass model control for walking and running. The ultimate real-world performance that we measure experimentally will be linked to our specific hardware, but will illustrate the extent of theoretical performance degradation that should be anticipated on a bipedal robot.

1.3 Expected Contributions

The goal of this thesis is twofold. First, we aim to find a theoretically optimal gait transition policy for the bipedal spring mass model using mathematical optimization. Second, we aim to evaluate how this policy translates to real-world legged systems. This process will
address an unanswered scientific question: what is the maximum theoretical robustness and practical utility of the unified spring mass model for walking and running? Once complete, this thesis will have

1. numerically quantified the maximum robustness attainable for gait transitions within the spring mass model framework;

2. demonstrated a unified spring mass model controller for walking, running, and gait transitions on robot hardware.

3. experimentally quantified the amount of performance degradation incurred when transferring theoretical spring mass model controls to hardware.
Chapter 2

Background

2.1 The Spring Mass Model

Over the last 30 years the spring mass model has evolved from a simple predictor of human running mechanics to a unified control framework for a variety of bipedal gaits. Although there exist numerous feasible parameterizations for walking and running, the spring mass model is important because it offers a single low-dimensional descriptor of essential degrees-of-freedom for legged locomotion. Furthermore, control policies have been identified within this framework that are both superstable and capable of blindly traversing frequent and large ground height changes. These features have two important implications. First, the spring mass model is simple enough that it can serve as a locomotion template for high degree-of-freedom robot morphologies without limiting extensibility for other high-level goals. Second, the framework lends itself well to mathematical optimization approaches that scale poorly with the size of state or action space. In this section, we will discuss the history of research on the spring mass model in order to describe the current state-of-the-art and highlight future research that remains to be done.
Early Biomechanical Studies

Compliant elements have a long history of providing simple models for complex mechanical phenomena, such as atomic forces and fluid mechanics. Experimental evidence that the center of mass motion during human running could also be described using elastic mechanisms began to emerge in the 1960s and 1970s. Cavagna and colleagues measured the mechanical work performed during human walking and running for more than a decade using force plates and cameras [16, 17, 19]. These studies observed that the variations in potential and kinetic energy of the center of mass were substantially in phase during running, similar to those of a bouncing rubber ball. This conclusion led to further hypotheses about the mechanisms of elastic energy storage in the human body [18]. Concurrently, these results were extended to include bipedal and quadrupedal animals [20], cementing elastic energy storage as a fundamental concept for legged running mechanics.

Early attempts to produce a mathematical model of biomechanical running focused on using spring laws to represent vertical ground reaction forces during stance. Alexander and colleagues began this work in the 1970s by modeling the running motions of kangaroos [2] and humans [1, 4] with complex cosine equations which could correspond to springs. Around the same time, McMahon and Greene presented a spring-damper model to describe vertical leg behavior during running [48]. McMahon simplified this model several years later by using a single spring mass system to represent the vertical center of mass motions during mammalian running [46, 49]. This “mass-spring” model was shown to be a good predictor of vertical dynamics during both bipedal and quadrupedal gaits. However, none of these early spring-based models attempted to describe horizontal running dynamics.

It wasn’t until the late 1980s that spring mass models were presented as predictors of both horizontal and vertical locomotion dynamics. Although not widely known at the time, van Gurp and colleagues were the first to publish such a spring mass model for describing
the hindlimb motion of a walking horse \cite{29}. This was shortly followed by two independent studies that presented the modern day planar spring mass model as a predictor of human running dynamics. Blickhan \cite{9} investigated which model parameters led to biologically plausible running gaits. McMahon and Cheng \cite{47} focused on how spring stiffness varied with different gait features. These two works introduced the two-dimensional spring mass model for running and inspired a decade of biomechanical studies confirming its predictive utility across a wide spectrum of animals \cite{3,10,22,26,27,28,30}.

Control of The Spring Mass Model for Running

Before discussing running controls that have been developed for the spring mass model, we will more formally introduce the model. The planar spring mass model for running (Figure 2.1a) consists of a point mass $m$ attached to a massless spring leg with stiffness $k$ and rest length $l_0$. During the flight phase, the point mass follows a purely ballistic trajectory. The flight dynamics governing the center of mass motion are

\begin{align}
    m\ddot{x} &= 0, \\
    m\ddot{z} &= -mg,
\end{align}

where $(x, z)$ describes the Cartesian coordinates of the point mass and $g$ is the acceleration due to gravity. The system instantaneously transitions into stance when the foot point touches the ground; the model does not consider any slipping of the foot. Once in stance, the system behaves as an inverted pendulum with an embedded spring. This results in the more complex center of mass stance dynamics

\begin{align}
    m\ddot{x} &= k \left[l_0 \left(x^2 + z^2\right)^{-1/2} - 1\right] x, \\
    m\ddot{z} &= k \left[l_0 \left(x^2 + z^2\right)^{-1/2} - 1\right] z - mg.
\end{align}
(a) The spring mass model for running.  
(b) The spring mass model for walking.

The model exits stance and returns to flight after rebound once the leg has fully extended back to its rest length.

Although the use of the spring mass model as a predictor of locomotion dynamics has its roots in biomechanics, the model has found widespread use in robotics as a template for designing and controlling running robots. Raibert was the first to recognize that springy mechanisms could be used to construct legged robots capable of running and hopping [60]. Throughout the 1980s, his lab produced a number of successful monopedal, bipedal, and even quadrupedal robots using pneumatic spring legs. Raibert’s lab also studied these machines in simulation using both complex spring-embedded rigid body models and the simple planar spring mass model [59]. Despite these simulation studies, all of Raibert’s robots utilized an intuitive control policy based on legged locomotion insights. This control scheme focused on regulating three essential running gait quantities: body orientation, apex height, and forward velocity. Body orientation was handled with a proportional-derivative controller to servo the torso in stance. Apex height was regulated by applying a fixed impulse along the leg during stance. And forward velocity was handled using a heuristic foot placement law for the horizontal foot position

\[ x_{\text{foot}} = x_{\text{neutral}} + k_i (\dot{x} - \dot{x}^*), \]  \hspace{1cm} (2.3)  

where \( k_i \) was an empirically tuned gain, \( \dot{x} \) was the robot’s forward velocity, and \( x_{\text{neutral}} \)
was the neutral foot point that would generate zero net forward acceleration. This last quantity was estimated as half of the previous stance time multiplied by the forward speed, which approximately equals the horizontal midpoint during a symmetric running stance.

The success of Raibert’s legged machines was shortly followed by a number of studies throughout the 1990s that formally investigated the control and stability of spring mass model running. This began with McGeer, who analyzed a three-link rigid body biped model with spring legs and spring joints. He found which mechanical parameters led to passive periodic running and which mechanical parameters could be actively stabilized using hip torque and leg thrust [44]. This analysis involved adjusting model parameters $p$ while numerically calculating the takeoff-to-takeoff return map $R(s, p)$ of the system using a five-dimensional state vector $s$. This technique of analyzing a system’s return map is often referred to as Poincaré analysis and is a common method of studying the stability of periodic spring mass model gaits. By defining a Poincaré section as a single event in the gait cycle, the corresponding return map explains how the dynamical system evolves from cycle to cycle. Fixed points on the return map confirm the presence of stable periodic behavior, while points on the return map whose Jacobian eigenvalue lies within the unit circle confirm local asymptotic stability. Ultimately, the shape of a return map is determined by both static model parameters $p$ and changing control inputs $u$. M’Closkey and Burdick used this technique to evaluate the stability of Raibert’s planar hopper by modeling a spring mass system with a nonlinear air spring along with Raibert’s apex height regulation and foot placement algorithm [45]. Schwind and Koditschek expanded on this analysis by investigating the system with only the foot placement algorithm [66]. They chose the moment of apex during flight as the Poincaré section, which can be fully described with just two state variables: vertical height and forward speed. This study identified a method of increasing the controller’s basin of attraction using a modified leg placement algorithm.
Numerous control policies for the spring mass model have been investigated using Poincaré analysis but those which can be formulated using optimal control are of particular interest. When a target Poincaré section state \( s^* \) is specified given a current section state \( s \), the controller can be formulated as an optimization problem

\[
\mathbf{u} = \arg\min_{\mathbf{u}} \| s^* - \mathbf{R}(s, \mathbf{p}, \mathbf{u}) \| ,
\]

which attempts to find the control input \( \mathbf{u} \) that minimize the distance between the target and current state after one cycle. In the case of the running spring mass model with foot placement as input, control strategies have been identified which bring this distance to zero after one step. This is often referred to as deadbeat control and has been used to create optimal running controllers for the spring mass model \[64\]. Generalizations of this control have led to swing leg retraction policies which remain deadbeat even in the presence of unobserved ground height changes \[67, 68\]. These controllers have also been extended beyond the sagittal plane to control running and turning for the 3-D spring mass model \[14, 15, 85\].

