IDENTIFICATION AND STABILITY ANALYSIS OF PERIODIC MOTIONS FOR A PLANAR LEGGED RUNNER WITH A RIGID BODY AND A COMPLIANT LEG

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Güneş Bayır

August, 2013

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Ömer Morgül(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Uluç Saranlı(Co-advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Sinan Gezici

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Melih Çakmakçı

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

IDENTIFICATION AND STABILITY ANALYSIS OF PERIODIC MOTIONS FOR A PLANAR LEGGED RUNNER WITH A RIGID BODY AND A COMPLIANT LEG

Güneş Bayır
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. Ömer Morgül
August, 2013

The Spring-Loaded Inverted Pendulum (SLIP) model is an extensively used and fundamental template for modeling human and animal locomotion. Despite its wide use, the SLIP is a very simple model and considering the effects of body dynamics only as a point mass. Although the assumption of a point mass for the upper body simplifies system dynamics, it prevents us from performing detailed analysis for more realistic robot platforms with upper trunks. Hence, we consider an extension to the classic SLIP model to include the upper body dynamics in order to better understand human and animal locomotion.

Due to its coupled rotational dynamics, extending the SLIP model to the Body-Attached Spring-Loaded Inverted Pendulum (BA-SLIP) brings additional difficulties in the analysis process, making it more difficult to obtain analytical solutions. Consequently, simulations have been used to reveal the periodic structure behind locomotion with this model, and to find fixed points of discretized system dynamics. These fixed points correspond to periodic motions of the system and are important in designing controllers since they are used as steady-state control targets for most applications. The main concern of this thesis is to find fixed points of the BA-SLIP model and to investigate the dimension of the fixed point manifold.

We performed extensive simulation studies to find fixed points of the system and the properties of the underlying space with a PD controller. Our simulations revealed the existence of periodic gaits, in which the upper body should be downward oriented for stable locomotion. Additionally, a region of stability is found such that the model sustains periodic gaits when it stays inside this region.

Finally, we show that fixed points for running with upright body orientation are unstable when system dynamics are regulated with a constant parameter controller. We also present some simulation results which indicate the existence of stable periodic motions when controllers with time varying parameters, that use current state information, are used.

Keywords: Spring-Mass Hopper, Spring-Loaded Inverted Pendulum (SLIP), Legged Locomotion, Fixed Point, PD Control.

ÖZET

ESNEK BACAKLI VE GÖVDE EKLENMİŞ DÜZLEMSEL BİR BAÇAKLI ROBOTUN PERİYODİK HAREKETLERİNİN BELİRLENMESİ VE KARARLILIK ANALİZİ

Güneş Bayır
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Ömer Morgül
Ağustos, 2013

Yaylı Ters Sarkaç (YTS) insan ve hayvan hareketini modellemek için tasarlanmış, geniş kullanım alanı olan, temel bir şablondur. Popülerliğine rağmen, vücut dinamiklerini temsil etmek için YTS basit kalmaktadır çünkü üst gövde sadece bir noktasal kütle olarak gösterilmiştir. Noktasal kütle varsayımı sistem dinamiklerini basitleştirse de, üst gövde dinamiklerinin de dahil olduğu daha gerçekçi robotik platformlarını detaylı bir analiz imkanından yoksun bırakmaktadır. Bu nedenle, insan ve hayvan hareketini daha iyi anlamak için, klasik YTS modelinin bir üst gövde ile genişletilmiş halini temel alıp çalışmalarımızı yürüttük.

Birleşik dairesel dinamikler yüzünden, YTS modelini Gövde-Eklenmiş Yaylı Ters Sarkaç (GE-YTS) modeline genişletmek analizine bazı zorluklar getirmiş olup, analitik çözümler elde etmeyi zorlaştırmıştır. Bu yüzden, modelin periyodik hareketlerini açığa çıkarmak için benzetim tabanlı analizler kullanılmıştır; bu analizler sistem dinamiklerinin sabit noktalarını bulmak için faydalı olacaktır. Sabit noktaların sistemin periyodik hareketlerine karşılık geldiklerini belirtmekte fayda vardır, bu sabit noktalar denetleyici tasarımında büyük önem taşımaktadırlar çünkü birçok uygulamada yatışkın durum denetleyici hedefleri olarak kullanılabilirler. Bu tezin ana hedefi GE-YTS modelinin sabit noktalarını bulmak ve bu noktaların içinde bulunduğu uzayı ve boyutunu incelemektir.

Sistemin sabit noktalarını bulmak ve bu noktaların içinde bulunduğu uzayın özelliklerini incelemek için Orantılı Türevli (OT) denetleyici kullanılarak, kapsamlı benzetim çalışmaları yapılmıştır. Sonuç ta, bu çalışmalarımız bazı periyodik adımların varlığını açığa çıkarmış olup, denetleyici parametrelerinde durum bilgisi kullanılmadığı zaman, bu adımlarda kararlılık için gövde açısının aşağıya

doğru olması gerektiği görülmüştür. Ek olarak, modelin periyodik adım atmaya devam etmesi için bir kararlılık bölgesi bulunmuştur. Son olarak, gövde açısının yukarıya doğru olduğu durumda koşma için bulunan sabit noktaların açık-döngü parametre kullanan denetleyicide kararsız olduğu görülmüştür. Ayrıca, bazı benzetim sonuçları kapalı döngü parametre kullanan, yani durum bilgisini kullanıp adım başı parametre düzeltmesi yapan, denetleyiciler varlığında vücut yukarı doğru meyilli iken periyodik hareketlerin varlığını göstermektedir.

Anahtar sözcükler: Yay-Kütle Zıplayanı,Yaylı Ters Sarkaç (YTS), Bacaklı Hareket, Sabit Nokta, Açık-Döngü Parametreli Kontrol.

Acknowledgement

I would firstly like to thank my supervisors, Ömer Morgül and Uluç Saranlı, for their guidance, encouragement and patience throughout my graduate experience. Their profound knowledge about theory and applications on my interest area directed me towards success.

The initial ideas of this thesis was proposed to me by Uluç Saranlı, and I'm very happy to gain his trust on me in this complicated and difficult problem. His stimulating discussions and vision always directed me to better solutions.

Additionally, I thank my close friends from department office, İsmail Uyanık especially for his conceptual contributions to my thesis and Ali Nail İnal, for their patience and guidance throughout my most stressful times. Moreover, I would like to thank my other close friends from same department Ali Alp Akyol, Fatih Emre Şimşek, Taha Ufuk Taşçı, Veli Tayfun Kılıç for always being nearby me to listen and motivate.

I also appreciate the financial support from Scientific and Technical Research Council of Turkey (TÜBİTAK).

Finally, I owe my thanks to my parents, Vahide and Arslan Bayır for their love and support.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

"Life begins with motion" - that is a famous motto of all times. Although there are many distinguishing features of living creatures with respect to objects surrounding them, motion is the most concrete and observable characteristic of animals and plants. In particular, locomotion talent of humans and animals make them the rulers of their environment since they have the capability of changing the world around them by locomotion and the use of their intelligence.

There are many reasons for humans and animals to locomote, such as finding food and a good habitat to live, or escape from predators in nature. Animals and humans do this task in very different forms such as walking, swimming and flying. The interesting notion here is that it does not matter how small intelligence an animal may have, it can perform extreme locomotion tasks with relative ease.

With the beginning of new technology age, humans tried to build robotic systems that can locomote like animals in nature to create extra labor. This concept is known as bio-inspired robotics. Recent advances in this field show much progress in legged and limbless locomotive, climber and jumper robots to make human life easier. These robots can be used in various fields, such as rough terrain surveillance, military intelligence, space research, etc.

One of the most important applications in bio-inspired robotics is locomotion

on rough terrain in a stable and robust manner because most applications require earth surface missions for robots. Although there are different approaches for land locomotion, such as wheeled or tracked systems, the utility of legged morphologies for robust and efficient locomotion have been observed [1, 2] if we want to build robots that can negotiate rough terrain as animals do.

In this thesis, we study periodic motions of and perform stability analysis for the locomotion patterns of a Body-Attached Spring-Loaded Inverted Pendulum (BA-SLIP). Such an analysis would be quite useful in implementing control algorithms for robots that have an upper trunk to be stabilized during locomotion. Periodic motions in such a system correspond to fixed points of an associated Poincaré map, namely the apex-return map, associated with locomotion dynamics. In contrast to existing studies in literature, we perform detailed analysis to find fixed points of this return map and investigate the dimension of the fixed point manifold.

1.1 Motivation and Background

Locomotion is achieved via complex, high-dimensional, nonlinear, dynamically coupled interactions between an organism and its environment [3]. In order to understand these complex dynamics, we use reduced-order mathematical models. From the perspective of biomechanics research, the Spring-Loaded Inverted Pendulum (SLIP) model is a successful descriptive tool for running animals [4].

The SLIP model, a point mass attached to a massless leg endowed with a linear spring, was established as a simple and accurate descriptive tool to analyze the dynamics of animal locomotion for different sizes and morphologies [5, 6, 7]. This idea paved the way towards building successful robot platforms such as Raibert's hoppers [8], the ARL-Monopods [9], the Bow-Leg design [10] and the BiMasc [11].

One of the main problems with the standard SLIP model is the representation of the body by just a single point mass [12]. As a result, one can neglect the

problem of trunk stabilization. However, this is a major issue for bipedal robotic platforms and cannot be ignored if we want to mimic animal locomotion.

Upright walking has significant advantages as observed in different scientific disciplines [13, 14, 15, 16, 17]. On the other hand, the addition of a trunk instead of a point mass requires more complex models and yields similarly complex control strategies. Although mathematical models have been proposed for the SLIP model with a trunk [18], they mostly focus on Virtual Pivot Point (VPP) concept for analysis and controller design.

Motivated from these studies in the literature, we start by introducing a lossless SLIP model with a trunk, called the Body-Attached Spring-Loaded Inverted Pendulum (BA-SLIP) model, for a monopedal robot platform as illustrated in Fig. 1.1. In contrast to existing studies on the analysis and control of SLIP model with a trunk [18, 19, 20], our goal is to focus on the identification and analysis of periodic solutions to the system and to investigate the dimension of the fixed point manifold for our BA-SLIP model.

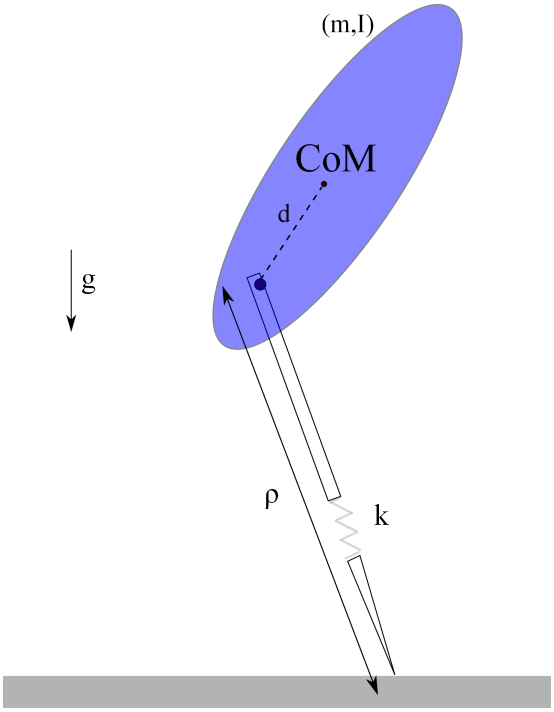

Figure 1.1: The Body-Attached Spring-Loaded Inverted Pendulum (BA-SLIP) model.

In Fig. 1.1, a trunk with mass m and inertia I is connected to a massless spring leg with stiffness k and length ρ, through a pivot point around which the trunk can freely rotate, at a distance d from the center of mass (CoM) of the trunk. This model consists of two important dynamical components: Linear and rotational. The effect between the CoM and the hip joint couples these two dynamics, resulting in more complex behavior than the classical SLIP model [21].

Our goal is to identify periodic motion patterns admitted by our model for sagittal plane locomotion. Since periodic motions correspond to fixed points of Poincaré maps associated with locomotion dynamics, in particular the apex-return map, we try to identify fixed points of this map to reveal periodic motions. These fixed points are mostly used as steady-state control targets for locomotion. Additionally, in the case of nonlinear systems, such as our model, we start the analysis by first deriving dynamical equations and examining the system around limit cycles, so called fixed points. These fixed points depend on system parameters, such as gravity, mass and inertia of the upper trunk, leg stiffness and distance between leg-body joint and center of mass. The resulting system dynamics are highly non-linear which complicates further analysis. Consequently, to decrease the dimension of the optimization problem, these parameters are held constant throughout the thesis. Finally, we seek to find a dimension for the fixed point manifold, so that we can build an intuition for controllers we may implement for further analysis.

1.2 Methodology

As discussed in Section 1.1, the point mass assumption adopted by SLIP models cannot represent animal-like locomotion as accurately as models with trunks. Therefore, a new model which considers the effect of an upper trunk is needed. In the first part of the thesis, the BA-SLIP model is considered to represent the system dynamics for animal-like locomotion. Afterwards, we derive the equations of motion for this model using Lagrangian dynamics and perform simulations on the second order differential equations.

Following the derivation of necessary equations, we identify fixed points of the apex return map function. As mentioned earlier, identification of fixed points is crucial for designing control algorithms since they are mostly used as steady-state control targets. Therefore, we performed step by step simulations to identify fixed points of the system for different states.

Finally, we focus on identifying the dimension of the fixed point manifold by using extensive simulation studies performed for different states. The results of this study revealed intuitional information about periodic structure behind animal locomotion. We also used eigenvalue analysis to prove stability of different cases to attribute a theocratical perspective to our findings.

1.3 Contributions

The very first contribution in this thesis is a different analysis, specifically int the choice of the coordinates, on the mathematical model, the BA-SLIP. The model considered in this paper may be utilized as a reasonably accurate descriptive tool for the analysis of sagittal plane locomotion, however it does not consider the effect of damping in the system. We derived the analytical expressions for our lossless model and performed extensive simulation studies to systematically analyze its behavior.

After obtaining analytical expressions for model dynamics, the most significant contribution in this thesis is to find the fixed points of a single-stride for planar locomotion. As mentioned earlier, identification of these fixed points is important in the linearization of system dynamics, furthermore they can be used as steady-state control targets for locomotion.

After finding fixed points of the system, their stability is investigated by calculating the numerical Jacobian and finding eigenvalues of the linearized system. As a result of this procedure, we found some important properties, such as the body incline patterns, of stable fixed points under the proposed control scheme.

The final contribution of this thesis is extensive simulation studies performed to identify the dimension of the fixed point manifold of the system dynamics. With this knowledge in mind, we may have the opportunity to define a reasonable domain and goal regions for our return map function in which we have fixed points, yielding stable locomotion.

Chapter 2

BACKGROUND AND EXISTING WORK

This chapter introduces background for the spring-mass hopper as well as a summary of existing work on the SLIP model with an upright trunk, including its stability analysis and controller design. The necessary background for models, controllers and optimization methods used throughout the thesis is explained at the beginning of each chapter.

2.1 The SLIP Model

Biomechanists discovered the Spring-Loaded Inverted Pendulum (SLIP) model, illustrated in Fig. 2.1, as a metaphor for the locomotion of running animals [4]. As mentioned earlier, subsequent research in biomechanics established the SLIP model as an accurate descriptive tool for different running animals as diverse as humans and cockroaches [5, 6, 7].

Despite its apparent simplicity, the SLIP model represents difficulties from an engineering point of view for conducting formal analysis and designing control algorithms. The SLIP model is a hybrid dynamical system with nonlinear stance

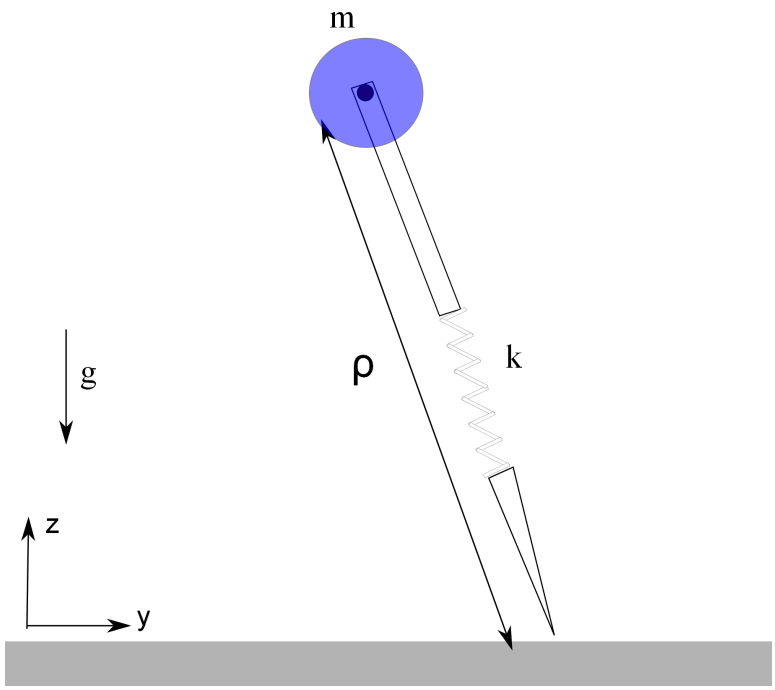

Figure 2.1: The Spring-Loaded Inverted Pendulum (SLIP) model.

dynamics that are non-integrable under the effect of gravity [22]. Motivated by this problem, several analytical approximations to support the analysis of its behaviors and the design of associated controllers have been proposed [12, 23, 24, 25, 26]. We give detailed information on the system model and dynamics for the BA-SLIP model in Chapter 3. Hence, we will end the discussion of the SLIP model here and continue with the existing work on the body attached spring loaded inverted pendulum model.