Unification with Walking and Gait Transitions

The spring mass model can also be utilized to describe walking by including a second massless spring leg (Figure 2.1b). While the flight and single stance phase dynamics remain identical to those of the running spring mass model, an additional set of dynamics equations is needed to represent the double stance phase

\[
\begin{bmatrix}
\dot{m}x \\
\dot{m}z
\end{bmatrix} = \begin{bmatrix}
\frac{x-x_L}{l_1} & \frac{x-x_L}{l_2} \\
\frac{z}{l_1} & \frac{z}{l_2}
\end{bmatrix} \begin{bmatrix}
k(l_0 - l_1) \\
k(l_0 - l_2)
\end{bmatrix} - \begin{bmatrix}
0 \\
mg
\end{bmatrix},
\]

(2.5)
where \( l_i \) represents the lengths of the two spring legs and \( x_{fi} \) represents the horizontal position of the two foot points. This extension allows the spring mass model to exhibit both running and walking gaits within a single simple mechanical system.

Despite early demonstrations of walking on Raibert’s bipedal spring mass robots [33], spring mass model walking was not formally investigated until the mid 2000s. Motivated by the inability of stiff leg models to truly predict walking dynamics [31], Geyer and colleagues examined how a bipedal spring mass model could be used instead [32]. They found that this compliant leg system could demonstrate a wide range of gaits including both walking and running. A later experimental study found that this model is capable of accurately describing human walking at moderate speeds, but deviates at slow and fast speeds [42].

Recently, the ability of the spring mass model to generate both walking and running behaviors has sparked interest in how it can also be used to study transitions between these gaits. Salazar and Carbajal were first to investigate if these transitions existed at a fixed energy level using only foot placement controls [63]. They demonstrated that gait transitions could indeed be achieved over the course of several steps, but did not explore how best to design a gait transition controller. This study also revealed that some of these transitions policies required the system to enter grounded running, a gait observed in humans and animals [49, 62]. Shahbazi and colleagues further examined how gait transitions could be achieved when the bipedal spring mass model was also allowed to change stiffness at certain events in the gait cycle [69, 70]. They utilized optimal control across multiple return maps to select spring stiffnesses and foot positions that quickly achieved desired gait transitions. However, several biologically inspired constraints and mathematical approximations were employed to simplify the optimization problem.
2.2 The Promise of Reinforcement Learning

The extensive use of Poincaré return maps to develop optimal control policies for the spring mass model can be attributed to both their ease of use and repeatedly successful application. However, the power of Poincaré techniques is limited by two factors. First, unless a closed form solution to the return map is known, an exhaustive state and action space search is required. As a result, the method extends poorly to systems with high dimensional states or several control inputs. Second, although choosing a Poincaré section collapses one of the state dimensions, it ultimately limits the control policies that can be generated. For example, when midstance is chosen as the point of analysis, the resulting control will never consider actions at different times in the gait cycle. This is acceptable in some cases, such as running, where deadbeat solutions exist for many states, but it can also limit the controller performance when no single step solutions exist.

Ideally, we wish to solve spring mass model control as an optimal control problem using tools that consider all of state space and scale well with problem dimensionality. Reinforcement learning is one tool that addresses how a goal-directed agent can interact with its environment to optimize a numerical reward. In general these methods involve computing a policy based on a continual reward signal in order to maximize the reward in the long run, represented as a value function. These techniques have been successful within the computer graphics community, even with high dimensional state and action spaces [56, 57]. In this section we will address how and why reinforcement learning can be utilized to develop optimal controls for the spring mass model.

Comparison of Approaches

Although reinforcement learning has become popular in recent years, several other approaches can also be used for attempting to find optimal control policies. Due to the long history of reinforcement learning within the field of artificial intelligence, the classifica-
tion of many optimal control methods is not well defined. Thus, a wide range of optimal control methods can be considered as part of reinforcement learning. The common root among these methods is the introduction of the Bellman optimality equation in the 1950s by Richard Bellman. Algorithms suitable for solving control problems using this equation became known as dynamic programming algorithms [6]. Dynamic programming is widely used for solving general optimal control problems, but it suffers from what Bellman called the “curse of dimensionality” because its computational complexity grows exponentially with the number of state dimensions. Nevertheless, dynamic programming has been utilized extensively along with many techniques to avoid this dimensionality issue. Dynamic programming and reinforcement learning solve the same type of problems, which makes it difficult to consider them as separate classes.

Before considering how dynamic programming could be used to solve unified spring mass model control, we will first address the feasibility of a brute force search. The bipedal spring mass model (Figure 2.1b) can be fully defined at any point in the gait cycle using a six dimensional state vector \((x, z, \dot{x}, \dot{z}, x_{f1}, x_{f2})\) where \(x_{f1}, x_{f2}\) describe the horizontal foot positions relative to the center of mass. One additional variable is needed to store the gait phase (flight, single stance, or double stance). There are many action parameterizations that can be used, but one example that has worked well for gait transitions is the three dimensional vector defining the two spring stiffnesses and the next leg placement angle. If each of these state and action vector quantities are coarsely discretized into 10 bins, the system will have \(n = 10^6\) states and \(k = 10^3\) control actions. This corresponds to a total of \(k^n = 10^{3000000}\) possible policies that would need to be evaluated.

Fortunately, dynamic programming allows us to solve this problem exponentially faster than an exhaustive search. Popular methods such as policy iteration or value iteration can iteratively find globally optimal solutions in polynomial time [73]. For instance, the computational complexity of value iteration is \(O(n^2k)\) per iteration. Despite this remarkable
decrease in complexity, conventional dynamic programming methods are still not feasible in many cases due to the need to iterate over every state. One popular extension is to use asynchronous methods which visit the states in any order. This can be noticeably more efficient when the optimal solution remains in a small subset of the state space.

Tabular reinforcement learning methods, such as conventional SARSA or Q-learning, can also be used to solve this problem efficiently using temporal difference learning. These algorithms, similar to dynamic programming, iteratively improve their solutions by using previous estimates to generate new solutions, a feature known as bootstrapping. These techniques do not require a model and can be implemented online in an incremental fashion. Most importantly, these tabular algorithms are still provably convergent to the global optimum [73, 81].

One pertinent disadvantage of the methods discussed so far is that they do not extend well to continuous state and action spaces. Dynamic programming can be improved in this case by randomly sampling actions over a continuous space [5]. Alternatively, approximate dynamic programming methods, such as fitted value iteration, offer compact memory-efficient representations of the value function or policy using function approximators [11]. Function approximators allow value functions to generalize single updates across multiple states, but can introduce a bias and even prevent guarantees of convergence. Nevertheless, function approximators have been widely used to address continuous systems and have demonstrated exceptional performance in practice.

Value function approximation is often accomplished using stochastic gradient descent methods to minimize the mean squared value error (MSVE). When a simple linear function approximator is used this can lead to a global optimum. A typical linear function approximator is $\hat{V}(\theta, s) = \theta^T \phi(s)$, where $\theta$ represents the weight vector and $\phi(s)$ is a feature vector for state $s$. The simple stochastic gradient descent update rule can be written as

$$\theta_{i+1} = \theta_i + \alpha(U_i - \hat{V}(\theta_i, s_i))\nabla\hat{V}(\theta_i, s_i),$$

(2.6)
where $\alpha$ is the learning rate and $U$ is the new function estimate. If $U$ is an unbiased estimate of the true value function and the function approximator is linear, then this algorithm is guaranteed to converge to the global optimum [8]. Unfortunately, bootstrapping methods are biased because they depend on the current weight vector $\theta$. When temporal difference methods are used with a linear function approximator, the gradient descent algorithm instead converges to the TD fixed point, which is within a bound of the optimum [73].

Policy-gradient algorithms also leverage gradient descent in order to update a functional approximation of the policy based on the gradient of a performance metric [58, 74, 84]. One strong benefit of these methods is that they allow continuous policies and action spaces to be utilized. Actor-critic algorithms arise when policy-gradients are combined with explicit approximations of the performance metric. The policy is referred to as the actor, while the performance metric is the critic. This has the benefit of potentially reducing variance of the gradient estimate compared to full Monte Carlo samples.