2.2 Existing Work: The Body-Attached SLIP Model

In addition to the approach introduced in this thesis, there are several studies on the stability analysis and control of stable upright walking using SLIP-like models. A widely used strategy for stabilizing the trunk is to measure the pitch angle with respect to the ground and apply a PD control [8, 27] or a higher level

control [28] in robots with spring-like behavior.

In [19], Maus et al. investigate trunk stability based on the bipedal SLIP model. The control strategy proposed in [19] is to apply a hip torque such that the Ground Reaction Force (GRF) acting on the toe is directed towards a point on the body axis above the center of mass (CoM). Using such a control strategy leads to a damped pendulum-like pitch motion during walking and running. The dynamic stability of the system is analyzed by using a Poincaré map the system variables $(z, \dot{y}, \theta_h, \dot{\theta}_h)$ at each apex state. The system is considered stable if a periodic solution exists and all eigenvalues of the Jacobian of the Poincaré map at the periodic solution have magnitudes less than one. A Newton-Raphson algorithm is utilized to find periodic solutions. Results of this study show that the proposed strategy [19] leads to a pendulum-like pitching motion mounted at a point P, which is called as Virtual Pivot Point (VPP). The model also predicts a hip torque profiles similar in shape and magnitude to observed in human walking.

Based on the conclusions presented in [19], Maus et al. introduced the concept of Virtual Pivot Point, which is used as a support point above the center of mass [20]. The goal of this study was to demonstrate how the VPP concept explains dynamic stability during an upright bipedal gait [20]. Experimental evidence is also provided, showing that this concept is not limited to human walking. In [20], the model of a body which represents an unstable inverted pendulum that needs to be stabilized on top of two springy legs, is used. As in [19], a hip torque during the stance of each leg is introduced to redirect its GRF to a VPP somewhere along the body's long axis [20].

Additionally, a second key role of the trunk model is proposed in [20]. According to [20], adjusting the VPP location offers a simple way to change the speed of the model. This is mainly due to the fact that GRF is always directed towards the VPP. Hence, the acceleration and deceleration of the trunk must be accompanied by a forward and backward lean, respectively. This is a simple way to control the speed of the model proposed in [20]. Experimental studies of [20] yield that VPP for each step can be defined as the single point at which the total

transferred angular momentum remains constant and the sum-of-squares difference to the original angular momentum over time is minimal. Additionally, it is shown that stabilizing posture by applying the VPP concept is not unique to human walking. Experiments suggest that similar observations have been made in other animals, including in dogs and chickens [20].

Following the introduction of the Virtual Pivot Point (VPP) concept in [20], Rummel and Seyfarth performed a deeper investigation of the stabilization of the trunk using VPP [29]. In [20], the fundamental control strategy to stabilize the trunk is to apply a hip torque such that the trunk is transferred to a virtual hanging pendulum about a VPP. However, this strategy requires the generation of a hip torque, which could possibly increase the energy consumption of the robot. Following this idea, the goal of [29] is to investigate if the required hip torque could partially be generated passively using hip springs. It is obviously clear that a passive contribution for trunk stabilization would reduce the robot's energy consumption during locomotion [29].

Although the implementation of stabilizing control strategies for a robot is a challenging problem, fundamental strategies can also be deduced from simple simulation models [29]. For instance, a bisecting strategy for passive stabilization of the trunk was first investigated on a simple passive walker model before it was implemented in [30]. Although this kind of control approaches guarantee trunk stability, it is not similar to the way humans keep their trunk upright [29]. Therefore, [29] uses the control strategy of [20] and tries to investigate the effect of hip springs on passive stabilization.

Actually, the implementation of hip springs in robots was found to be helpful to facilitate swing leg motion and stabilize the gait [31, 32]. However, since hip springs are attached between the legs, they do not contribute to trunk stability. In [33], it is mentioned that tendons, when attached between upper body and legs, could positively contribute to the swing leg motion. This idea was previously used in ARL-monopod II and it has been shown that a single torsional spring decreases the energy consumption of the hip actuator [34]. Motivated from these examples, [29] tries to investigate the effect of hip springs, attached between upper body and

legs, to the stabilization of the upright trunk. Extensive simulation studies show that in a highly reduced model, two separated springs per leg surrounding the hip facilitate stabilization of the trunk during walking [29]. It has also been shown that these passive elements reduce the energy consumption of the hip actuator when arranged in parallel [29].

Following developments on the analysis and control of trunk stabilization, Sharbafi et al. proposed a new leg adjustment strategy, which is combined with the previous Virtual Pivot Point (VPP) concept, to induce stable hopping of the SLIP model extended with an upright trunk [35]. The main goal of [35] is to achieve robust stable hopping with a trunk in the sagittal plane, defined with zero forward velocity. In contrast to [36], placing the leg at a given fixed angle with the ground will not be sufficient to stabilize hopping in this case. Hence, an extra control layer adjusting leg angle during the swing phase should be introduced with respect to standard VPP control strategies [19, 20].

Previous studies on the Virtual Pivot Point (VPP) concept mostly focused on a fixed VPP in a frame attached to the trunk [19, 20]. This strategy was enough to stabilize the posture but it results in slow steady-state convergence and moderate robustness against the perturbations [35]. It has been shown that placing the VPP out of the body frame axis could be used for maneuvers [19] and the compensation of energy losses [37]. Similar approaches are used in [35] to solve issues regarding disturbance rejection and robustness.

The simplest leg placement strategy using a pre-defined angle of attack with respect to ground cannot yield a stable hopping for a SLIP model with an upright trunk as shown for running in [36] and walking in [38]. There are studies, focusing on leg adjustment strategies based on CoM velocity, inspired from Raibert's approach for adjusting foot landing position based on horizontal velocity [39]. Peuker et al. also investigated various strategies for leg adjustment [40]. However, the most robust strategy is to adjust the leg angle with respect to both COM velocity and gravity, which is also used in [35].

In the former control strategy based on the VPP concept, the VPP position is held constant and torque is used to redirect the GRF to this fixed point on

the trunk axis [20]. In contrast to this approach, [35] proposes a new event-based control strategy in which the VPP position is adapted in each apex for the next stance phase using the current system state. This is crucial to improve performance and robustness of the hopping motion. After defining controllable system dynamics, [35] uses two control strategies to choose the state feedback gain; Deadbeat control and Discrete LQR. Both of these controllers resulted in considerable improvements regarding disturbance rejection and robustness against perturbations.

Following the stabilizing controller of [35], Sharbafi et al. investigated possible applications of such control strategies on real robot platforms [41]. To accomplish this goal, they first extended the SLIP model with trunk to a more realistic physical model by adding leg damping and mass to the model presented in [35]. Another contribution of [41] is to add a third layer controller to the two-layer control strategy of [35], so that they can regulate the apex height during locomotion.

Linearizing system dynamics around a fixed point decouples the vertical position dynamics as discussed in [41]. The goal of the hopping height control is to stabilize vertical position dynamics, ensuring that absolute values of eigenvalues are smaller than one [41]. There exist different control strategies using leg rest length and stiffness adjustment [42, 43, 44]. The height control layer of [41] uses the leg rest length adjustment strategy at each apex. They evaluate stable hopping by quantifying the largest perturbation from which the system can recover. Results in [41] show that the controller is capable of stabilizing the system and can handle large perturbations. However, disturbance rejection of such a control strategy is generally slow.

Different from the previous research on the control of the SLIP model with an upright trunk based on the Virtual Pivot Point concept, Poulakakis et al. developed a hybrid controller that induces stable running gaits on an asymmetric spring-loaded inverted pendulum (ASLIP) model [45]. The ASLIP model includes a torso pitch, whose dynamics are coupled to the leg motion [45]. The proposed controller for ASLIP acts on two levels. On the first level, a continuous controller

within a stride regulates the desired torso posture. On the second level, an event-based controller stabilizes closed-loop system dynamics along a periodic SLIP orbit [45]. Results of this study can be treated as a first step toward a general framework of controller design exhibiting compliant hybrid zero dynamics.

2.3 Open Problems

Section 2.2 gives detailed information about existing literature on the analysis and control of Spring-Loaded Inverted Pendulum (SLIP) model with an upper trunk. The VPP concept of [20] provided a good basis for most studies on the body-attached SLIP models. However, we require further analysis of the BA-SLIP model if we want to introduce feedback controller models, which use closed form solutions and limit cycles of the system dynamics. Therefore, a systematic investigation of the periodic solutions related to the SLIP models with an upper trunk must be introduced in order to initiate novel studies on the control of BA-SLIP models.

Motivated from this problem, we introduce our BA-SLIP model with a controller to characterize the fixed points of the dynamical system. Our goal here is to get some intuition about the fixed points of the BA-SLIP model in order to design new controllers that regulates in the fixed point manifolds. We also investigate the stability properties of these fixed points and try to find the dimension of the fixed point manifold in the following chapters.

Chapter 3

BODY ATTACHED SLIP MODEL

3.1 Body Attached SLIP Template

The SLIP Model illustrated in Fig. 3.1, consists of a rigid body attached to a massless leg with a linear spring of stiffness k through a pivot point d away from its center of mass. During locomotion, this model alternates between *stance* phase, during which the toe is fixed on the ground, and *flight* phase, during which the body follows a ballistic trajectory while freely rotating around its pivot point. The *flight* phase is divided into two sub phases: *ascent* and *descent*, according to vertical velocity of the body. The *stance* phase is also divided into two sub phases: *compression* and *decompression*. Fig. 3.2 illustrates a complete step from one apex state to the next, labeling all relevant phases, sub phases and transition events. Properties of these events will be explained in detail in the following paragraphs.

Flight : The time interval when the robot is completely in the air, and does not have any physical contact with the ground. In this phase, the whole body rotates around its center of mass. In our model, since the leg is assumed

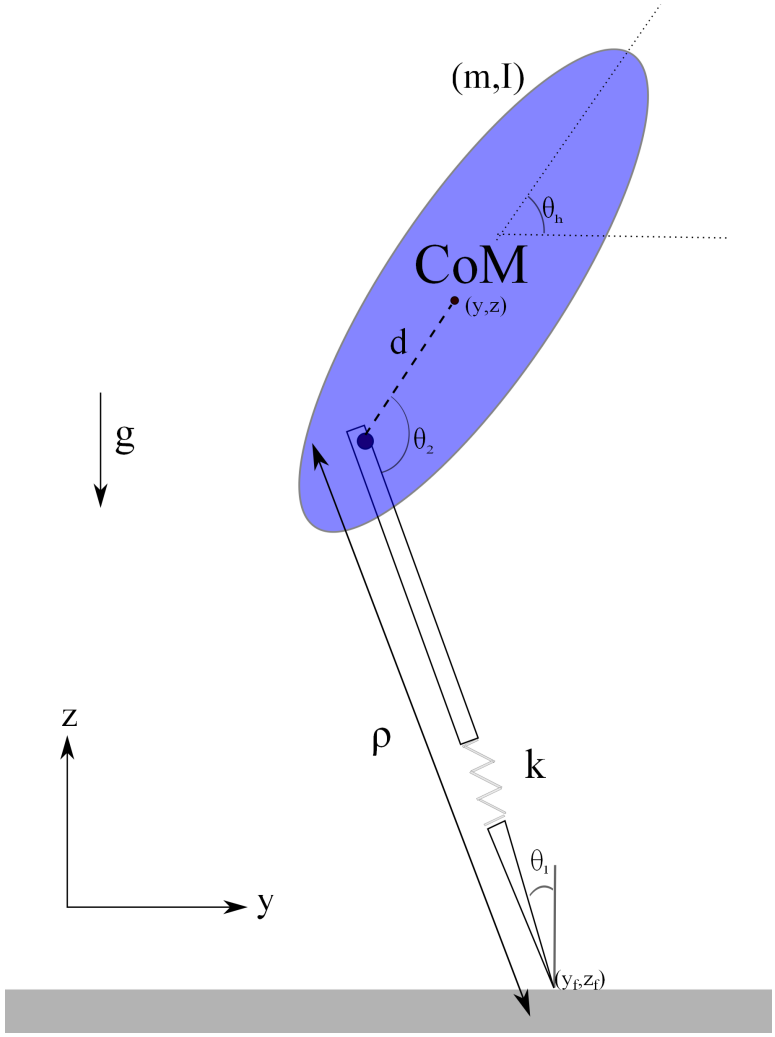

Figure 3.1: The Body-Attached Spring-Loaded Inverted Pendulum (BA-SLIP) model with cartesian and polar coordinates.

to be massless, the total center of mass is the center of mass of the upper body, and therefore rotational dynamics are governed around this point. While freely rotating around the center of mass, the robot also follows a ballistic trajectory under the effect of gravity.

Ascent : A portion of the *flight* phase, where the body gains gravitational potential energy, i.e. is moving up. In this phase, the vertical velocity is positive but decreasing.

Descent : A portion of the *flight* phase, where the body loses gravitational potential energy, i.e. is moving towards the ground. In this phase, the

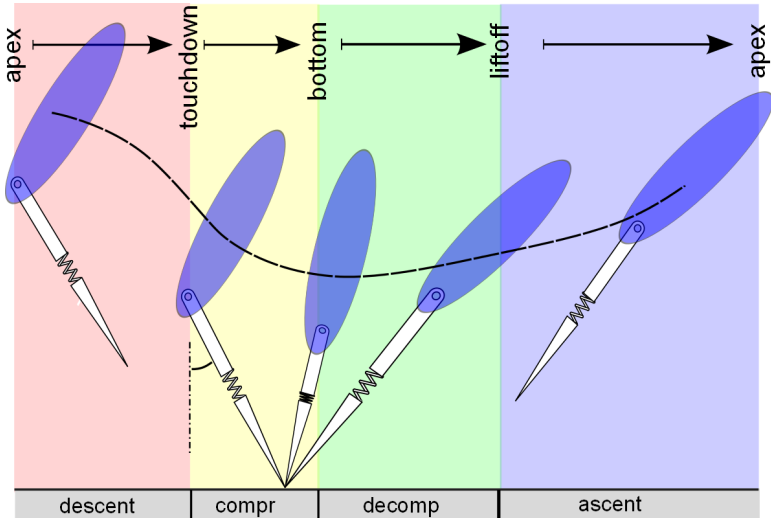

Figure 3.2: All phases and transition events related to the Body-Attached Spring-Loaded Inverted Pendulum (BA-SLIP) model.

vertical velocity is negative and increases in magnitude.

Stance : The time interval when the toe is completely in physical contact with the ground. In this phase, the body rotates around its leg joint (pivot point), following an arc-like trajectory resulting in more complex and coupled dynamical equations.

Compression : A portion of the *stance* phase, where the spring length is smaller than the rest length and is continuing to get smaller. In this phase, the potential energy of the spring increases, and the system stores energy for liftoff.

Decompression : A portion of *flight* phase, where the spring length is smaller than the rest length but increasing. In this phase the potential energy of the spring decreases, being transferred into kinetic energy.

In addition to these phases and sub phases, the model also includes four transition events resulting in phase changes during locomotion. Now, we focus on general characteristics of these events.

Apex : This is the transition event from ascent and to descent. During ascent, the body moves upward with positive vertical velocity, and in descent the

Table 3.1: Notation associated with the BA-SLIP model used throughout the thesis

BA-SLIP States, Event States	
$\rho, \dot{\rho}$	Leg length and its rate of change
$\theta_1, \dot{\theta}_1$	Leg angle with the vertical and its rate of change
$\theta_2, \dot{\theta}_2$	Angle between body and leg and its rate of change
q_s	Stance configuration in polar coordinates, $q_s = [\rho\ \theta_1\ \theta_2]^T$
X_s	Stance state vector in polar coordinates, $X_s = [\rho\ \theta_1\ \theta_2\ \dot{\rho}\ \dot{\theta}_1\ \dot{\theta}_2]^T$
y, z	Horizontal and vertical body positions
$\dot{y}, \dot{z}$	Horizontal and vertical body velocities
y_f, z_f	Horizontal and vertical foot positions
$\theta_h, \dot{\theta}_h$	Body angle with the horizontal and its rate of change
q_f	Flight configuration in cartesian coordinates, $q_f = [y\ z\ \theta_h]^T$
X_f	Flight state vector in cartesian coordinates, $X_f = [y\ z\ \theta_h\ \dot{y}\ \dot{z}\ \dot{\theta}_h]^T$
SLIP Parameters	
m, g	Body mass and gravitational acceleration
I	Body inertia
d	Distance between the CoM and the pivot point
ρ_o	Leg rest length
k	Spring constant
$\mathbf{V}$	Total potential energy
$\mathbf{T}$	Total kinetic energy
$\mathbf{E}$	Total mechanical energy
Torque Parameters	
θ_{td}	Touchdown leg angle
τ	Hip torque command during stance
K_p	Torque proportional constant
K_a	Torque desired velocity
Return maps	
$\mathbf{r}()$	Return map for the BA-SLIP model

body moves downward with negative vertical velocity. So, the sign change in the vertical velocity triggers the apex event and can be detected by the zero crossing of the following function during flight:

$$\mathbf{g}_a(t) := \dot{z}(t). \tag{3.1}$$

Touchdown : This is the transition event from flight to stance. During descent, the robot is completely in the air and moves downward. While moving down, the foot touches the ground at some instant. This moment is referred to as touchdown and it can be detected by checking the zero crossing of the vertical foot position during flight:

$$\mathbf{g}_{td}(t) := z(t) - \rho cos(\theta_1) + dcos(\theta_1 + \theta_2). \tag{3.2}$$

Bottom : This is the transition event from compression and to decompression. During compression, the leg length is decreasing, and during decompression the leg length is increasing. So, the sign change in the leg length rate triggers the bottom event and can be detected by checking the zero crossing of the following function during stance:

$$\mathbf{g}_b(t) := \dot{\rho}(t). \tag{3.3}$$

Liftoff : This is the transition event from stance to flight. It occurs when the foot is about to leave the ground. The foot can only leave the ground when the ground reaction force on it is equal to zero. This event can be detected by the zero crossing of the following function during decompression:

$$\mathbf{g}_{lo}(t) := -k(\rho(t) - \rho_0). \tag{3.4}$$

As described in transition event definitions, the highest point during flight is defined as *the apex* point for each stride. Using the system parameters, the apex point for the n^{th} stride is defined as

$$q_n := [z_n, \theta_h, \dot{y}_n, \dot{\theta}_h]^T. \tag{3.5}$$

3.2 BA-SLIP Dynamics

As mentioned in previous sections, the BA-SLIP model has hybrid properties which requires separate analysis of stance and flight dynamics. This section provides a complete overview of motion dynamics.