One last reinforcement learning algorithm that is worth mentioning in the context of legged locomotion is the continuous actor-critic learning automaton (CACLA) [80]. This algorithm has been particularly successful in demonstrating dynamic legged locomotion behaviors in simulation [56, 57]. CACLA operates similar to policy-gradient actor-critic methods, but rather than updating the actor in policy space it updates the actor in action space. In addition, actor updates are only performed when the sampled action improves the performance metric. This prevents the actor from moving towards unsampled actions.

**Reinforcement Learning Applied to Bipedal Robots**

Reinforcement learning has been applied to both simulated and real-world systems for solving a variety of complex dynamic tasks. Learning to walk on bipedal robot hardware has been demonstrated numerous times using conventional reinforcement learning techniques. Although the approach taken in this thesis focuses on offline computation of control policies,
many successful walking controllers have been learned online by leveraging the model-free ability of reinforcement learning algorithms.

Benbrahim was one of the earliest to demonstrate using modern reinforcement learning methods to walk on a physical biped [7]. His doctoral dissertation involved implementing an actor-critic framework using neural networks on a small biped during the mid-1990s. He reduced the dimensionality of the problem using central pattern generators (CPGs). Although the robot he worked on utilized weak electric actuators and was attached to a walker, it demonstrated a slow walking gait after three hours of online learning.

CPGs have been used several times with reinforcement learning methods on biped hardware [24, 51, 54]. Researchers have utilized this approach to demonstrate slow stable walking gaits on small humanoids after significant numbers of trials. Tedrake and colleagues demonstrated walking from actor-critic reinforcement learning on biped hardware without the use of CPGs, but on a small passive dynamic walker outfitted with just two degrees of actuation [77]. Morimoto and Atkeson demonstrated planar walking on a small five-link biped using reinforcement learning to approximate Poincaré maps and value functions [52]. This system was capable of stable walking within 100 trials.

A notable recent example of gait optimization on robot hardware is the work by Calandra and colleagues using Bayesian optimization [12, 13]. Bayesian optimization is an alternative black-box optimizer that is particularly efficient when policy evaluations are expensive, such as with robot hardware trials. This work is pertinent to this thesis because it addresses automatic gait optimization on a small planar biped with spring legs. Bayesian optimization may potentially learn more efficiently than reinforcement learning on hardware, but the successful performance in this work shows promise for gait optimization on springy robots.
2.3 Realization on Hardware

The reality of developing control policies for a virtual model is that it does not directly transfer to hardware. Physical legged robots typically have more degrees of freedom and more dynamics, which can require additional models and layers of control. This section will address how control policies have been realized on humanoid robots in the context of simplified models and gait transitions.

Methods of Transferring Center of Mass Behaviors to Hardware

Heuristic strategies are some of the least complex yet best performing methods for translating a general center of mass behavior to robot hardware. Raibert’s running machines [60] regulate their hopping height, and thus energy, using a fixed leg thrust force during stance coupled with a feedback controller to stabilize body pitch. Similarly, Siavash and colleagues presented a bipedal walking controller on ATRIAS using a spring law with variable rest length for energy regulation and a separate orientation feedback controller [61]. The simple biped control (SIMBICON) framework [87] is another heuristic strategy for walking and running that has been used to demonstrate a variety of gaits on simulated systems. This framework combines decoupled controllers for foot placement, torso regulation, and joint-level position feedback.

Despite the success of heuristic controllers, they do not explicitly compute the motion of the center of mass. Operational space control, also referred to as task space control, uses a full-order model of the system dynamics to find the control inputs which best achieve desired accelerations [37]. This approach allows center of mass motion goals to be defined explicitly using accelerations without having to define individual joint trajectories. Quadratic programming (QP) is a popular and effective way to formulate task space control for high degree-of-freedom humanoid robots. This approach uses numerical optimization to compute joint torques given a set of chosen tasks prioritized for legged locomotion. Feng
and colleagues used this approach to track specific center of mass motions and centroidal momentum on an ATLAS robot [29]. Several other groups have also used QP task space control on ATLAS robots to track quantities such as the instantaneous capture point [38] and desired body coordinate frames [39]. This class of full body controllers has even been utilized for tracking spring mass model foot placement targets on simulated 3D humanoid robots [82].

The optimization involved in full body QP task space control typically has high computational costs, and thus is only applied on the current time step. The disadvantage of this approach is that it neglects future dynamics of the system that could be leveraged. One technique for avoiding this problem is to use a less complex, intermediate model with essential degrees-of-freedom that can be efficiently simulated. For instance, Kuindersma and colleagues precede their full body task space control with a time-varying linear quadratic regulator (LQR) to stabilize a simplified model [39]. Centroidal dynamics models [40, 55] are an attractive choice of simplified model for capturing the rotational dynamics of humanoid robots. This approach can provide greater control over the angular momentum of legged systems.

The Lack of Controlled Gait Transitions for Bipedal Robots

Relatively few bipedal robots have demonstrated controlled gait transitions between walking and running. Hodgins presented the earliest example of controlled gait transitions on a planar two-legged machine [33]. This work adopted an event-based heuristic strategy for switching between walking and running gaits. During the middle of the single stance phase, the leg was either lengthened or shortened to transform between vertical walking and running motions. Running was controlled using the classic Raibert hopping approach, while walking used constant length legs and constant foot placement without velocity feedback.

Honda’s humanoid ASIMO robot platform has demonstrated 3D walking, running, and
.transitions between using a model-based approach \cite{75, 76}. Takenaka and colleagues designed ASIMO’s walking and running gaits as cyclic trajectories for the zero moment point, feet, and upper body using simplified dynamics models. This approach relies on matching specific boundary conditions between cycles to change gait patterns. Scientists have also demonstrated model-based gait transitions on the MABEL \cite{72} and RABBIT \cite{53} planar bipeds using hybrid zero dynamic frameworks. Walk-to-run transitions are accomplished by blending the virtual constraint at the end of the walking gait with the virtual constraint at the beginning of the running gait. Run-to-walk transitions rely on discretely switching to the walking controller modified with a virtual compliant element. These strategies result in gait transitions within one or two steps.

Gait transitions have also been shown in 3D using the ATRIAS biped platform \cite{34}. Researchers at Oregon State University developed a heuristic control strategy that can transition from walking to running when the desired forward velocity is increased beyond 2.0 m/s. This controller provides a smooth, continuous transition between walking and running gaits, but does not explicitly choose whether it walks or runs.

Despite this handful of gait transition examples, there are a considerable number of humanoid robots that only target single gaits. Since gait transitions have been successfully demonstrated on bipeds, it is reasonable to ask why gait transitions controllers are not more common. The simplest explanation is that many biped platforms do not yet have the necessary power density to run. For example, the ATLAS robot that was used during the 2015 DARPA Robotics Challenge weighs 150 kg. In order to generate running motions similar to a human, it would need to create peak vertical ground reaction forces approximately three times its body weight. This would require greater than 4400 N of force to be generated within fractions of a second. Bipedal robots and humans alike must create very large ground reaction forces in order to run.
Chapter 3

Deriving Optimal Gait Transition Policies

3.1 Completed Work

This section will address work completed towards developing optimal gait transition policies. The objective of this research branch is to utilize reinforcement learning as a tool for studying the limits of the spring mass model. Initial work has focused on developing implementations of learning algorithms in order to determine which are most appropriate for this system.

As discussed in chapter 2, the bipedal spring mass model for walking and running can be described with a six-dimensional state vector. There are several possible parameterizations of control input including stiffness of each spring, damping of each spring, and leg touch down angle. We wish to optimize over continuous state and action spaces. Thus, we have investigated different function approximators that may be appropriate for storing spring mass model actors and critics. Deep neural networks have shown remarkable learning ability with continuous systems in recent studies [41, 57], however, their nonlinearity and complexity prevents formal convergence guarantees. On the other hand, linear function
approximators that consist of a weighted summation of features do have global convergence guarantees when used with gradient methods [73].

Even if convergence to a global optimum can be guaranteed using linear function approximators, the resulting optimum is only global in the sense that it is the best linear function of the chosen features. Features must be general enough to not bias the solution. We have taken the coarse coding approach of using state space tiles that are equal to 1 at the center and drop exponentially towards the edge. These features can be computed quickly and at a fine enough resolution to represent complex control policies. In fact, if the tiles are made small enough, such a linear function approximator can exactly represent a policy found with Poincaré analysis.

We have focused on utilizing the actor-critic method known as CACLA to perform fast controller optimization. This algorithm can be improved using ensemble methods to address portions of the state space with overlapping optimal policies. For instance, figure 3.1a shows an optimal leg placement controller for reaching zero vertical velocity at midstance. Notice that the right hand side features overlapping green and blue solutions. Ensemble methods address this problem by using multiple actor-critic pairs with biased action exploration [23, 57, 83]. This allows CACLA to store multiple policies. Initial results in figure 3.1 show that this optimization algorithm is capable of reproducing Poincaré policies.

3.2 Proposed Work

Our ultimate goal is to find optimal policies for the spring mass model which optimize an objective measure of robustness. There are many ways to define robustness in legged locomotion controllers, but one that lends itself well to gait transitions is the basin of attraction between gaits. Given a policy, if we define the set of all states which attract to walking gaits as $W$ and the set of all states which attract to running gaits as $R$, then our
Figure 3.1: (a) Foot placement policy calculated from Poincaré analysis at vertical leg orientation during stance. The colors correspond to different touchdown angles. (b) Foot placement policy learned using ensemble CACLA methods with linear function approximators.

The proposed workflow for this research will alternate between finding gait transition policies that maximize robustness and changing the control parameterization of the spring mass model. The aim is to analyze how gait transition robustness changes as more degrees of freedom are added. The proposed work is summarized by the following questions and hypotheses.

Question 1: Is the spring mass model capable of robust gait transitions?
**Hypothesis 1:** Yes, the size of the basin of attraction shared by walking and running gaits is significant for several control parameterizations.

**Question 2:** Which control parameterization maximizes robustness?

**Hypothesis 2:** Use of landing angle and spring stiffness maximizes robustness with the fewest number of parameters.

**Question 3:** Do optimal gait transition policies significantly improve robustness over existing heuristic policies?

**Hypothesis 3:** Yes, there is a meaningful increase in robustness.
Chapter 4

Implementation and Demonstration on Robot Hardware

4.1 Completed Work

This section will cover realizing deadbeat spring mass model running control on the ATRIAS biped.

The simple spring mass model (SMM) describes a point mass rebounding on massless spring legs. Research on this model has led to deadbeat foot placement strategies that produce highly robust SMM running in the presence of large and frequent, unexpected gait disturbances [14, 64, 68, 85]. This theoretical performance goes far beyond what has been demonstrated in running robots [34, 36, 60, 72]. However, these robots are clearly more complex systems than the conceptual SMM. They possess more degrees of freedom leading to additional dynamics, are limited by actuator saturation, and experience sensory noise that produces uncertainty about the actual state of the system. As a result, the utility of the SMM theories for the control of complex running robots remains largely unclear.

Addressing this gap in understanding, several researchers have investigated the foot placement strategies of the SMM on more simplified hopping robots. For an early example,
Zeglin [88] investigated state space planning algorithms based on the SMM for a bow-legged hopper with a compressible spring and passively stabilized trunk. More recently, Shemer and Degani [71] investigated deadbeat hopping policies for a similar monopod robot with a gyroscopically stabilized trunk in a low gravity environment. They used an analytical approximation of the SMM to compare the effect of constant deadbeat impact angles to swing leg retraction policies. Finally, Uyanık and colleagues [78] quantified the predictive performance of analytical approximations of the SMM in achieving deadbeat behavior using a monopedal spring leg with no attached trunk. All these studies have in common that they were performed with small and specialized one-legged platforms, characterized by prismatic legs, passively stabilized trunk motion in stance, and external sensor measurements. In contrast, we are interested in understanding if the SMM leg placement theories can be transferred to more humanoid robots and attempt the transfer on ATRIAS, a bipedal machine of human scale and weight with an actively controlled trunk and without external sensing (Fig. 4.1).

For the transfer, we focus on rendering the best possible behavior match between the SMM and ATRIAS. To achieve this goal, we use a model-based force control approach during the stance phase of running. Controllers of this type have been implemented successfully on legged robots for tracking desired forces during locomotion [29, 38, 39]. Combined with tracking the deadbeat foot placements of the SMM in flight, ATRIAS should match the behavior and robustness observed in the simplified model. However, we expect deviations from this ideal behavior due to the real world challenges faced by the machine. We perform planar running experiments to quantify these deviations, and thus, the utility of the SMM theories for more complex robots.
Control Approach

The SMM consists of a point mass $m$ rebounding on a massless spring leg of stiffness $k$ and rest length $l_0$. This system behaves as a purely ballistic projectile during flight and as a spring-loaded inverted pendulum during stance with

$$
\begin{align*}
    m\ddot{x} &= k\left[l_0 \left(x^2 + z^2\right)^{-1/2} - 1\right]x, \\
    m\ddot{z} &= k\left[l_0 \left(x^2 + z^2\right)^{-1/2} - 1\right]z - mg,
\end{align*}
$$

(4.1)

where $(x, z)$ are the coordinates of the point mass in the horizontal and vertical dimensions. The model does not consider sliding during stance. Stance occurs when the foot point strikes the ground and flight resumes once the leg length reaches $l_0$ during rebound. The model’s trajectory in flight is fully determined by the horizontal speed $\dot{x}$ and the system energy $E_s$, which is a constant parameter of the model. Given the speed in one flight phase, the model behavior in the ensuing stance and flight phases is controlled by the leg angle $\alpha_{TD}$ at touchdown [68]. This influence of the landing angle on the model behavior can be captured with the apex return map, $\dot{x}_{i+1} = f(\dot{x}_i, \alpha_{TD,i})$, which relates the state of the model between the apexes of two subsequent flight phases ($i$ and $i+1$). Inverting this function yields a deadbeat touchdown angle that takes the system from the current forward velocity $\dot{x}_i$ to a desired forward velocity $\dot{x}_{i+1} = \dot{x}_a^*$ in a single step,

$$
\alpha_{TD,i}^* = f^{-1}(\dot{x}_i, \dot{x}_a^*).
$$

(4.2)

Deadbeat controllers based on this theory have been identified that provide robustness to unobserved rough terrain for the SMM in simulation [13, 68, 85].

Our target platform for translating this theory is CMU’s ATRIAS (Fig. 4.1), one of three identical copies of a human-sized bipedal robot developed by the Dynamic Robotics Laboratory at Oregon State University [35]. The centroidal dynamics of ATRIAS has
inertial properties similar to that of human locomotion. The robot weighs about 64 kg with its mass concentrated in the trunk, 0.19 m above the pelvis. The trunk’s rotational inertia is about 2.2 kg \cdot m^2. Each leg of this bipedal robot is constructed from four lightweight carbon fiber segments. The proximal segments are driven in the sagittal plane by series elastic actuators (SEA) composed of a fiberglass leaf spring and a geared electric DC motor. The reflected inertia of these hip-anchored motors is about 3.75 kg \cdot m^2 after gearing. With a power supply stabilized by a 0.12 F electrolytic capacitor, these motors can draw peak currents of 165 A each, which translates into peak torques of about 600 N \cdot m per actuator at the joint level. In addition, frontal plane abduction and adduction of each hip is provided by a third DC motor mounted on the trunk. Although ATRIAS is capable of untethered 3-D locomotion, this paper focuses on planar control theory of the SMM; hence, the trunk is attached to a boom but is free to pitch in the sagittal plane. The boom constrains the robot to a sphere, which causes small disturbances to the vertical ground reaction forces. However, the mass and inertia of the boom are small and thus do not significantly affect the rotational dynamics of the robot.
Implicit regulation of system energy

Two points complicate the transfer of control theories developed for the SMM onto legged robots such as ATRIAS. The first point is that the system energy is constant in the model but will change in a robot due to the desire to accelerate and brake as well as internal friction. One way of changing energy in the SMM is to introduce another control input, such as a variable leg stiffness during stance [86]. However, we adopt a different approach. We approximate the SMM dynamics (4.1) around the vertical pose \((x, z) = (0, z_0^*)\) with \(z_0^* < l_0\) by

\[
\begin{align*}
m\ddot{x} &= k \left( z_0^* - z \right) \frac{x}{z}, \\
m\ddot{z} &= k \left( z_0^* - z \right) - mg.
\end{align*}
\]

This approximate SMM is similar to the one used in [50]; it has decoupled vertical dynamics, which enables independent control of apex height and horizontal speed achieved during flight, implicitly regulating system energy with more natural gait variables. Specifically, we use (4.4) to prescribe a desired vertical motion \(z^*(t)\) for ATRIAS with apex height \(z_a^*\) and landing and takeoff height \(z_0^*\) (Fig. 4.2). Given this reference, we compute the corresponding return map of the horizontal motion from (4.3). Thus, the updated deadbeat control law for leg placement in flight becomes

\[
a_{TD,i}^* = f^{-1}(\dot{x}_i, \ddot{x}_i^*, z^*(t)),
\]

which regulates running speed on ATRIAS.