3.2.1 Analysis Methodology

Dynamic equations of the model are written using the Lagrangian method, in which the Lagrangian of a system is found and replaced in the Euler-Lagrange Equation

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_{q_i}, \tag{3.6}$$

where $L = T - V$ is the *Lagrangian*, T is the total kinetic energy and V is the total potential energy of the system. The variables q_i are the generalized coordinates for the system. In each phase (flight and stance), there are three of them, resulting in a 6 element state vector. The details of these variables are given in the next subsections. τ_{q_i} is the torque component defined for coordinate q_i.

3.2.2 Flight Dynamics of BA-SLIP

The BA-SLIP model has a rigid body which can freely rotate. Therefore, during the flight phase, while following a ballistic trajectory under the effect of gravity, this body independently rotates around its center mass.

During the flight phase, since the spring/leg length and angle between the leg and the normal is not important, it's more convenient to use cartesian coordinates in the dynamical analysis. Therefore, the coordinate vector is defined as

$$\mathbf{q} = \begin{pmatrix} y \\ z \\ \theta_h \end{pmatrix}. \tag{3.7}$$

During flight, the kinetic energy, T of the system can be written as

$$T = \frac{1}{2}m\dot{y}^2 + \frac{1}{2}m\dot{z}^2 + \frac{1}{2}I\dot{\theta}_h^2, \tag{3.8}$$

where the first term stands for the horizontal kinetic energy, the second term stands for the vertical kinetic energy and the last term stands for the rotational kinetic energy of the trunk.

The potential energy can be written as

$$V = mgz, \tag{3.9}$$

which is due to the potential energy of the upper trunk with respect to the ground, where the leg is massless.

So, the Lagrangian of the system is

$$L = T - V = \frac{1}{2}m\dot{y}^2 + \frac{1}{2}m\dot{z}^2 + \frac{1}{2}I\dot{\theta}_h^2 - mgz. \tag{3.10}$$

After substituting in Euler-Lagrange Equation for every coordinate component, we get the following dynamical equations (Details of the derivation are given in Appendix A):

$$\begin{bmatrix} \ddot{y} \\ \ddot{z} \\ \ddot{\theta}_h \end{bmatrix} = \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix}. \tag{3.11}$$

From the equations above, the state vector of cartesian coordinates can be defined as

$$\mathbf{X_f} := \begin{bmatrix} y & z & \theta_h & \dot{y} & \dot{z} & \dot{\theta}_h \end{bmatrix}^T, \tag{3.12}$$

and the corresponding flight dynamics are

$$\dot{\mathbf{X}}_\mathbf{f} := \begin{bmatrix} \dot{y} & \dot{z} & \dot{\theta}_h & 0 & -g & 0 \end{bmatrix}^T, \tag{3.13}$$

which can be written as

$$\dot{\mathbf{X}}_\mathbf{f} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{X}_\mathbf{f} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -g \\ 0 \end{bmatrix}. \tag{3.14}$$

3.2.3 Stance Dynamics of BA-SLIP

During stance, the robot follows an arc like trajectory since the leg is rotating around the ground contact point without slipping. Polar coordinates will be better suited for this trajectory, so we define a coordinate vector with polar variables as

$$\mathbf{q} = \begin{pmatrix} \rho \\ \theta_1 \\ \theta_2 \end{pmatrix}. \tag{3.15}$$

The associated polar coordinate transformation is given as

$$\begin{aligned} y &= y_f - \rho \sin\theta_1 + d\sin(\theta_1 + \theta_2) \\ z &= z_f + \rho\cos\theta_1 - d\cos(\theta_1 + \theta_2) \end{aligned}. \tag{3.16}$$

Here, the cartesian coordinates, vertical and horizontal positions, are written in terms of polar coordinates and system parameters, leg length, leg angle and the body-leg angle. This transformation is important because in the derivation of dynamical equations for the stance phase we need the Lagrangian formulation in terms of only polar coordinates.

Similarly, the leg length can be written in terms of vertical and horizontal positions and body-leg angle as

$$\rho = \sqrt{y^2 + z^2 - d^2\sin^2\theta_2} + dcos\theta_2. \tag{3.17}$$

Finally the body reference frame angle can be written in polar coordinates using geometric arguments as

$$\theta_h = \theta_1 + \theta_2 - \pi/2. \tag{3.18}$$

Before writing the Lagrangian of the system, we need to find the counterparts of vertical, horizontal and rotational velocities' in polar coordinates. Taking derivatives of both sides of equations (3.16),(3.18), we obtain the following first order velocity equations in polar coordinates:

$$\dot{y} = -\dot{\rho}\sin\theta_1 - \rho\cos\theta_1\dot{\theta}_1 + d\cos(\theta_1+\theta_2)(\dot{\theta}_1+\dot{\theta}_2), \tag{3.19}$$

$$\dot{z} = \dot{\rho}\cos\theta_1 - \rho\sin\theta_1\dot{\theta}_1 + d\sin(\theta_1+\theta_2)(\dot{\theta}_1+\dot{\theta}_2), \tag{3.20}$$

$$\dot{\theta}_h = \dot{\theta}_1 + \dot{\theta}_2. \tag{3.21}$$

During stance, the Lagrangian of the system slightly changes, because this time, we must account for spring dynamics as well. Kinetic energy for the stance phase can be written as

$$T = \frac{1}{2}m\dot{y}^2 + \frac{1}{2}m\dot{z}^2 + \frac{1}{2}I(\dot{\theta}_1+\dot{\theta}_2)^2, \tag{3.22}$$

where the first two terms are due to kinetic energies resulting from horizontal and vertical velocities, respectively, and the last term stands for the rotational kinetic energy. Different from equation (3.8), stance rotational kinetic energy is written in terms of polar coordinate variables using equation (3.18).

The potential energy, V, of the system is

$$V = mgz + \frac{1}{2}k(\rho-\rho_0)^2, \tag{3.23}$$

where the first term stands for the trunk's gravitational potential energy with respect to ground reference frame, and the second term stands for the spring potential energy of the leg. So, the Lagrangian of the system is

$$L = T - V = \frac{1}{2}m\dot{y}^2 + \frac{1}{2}m\dot{z}^2 + \frac{1}{2}I(\dot{\theta}_1+\dot{\theta}_2)^2 - mgz - \frac{1}{2}k(\rho-\rho_0)^2. \tag{3.24}$$

After applying the polar coordinate transformations given in Equation (3.16), we obtain the polar Lagrangian equation as

$$L = \frac{1}{2}m\Big(\dot{\rho}^2 + \rho^2\dot{\theta}_1^2 + d^2(\dot{\theta}_1+\dot{\theta}_2)^2 + 2d(\dot{\theta}_1+\dot{\theta}_2)(\dot{\rho}\sin\theta_2 - \rho\dot{\theta}_1\cos\theta_2)\Big) + \frac{1}{2}I(\dot{\theta}_1+\dot{\theta}_2)^2 - mg\rho\cos\theta_1 + mgd\cos(\theta_1+\theta_2) - \frac{1}{2}k(\rho-\rho_0)^2. \quad (3.25)$$

Subsequently, applying Lagrange equation (3.6), to each coordinate component leaves us with the second order dynamical equation

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}). \quad (3.26)$$

where $\mathbf{M}$ and $\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}})$ are given as

$$\mathbf{M} = \begin{bmatrix} 1 & d\sin\theta_2 & d\sin\theta_2 \\ d\sin\theta_2 & \rho^2 - 2d\rho\cos\theta_2 + d^2 + I/m & d^2 - d\rho\cos\theta_2 + I/m \\ d\sin\theta_2 & d^2 - d\rho\cos\theta_2 + I/m & d^2 + I/m \end{bmatrix}, \quad (3.27)$$

$$\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \rho\dot{\theta}_1^2 - g\cos\theta_1 - \frac{k}{m}(\rho-\rho_0) - d\cos\theta_2(\dot{\theta}_1+\dot{\theta}_2)^2 \\ -2\dot{\rho}\dot{\theta}_1(\rho - d\cos\theta_2) + g\rho\sin\theta_1 - \\ \dot{\theta}_1\dot{\theta}_2 2d\rho\sin\theta_2 - \dot{\theta}_2^2 d\rho\sin\theta_2 - gd\sin(\theta_1+\theta_2) \\ \frac{\tau}{m} + \dot{\rho}\dot{\theta}_1 2d\cos\theta_2 + \dot{\theta}_1^2 d\rho\sin\theta_2 - gd\sin(\theta_1+\theta_2) \end{bmatrix}. \quad (3.28)$$

We can write this system in the standard form as

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{T}, \quad (3.29)$$

where $\mathbf{M}$ is the same matrix, and $\mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})$, $\mathbf{G}(\mathbf{q})$ and $\mathbf{T}$ are given as

$$\mathbf{B} = \begin{bmatrix} 0 & d\cos\theta_2\dot{\theta}_1 - \rho\dot{\theta}_1 & d\cos\theta_2(\dot{\theta}_2 + 2\dot{\theta}_1) \\ 2\dot{\theta}_1(\rho - d\cos\theta_2) & -\dot{\theta}_2 2d\rho\sin\theta_2 & \dot{\theta}_2 d\rho\sin\theta_2 \\ -\dot{\theta}_1 2d\cos\theta_2 & -\dot{\theta}_1 d\rho\sin\theta_2 & 0 \end{bmatrix}, \quad (3.30)$$

$$\mathbf{G} = \begin{bmatrix} \frac{k}{m}(\rho - \rho_0) + g\cos\theta_1 \\ -g\rho\sin\theta_1 + gd\sin(\theta_1 + \theta_2) \\ gd\sin(\theta_1 + \theta_2) \end{bmatrix}, \tag{3.31}$$

$$\mathbf{T} = \begin{bmatrix} 0 \\ 0 \\ \frac{\tau}{m} \end{bmatrix}. \tag{3.32}$$

The details of the derivation procedure can be found in Appendix B.

From the equations above, the state vector for polar coordinates can be defined as

$$\mathbf{X_s} := \begin{bmatrix} \rho & \theta_1 & \theta_2 & \dot{\rho} & \dot{\theta}_1 & \dot{\theta}_2 \end{bmatrix}^T. \tag{3.33}$$

3.3 Formulation of the Control Problem

This section focuses on the formulation of the control problem for the BA-SLIP template. It briefly describes control parameters used and how they were chosen. Control of BA-SLIP model seeks to reach stable apex states during locomotion through the use of discrete, per-step control inputs, as well as the continuous torque during stance.

3.3.1 Definition of Fixed Point and Its Importance

BA-SLIP state $\mathbf{X}$ in Cartesian coordinates is defined in (3.12). Let us define $\mathbf{X_n}$ as the apex state at the n^{th} stride, $\mathbf{X_{n+1}}$ as the apex state at the $(n+1)^{th}$ stride, and assume that $\mathbf{u_n}$ is the control input vector applied at the n^{th} stride of Poincaré section at apex with $\dot{z} = 0$ enables us to define a discrete apex return map of the form

$$\mathbf{X_{n+1}} = \mathbf{r}(\mathbf{X_n}, \mathbf{u_n}). \tag{3.34}$$

Here, the function $\mathbf{r}$ depends on the composition of the dynamics given by (3.13) and (3.26). Since the closed form solutions of such function is not possible, we rely on its numerical computation by using the dynamics defined in Section 3.2. Hence, a fixed point $\mathbf{X}^*$ of the system given by (3.34) is a vector in $\mathbf{R^m}$ which satisfies the equation

$$\mathbf{X}^* = \mathbf{r}(\mathbf{X}^*, \mathbf{u}^*). \tag{3.35}$$

Since horizontal position is non-cyclic and vertical velocity is always constant and zero at apex points, as defined in event transition sections, its form can be defined as

$$\mathbf{X}^* := \begin{bmatrix} z^* \\ \theta_h^* \\ \dot{y}^* \\ \dot{\theta}_h^* \end{bmatrix}. \tag{3.36}$$

The objective here is to find control inputs using the above return map for a given apex state to make it a fixed point of this dynamical system. Fixed points are very important since they correspond to periodic motions and can be used as steady state control targets.

3.3.2 Possible Modes of Control

Based on the SLIP and BA-SLIP models, two important control parameters are common in many control techniques:

- The touchdown leg angle with the vertical, θ_{td}: This is an essential component for most legged systems. Its control is a relatively simple objective because the only thing to consider is making sure that the leg reaches the desired angle before touchdown occurs.
- The mechanical energy change, ΔE: In many applications, it's indispensable to control the total mechanical energy by injecting into the system or ejecting from it. This component generally requires more effort to control and may be implemented in various ways [10, 25, 46]. Some of these control

methods include Leg-Stiffness Control (LSC), Two-Phase Stiffness Control (TPSC), Leg Length Control (LLC), Torque Actuated Control(TAC).

In LLC, injection of mechanical energy is achieved by compressing the spring leg during flight and effectively releasing it during stance. An example application is the Bow-Leg robot [47]. In LSC, ΔE is controlled by changing the leg stiffness during flight as in [48]. TPSC is the inverse version of LSC, where the leg stiffness is changed in stance [49]. The last one, TAC, is of more interest to us since our model uses this strategy through an actuator at the hip joint. A negative or positive torque is applied to the body during stance phase resulting changes in the total mechanical energy. Example applications using this model include quadrupeds [50], RHex hexapod [51], as well as a number of other monopedal platforms [21, 46, 52, 53], that use a single rotary actuator for each leg. The reason for choosing torque actuated control (TAC) scheme is to be able to easily utilize the robot model given in [21].

3.3.3 Torque Model

As described in the previous section, we will utilize torque actuation control scheme (TAC). For the torque model, we use the following control law, which consists of four parameters:

$$\tau = -K_p(\theta_2 - \theta_{2d}) - K_d(\dot{\theta}_2 - \dot{\theta}_{2d}). \tag{3.37}$$

In this torque formulation, K_p and K_d are proportional and derivative gains, θ_{2d} and $\dot{\theta}_{2d}$ are desired body-leg angle and its desired rate of change. It's a classic PD controller that aims to reach steady state behavior by applying negative feedback. To be more specific, assuming that we want $\theta_2 = \theta_{2d}$, if $\theta_2 > \theta_{2d}$, we apply a negative torque on the body, resulting in clockwise motion of the trunk, which decreases θ_2 and make it closer to the desired value. If $\theta_2 < \theta_{2d}$, we apply counter-clockwise torque on the trunk, which results increase on body-leg angle,

θ_2, again making it closer to the desired value. Similar intuition is valid on the derivative component.

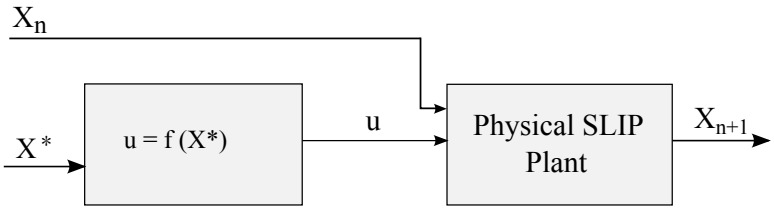

Figure 3.3: Control diagram, X^* is the desired state, u is the control parameters for reaching X^* and, X_n is the current apex state.

Fig. 3.3 represents the main control diagram implemented in this thesis. Here, $\mathbf{X_n}$ is the current apex state vector, $\mathbf{X_{n+1}}$ is the next apex state vector, $\mathbf{X}^*$ is the desired apex state vector, and $\mathbf{u}$ is the control input in order to reach the desired apex state. This vector consists of torque parameters of (3.37), and the touchdown leg angle, θ_{td}. The function f represents a mapping between desired apex states and control inputs. The control input is only specified by the desired apex state vector, $\mathbf{X}^*$, and does not utilize the state information, $\mathbf{X_n}$. Hence, the resulting control algorithm is open-loop. Since the closed form expression of the function f requires the integration of system dynamics and is very difficult to write, we try to investigate it numerically in Chapter 4 by performing various simulations.

Chapter 4

SIMULATIONS AND RESULTS

In this chapter, we present simulation results regarding the model given in Chapter 3. More specifically, brief subsections for the following subjects of interest are given:

- Fixed points of the control problem described in Section 3.3,
- Stability analysis of fixed points,
- Stability regions around fixed points,
- Results regarding the fixed point manifold dimension,
- Time-varying control strategies for unstable fixed points

4.1 Simulation Environment

This subsection contains a brief information about the simulation environment, utilities used, algorithm choices and finally performance criterion. First, the development environment and utilities are briefly discussed, then for the optimization problem, the algorithm choices are presented and advantages and disadvantages of each of them are listed, and finally a short discussion on tolerances

of optimization algorithms is given.

4.1.1 Simulation Environment and Toolboxes

All of the simulations, graphics, plots and numerical analysis presented in this chapter are obtained by using MATLAB. This chapter focuses on the problem of finding the fixed points of the model in the previous chapter, i.e. the points which satisfy the global minimum of the objective function which will be explained in section 4.2. MATLAB has two important toolboxes for this purpose; Optimization Toolbox and Global Optimization Toolbox.

Global Optimization Toolbox has the following algorithm options: Global Search, Direct Search, Genetic Search, Simulated Annealing, Multi objective Optimization. Most of these algorithms are good with objective functions with less points satisfying global minimum, and start searching for optimum solutions using randomized choices in order to scan all basins of attraction, instead of derivative search.

Optimization Toolbox contain more alternatives for standard and large-scale optimization algorithms, which generally use gradient information. This is beneficial when we have an intuitive idea about the global solution neighborhood to begin the search. As it will be discussed in more detail in the following sections of this chapter, since the BA-SLIP model has a lot of fixed points (points that satisfy the global minimum), and we have some intuitive idea about how these solutions might lie on the space, the latter toolbox (Optimization Toolbox) is more appropriate for our problem.

4.1.2 Algorithms

The Optimization Toolbox has a variety of functions for optimization problems. Since, in our model, some of the solution parameters must contain lower and upper bounds in order to comply with the physical constrains, constrained minimization

algorithms are of interest to us. For this problem, there are four types of algorithm choices available:

- **Trust-region-reflective Algorithm:** This algorithm type requires the analytical gradient of the objective function and therefore does not suit to our problem.
- **Interior-point Algorithm:** This is a large-scale algorithm and mostly specialized to operate on sparse matrices [54]. On the other hand, our model mass matrix presented in equation (3.27) is not primarily populated with zeros.
- **Active-set Algorithm:** This is not a large-scale algorithm and can take large steps which speeds up the global minimum search. This property is very important and might be extremely useful when working with a high dimensional input space.
- **SQP:** Like active-set algorithm, this one is not a large-scale method and similar to active-set in the problem formulation and might sometimes be advantageous for fast convergence [55] .

Our optimization problem has a parameter space having at least seven dimensions. It's a high dimensional space with many fixed points, in other words, points satisfying global minimum value of the objective function. According to above algorithm descriptions, our optimization problem needs algorithms which are flexible and fast in parameter spaces with high dimensions, does not need an analytical gradient, does not need sparse linear algebra, and therefore best fits with Active-Set and SQP algorithms. All of the simulations in Chapter 4, use these two main algorithms.

4.1.3 Tolerances

As explained in subsection 4.1.2, SQP and Active-Set nonlinear optimization algorithms were chosen for our optimization problem. For these algorithms, there

are two important tolerance factors that needs to be considered carefully while searching for fixed points:

- TolFun: Lower bound on the change of the objective function during an iteration step,
- TolX: Lower bound on the size of the iteration step, in other words, the Euclidian-norm of the distance between the two input vectors at consecutive iteration steps.

Iterations end when the corresponding objective function values and step size at the last step is smaller than TolX or TolFun. So, although an infinite precision simulation is not possible in order to get accurate results, these values should be as small as they can while complying with differential solver's tolerances. So, in most of our simulations, TolFun and TolX values are taken as 10^{-11}. Only when searching for fixed points in a relatively small domain, they have been changed to 10^{-12} for finding better results.

4.2 Fixed Point Characterization and Performance Criterion

Fixed points are important because they reveal periodic motion patterns in the locomotion, and can be used as steady state targets. A fixed point of our dynamical system is a vector $\mathbf{X}^*$ that satisfy the equation $\mathbf{X}^* = \mathbf{r}(\mathbf{X}^*, \mathbf{u}^*)$, where $\mathbf{u}^*$ is the steady state control input and $\mathbf{r}$ is the vector return map defined in Chapter 3. In order to find the fixed points of our model, extensive simulations have been made using the chosen algorithms explained in detail in section 4.1. Considering our return map given in equation (3.34), a fixed point theoretically satisfies $X^* = X_{n+1} = X_n$, i.e. there is no error between consecutive steps. However, in simulative approaches this is not reasonable in most cases, since computers work with finite precision. Since the return map domain and range vector elements does not constitute a countable set, we have to put an error threshold in order to

classify the resulting points as fixed or not. Therefore, an error function given as below should be defined:

$$\epsilon_n = \frac{\|X_{n+1} - X_n\|}{\|X_n\|}. \tag{4.1}$$

The intuition behind dividing with X_n is calculating the percentage error made in consecutive steps, i.e. normalizing the error function, because absolute error may not be very useful in most cases, especially where we don't have a clue about the magnitude of X_n. This error function will be used as the objective function for the optimization, and the global minimum value we're looking for is 0, because of the definition of fixed point. However, as it will be explained in more depth in the following sections, considering stable and unstable fixed point candidates acquired from simulations, an error threshold 10^{-5} will be enough for being a candidate fixed point. In other words, if $\epsilon_n < 10^{-5}$, most probably we are near a fixed point, and couldn't find it precisely. Due to optimization algorithms, differential solver precisions and small numerical noise in finite-precision arithmetics in Matlab, the given point can be characterized as a fixed point of the system. This hypothesis is supported even more clearly in the following sections where stable fixed points are investigated, because optimization algorithms give global minimum as nearly 10^{-5} for these points, however since they're stable, after a few steps, the objective function decreases below 10^{-12}.

4.3 Fixed Points for a Given Horizontal Velocity

In this section we explain the motivation behind finding fixed points for a given horizontal velocity, present simulation results regarding our BA-SLIP model, and investigate the stability behavior of these fixed points.

4.3.1 Motivation

It's very important to find a fixed point for any, in some reasonable range, given horizontal velocity because it means that we can find a periodic motion pattern,

and run the robot in any velocity we want. Since the locomotion of interest is in the sagittal plane, to be able to run the robot in a desired horizontal velocity constitutes an important steady state control target.

For an average human body, maximum locomotion speed is 22 km/h [56]. In [57], some experimental results supporting linear correlation between comfortable locomotion speed and leg length of human body are presented. Since human leg length is nearly 1 meters [58], and our robot has leg length 20 cm, [21], we might consider our robot speed as nearly at most 22 / 5 = 4.4 km / h, which corresponds to 1.2 m/s approximately. Since this is an upper bound, we might round down this number to 1 m/s in order to decrease simulation time. Therefore, a reasonable range for the horizontal velocity can be determined as

$$0 \leq \dot{y} \leq 1. \tag{4.2}$$

In order to show that, for any given horizontal velocity in the range given in equation (4.2), there exist a fixed point; we must choose a step size as small as we can. Considering the dimension of our optimization problem, step size of horizontal velocity is chosen as 0.01, dividing the range in equation (4.2) into 100 equal pieces.

4.3.2 Algorithm Diagram

As described in section 4.2, in order to find a fixed point, we have to minimize the objective function given in equation (4.1); which is a classical optimization problem and we can use the algorithms discussed in Algorithms section. Fig. 4.1 represents an overview of our algorithm. It takes an initial guess on the parameters, $\mathbf{p_0}$, consisting of torque coefficients, touchdown leg angle, apex height, body angle and body angle rate; and converges to optimum configuration by using Active-Set or SQP Algorithms.

In the above figure, a horizontal velocity in the range [0,1] is given as input to the optimization; and Matlab finds a fixed point candidate with given horizontal velocity. The stopping criteria block controls if the objective function is less than

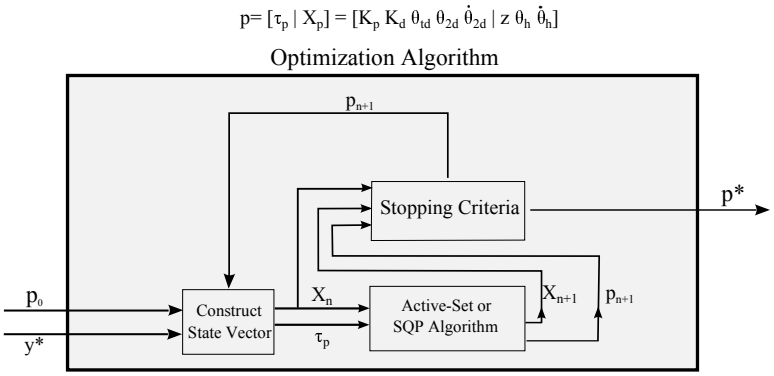

Figure 4.1: Optimization diagram constructed for finding the fixed point configuration, p^*; height, body angle, body angle rate, torque parameters and touchdown angle, given the desired horizontal velocity, $\dot{y}^*$.

TolFun or TolX. In order to enhance the performance, optimizations are run again by using the previous optimization's output as initial condition to the next optimization, until the error between two consecutive steps is below an acceptable threshold. Fig. 4.2 illustrates this concept clearly.

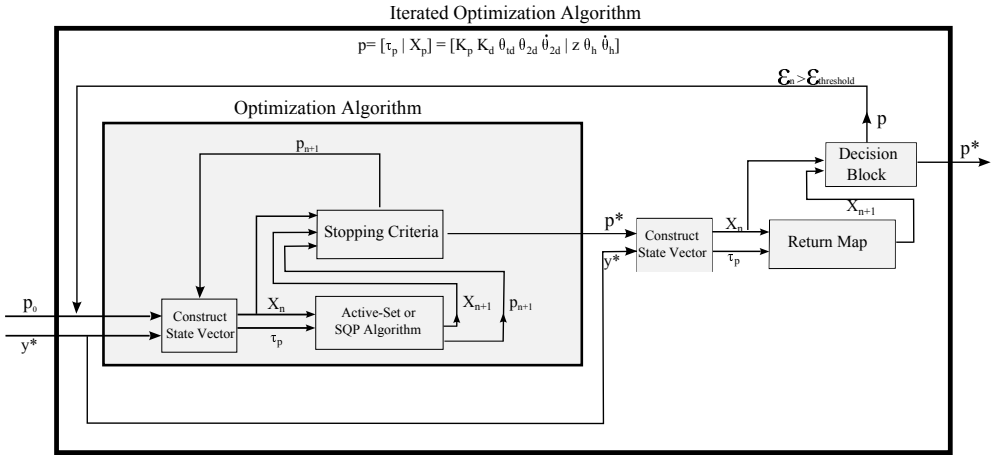

Figure 4.2: Enhanced version of the diagram given in Fig. 4.1. This diagram contains an additional step which checks the output parameters if they satisfy the fixed point criterion or not, and continues optimization if they do not satisfy.

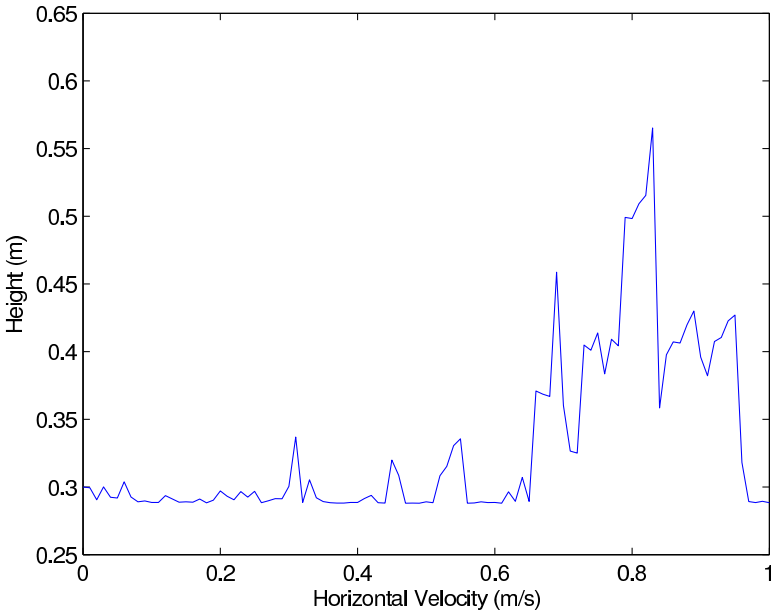

Figure 4.3: Relation between horizontal velocity, $\dot{y}$, and apex height, z for fixed points found. In the optimization procedure, horizontal velocity is given as in input to the loop and remaining parameters are read as output. Therefore, the apex height values are the results of the optimization loop.

4.3.3 Results and Discussion on the Stability

Fig. 4.3 shows the relationship between horizontal velocity and apex height of the fixed points. One important observation that can be made from this figure is most apex height values are nearly 0.3 meters. That's because of the lower limit on apex value in the constrained optimization algorithm, since below some height threshold robot can not actually open its leg or jump due to physical constraints such as its actual leg and body length. Apparently, if the robot jumps from lower heights, the error that's made between consecutive steps will be less. This result is intuitive however while at the same time being speculative because we might miss other periodic motion patterns. There might be different height - horizontal velocity combinations that lead us to better and more stable locomotions. Proof of this concept is obvious when we look at bigger horizontal velocities in this figure. At $\dot{y}$ between 0.65 and 0.95, we have apex height, z, between 0.4 and 0.55 meters. After $\dot{y} = 0.95$, apex height again becomes nearly 0.3 meters. Upon this result, we might infer that apex height is not unique for a given horizontal

velocity, in other words, fixed point manifold dimension is bigger than 1.

Fig. 4.4 shows the body angle versus horizontal velocity. According to this plot, we can see that the trunk is in almost at upright position at the fixed points, i.e. $\theta_h \approx \pi/2$. Some simulation results, as we will present in section 4.4, show that this is not the only case for fixed point distribution. There is another distribution we encounter when the body is downward oriented, i.e. $\theta_h \approx -\pi/2$. However, in both situations there is not much deviation from $-\pi/2$ or $\pi/2$.

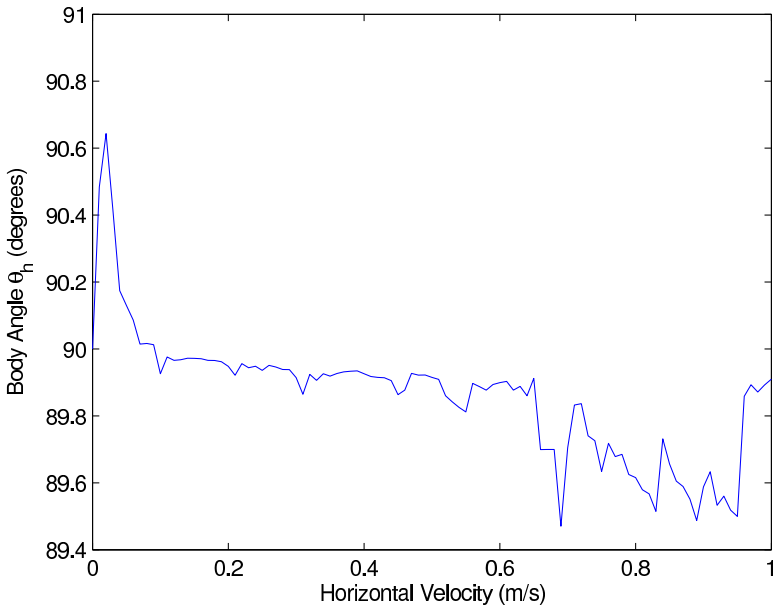

Figure 4.4: Relation between horizontal velocity, $\dot{y}$, and body angle, θ_h for fixed points found. Body angle is the trunk orientation with respect to the ground reference frame, and independent of leg orientation.

Finally, Fig. 4.5 shows the maximum absolute eigenvalues of the Jacobian matrix with respect to changing horizontal velocity, numerically found to investigate the stability behaviors of our fixed points. According to this plot, eigenvalues are all greater than nearly 3, resulting in unstable behavior.

For stable motion we need to have maximum absolute eigenvalue less than 1, since this is a discrete map. Unstable fixed points are only important for themselves, not the points in their neighborhood, because if we start near an unstable fixed point, its vector field pushes us out, and therefore apex vectors, $\mathbf{X_n}$ diverge. Because of finite precision arithmetics due to computer hardware, it's

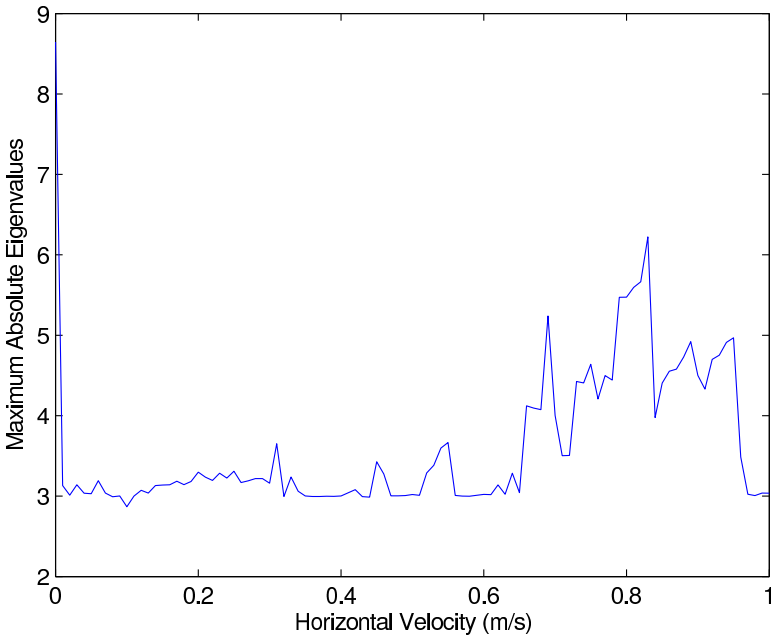

Figure 4.5: Relation between horizontal velocity, $\dot{y}$, and maximum of the absolute eigenvalues of the numeric Jacobian matrix, J. These eigenvalues show the stability patterns of given fixed points.

nearly impossible to have the fixed point with no numerical noise. For instance, the apex configuration shown in Fig. 4.6, is one of the very intuitive fixed points; hopping on the ground with vertical body alignment and no horizontal velocity. It's physically obvious that if we do not have damping and do not apply any torque, the robot will hop forever. This result is also found in our optimization for $\dot{y} = 0$, however, the error between consecutive apexes is found to be nearly 10^{-12} in Matlab due to numerical roundoff errors(e.g. we cannot represent $\pi/2$ exactly in numerical analysis because the number is irrational).