Besides implicit regulation of system energy, the approximation of the SMM with (4.3) and (4.4) allows us to easily generalize this model from a point mass to a rigid body, which we address in the next section.
Explicit stabilization of trunk orientation

A second point complicating the transfer of SMM theories onto bipedal robots is that they require stabilization of trunk orientation, which is ignored in the SMM. This is a common problem in humanoid walking control based on the linear inverted pendulum model. It is often solved using a nonlinear quadratic program for a full-order dynamics model of the robot [29, 38, 39]. This optimization balances different goals, such as the center of mass (CoM) behavior, trunk orientation, and other constraints on the robot motion. Due to computational costs, the optimization typically applies to only the current time step without taking advantage of future dynamics. In contrast to this approach, we introduce an intermediate model of reduced order (Fig. 4.3b) that allows us to consider the future dynamics of a floating rigid body with orientation $\theta$, inertia $I$, and dynamics

$$
\begin{align*}
    m\ddot{x} &= F_x, \\
    m\ddot{z} &= F_z - mg, \\
    I\ddot{\theta} &= -zF_x + xF_z,
\end{align*}
$$

(4.6)

using finite-horizon linear quadratic regulation (LQR), as detailed in the stance control section. Here $F_x$ and $F_z$ are the ground reaction forces of the approximate SMM model modified by a stabilizing control for the trunk orientation (detailed in the stance control
section below, equation 4.9). We assume that the centralized inertia $I$ on ATRIAS is constant, rather than configuration-dependent, because the robot’s legs are light relative to its body.

**Overview of control flow**

Given the approximate spring mass model, the intermediate complexity model distributes translational and rotational motion. However, a third layer of model complexity is required to translate this centroidal motion into robot control. Overall this leads to a three layer control structure.

Figure 4.3 summarizes the flow of this control structure for the transfer of SMM control theory onto the ATRIAS biped. At the highest level, we define a spring mass gait based on desired speed and desired apex height. The corresponding approximate SMM provides the
desired CoM trajectory in stance and the desired deadbeat angle in flight (Fig. 4.3a). In stance, the intermediate implementation level then generates GRFs that optimally trade off the desired CoM behavior against the desired trunk orientation (Fig. 4.3b). These GRFs are mapped in the next level by a dynamics model of the ATRIAS robot (detailed in [86]),

\[ M\ddot{q} + h = S\tau + J^TF, \]

(4.7)
to the required joint torques (Fig. 4.3c), which are finally converted into desired motor velocities for the torque control of ATRIAS’ SEAs (Fig. 4.3d).

In flight, the deadbeat angle from the approximate SMM is used to generate a foot point trajectory for the leg that achieves the target angle at a designated touchdown time (Fig. 4.2). This foot trajectory is converted into joint trajectories using a kinematics model of ATRIAS (Fig. 4.3c). The joints are then position-controlled by sending a velocity command to the robot SEAs (Fig. 4.3d).

**Implementation**

The control implementation on the ATRIAS biped requires more consideration. Besides detailing the individual layers of the control flow presented in the last section, this section explains how we address state estimation, external disturbances, and model inaccuracies. All described control is implemented onboard ATRIAS using MATLAB Simulink Realtime software with an update rate of 1 kHz.

**Estimation of CoM and contact states**

Tracking the CoM of ATRIAS and knowledge about its ground contact state are prerequisites for implementing the SMM control. For the first, we use two independent but identically structured Kalman filters estimating the horizontal and vertical CoM states. In both filters, the underlying model is a point mass \( m \) influenced by an applied force \( F \). For
instance, the resulting discrete time process equation of the filter for the horizontal states is

\[
\begin{bmatrix}
    x_{t+1} \\ \\
    \dot{x}_{t+1} \\ \\
    \ddot{x}_{t+1}
\end{bmatrix} =
\begin{bmatrix}
    1 & \Delta t & 0 \\ \\
    0 & 1 & \Delta t \\ \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x_t \\ \\
    \dot{x}_t \\ \\
    \ddot{x}_t
\end{bmatrix} +
\begin{bmatrix}
    0 \\ \\
    0 \\ \\
    \frac{1}{m}
\end{bmatrix}
(\hat{F}^x_t - \hat{F}^x_{t-1}) +
\begin{bmatrix}
    0 \\ \\
    0 \\ \\
    \frac{1}{m}
\end{bmatrix}
w_t,
\]

where \( \Delta t = 1 \text{ ms} \) is the time step and \( w \) is Gaussian white process noise with covariance\( Q = 25 \text{ N}^2 \). The force \( \hat{F}^x \) is estimated from the measured torques of the hip SEAs and the commanded torques of the lateral motors (ATRIAS has no torque sensing for its lateral motors). This is accomplished by solving for \( F \) in equation 4.7 with the constraint \( J\ddot{q} = -J\dot{q} \), which assumes a static point of contact, yielding,

\[ \hat{F} = f(\tau, q, \dot{q}). \tag{4.8} \]

The measurement equation of the filter is

\[
\begin{bmatrix}
    \hat{x}^R_t \\ \\
    \hat{x}^L_t \\ \\
    \hat{x}^A_t
\end{bmatrix} =
\begin{bmatrix}
    1 & 0 & 0 \\ \\
    1 & 0 & 0 \\ \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x_t \\ \\
    \dot{x}_t \\ \\
    \ddot{x}_t
\end{bmatrix} + \nu_t,
\]

where \( \hat{x}^L/R \) are estimates of the horizontal distances from the left and right feet to the CoM, respectively, \( \hat{x}^A \) is an estimate of the horizontal acceleration of the CoM, and \( \nu \) is measurement noise. The distances are computed from the kinematics model of ATRIAS using the robot’s measured joint angles and trunk orientation. The acceleration is calculated using acceleration measurements from an IMU attached to ATRIAS’ trunk. The horizontal filter is initialized using these measurements on every touchdown to account for a changing foot point. The vertical filter is only initialized once on the first touchdown. The covariance
matrix

\[
R_t = \frac{1}{\Delta t} \begin{bmatrix}
R_{mx} - \mu_R(R_{mx} - R_{mn}) & 0 & 0 \\
0 & R_{mx} - \mu_L(R_{mx} - R_{mn}) & 0 \\
0 & 0 & R_A
\end{bmatrix}
\]

of the measurement noise is adaptive. Specifically, the covariance for the distance measurement noise is inversely proportional to the estimated load on each leg in units of body weight, \(\mu_{R/L} = 10 \frac{\hat{F}_z}{mg} (\mu_{R/L} \text{ clamped to } [0, 1])\), \(R_{mn} = 5 \times 10^{-5} \text{ m}^2\), \(R_{mx} = 1 \text{ m}^2\), and \(R_A = 4 \text{ m}^2/\text{s}^4\).

The contact state of each leg is determined from the estimated vertical GRF, \(\hat{F}_z\). ATRIAS has no explicit contact or force sensing at its feet; instead, the force estimate (4.8) is used to determine if a leg is in stance. An \(\hat{F}_z\) exceeding 50\% of body weight triggers the touchdown event and causes the control to enter the stance phase. Conversely, once the vertical CoM velocity \(\hat{\dot{z}}\) crosses from negative to positive values, indicating rebound, a drop in \(\hat{F}_z\) below the 50\% threshold triggers take off and the exit from stance control. This threshold level creates a small delay of approximately 15 ms (about 5\% of stance duration) in contact detection.