To sum up, we could find a fixed point for any given horizontal velocity in the specified range but these fixed points turned out to be unstable and discontinuous with respect to apex height. Therefore, it's better to look for fixed points for a given height and horizontal velocity pair, and examine stability behaviors for both upright and downward body angle orientations. Related simulations are presented in the next section.

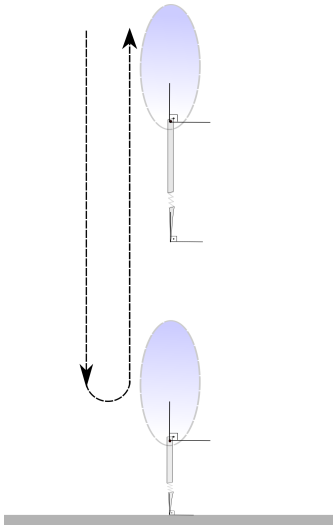

Figure 4.6: Realization of hopping in place. Body angle is 90 degrees, leg angle is 0 degrees, no horizontal speed is present and no torque is applied. The body falls from a specified height with vertical orientation, touches the ground and lifts off after some time and goes back to the height where it's been thrown since no damping factor is present in the system.

4.4 Fixed Points for a Given Height and Horizontal Velocity

In this section we explain the motivation behind finding fixed points for a given horizontal velocity and apex height pair, present simulation results regarding our BA-SLIP model, and investigate the stability behavior of these fixed points. Also, different from the previous section, we consider the two cases of fixed point spaces: upward body orientation and downward body orientation, because stability behaviors of these two spaces will be totally different in the proposed control scheme.

4.4.1 Motivation

As described in the previous section, we could find fixed points for any given horizontal velocity, but there were numerical evidence showing that these fixed points are not unique, and there might exist multiple apex heights satisfying fixed

point criterion for a given horizontal velocity. This time a similar grid search on the space of horizontal velocity, $\dot{y}$, and apex height, z, must be conducted so that for reasonable values of these state parameters, we can find fixed points and their control parameters. The horizontal velocity and apex height ranges are chosen to be as follows:

$$0.1 \leq \dot{y} \leq 1, \qquad 0.3 \leq z \leq 0.7 \tag{4.3}$$

because, as described in section 4.3.1, $\dot{y}$ range is intuitive, and apex height must be greater than the leg length + pivot-COM distance to be able to physically jump, which corresponds to 0.24 meters in our case. An average human cannot jump to heights three times his/her leg length, so, our range for BA-SLIP model with 0.2 meters is far more greater than the reasonable range for human body jump. For these simulations, the step size for horizontal velocity is taken as 0.1, and for apex height as 0.05 meters, resulting in 90 grid squares in the specified region.

4.4.2 Algorithm Diagram

The optimization problem here is very similar to the one described in section 4.4.2 and can be depicted as in Fig. 4.7.

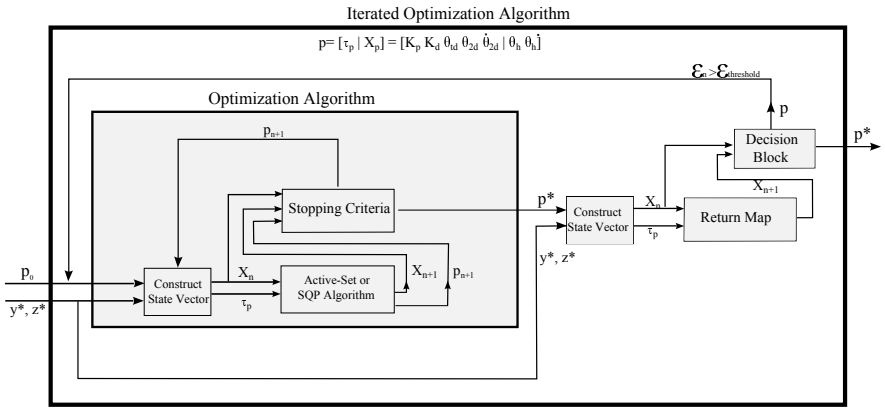

Figure 4.7: Optimization diagram constructed for finding the fixed point configuration; body angle, body angle rate, torque parameters and touchdown angle, given the desired horizontal velocity, $\dot{y}^*$, and apex height, z.

In Fig. 4.7, as described in the preceding subsection, a horizontal velocity in the range [0.1,1] and apex height in the range [0.3,0.7] is given as input to the optimization; and Matlab finds a fixed point candidate at that horizontal velocity-apex height pair. In order to enhance the performance, we do the simulations again with the initial conditions found in the previous optimization up to an acceptable between the two consecutive apex states, as we did in section 4.4.2.

4.4.3 Results and Discussion on the Stability

In this section simulation results are presented in two subsections; upward and downward body orientation because of their different stability properties.

4.4.3.1 Upward Body Orientation

Simulation results show that for almost any given apex height-horizontal velocity pair in the range given in equation (4.3), there exists a fixed point with upward body orientation, with nearly body angle, θ_h, is being equal to 90 degrees.

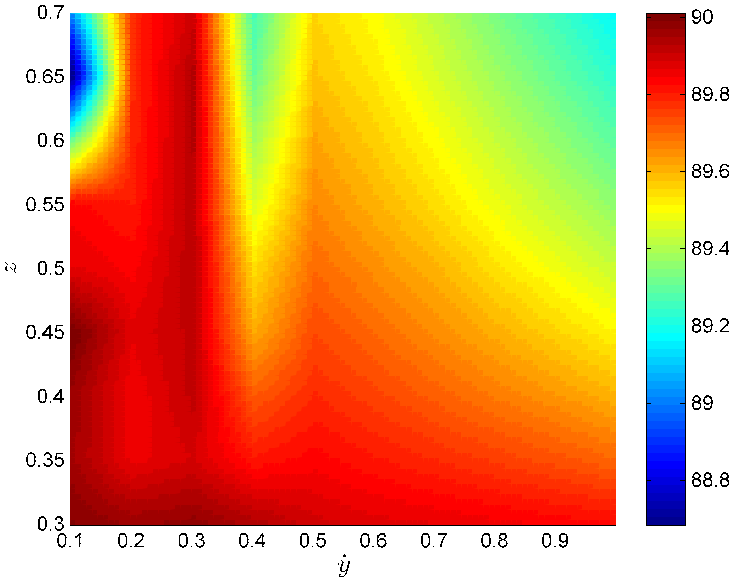

Figure 4.8: Colormap of the body angle, θ_h with respect to changing horizontal velocity, apex height pairs.

There are only two $(\dot{y}, z)$ pairs which the optimization couldn't find a fixed point. These pairs are (0.1, 0.65) and (0.1, 0.7) and their neighborhood is shown as blue region in Fig. 4.8. After all, this is rather intuitive since the robot does not have much horizontal velocity but large potential energy due to its apex height.

Fig. 4.8 shows a color map of the body angle versus horizontal velocity and apex height with respect to the color bar given in the right. According to this plot, we can see that the trunk is almost at the upright position, i.e. $\theta_h \approx \pi/2$. It can be seen that all of the fixed point angles are in the neighborhood 1 degree neighborhood of 90 degrees. Other angles such as 30, 45 and 60 degrees don't appear in the solutions. This is rather intuitive because of the fact that when the body starts with vertically aligned, the gravity force acting on the trunk is less likely to affect towards rotating it around the leg joint. Similar logic can be used to justify the fixed points with body angle nearly -90 degrees.

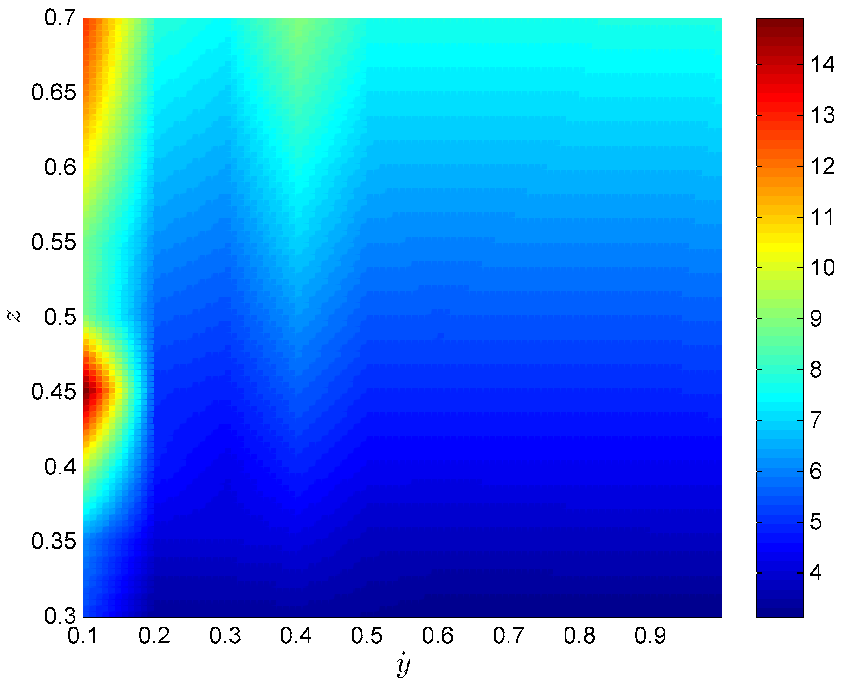

Figure 4.9: Maximum of the absolute eigenvalues of Jacobian matrix numerically calculated at fixed points, λ_{max}, with respect to changing apex height, z, and horizontal velocity, $\dot{y}$. Apex height is in meters and horizontal velocity is in m/s.

Moreover, Fig. 4.9 shows the maximum absolute eigenvalues of the Jacobian matrix with respect to changing horizontal velocity and apex height, numerically

found to investigate the stability behaviors of our fixed points. According to this plot, eigenvalues are all greater than nearly 3, resulting in unstable behavior.

4.4.3.2 Downward Body Orientation

Simulation results show that for any given apex height-horizontal velocity pair in the range given in equation (4.3), there exists a fixed point with downward body orientation, with nearly body angle, θ_h, is being equal to -90 degrees.

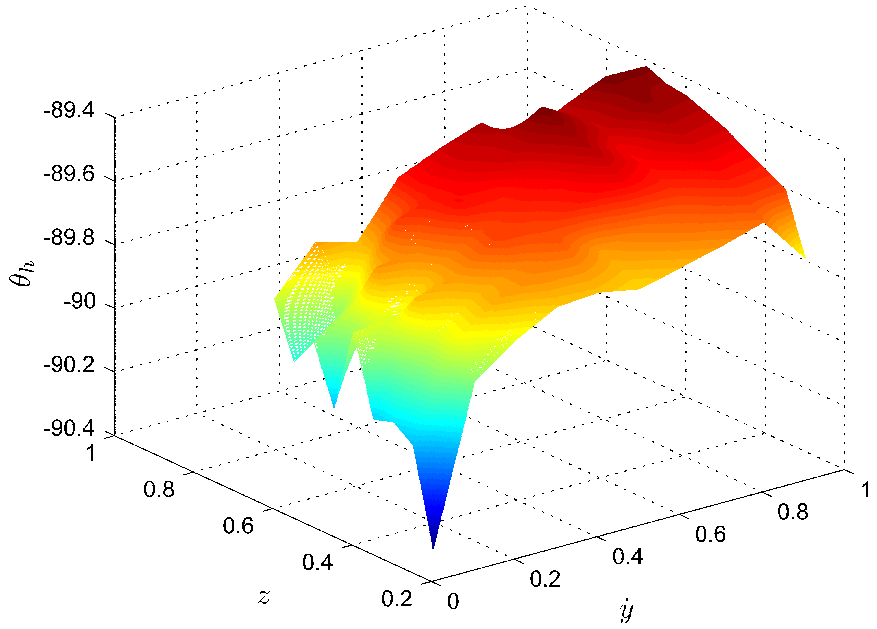

Figure 4.10: Body angle, θ_h, with respect to changing apex height, z, and horizontal velocity, $\dot{y}$. Apex height is in meters, horizontal velocity is in meters/second and body angle is in degrees.

Fig. 4.10 shows the body angle versus horizontal velocity and apex height. According to this plot, we can see that the trunk orientation, θ_h, is nearly -90 degrees in all fixed points; and the maximum deviation on the fixed point body angles is 0.5 degrees, which means the body angle does not deviate much from -90 degrees, while $\dot{y}$ and z change in the specified domain.

Fig. 4.11 examines the stability property of these fixed points, calculating numerical Jacobian and finding the eigenvalues with maximum absolute value

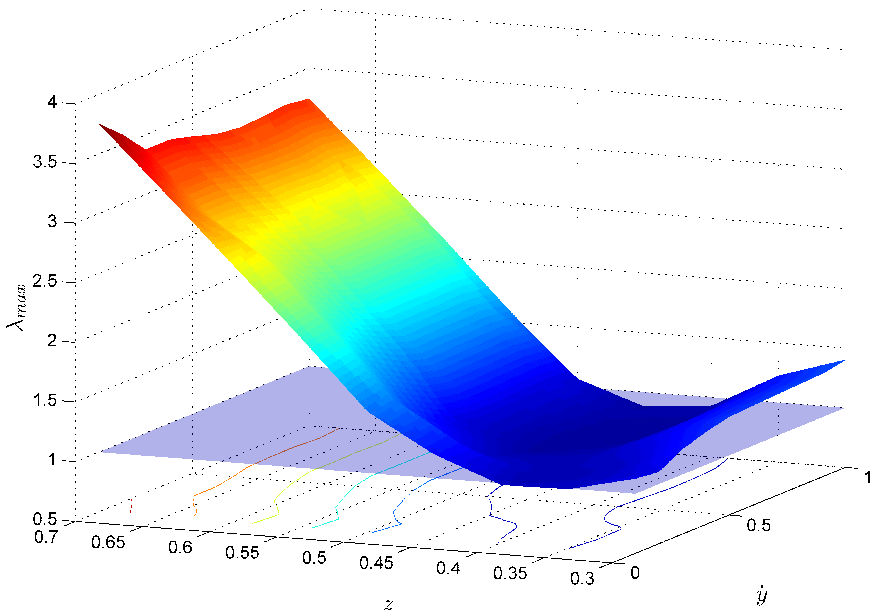

Figure 4.11: Maximum of the absolute eigenvalues of jacobian matrix numerically calculated at fixed points, λ_{max}, with respect to changing apex height, z, and horizontal velocity, $\dot{y}$. Apex height is in meters and horizontal velocity is in m/s. The dark blue corresponds to eigenvalues with magnitude less than 1, which are stable fixed points.

and comparing them with 1 for each apex height and horizontal velocity pair. It seems that stable fixed points form a continuous region on the $\dot{y} - z$ plane. Next subsection investigates some of the properties of this region in a deeper manner.

4.5 Stability Region

In the previous sections, it was numerically shown that for upward body orientation, there were no stable fixed points; however, for downward body orientation, there seems we have a stable region of points in $\dot{y} - z$ plane. To further investigate this region, the following color map could be used.

According to Fig. 4.12, the blue regions indicate stability and as it goes to red regions, fixed points become more unstable. According to this plot, we have our fixed points mostly in the regions where z is in the interval [0.35, 0.45] meters.

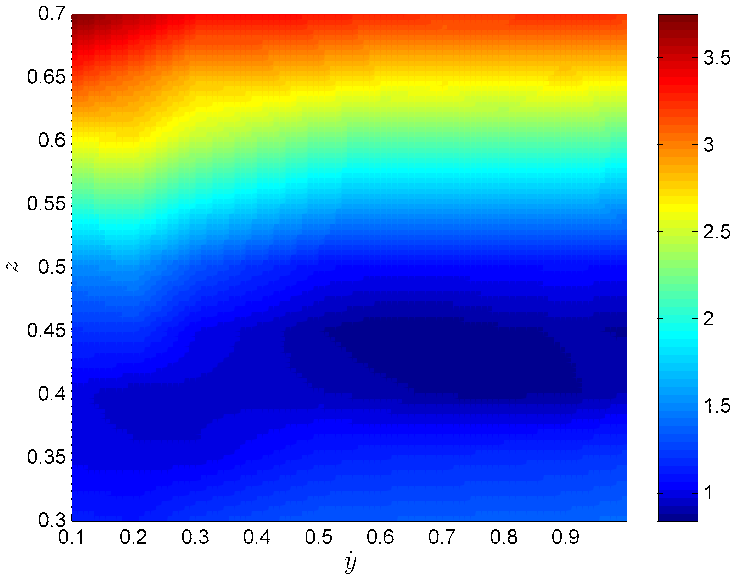

Figure 4.12: A color map version of eigenvalues calculated in Fig. 4.11. This 2D version plot is useful for better investigating the stability region of fixed points in the $\dot{y} - z$ plane.

When apex height, z, is 0.4 all horizontal velocities, result in stable periodic motion with proper control inputs. When apex height increases, z becomes 0.45, horizontal velocities bigger than 0.4 result in stable locomotion; and when apex height decreases, z becomes 0.35, horizontal velocities smaller than 0.4 results in stable locomotion. In order to support this argument, randomly chosen 150 points in this region are taken into optimization shown in Fig. 4.7, and their stability properties are investigated, using eigenvalue analysis of the numerical Jacobian matrix. Randomized search of these stable points resulted in all success, i.e. all turned out to be stable. Therefore, the region shown in the colormap Fig. 4.12, is supported by randomized trials.

Besides stable fixed points, there exist a region in $\dot{y} - z$ plane such that a point in this region does not necessarily need to be a fixed point, but, its control target can be adjusted as the nearest stable fixed point. In other words, a robot configuration for a given horizontal velocity and apex height, we have an unstable fixed point; and by applying the control parameters of the nearest stable fixed point, the robot can locomote with different velocity and apex height, but in a stable manner.

For instance, in Fig. 4.13, the point shown with red cross is an unstable fixed point. Applying its control inputs found from the optimization leads to falling, meaning $\dot{y} = 0$ and $z = 0$ in the plot.

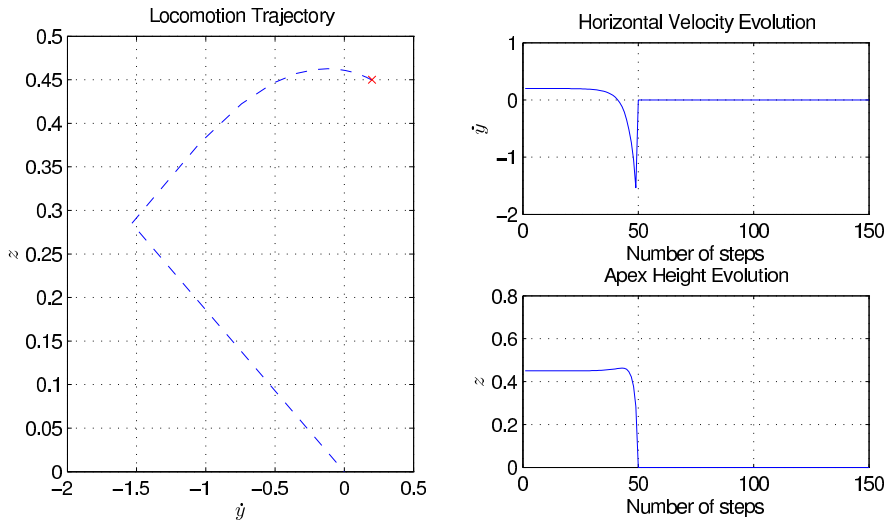

Figure 4.13: In the left figure, the red cross represents an unstable fixed point, and the blue dashed curve represents $\dot{y} - z$ trajectory under its corresponding control inputs found from the optimization. The trajectory ends at (0,0) point which corresponds to falling. In the right figure, horizontal velocity and apex heights are plotted with respect to number of steps.

However, by applying the control inputs of the nearest stable fixed point, the same unstable point converges to the stable one and continues moving forward. In Fig. 4.14, the point shown with red cross is an unstable fixed point and the point shown with blue circle is a stable fixed point, both found from the optimization. Applying the control inputs associated with the circular point, robot can locomote stably at the desired velocity and apex height as illustrated.

Example given in Fig. 4.14 illustrates that some unstable fixed points can converge to nearby stable fixed points by applying the right control parameters. However, that may not be possible for every unstable and stable fixed point pairs. The reason of convergence illustrated in Fig. 4.14 is that the unstable fixed point projected onto the control parameters' plane of the stable fixed point is in the domain of attraction of the stable fixed point. The domain of attraction of these fixed points is a high dimensional region and cannot be visualized easily. Therefore, projections onto some useful and understandable lower dimensional

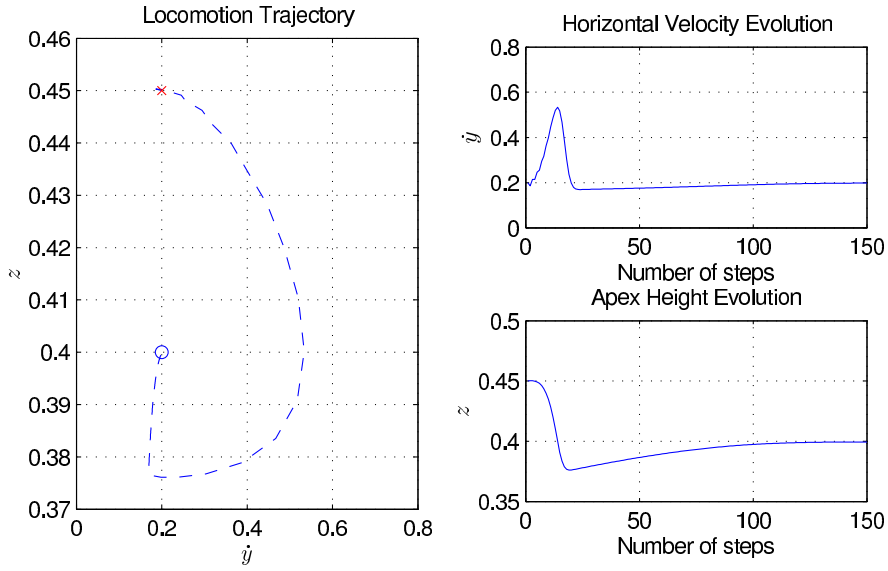

Figure 4.14: In the left figure, the red cross represents an unstable fixed point, blue circular point represents a stable fixed point, and the blue dashed curve represents $\dot{y} - z$ trajectory under the stable point's control inputs found from the optimization. The unstable fixed point converges to stable one after some number of steps. In the right figure, horizontal velocity and apex heights are plotted with respect to number of steps.

spaces should be used, to get a better intuition. Fig. 4.15 illustrates the projection of the domain of attraction of the fixed point given by blue circle in the figure to the horizontal velocity - apex height plane. To be short, let us define these regions as *convergence regions*. So, if a point is inside the convergence region of one stable fixed point, we can apply the control input associated with the stable fixed point and make the particular point converge to the stable fixed point.

In order to generalize, it's better to show all stable fixed points' convergence regions in one plot. Fig. 4.16 shows the union of all these convergence regions. Here, blue circles represent all of the stable fixed points found from the optimization given in section 4.4.2, and the blue solid line is the bound for the total unified convergence region. This figure indicates that if our initial apex state's horizontal velocity and apex height coordinate pair is in this region, it's guaranteed that this state can be made to converge to one of the stable fixed point configurations represented by blue circles. And note that the convergence region found in Fig. 4.15 is a subset of this unified region.

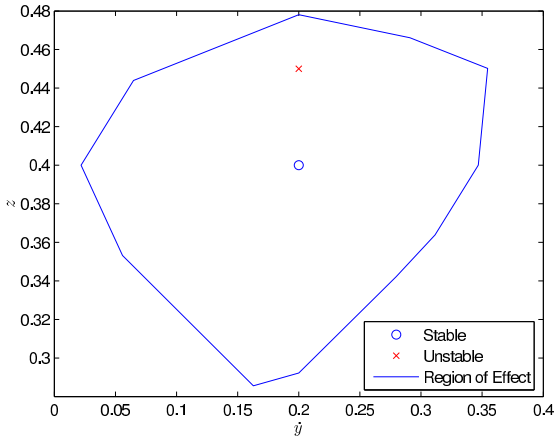

Figure 4.15: The red cross represents the unstable fixed point, blue circular point represents a stable fixed point, and the blue solid curve is the bound for stability effect region of the stable fixed point.

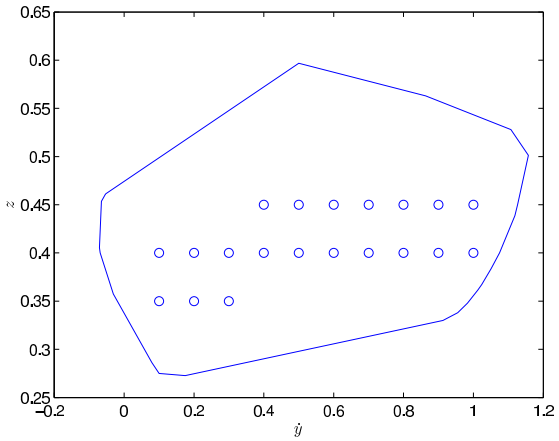

Figure 4.16: Union of convergence regions of all the stable fixed points.

In the light of the findings presented in this section if body orientation is chosen as downward, we can apply control inputs to many of the $\dot{y} - z$ pairs for stable and periodical locomotion pattern, resulting in a fixed point space with dimension at least 2.

4.6 Fixed Point Manifold Dimension

From the simulations shown up to here, we cannot conclude with certainty but it can be inferred that the fixed point manifold dimension is at least 2, because

we have numerical evidence that the stable fixed points form a two dimensional region in $\dot{y} - z$ plane as shown in section 4.5. In order to better understand the fixed point space and investigate its dimension, it's necessary to introduce a third variable in our grid search. In this section, it will be important to check whether the dimension could be three or not. A necessary condition for a three dimensional space is that it must constitute a volume. If the fixed point space has a volume, then for a given fixed point with $\dot{y}$ and z, we must find a continuous region of body angles, θ_h, such that they all are again fixed points. In order to check this concept, θ_h is swept all over the range $[-\pi, \pi]$ for a chosen fixed point. In Fig. 4.17, resulting error versus body angle, θ_h is plotted.

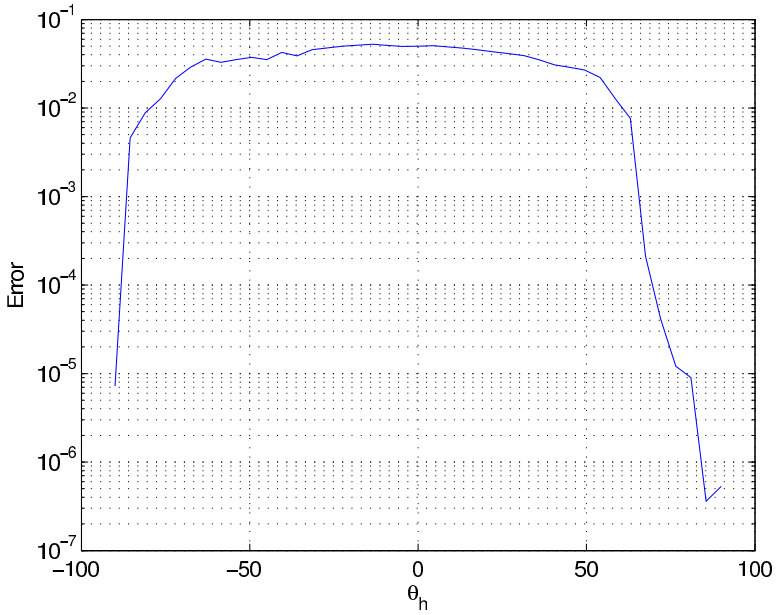

Figure 4.17: For $\dot{y} = 0.9$, $z = 0.45$, and changing θ_h, optimization converges to nearby stable points but different $\dot{y} - z$ pairs.

Here, as we expect, the error is minimum when the body angle is nearly near $\pi/2$ or $-\pi/2$. Therefore, based on numerical evidence, it's not guaranteed to find control parameters for any given ($\dot{y}$, z, θ_h) triple, to make it a fixed point of the system. However, we might consider finding fixed points for any given ($\dot{y}$, z) and θ_h is being in some small continuous interval. In order to investigate this possibility, a small scale search on the body angle with step size $\pi/3200$ is performed near a chosen stable fixed point.

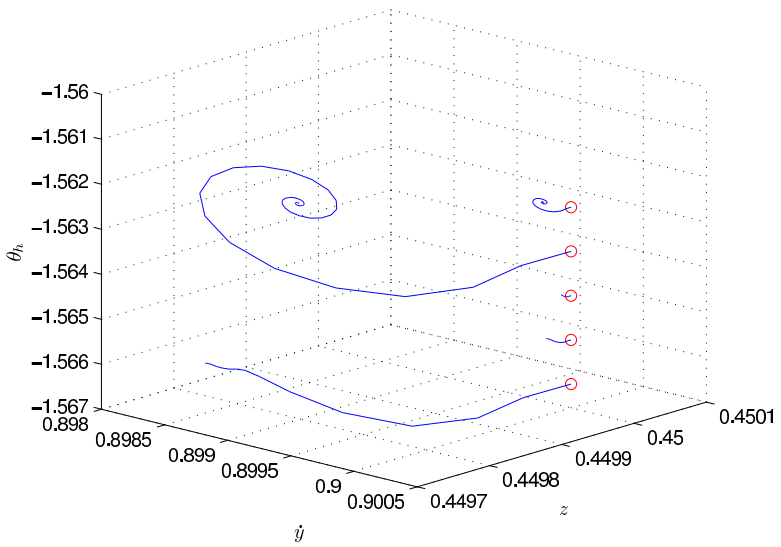

Figure 4.18: For $\dot{y} = 0.9$, $z = 0.45$, and changing θ_h, optimization converges to nearby stable points but with different $\dot{y} - z$ pairs.

As shown in Fig. 4.18, for different but close θ_h, there exist fixed points because the error between two consecutive apex states are very small. However, when looked at these points' vector fields, they converge to different but close $\dot{y}$ and z pairs as explicitly shown in Fig. 4.18. This implies a small perturbation in the body angle results in small perturbations in $\dot{y}$ and z pairs, which does not indicate a volume but may be a surface in $\dot{y}$, z, θ_h space because changing body angle results in changes in other variables, which may indicate a relationship between them. In other words, these three variables may depend on each other in some manner, and therefore there is not enough evidence to say that the fixed point manifold is a three dimensional space, however it's supported by lots of numerical evidence that it's greater than or equal to 2.

4.7 Time-Varying Control Strategies for Unstable Fixed Points

4.7.1 Motivation

In the previous sections, all of our focus was on the stable fixed points with downward body orientation. We found that many of the downward oriented robot configurations could result in stable periodic locomotion by applying the right control inputs; however unstable fixed points with upward body orientation still remains as a problem. According to simulations in the preceding sections made for these points, it can be inferred that a constant parameter torque model may not be enough for upright locomotion. Therefore, it would be useful to investigate time varying control inputs. Basically, our upright fixed points are unstable even with small numerical errors such as 10^{-12} are present between consecutive apex states. Since these points are unstable, the robot falls if it starts moving with one of these configurations after some number of steps (sometimes less than 10 steps). Here, the number of steps is directly correlated with the maximum absolute eigenvalue being close to 1 or not. These unstable fixed points were found by using our optimization procedure described in section 4.3.2. The idea is using this algorithm iteratively at each step to find the control parameters that best fit to the current state, as illustrated in Fig. 4.19.

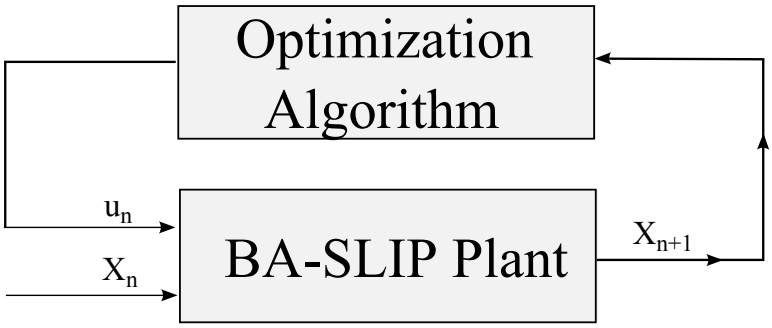

Figure 4.19: Time varying control algorithm. The control input u_n is calculated at each apex state, again.

As depicted in the above figure, this time the control input vector is not constant, and real time parameter adjustments are made possible. The control input to be applied is calculated using the current apex state, but from optimization. In other words, let initial apex configuration be $\mathbf{X_1}$, and by applying control input, $\mathbf{u_1}$, next apex configuration, $\mathbf{X_2}$, is obtained with very small error. In order to decrease the next step error, $\mathbf{X_2}$, is given to our optimization and a new control input, $\mathbf{u_2}$ is found for minimizing the error between $\mathbf{X_2}$ and $\mathbf{X_3}$, and the algorithm continues similarly. This work requires much computation, because at each step a new optimization begins in a 7 dimensional space. However, these results are important for building some intuition on controller templates we can implement in future.

4.7.2 Simulation Results

The time varying strategy described in the previous subsection is applied to some of the unstable fixed points and resulted in stable motion for large number of steps as shown in Fig. 4.20.