### Stance control

Upon transition of ATRIAS into the stance phase, the approximate SMM layer of the control (Fig. 4.3a) generates a desired CoM trajectory \([x^*(t), z^*(t)]\) based on the previous takeoff velocity \(\hat{x}_t\), the next desired takeoff velocity \(\hat{x}^*_{t+1}\), the prescribed vertical motion \(z^*(t)\), and equation (4.3). Note, although \(x^*(t)\) describes the horizontal motion in stance, it is chosen along with the foot placement based on the system state at the previous takeoff. The layer also generates a corresponding force input \(u^*(t) = [F^*_x(t), F^*_z(t)]\) from the GRFs of the approximate SMM with \(F^*_z(t) = k(z_0^* - z^*(t))\) and \(F^*_x(t) = F^*_z(t) \frac{x^*(t)}{z^*(t)}\).
In the second control layer (Fig. 4.3b), the force input is modified to account for trunk stabilization. The desired CoM trajectory is combined with a desired trunk orientation \( \theta^*(t) = 0 \) into a reference motion \( \xi^*(t) = [x^*(t), \dot{x}^*(t), \theta^*(t), \dot{\theta}^*(t), z^*(t), \dot{z}^*(t)] \) for a floating rigid body. We convert the floating rigid body dynamics (equation 4.6) to state space form and linearize the error dynamics around this reference trajectory, which yields,

\[
\Delta \dot{\xi} = F^* \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{F^*_z(t)}{l} & 0 & 0 & -\frac{F^*_y(t)}{l} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \Delta \xi + \begin{bmatrix} 0 & 0 \\ 0 & 1/m \\ -z^*(t) & x^*(t) \\ 0 & 0 \\ 0 & 1/m \end{bmatrix} \Delta u,
\]

where \( \Delta \xi = \xi - \xi^* \) and \( \Delta u = u - u^* \). We approximate \( F^*_x(t) = 0 \) and \( x^*(t) = 0 \), which decouples the vertical state error dynamics. As this approximate model is linear, finite horizon LQR yields the optimal force control input \( u(t) = u^*(t) - K(t)(\xi(t) - \xi^*(t)) \) for trading off tracking the SMM behavior against balancing the trunk, with the feedback gain \( K(t) = [K_x(t), K_z] \). Thus, the second control layer generates the modified desired GRF \( u(t) = [F_x(t), F_z(t)] \) with components

\[
F_x = F^*_x - K_x \begin{bmatrix} x - x^* \\ \dot{x} - \dot{x}^* \\ \theta - \theta^* \\ \dot{\theta} - \dot{\theta}^* \end{bmatrix}, \quad F_z = F^*_z - K_z \begin{bmatrix} z - z^* \\ \dot{z} - \dot{z}^* \end{bmatrix}.
\] (4.9)

The third control layer (Fig. 4.3c) converts the desired GRFs (4.9) into motor commands of the robot’s SEAs in three steps. First, the forces are passed through a safety check to ensure that the foot does not slip on the ground. This requires the vertical force to remain...
positive, \( F_z \geq 0 \), and the horizontal force \( F_x \) to be inside a friction cone with stiction coefficient of 0.5. It is important to note that these two constraints are almost never active on ATRIAS because stance begins when \( F_z \) exceeds 50% of body weight and the stance leg is typically near the vertical. Second, the control compensates for the vertical constraint forces caused by the boom (Fig. 4.1), \( F_z^\dagger = F_z \hat{F}_z/(m \hat{z}) \), where \( \hat{z} \) is provided by the CoM state estimator. Third, the modified desired forces, \( F_x^\dagger \) and \( F_z^\dagger \), are then mapped based on the dynamics model of ATRIAS (4.7) into joint torques using

\[
\zeta = \begin{bmatrix}
M_{5 \times 5} & -S_{5 \times 2} \\
J_{2 \times 5} & 0_{2 \times 2}
\end{bmatrix}^{-1}
\left[
\begin{bmatrix}
h_{5 \times 1} \\
-(J\dot{q})_{2 \times 1}
\end{bmatrix} +
\begin{bmatrix}J^T_{5 \times 2} \\
0_{2 \times 2}
\end{bmatrix}
\begin{bmatrix}
F_x^\dagger \\
F_z^\dagger
\end{bmatrix}
\right]
\]

with \( \zeta = [\ddot{x} \ \ddot{z} \ \dot{\theta}_l \ \dot{\gamma}_l \ \tau_f \ \tau_b]^T \) specifying accelerations as well as joint torques, and \( \dot{I} \) and \( \dot{\gamma}_l \) being the leg length and angle accelerations, respectively (detailed in [86]). Although the solution vector \( \zeta \) contains both accelerations and torques, the accelerations are not used by the controller. Furthermore, because the swing leg is light, we do not account for its accelerations and control it independently as described in the next section. Finally, the resulting joint torques, \( \tau_f \) and \( \tau_b \), are tracked on ATRIAS using the velocity-based SEA controller described in [65].

**Flight control**

Once ATRIAS transitions out of stance, the SMM prescribes only a desired landing angle \( \alpha_{TDf}^* \) (4.5). The model does not specify which of the robot’s two legs is to land or how it shall reach the target. We solve the first problem by introducing a transitory control phase and the second by defining a kinematic trajectory for the foot point.

Figure 4.4 summarizes the state machine of the ATRIAS controller. While one leg follows the SMM stance and flight behaviors, the other leg remains in a mirror control phase. For instance, when the right leg takes off (RTO), it enters this mirror phase and...
Figure 4.4: State-machine of ATRIAS biped control. While the left (L) or right (R) leg cycles through an SMM flight and stance phase, the other leg remains in the transitory mirror phase. Both legs switch roles when take off occurs. TD/TO: touchdown/takeoff.

Figure 4.5: Desired foot point trajectory in the flight control phase. The horizontal trajectory is a quadratic function. The vertical trajectory is composed of a quadratic function, which increases $z_{\text{flight}}$ to encourage leg retraction, and a cosine function to approach the landing condition.

the left leg simultaneously transitions from it into the flight control phase. The right leg remains in the mirror phase while the left leg cycles through an entire step until it takes off (LTO), at which point both legs switch roles.

When a leg is in the mirror control phase, its motion reflects that of the other leg. Raibert [60] introduced this concept of symmetric leg motion, which reduces trunk pitching and prepares the leg for landing. Specifically on ATRIAS, we reflect the foot-to-CoM reference $[x^*(t), z^*(t)]$ from the simplified model across the line $x = 0$. We then negate this reflection to transform it into a CoM-to-foot trajectory,

$$
\mathbf{r}_{\text{mirror}}(t) = [x^*(t), -(z^*(t) - \rho)],
$$

where $\rho = 0.2 \text{ m}$ ensures leg retraction for ground clearance.

When a leg switches into the flight control phase at takeoff time $t_{\text{TO}}$, a new CoM-to-foot
trajectory is engaged,

\[
\mathbf{r}_{\text{flight}}(t) = \begin{bmatrix} x_{\text{flight}}(t) \\ z_{\text{flight}}(t) \end{bmatrix},
\]

where \(x_{\text{flight}}(t)\) and \(z_{\text{flight}}(t)\) are analytic functions that guide the foot to the desired landing condition at a predicted touchdown time \(\hat{t}_{TD}\) (Fig. 4.5). These functions begin at the takeoff mirror position, \(\mathbf{r}_{\text{mirror}}(t_{TO})\), and end at the desired deadbeat landing position, \(\mathbf{r}_{\text{flight}}(\hat{t}_{TD}) = -[\frac{z_{\ast}^\prime}{\tan(\alpha_{TD,i}^\ast)}, z_{0}^\ast]\). The velocity at the expected touchdown time is chosen to match the ground speed, \(\dot{\mathbf{r}}_{\text{flight}}(\hat{t}_{TD}) = [-\dot{x}_{TO}, 0]\), based on the estimated horizontal velocity at takeoff. The predicted touchdown time

\[
\hat{t}_{TD} = \frac{\dot{z}_{TO}}{g} + \frac{1}{g} \sqrt{\frac{\dot{z}_{TO}^2}{2} - 2g(z_{0}^\ast - \hat{z}_{TO})}
\]

is calculated from the expected touchdown state and from the estimated vertical CoM position \(\hat{z}_{TO}\) and velocity \(\dot{z}_{TO}\) at takeoff.

Both the mirror and flight foot trajectories are mapped through the kinematics model of ATRIAS into leg joint trajectories \(\mathbf{q}(t)\) that are tracked with position control (Fig. 4.3c). Here the compliance of the SEAs is ignored by assuming the motor output shafts are rigidly connected to the joints.