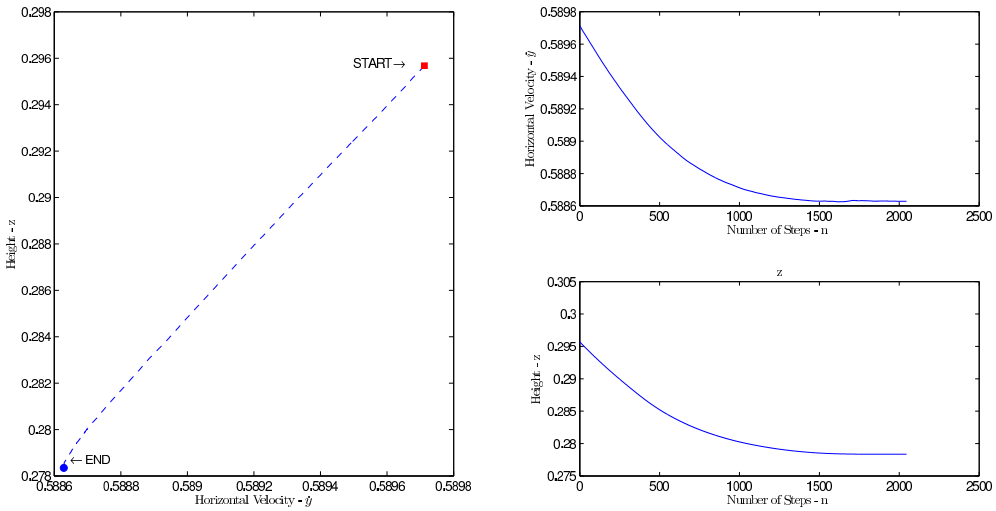

Figure 4.20: Right figures represent horizontal velocity, $\dot{y}$, and apex height, z, changes with respect to number of steps, n, under time varying control. The left figure is the 2D trajectory of this change.

Fig. 4.20 indicates that under time varying control, the robot model can hop

over 2000 steps, and can continue its locomotion. In the end of its motion, the error between two consecutive steps becomes nearly 10^{-10}. However, applying the parameters found for the 2048^{th} step does not resolve unstability problem, because when this parameters are applied, the robot hops at most 23 times as depicted in Fig. 4.21, and then again falls. Reason for this is the unstability behavior of the fixed point configuration under constant control template we applied throughout this thesis. These results support the need for time varying and state adaptive control templates, that the parameters can slightly change in order to stably move.

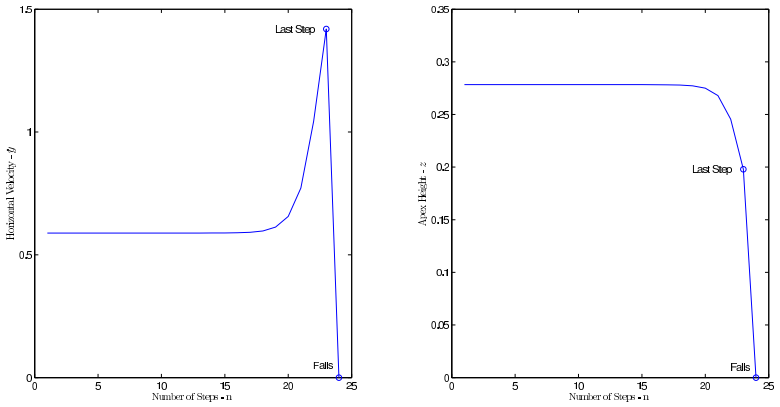

Figure 4.21: The apex configuration at $\mathbf{X_{2048}}$ is applied to control input found $\mathbf{u_{2048}}$, and the robot falls after 23^{rd} step, i.e. at apex configuration $\mathbf{X_{2061}}$.

While the robot converges to its steady state target as depicted in Fig. 4.20, the parameters change slowly in each step, forming curves that look like continuous functions. The parameter changes of this 2048 step movement are shown in Fig. 4.22.

To sum up, comparing Fig. 4.20 and Fig. 4.21 indicates the need for time varying control inputs; and Fig. 4.22 supports the idea that control parameters may have underlying functional forms that can be exploited to guess the next control input to be applied. However, the main problem here is calculating the next step control input in a reasonable amount of time since the robot will need to move fast, and the optimization algorithm performs this job very slowly. To be able to do that, a mathematical study must be performed over the change of parameters of many unstable fixed points, and this work is left for future studies.

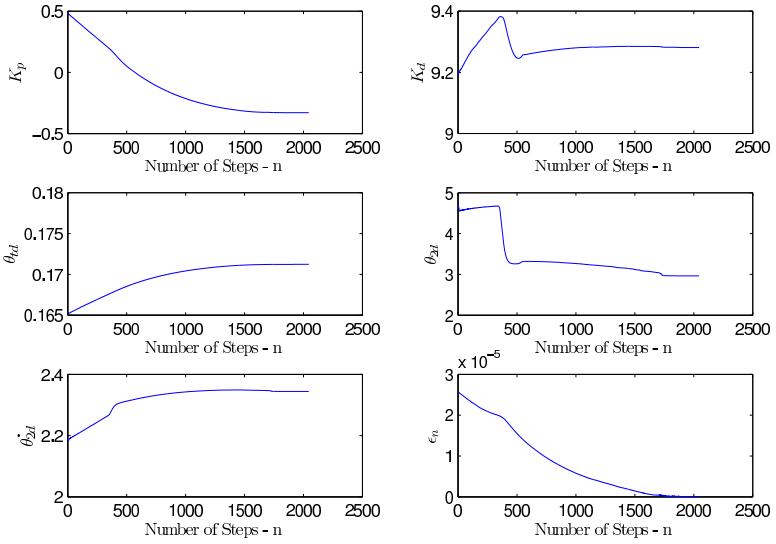

Figure 4.22: Torque parameters, $K_p, K_d, \theta_{td}, \theta_{2d}, \dot{\theta}_{2d}$ and error ϵ_n, with respect to number of steps, n, under time varying control, respectively from left to right and up to down.

Chapter 5

CONCLUSION

In this thesis, we performed a different mathematical analysis on the model, Body-Attached Spring-Loaded Inverted Pendulum (BA-SLIP). Based on this analysis we considered some control strategies in order to reveal the periodic motion behaviors behind this locomotion. In the controller design, due to complex and nonlinear characteristics of the model dynamics, we focused on simulative approaches, which shows some numerical evidence of stable and periodic running.

Periodic motion in such a system corresponds to a fixed point of a special map, namely the apex-return map, associated with the locomotion dynamics. Under this knowledge, we performed extensive simulation studies on finding the fixed points of the system for any given horizontal velocity and apex height pair in some reasonable range. The fixed points found were categorized into two subtitles: fixed points with upward body orientation and fixed points with downward body orientation. There is enough numerical evidence which allows us to assume that these fixed points form a region in a high dimensional space, namely manifold, and we investigated the dimension of it by presenting some numerical evidence.

In addition, after finding the fixed points of our apex-return map, their stability properties are investigated using numerical Jacobian. The fixed points with upward body orientation were found to be unstable under our control strategy. However, some of the points with downward body orientation were stable. We

tried to find the basins of attraction of these stable fixed points and projected the regions into horizontal velocity - apex height plane, therefore constructed a unified region which involves some of horizontal velocity - apex height pairs that can be stabilized under corresponding control inputs.

Finally, we performed simulations on some of the fixed points with upward body orientation that were unstable in our proposed control scheme, to investigate their locomotive patterns. We presented some numerical results showing that the upward body oriented fixed points can be stabilized using time-varying control inputs. These results may stimulate more discussion on adaptive and time varying control strategies, and can be a basis for future implementations and ideas, in order to mimic upright locomotion.

Bibliography

[1] P. Holmes, R. Full, D. Koditschek, and J. Guckenheimer, "The dynamics of legged locomotion: Models, analyses, and challenges," *SIAM Review*, vol. 48, pp. 207–304, May 2006.

[2] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. A. Rizzi, and M. Raibert, "Autonomous navigation for BigDog," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Anchorage, Alaska), May 3-8 2010.

[3] R. J. Full and D. E. Koditschek, "Templates and anchors: Neuromechanical hypotheses of legged locomotion," *Journal of Experimental Biology*, vol. 202, pp. 3325–3332, 1999.

[4] R. M. Alexander and A. S. Jayes, "Vertical movement in walking and running," *Journal of Zoology, London*, vol. 185, pp. 27–40, 1978.

[5] R. M. Alexander, "Three uses for springs in legged locomotion," *International Journal of Robotics Research*, vol. 9, no. 2, pp. 53–61, 1990.

[6] R. Blickhan and R. J. Full, "Similarity in multilegged locomotion: Bouncing like a monopode," *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, vol. 173, no. 5, pp. 509–517, 1993.

[7] C. T. Farley and D. P. Ferris, "Biomechanics of walking and running: Center of mass movements to muscle action," *Exercise and Sport Science Rev.*, vol. 26, pp. 253–283, 1998.

[8] M. H. Raibert, *Legged Robots That Balance*. Cambridge, MA, USA: MIT Press, 1986.

[9] P. Gregorio, M. Ahmadi, and M. Buehler, "Design, control, and energetics of an electrically actuated legged robot," *Transactions on Systems, Man, and Cybernetics*, vol. 27, pp. 626–634, August 1997.

[10] G. Zeglin, *The Bow Leg Hopping Robot*. PhD Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, October 1999.

[11] J. W. Hurst, J. Chestnutt, and A. Rizzi, "Design and philosophy of the bimasc, a highly dynamic biped," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Roma, Italy), April 10-14 2007.

[12] W. J. Schwind, *Spring Loaded Inverted Pendulum Running: A Plant Model*. PhD Thesis, University of Michigan, Ann Arbor, MI, USA, 1998.

[13] T. Weaver and R. Klein, "The evolution of human walking," in *In Human Walking*, Lippincott: Williams & Wilkins, 2006.

[14] E. D. Brul, "The general phenomenon of bipedalism," *American Zoologist*, vol. 2, no. 2, pp. 205–208, 1962.

[15] H. Pontzer, D. Raichlen, and M. Sockol, "The metabolic cost of walking in humans, chimpanzees, and early hominins," *Journal of Human Evolution*, vol. 56, no. 1, pp. 43–54, 2009.

[16] W. Wang and R. Crompton, "The role of load-carrying in the evolution of modern body proportions," *Journal of Anatomy*, vol. 204, no. 5, pp. 417–430, 2004.

[17] W. Wang and R. Crompton, "Energy transformation during erect and 'bent-hip, bent-knee' walking by humans with implications for the evolution of bipedalism," *Journal of Human Evolution*, vol. 44, no. 5, pp. 563–579, 2003.

[18] M. Horst-Moritz, *Towards understanding human locomotion*. PhD Thesis, Technische Universität Ilmenau, Germany, 2013.

[19] H.-M. Maus, J. Rummel, and A. Seyfarth, "Stable upright walking and running using a simple pendulum based control scheme," in *International Conference on Climbing and Walking Robots*, (Coimbra, Portugal), September 08-10 2008.

[20] H. M. Maus, S. W. Lipfert, M. Gross, J. Rummel, and A. Seyfarth, "Upright human gait did not provide a major mechanical challenge for our ancestors," *Nature Communications*, vol. 1, no. 6, p. 70, 2010.

[21] I. Uyanık, *Adaptive Control of a one-legged hopping robot through dynamically embedded spring-loaded inverted pendulum template.* M.S. Thesis, Bilkent University, Ankara, Turkey, August 2011.

[22] P. Holmes, "Poincaré, Celestial Mechanics, Dynamical-Systems Theory and "Chaos"," *Physics Reports (Review Sect. of Physics Lett.)*, vol. 193, pp. 137–163, September 1990.

[23] H. Geyer, A. Seyfarth, and R. Blickhan, "Spring-mass running: Simple approximate solution and application to gait stability," *Journal of Theoretical Biology*, vol. 232, pp. 315–328, February 2005.

[24] M. M. Ankaralı, O. Arslan, and U. Saranlı, "An analytical solution to the stance dynamics of passive spring-loaded inverted pendulum with damping," in *12th International Conference on Climbing and Walking Robots and The Support Technologies for Mobile Machines (CLAWAR'09)*, (Istanbul, Turkey), September 2009.

[25] U. Saranlı, *Dynamic Locomotion with a Hexapod Robot.* PhD Thesis, The University of Michigan, Ann Arbor, MI, USA, September 2002.

[26] O. Arslan, U. Saranlı, and O. Morgül, "An Aproximate Stance Map of The Spring Mass Hopper with Gravity Correction For Nonsymmetric Locomotions," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Kobe, Japan), May 2009.

[27] N. Neville, M. Buehler, and I. Sharf, "A bipedal running robot with one actuator per leg," in *IEEE International Conference on Robotics and Automation*, (Orlando, USA), May 2006.

[28] I. Poulakakis and J. W. Grizzle, "Monopedal running control: Slip embedding and virtual constraint controllers," in *IROS*, pp. 323–330, 2007.

[29] J. Rummel and A. Seyfarth, "Passive stabilization of the trunk in walking," in *Proceedings of the International Conference on Simulation, Modeling and Programming for Autonomous Robots 2010 Workshops*, (Darmstadt, Germany), pp. 127–136, November 15-16 2010.

[30] M. Wisse, D. G. E. Hobbelen, and A. L. Schwab, "Adding an upper body to passive dynamic walking robots by means of a bisecting hip mechanism," *Transactions on Robotics*, vol. 23, pp. 112–123, February 2007.

[31] D. G. E. Hobbelen and M. Wisse, "A Disturbance Rejection Measure for Limit Cycle Walkers: The Gait Sensitivity Norm," *IEEE Transactions on Robotics*, vol. 23, pp. 1213–1224, Dec. 2007.

[32] D. Owaki, M. Koyama, S. Yamaguchi, S. Kubo, and A. Ishiguro, "A two-dimensional passive dynamic running biped with knees," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5237–5242, 2010.

[33] A. Kuo, "The six determinants of gait and the inverted pendulum analogy: A dynamic walking perspective," *Human Movement Science*, vol. 26, pp. 617–656, Aug. 2007.

[34] M. Ahmadi and M. Buehler, "Controlled passive dynamic running experiments with the ARL-Monopod II," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 974–986, 2006.

[35] M. Sharbafi, C. Maufroy, H. Maus, A. Seyfarth, M. Ahmadabadi, and M.-J. Yazdanpanah, "Controllers for robust hopping with upright trunk based on the virtual pendulum concept," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2222–2227, 2012.

[36] A. Seyfarth, H. Geyer, M. Gunther, and R. Blickhan, "A movement criterion for running," *Journal of Biomechanics*, vol. 35, no. 5, pp. 649–655, 2002.

[37] C. Maufroy, H. Maus, and A. Seyfarth, "Simplified control of upright walking by exploring asymmetric gaits induced by leg damping," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 491–496, 2011.

[38] H. Geyer, A. Seyfarth, and R. Blickhan, "Compliant leg behaviour explains basic dynamics of walking and running," *Proceedings. Biological sciences / The Royal Society*, vol. 273, pp. 2861–2867, November 2006.

[39] M. H. Raibert, H. B. Brown, E. Hastings, J. Koechling, K. N. Murphy, S. S. Murthy, and A. J. Stentz, "Dynamically stable legged locomotion," tech. rep., MIT Artificial Intelligence Laboratory, 1989.

[40] F. Peuker, C. Maufroy, and A. Seyfarth, "Leg-adjustment strategies for stable running in three dimensions," *Bioinspiration and Biomimetics*, vol. 7, no. 3, p. 036002, 2012.

[41] M. A. Sharbafi, C. Maufroy, M. N. Ahmadabadi, M. J. Yazdanpanah, and A. Seyfarth, "Robust hopping based on virtual pendulum posture control," *Bioinspiration and Biomimetics*, vol. 8, no. 3, p. 036002, 2013.

[42] S. Riese and A. Seyfarth, "Stance leg control: variation of leg parameters supports stable hopping," *Bioinspiration and Biomimetics*, vol. 7, no. 1, p. 016006, 2012.

[43] J. Schmitt and J. Clark, "Modeling posture-dependent leg actuation in sagittal plane locomotion," *Bioinspiration and Biomimetics*, vol. 4, no. 4, p. 046005, 2009.

[44] H. B and C. M, "Safe control of hopping in uneven terrain," *J. Dyn. Syst. Meas. Control*, vol. 131, no. 1, 2009.

[45] I. Poulakakis and J. W. Grizzle, "The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper," *IEEE Trans. Automat. Contr.*, vol. 54, no. 8, pp. 1779–1793, 2009.

[46] M. M. Ankaralı and U. Saranlı, "Stride-to-stride energy regulation for robust self-stability of a torque-actuated dissipative spring-mass hopper," *Chaos:*

An Interdisciplinary Journal of Nonlinear Science, vol. 20, no. 3, p. 033121, 2010.

[47] G. Zeglin and H. B. Brown, "Control of a Bow Leg Hopping Robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Leuven, Belgium), May 1998.

[48] H. Yu, M. Li, W. Guo, and H. Cai, "Stance Control of the SLIP Hopper with Adjustable Stiffness of Leg Spring," in *Proceedings of 2012 IEEE International Conference on Mechatronics and Automation*, (Chengdu, China), August 2012.

[49] O. Arslan, *Model Based Methods for the Control and Planning of Running Robots*. M.S. Thesis, Bilkent University, Ankara, Turkey, July 2009.

[50] I. Poulakakis, E. Papadopoulos, and M. Buehler, "On the stability of the passive dynamics of quadrupedal running with a bounding gait," *The International Journal of Robotics Research*, vol. 25, no. 7, pp. 669–687, 2006.

[51] U. Saranlı, M. Buehler, and D. E. Koditschek, "RHex: A simple and highly mobile robot," *The International Journal of Robotics Research*, vol. 20, pp. 616–631, July 2001.