**Online adaptation of return map**

The final piece of control implementation is the online adaptation of the deadbeat control (4.5) derived from the approximate SMM. To counter small systematic modeling errors and imperfect torque tracking of ATRIAS, the observed error in the return map behavior of ATRIAS is approximated by a linear model

\[
\dot{x}_{i+1} - \dot{x}_{a}^\ast = \epsilon_1 \dot{x}_{a}^\ast + \epsilon_0,
\]
where $\dot{x}_{i+1}$ is the observed speed in the flight phase $i+1$ and $\epsilon_0$ and $\epsilon_1$ are obtained online through linear regression,

$$
\begin{bmatrix}
\epsilon_1 \\
\epsilon_0
\end{bmatrix} = (X^T X)^{-1} X^T Y,
$$

with $X_i = [\dot{x}_a^* 1]$ and $Y_i = \dot{x}_{i+1} - \dot{x}_a^*$. This error is compensated for by adapting the landing angle. For small deviations, the return map of the approximate SMM generates an error

$$
\dot{x}_{i+1} - \dot{x}_a^* = \partial_\dot{x} f^* (\dot{x}_i - \dot{x}_a^*) + \partial_\alpha f^* (\alpha_{TD,i} - \alpha_{TD,i}^*),
$$

with the partial derivatives pre-computed from the SMM return map. Hence, the observed error (4.10) is compensated for by the adapted landing angle,

$$
\alpha_{TD,i}^* = \alpha_{TD,i}^* - (\epsilon_1 \dot{x}_a^* + \epsilon_0 + \partial_\dot{x} f^* (\dot{x}_i - \dot{x}_a^*)) / \partial_\alpha f^*.
$$

**Hardware Experiments**

To evaluate the planar running control developed in the approach and implementation sections, we perform several experiments on undisturbed and disturbed locomotion using the ATRIAS biped attached to the boom (Fig. 4.1 and supplementary video). In this setup, power is supplied to the robot externally; however, all sensing and computation is performed on-board. Each experiment starts with ATRIAS standing still in a reference pose on one leg. A human operator then holds the boom to stabilize the robot while it follows a reference chirp signal for its CoM height. When takeoff occurs, the actual controller engages and the operator releases the boom. Besides the constant apex height target $z_a^*$ (Fig. 4.2), the input provided in each trial by the operator to the ATRIAS controller is a profile of apex velocity targets $\dot{x}_a^*$ indexed by step number. The first and last velocity
targets are always zero, and each experiment ends once the robot reaches the last step.

**Undisturbed running**

First, we evaluate the performance of the proposed controller in undisturbed running over level ground at a speed of 1 m·s⁻¹. In this gait, about 80% of the available torque of ATRIAS’ SEAs is consumed for generating the desired spring mass rebound behavior (eqs. 4.3 and 4.4) with a stiffness $k =16$ kN·m⁻¹. This stiffness optimally trades off longer stance phases (larger vertical impulses) against the reduced mechanical advantage of ATRIAS’ legs with increasing compression [43]. As a result, it enables the largest hopping heights of about 3 cm with appreciable flight times of about 150 ms (Fig. 4.2). The remaining 20% of torque capacity is available for error compensation. ATRIAS utilizes a large amount of torque to achieve this gait due to the low mechanical advantage in its legs. Other similarly sized robots with different geometries would require different torques, but the ground reaction forces for this spring mass behavior would remain the same.
The tracking performance of the controller is summarized in figure 4.6. At the SMM level, the controller tracks the desired CoM trajectory \([x^*(t), z^*(t)]\) in stance with an error (mean and standard deviation) of 4.5 ± 4.7 cm in \(x\) and 2.6 ± 2.0 cm in \(z\), and tracks the target leg angle at touchdown with an error of 0.99 ± 0.80 ° (Fig. 4.6a). The model deviations originate from two primary sources. The first source is the GRF tracking error due to ground impacts, delayed contact detection, and limited actuator bandwidth. These force errors are reflected in the tracking of the desired SEA torques \((53 ± 68 \text{ N} \cdot \text{m} \text{ error, Fig. 4.6d})\), which is limited by a 20 Hz closed-loop bandwidth of ATRIAS’ SEAs. The second source is the stance feedback control, which creates deviations from the simplified model in order to stabilize the trunk orientation (error of 7.6 ± 6.2 °, Fig. 4.6b) as shown by the deviation in the GRF from the reference GRF of the SMM (error 110 ± 130 N, Fig. 4.6c).

Tracking SMM deadbeat velocity targets

In a second series of experiments we quantify how closely the implemented controller can realize the deadbeat behavior of the theoretical SMM model when the desired velocity \(\dot{x}_a\) changes. We perform two sets of five repeated trials, in which ATRIAS runs over flat ground with desired apex velocities that change every five steps (Fig. 4.7a). In the first set, the change is 0.2 m⋅s\(^{-1}\) with a maximum base velocity of 1.0 m⋅s\(^{-1}\). In the second set, the change and maximum base velocity are 0.4 m⋅s\(^{-1}\) and 1.6 m⋅s\(^{-1}\), respectively. Larger step sizes require deadbeat foot targets beyond the mechanical limits of ATRIAS at high velocities. These limits impose a maximum possible velocity of 2.6 m⋅s\(^{-1}\) for the chosen spring mass gait.

The observed velocity tracking performance is summarized in figure 4.7b. ATRIAS tracks desired velocity changes of 0.2 m⋅s\(^{-1}\) (circles) with the average error observed in undisturbed running (0.05 m⋅s\(^{-1}\), dashed line) after one step, indicating spring-mass-like
Figure 4.7: Tracking of SMM deadbeat velocity targets. (a) Profile of desired apex velocities $\dot{x}^*_a$ for changes of $0.2 \text{ m} \cdot \text{s}^{-1}$ (circles) and $0.4 \text{ m} \cdot \text{s}^{-1}$ (crosses). The first 20 steps are used in each trial for the online adaptation of the return map (Sec. 4.1) and do not count toward the experiments. (b) Root-mean-square error between the target velocity and the robot’s velocity in flight over the number of consecutive steps taken after a change in the velocity target. Averages over all five trials are shown for the entire experiment (black) and separated out based on the different trials (gray). The dashed line indicates the average tracking error in undisturbed locomotion at $1 \text{ m} \cdot \text{s}^{-1}$.

deadbeat behavior within the performance bounds of undisturbed gait. However, the robot requires more steps for tracking $0.4 \text{ m} \cdot \text{s}^{-1}$ changes (crosses), caused mainly by increased ground impacts at the higher horizontal velocities. We measure a peak horizontal impact force of approximately 200 N when running at $1.0 \text{ m} \cdot \text{s}^{-1}$. This impact force increases to nearly 300 N when running at $1.6 \text{ m} \cdot \text{s}^{-1}$.

As described in the undisturbed running section, the deviations from the simplified model are due to force tracking errors and trunk stabilization. For comparison to the hardware results, we quantify these two sources of deviation in simulation. When the sim-
Figure 4.8: Ground disturbance rejection. Shown are the averages over three trials for the root-mean-square error between the desired velocity $v^*_x = 1.0 \, \text{m} \cdot \text{s}^{-1}$ and the velocity achieved by the robot during flight over the number of consecutive steps taken after experiencing an unexpected ground height change $\Delta z$.

plified model is simulated with the same force errors measured on hardware, we observe similar velocity tracking errors of up to $0.15 \, \text{m} \cdot \text{s}^{-1}$ at $1.6 \, \text{m} \cdot \text{s}^{-1}$. When the intermediate complexity model (Fig. 4.3b) is simulated with an initial orientation error of $10^\circ$, we observe a velocity error of $0.05 \, \text{m} \cdot \text{s}^{-1}$ after one step. The system completely recovers after three steps. Thus, errors in force tracking and trunk orientation lead to a substantial performance deterioration compared to the SMM deadbeat control theory. This suggests that performance could be improved by extending the simplified model to account for ground impacts and rotational dynamics of the center body in legged locomotion.

**Rejecting unexpected ground changes**

With the third series of experiments, we explore how closely the implemented controller follows the deadbeat behavior of the SMM when the robot encounters unexpected changes
in ground height. We perform experiments for six different ground height changes of \( \pm 6 \text{ cm} \), \( \pm 11 \text{ cm} \) and \( \pm 15 \text{ cm} \), each repeated for three trials. In all trials, ATRIAS encounters the ground disturbance while running at \( 1.0 \text{ m} \cdot \text{s}^{-1} \) with its reference gait.

Figure 4.8 shows the velocity tracking performance of ATRIAS after encountering a ground height change measured as the error in velocity over the steps taken. Deadbeat behavior would result in an error no larger than the average error of \( 0.05 \text{ m} \cdot \text{s}^{-1} \) observed in undisturbed running at \( 1.0 \text{ m} \cdot \text{s}^{-1} \) from the first step on. However, each of the ground height changes results in a substantial velocity error in the first step of about the same size (\( 0.2 \text{ m} \cdot \text{s}^{-1} \) to \( 0.4 \text{ m} \cdot \text{s}^{-1} \)), which only gradually diminishes in the next steps.