[52] A. Sato and M. Buehler, "A planar hopping robot with one actuator: Design, simulation, and experimental results," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 4, (Sendai, Japan), 2004.

[53] N. Cherouvim and E. Papadopoulos, "Control of Hopping Speed and Height Over Unkown Rough Terrain Using a Single Actuator," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (Kobe, Japan), May 2009.

[54] R. H. Byrd, M. E. Hribar, and J. Nocedal, "An interior point algorithm for large scale nonlinear programming," *SIAM Journal on Optimization*, vol. 9, pp. 877–900, 1997.

[55] K. Schittkowski, "Nlpql: A fortran subroutine solving constrained nonlinear programming problems," *Annals of Operations Research*, vol. 5, no. 2, pp. 485–500, 1986.

[56] G. A. Cavagna, P. A. Willems, F. P., and C. Detrembleur, "The two power limits conditioning step frequency in human running," *Journal of Physiology*, vol. 487, pp. 95–108, June 1991.

[57] D. Webb, "Maximum walking speed and lower limb length in hominids," *American Journal of Physical Anthropology*, vol. 101, pp. 515–525, December 1998.

[58] A. Thorstensson and H. Roberthson, "Adaptations to changing speed in human locomotion: speed of transition between walking and running," *Acta Physiologica Scandinavica*, vol. 131, no. 2, pp. 211–214, 1987.

Appendix A

Derivation Of Flight Dynamics

The Lagrangian of the system in flight coordinates is

$$L = T - V = \frac{1}{2}m\dot{y}^2 + \frac{1}{2}m\dot{z}^2 + \frac{1}{2}I\dot{\theta}_h^2 - mgz. \tag{A.1}$$

Recall that the Euler-Lagrange equation is

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_{q_i}, \tag{A.2}$$

where q_i is the coordinate components and τ_{q_i} is the torque related with coordinate q_i. For $q_i = y$, the first component of Euler-Lagrange equation is

$$\frac{\partial L}{\partial \dot{y}} = m\dot{y}. \tag{A.3}$$

Taking time derivative of (A.3) gives

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{y}}\right) = m\ddot{y}. \tag{A.4}$$

The second component is

$$\frac{\partial L}{\partial y} = 0. \tag{A.5}$$

Substituting (A.3), (A.4) and (A.5) into (A.2) gives the first equation of motion where $\tau_y = 0$:

$$\ddot{y} = 0. \tag{A.6}$$

Now, for $q_i = z$, the first component of Euler-Lagrange equation is

$$\frac{\partial L}{\partial \dot{z}} = m\dot{z}. \tag{A.7}$$

Taking time derivative of (A.7) gives

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{z}}\right) = m\ddot{z}. \tag{A.8}$$

The second component is

$$\frac{\partial L}{\partial z} = -mg. \tag{A.9}$$

Substituting (A.7), (A.8) and (A.9) into (A.2) gives the second equation of motion where $\tau_z = 0$:

$$\ddot{z} = -g. \tag{A.10}$$

Finally, letting $q_i = \theta_h$

$$\frac{\partial L}{\partial \dot{\theta}_h} = I\dot{\theta}_h. \tag{A.11}$$

Taking time derivative of (A.11) gives

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_h}\right) = I\ddot{\theta}_h. \tag{A.12}$$

The second component is

$$\frac{\partial L}{\partial \theta_h} = 0. \tag{A.13}$$

Substituting (A.11), (A.12) and (A.13) into (A.2) gives the third equation of motion where $\tau_{\theta_h} = 0$:

$$\ddot{\theta}_h = 0. \tag{A.14}$$

Combining (A.6), (A.10), and (A.14), gives the equations in closed form as

$$\begin{bmatrix} \ddot{y} \\ \ddot{z} \\ \ddot{\theta}_h \end{bmatrix} = \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix}. \tag{A.15}$$

Appendix B

Derivation Of Stance Dynamics

Cartesian coordinates can be written in terms of polar coordinates as shown below:

$$\begin{aligned} y &= y_f - \rho \sin\theta_1 + d\sin(\theta_1+\theta_2) \\ z &= z_f + \rho\cos\theta_1 - d\cos(\theta_1+\theta_2) \\ \theta_h &= \theta_1 + \theta_2 - \pi/2. \end{aligned} \tag{B.1}$$

Taking derivative of both sides yield

$$\begin{aligned} \dot{y} &= -\dot{\rho}\sin\theta_1 - \rho\cos\theta_1\dot{\theta}_1 + d\cos(\theta_1+\theta_2)(\dot{\theta}_1+\dot{\theta}_2) \\ \dot{z} &= \dot{\rho}\cos\theta_1 - \rho\sin\theta_1\dot{\theta}_1 + d\sin(\theta_1+\theta_2)(\dot{\theta}_1+\dot{\theta}_2) \\ \dot{\theta}_h &= \dot{\theta}_1 + \dot{\theta}_2. \end{aligned} \tag{B.2}$$

Squaring $\dot{y}$ and $\dot{z}$ gives

$$\begin{aligned} \dot{y}^2 =& \dot{\rho}^2\sin^2\theta_1 + \rho^2\cos^2\theta_1\dot{\theta_1}^2 + d^2(\dot{\theta}_1+\dot{\theta}_2)^2\cos^2(\theta_1+\theta_2) \\ &+ 2\rho\dot{\rho}\dot{\theta_1}\sin\theta_1\cos\theta_1 - 2d\cos(\theta_1+\theta_2)\sin\theta_1\dot{\rho}(\dot{\theta}_1+\dot{\theta}_2) \\ &- 2\rho d\cos\theta_1\cos(\theta_1+\theta_2)(\dot{\theta}_1+\dot{\theta}_2)\dot{\theta}_1, \end{aligned} \tag{B.3}$$

$$\begin{aligned} \dot{z}^2 =& \dot{\rho}^2\cos^2\theta_1 + \rho^2\sin^2\theta_1\dot{\theta_1}^2 + d^2(\dot{\theta}_1+\dot{\theta}_2)^2\sin^2(\theta_1+\theta_2) \\ &- 2\rho\dot{\rho}\dot{\theta_1}\sin\theta_1\cos\theta_1 + 2d\sin(\theta_1+\theta_2)\cos\theta_1\dot{\rho}(\dot{\theta}_1+\dot{\theta}_2) \\ &- 2\rho d\sin\theta_1\sin(\theta_1+\theta_2)(\dot{\theta}_1+\dot{\theta}_2)\dot{\theta}_1. \end{aligned} \tag{B.4}$$

Then, their sum is

$$\begin{aligned}\dot{y}^2 + \dot{z}^2 =& \dot{\rho}^2 + \rho^2 \dot{\theta_1}^2 + d^2(\dot{\theta}_1 + \dot{\theta}_2)^2 + 2d\dot{\rho}(\dot{\theta}_1 + \dot{\theta}_2)\sin\theta_2 \\ & - 2\rho d\dot{\theta_1}(\dot{\theta}_1 + \dot{\theta}_2)\cos\theta_2.\end{aligned} \tag{B.5}$$

The Lagrangian of the system is

$$L = T - V = \frac{1}{2}m\dot{y}^2 + \frac{1}{2}m\dot{z}^2 + \frac{1}{2}I(\dot{\theta}_1 + \dot{\theta}_2)^2 - mgz - \frac{1}{2}k(\rho - \rho_0)^2. \tag{B.6}$$

This equation can be written in polar coordinates as follows using equation (B.5):

$$\begin{aligned}L = \frac{1}{2}m\Big(\dot{\rho}^2 + \rho^2\dot{\theta}_1^2 + d^2(\dot{\theta}_1 + \dot{\theta}_2)^2 + 2d(\dot{\theta}_1 + \dot{\theta}_2)(\dot{\rho}\sin\theta_2 - \rho\dot{\theta}_1\cos\theta_2)\Big) \\ + \frac{1}{2}I(\dot{\theta}_1 + \dot{\theta}_2)^2 - mg\rho\cos\theta_1 + mgd\cos(\theta_1 + \theta_2) - \frac{1}{2}k(\rho - \rho_0)^2.\end{aligned} \tag{B.7}$$

Recall that the Euler-Lagrange equation is

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_{q_i}, \tag{B.8}$$

where q_i is the coordinate components and τ_{q_i} is the torque related with coordinate q_i.

For $q_i = \rho$, the first component of Euler-Lagrange equation is

$$\frac{\partial L}{\partial \dot{\rho}} = \frac{1}{2}m\Big[2\dot{\rho} + 2d(\dot{\theta}_1 + \dot{\theta}_2)\sin\theta_2\Big] = m\dot{\rho} + md(\dot{\theta}_1 + \dot{\theta}_2)\sin\theta_2. \tag{B.9}$$

Taking time derivative of (B.9) gives

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\rho}}\right) = m\ddot{\rho} + md(\ddot{\theta}_1 + \ddot{\theta}_2)\sin\theta_2 + md\cos\theta_2(\dot{\theta}_1 + \dot{\theta}_2)\dot{\theta}_2. \tag{B.10}$$

The second component is

$$\frac{\partial L}{\partial \rho} = m\rho\dot{\theta}_1^2 - md\cos\theta_2\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2) - mg\cos\theta_1 - k(\rho - \rho_0). \tag{B.11}$$

Substituting (B.9), (B.10) and (B.11) into (B.8) gives the first equation of motion where $\tau_\rho = 0$:

$$\begin{aligned}m\ddot{\rho} + md(\ddot{\theta}_1 + \ddot{\theta}_2)sin\theta_2 + mdcos\theta_2(\dot{\theta}_1 + \dot{\theta}_2)^2 - m\rho\dot{\theta_1}^2 + \\ mgcos\theta_1 + k(\rho - \rho_0) = 0.\end{aligned} \tag{B.12}$$

Now, we repeat the same procedure for θ_1. Let $q_i = \theta_1$.

$$\begin{aligned}\frac{\partial L}{\partial \dot{\theta}_1} =& m\Big[\rho^2\dot{\theta}_1^2 + d^2(\dot{\theta}_1 + \dot{\theta}_2) + d\sin\theta_2\dot{\rho} - d\rho\cos\theta_2(\dot{\theta}_1 + \dot{\theta}_2) - \\ & d\rho\cos\theta_2\dot{\theta}_1\Big] + I(\dot{\theta}_1 + \dot{\theta}_2).\end{aligned} \tag{B.13}$$

Taking time derivative of (B.13) gives

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) =& m\Big[2\rho\dot{\rho}\dot{\theta}_1 + \rho^2\ddot{\theta}_1 + d^2(\ddot{\theta}_1 + \ddot{\theta}_2) + d\cos\theta_2\dot{\rho}\dot{\theta}_2 \\ & + d\sin\theta_2\ddot{\rho} - d\rho\cos\theta_2(\ddot{\theta}_1 + \ddot{\theta}_2) + d\rho\sin\theta_2\dot{\theta}_2(\dot{\theta}_1 + \dot{\theta}_2) \\ & - d\dot{\rho}\cos\theta_2(\dot{\theta}_1 + \dot{\theta}_2) - d\rho\cos\theta_2\ddot{\theta}_1 + d\rho\sin\theta_2\dot{\theta}_1\dot{\theta}_2 \\ & - d\cos\theta_2\dot{\rho}\dot{\theta}_1\Big] + I(\ddot{\theta}_1 + \ddot{\theta}_2),\end{aligned} \tag{B.14}$$

which yields

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) =& \ddot{\rho}(md\sin\theta_2) + \ddot{\theta}_1(m\rho^2 + md^2 - 2md\cos\theta_2 + I) \\ & + \ddot{\theta}_2(md^2 - md\rho\cos\theta_2 + I) + \dot{\rho}\dot{\theta}_1(2mg - 2md\cos\theta_2) \\ & + \dot{\theta}_1\dot{\theta}_2(2md\rho\sin\theta_2) + \dot{\theta}_2{}^2(md\rho\sin\theta_2).\end{aligned} \tag{B.15}$$

The second component is

$$\frac{\partial L}{\partial \theta_1} = mg\rho\sin\theta_1 - mgd\sin(\theta_1 + \theta_2). \tag{B.16}$$

Substituting (B.13), (B.15) and (B.16) into (B.8) gives the second equation of motion where $\tau_{\theta_1} = 0$ as

$$\begin{aligned}& \ddot{\rho}(md\sin\theta_2) + \ddot{\theta}_1(m\rho^2 + md^2 - 2md\cos\theta_2 + I) \\ & + \ddot{\theta}_2(md^2 - md\rho\cos\theta_2 + I) + \dot{\rho}\dot{\theta}_1(2mg - 2md\cos\theta_2) \\ & + \dot{\theta}_1\dot{\theta}_2(2md\rho\sin\theta_2) + \dot{\theta}_2{}^2(md\rho\sin\theta_2) - mg\rho\sin\theta_1 \\ & + mgd\sin(\theta_1 + \theta_2) = 0.\end{aligned} \tag{B.17}$$

Finally, we repeat the same procedure for θ_2. Let $q_i = \theta_2$.

$$\frac{\partial L}{\partial \dot{\theta}_2} = m\Big[d^2(\dot{\theta}_1 + \dot{\theta}_2) + d\sin\theta_2\dot{\rho} - d\rho\cos\theta_2\dot{\theta}_1\Big] + I(\dot{\theta}_1 + \dot{\theta}_2). \tag{B.18}$$

Taking time derivative of (B.18) gives

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) =&\ddot{\rho}(md\sin\theta_2) + \ddot{\theta}_1(md^2 + I - md\rho\cos\theta_2) + \ddot{\theta}_2(md^2 + I) \\ &- \dot{\rho}\dot{\theta}_1(md\cos\theta_2) + \dot{\rho}\dot{\theta}_2(md\cos\theta_2) + \dot{\theta}_1\dot{\theta}_2(md\rho\sin\theta_2).\end{aligned} \tag{B.19}$$

The second component is

$$\frac{\partial L}{\partial \theta_2} = md\dot{\rho}(\dot{\theta}_1 + \dot{\theta}_2)\cos\theta_2 + md\rho\dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2)\sin\theta_2 - mgd\sin(\theta_1 + \theta_2). \tag{B.20}$$

Substituting (B.18), (B.19) and (B.20) into (B.8) gives the third equation of motion where $\tau_{\theta_2} = \tau$ as

$$\begin{aligned}\tau =&\ddot{\rho}(md\sin\theta_2) + \ddot{\theta}_1(md^2 + I - md\rho\cos\theta_2) + \ddot{\theta}_2(md^2 + I) \\ &- \dot{\rho}\dot{\theta}_1 2md\cos\theta_2 - \dot{\theta}_1^2 md\rho\sin\theta_2 + mgd\sin(\theta_1 + \theta_2).\end{aligned} \tag{B.21}$$

Combining (B.12), (B.17) and (B.21) into matrix form leaves us with the second order dynamical system

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}), \tag{B.22}$$

where $\mathbf{M}$ and $\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}})$ are given as

$$\mathbf{M} = \begin{bmatrix} 1 & d\sin\theta_2 & d\sin\theta_2 \\ d\sin\theta_2 & \rho^2 - 2d\rho\cos\theta_2 + d^2 + I/m & d^2 - d\rho\cos\theta_2 + I/m \\ d\sin\theta_2 & d^2 - d\rho\cos\theta_2 + I/m & d^2 + I/m \end{bmatrix}, \tag{B.23}$$

$$\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}) \quad = \quad \begin{bmatrix} \rho\dot{\theta}_1^2 - g\cos\theta_1 - \frac{k}{m}(\rho - \rho_0) - d\cos\theta_2(\dot{\theta}_1 + \dot{\theta}_2)^2 \\ \\ -2\dot{\rho}\dot{\theta}_1(\rho - d\cos\theta_2) + g\rho\sin\theta_1 - \\ \dot{\theta}_1\dot{\theta}_2 2d\rho\sin\theta_2 - \dot{\theta}_2^2 d\rho\sin\theta_2 - gd\sin(\theta_1 + \theta_2) \\ \\ \frac{\tau}{m} + \dot{\rho}\dot{\theta}_1 2d\cos\theta_2 + \dot{\theta}_1^2 d\rho\sin\theta_2 - gd\sin(\theta_1 + \theta_2) \end{bmatrix}. \tag{B.24}$$

We can write this system in the standard form as

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{T}, \tag{B.25}$$

where $\mathbf{M}$ is the same matrix, and $\mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})$, $\mathbf{G}(\mathbf{q})$ and $\mathbf{T}$ are given as

$$\mathbf{B} = \begin{bmatrix} 0 & d\cos\theta_2\dot{\theta}_1 - \rho\dot{\theta}_1 & d\cos\theta_2(\dot{\theta}_2 + 2\dot{\theta}_1) \\ 2\dot{\theta}_1(\rho - d\cos\theta_2) & -\dot{\theta}_2 2d\rho\sin\theta_2 & \dot{\theta}_2 d\rho\sin\theta_2 \\ -\dot{\theta}_1 2d\cos\theta_2 & -\dot{\theta}_1 d\rho\sin\theta_2 & 0 \end{bmatrix}, \tag{B.26}$$

$$\mathbf{G} = \begin{bmatrix} \frac{k}{m}(\rho - \rho_0) + g\cos\theta_1 \\ -g\rho\sin\theta_1 + gd\sin(\theta_1 + \theta_2) \\ gd\sin(\theta_1 + \theta_2) \end{bmatrix}, \tag{B.27}$$

$$\mathbf{T} = \begin{bmatrix} 0 \\ 0 \\ \frac{\tau}{m} \end{bmatrix}. \tag{B.28}$$