The velocity error and its gradual decay are largely independent of the direction and size of the ground height change, which seems counterintuitive. For instance, a height drop of 15 cm results in an increase in speed to \( 2 \text{ m} \cdot \text{s}^{-1} \) if maintaining the same total system energy. Similarly, a height increase of the same amount cannot be achieved without increasing system energy, even with zero speed. Comparing the two cases, it seems they should lead to very different behaviors, and thus velocity errors, after the disturbance.

The reason why the errors behave similarly is because they are due to the increased ground impacts and trunk orientation errors that are common to all of the height changes. The sudden ground height changes are implemented as sheer jumps in the floor surface using concrete blocks (Fig. 4.1). This leads to increased impact forces of nearly 500 N and swing foot impacts with the side of the elevated ground. Thus, most of the observed performance degradation compared to the SMM deadbeat control theory is again due to the increased ground impacts and required trunk stabilization. The detrimental effect of swing leg impacts suggests that hardware implementations should focus on more compliant swing leg motions than stiff kinematic control provides.
Discussion

We investigated if the SMM leg placement theory can be transferred to running robots beyond the simplified one-legged test platforms used in previous studies. Specifically, we have evaluated the utility of spring mass theory on a robot of human scale and weight with an actively controlled trunk, articulated legs, and without external sensing. To this end, we focused on the ATRIAS biped platform and implemented on it a controller that transfers the SMM behavior through model-based force control in stance and kinematic control of foot placement in flight. We found that the proposed controller achieves on ATRIAS SMM-like deadbeat performance for velocity changes up to $\pm 0.2 \text{ m} \cdot \text{s}^{-1}$. For larger velocity changes and for ground height changes ranging from $\pm 6 \text{ cm}$ to $\pm 15 \text{ cm}$, the controller performance degraded, albeit without compromising gait robustness. The degradation was in large part due to ground impacts and the incessant need to stabilize the robot’s trunk, neither of which are considered in the SMM theory. The results highlight the limited utility of this theory for the control of more complex running machines; on the other hand, they also point to the potential of such an SMM-based control for generating robust and versatile behavior in running robots.

The achieved performance mirrors the performance observed for the implementation of SMM-based deadbeat control strategies on the much simpler robot platforms. The velocity tracking error of 5% on ATRIAS during undisturbed running is in line with the results obtained by Zeglin, who demonstrated deadbeat hopping with a mean velocity error of approximately 15% between steps [88]. The ability of the proposed controller to tolerate unobserved rough terrain of at least 17% of the nominal leg length (0.9 m) is similar to the performance described by Shemer and Degani [71], who demonstrated deadbeat hopping over terrain height changes of about 15% of leg length. In contrast to the previous results, however, the demonstration of these capabilities on ATRIAS with similar performance shows that they generalize to more complex and human-like bipedal robots.
One key advantage of the model-based control framework [29, 38, 39] also pursued in this work is that it is easier to generalize behavior beyond scripted motions. For example, the MABEL robot is capable of planar running using a control framework based on hybrid zero dynamics. However, as the robot encounters perturbations, its controller must adapt speed to maintain stability leading to “considerable variation” in forward velocity [72]. Similarly, Hubicki and colleagues [34] discovered that ATRIAS is capable of 3-D running (although with very short flight phases of about 30 ms) when a heuristic controller designed for walking was commanded higher desired velocities. In contrast, our proposed controller can (within the bounds provided by the torque capacity of the actuators) freely choose the speed at which it runs from step to step, whether on flat ground or after a disturbance, by taking advantage of the underlying gait model and its deadbeat foot placement strategies.

Several research directions will help to further the model-based control framework for running robots. First, the SMM theory remains to be evaluated on robots running in 3-D environments. Second, performance degradation due to force errors and trunk stabilization suggests that the utility of the SMM theory could be increased by extending it to account for ground impacts and the rotational dynamics of a trunk. These force errors could also be mitigated by designing a more compliant swing leg control. Third, the mechanical limits of real robots prevent reaching certain target states in a single step. Robustness could potentially be improved in this case by considering these actuation limits [21, 25]. Finally, the transfer of SMM-based control to walking robots could substantially enlarge the range of robust behaviors that can be addressed. It is our goal to pursue these research directions in order to demonstrate highly robust 3-D running and walking on ATRIAS over uncertain terrain.
4.2 Proposed Work

We propose a number of hardware experiments in order to investigate the practical utility of optimal spring mass model policies for human-scale bipedal robots.

Before unified gait transition controllers can be demonstrated on ATRIAS, we must implement spring mass model walking control on the system. We anticipate following one of two paths depending on their success. The first potential controller we intend to attempt is a walking extension of our existing deadbeat running control developed with Poincaré analysis. This is the preferred method because it is straightforward and has been successful for running control. However, the optimization framework developed for gait transitions could also be utilized to generate higher-dimensional walking control policies. In either case, we aim to embed bipedal spring mass model dynamics on ATRIAS, using a method similar to the running controller described above.

Once we have successfully demonstrated both walking and running on the system, our focus will shift to demonstrating optimal gait transitions on ATRIAS. The challenges of translating these controls to hardware will revolve around addressing model inaccuracies and control errors. Our deadbeat running implementation benefited from using online regression to improve the foot placement policy. Although an online regression scheme may also benefit our gait transition policies, the reinforcement learning framework naturally extends to online model-free learning. As a result, we will address these challenges initially using the developed optimization scheme on ATRIAS.

Our final goal for the hardware will be to demonstrate our planar spring mass model controllers on ATRIAS in three dimensions. This is an important step that is required to evaluate the practical utility of this control framework. Real-world humanoid robots need to operate beyond two dimensions, and modern deadbeat spring mass model controllers have not yet been demonstrated on a 3-D biped.
The proposed work is summarized by the questions and hypotheses below.

**Question 1**: Can embedded spring mass model control generate gait transitions on robot hardware?

**Hypothesis 1**: Yes, gait transitions can be accomplished on a physical robot using the spring mass model as a control template.

**Question 2**: What is the practical utility of using the spring mass model as a unified control framework in the real-world?

**Hypothesis 2**: The resulting controller will generate behaviors that are both robust to disturbances and able to create diverse robot motions.

**Question 3**: Can the spring mass model generate robust walking and running on a 3D biped robot?

**Hypothesis 3**: Yes, the planar spring mass model behaviors can be used for useful and robust 3D locomotion.
Chapter 5

Proposed Timeline

This section will present a proposed timeline for completing the work described in this document. A descriptive high-level summary of the involved goals and approaches can be found in the introduction under section 1.2.

We seek to evaluate the theoretical limitations and practical utility of using the spring mass model as a unified control framework for walking, running, and gait transitions. We have proposed specific optimization tools for performing this analysis and detailed what hardware demonstrations must be accomplished to evaluate utility. The following tentative schedule summarizes important milestones and an approximate time table for completing them between January 2017 and June 2018.
Task: Optimal Gait Transition Policies

(January to July 2017)

- January to March - Algorithm refinement
  This will focus on implementation and testing of reinforcement learning algorithms to determine which variation is most appropriate for this work.

- March to June - Optimal policy evaluation
  Here we will evaluate the theoretical robustness of optimal gait transition policies for different spring mass model parameterizations.

Task: Robot Hardware Experiments

(July 2017 to March 2018)

- June to September - SMM walking on robot
  This task will involve developing and demonstrating planar SMM walking policies on ATRIAS.

- September to December - SMM gait transitions on robot
  Once SMM walking is complete, this task will demonstrate and evaluate the optimal planar SMM gait transition policies on ATRIAS.

- December to March - SMM on 3D robot
  The final component will experimentally validate these planar SMM policies in 3D on ATRIAS.

Final Task: Write Thesis Document

(March to June 2018)
Bibliography


[26] Claire T Farley and Daniel P Ferris. 10 biomechanics of walking and running: Center


[41] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez,


[51] Takeshi Mori, Yutaka Nakamura, Masa-Aki Sato, and Shin Ishii. Reinforcement learn-


[57] Xue Bin Peng, Glen Berseth, and Michiel van de Panne. Terrain-adaptive locomotion skills using deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 35(4):81, 2016. 2.2 2.2 3.1


[61] Siavash Rezazadeh, Christian Hubicki, Mikhail Jones, Andrew Peekema, Johnathan Van Why, Andy Abate, and Jonathan Hurst. Spring-mass walking with atrias


[69] M Shahbazi, GAD Lopes, and Robert Babuska. Automated transitions between walk-


[86] Albert Wu and Hartmut Geyer. Highly robust running of articulated bipeds in un-
