

ANALYSIS AND CONTROL OF BODY ATTACHED UNDERACTUATED
SPRING MASS RUNNER MORPHOLOGIES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

İZEL SEVER GÖKMEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

FEBRUARY 2022

Approval of the thesis:

**ANALYSIS AND CONTROL OF BODY ATTACHED UNDERACTUATED
SPRING MASS RUNNER MORPHOLOGIES**

submitted by **İZEL SEVER GÖKMEN** in partial fulfillment of the requirements for
the degree of **Master of Science in Electrical and Electronics Engineering De-
partment, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** _____

Assist. Prof. Dr. Mustafa Mert Ankaralı
Supervisor, **Electrical and Electronics Engineering, METU** _____

Prof. Dr. Uluç Saranlı
Co-supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering, METU _____

Assist. Prof. Dr. Mustafa Mert Ankaralı
Electrical and Electronics Engineering, METU _____

Prof. Dr. Uluç Saranlı
Computer Engineering, METU _____

Prof. Dr. Yiğit Yazıcıoğlu
Mechanical Engineering, METU _____

Prof. Dr. Ömer Morgül
Electrical and Electronics Engineering, Bilkent University _____

Date: 10.02.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: İzel Sever Gökmen

Signature :

ABSTRACT

ANALYSIS AND CONTROL OF BODY ATTACHED UNDERACTUATED SPRING MASS RUNNER MORPHOLOGIES

Sever Gökmen, İzel

M.S., Department of Electrical and Electronics Engineering

Supervisor: Assist. Prof. Dr. Mustafa Mert Ankaralı

Co-Supervisor: Prof. Dr. Uluç Saranlı

February 2022, 80 pages

One of the benchmark models for analyzing legged systems in biology and robotics is the Spring-Loaded Inverted Pendulum (SLIP) template and its extensions. The basic SLIP model consists of a single point mass with an ideal spring connecting it to the ground during the stance phase. After its introduction, this model has received numerous extensions to handle physical constraints that exist in practical configurations, such as the upper body's effect on the system dynamics. Although the SLIP template can describe COM behavior in its primary form, it fails to provide a framework for describing full-body stabilization and control.

In the first part of the thesis, we present a new control policy called the Central Pivot Point (CPP) for the body-attached spring-mass runners. In the stance phase, CPP directs ground reaction forces through the center of mass and cancels the torque created by these forces on the body. In this way, the CPP model makes it possible to develop different controllers for both the body's rotational and euclidean dynamics. Moreover, we analyze the characteristics and stability of the periodic solutions of the CPP model. Then, we develop a Proportional-Derivative (PD) controller for pitch

dynamics and a Linear Quadratic Regulator (LQR) for gait-level apex-to-apex discrete dynamics to stabilize the system's periodic solutions. We compute the basin of attraction of the proposed control scheme and show how the model behaves under disturbances.

Although our model has a compact mathematical form for its dynamics, there is no qualifying analytical expression due to its nonlinear nature. In the second part, we present a precise analytical approximation to the stance dynamics of the model in the case of no damping and non-symmetric trajectories. Our approach is based on radial actuation and the partial feedback linearization that embeds a simple template to linearize the radial dynamics and approximate the angular dynamics for handling the nontrivial body dynamics of TSLIP. The next step is simulating the model under different gravity correction methods to study their prediction performance for a comprehensive set of trajectories, including non-symmetric ones. Finally, we analyze the extended model in terms of the characteristics and stability of the periodic solutions.

Results obtained throughout the analysis of the TSLIP model and the proposed control scheme substantiate the model's prospect to ease the design and control of humanoid systems.

Keywords: legged locomotion, spring-mass hopper, periodic solution analysis, template model embedding, analytic approximate solutions

ÖZ

EKSİK TAHRİKLİ GÖVDELİ YAYLI KÜTLE KOŞUCU MORFOLOJİLERİNİN ANALİZİ VE KONTROLÜ

Sever Gökmen, İzel

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Dr. Öğr. Üyesi. Mustafa Mert Ankaralı

Ortak Tez Yöneticisi: Prof. Dr. Uluç Saranlı

Şubat 2022 , 80 sayfa

Yaylı Ters Sarkaç (SLIP) şablonu ve uzantıları, biyoloji ve robotikte bacaklı sistemlerin hareketini analiz etmek için uzun süredir kullanılmakta olan referans modellerdir. Temelde SLIP modeli kayıpsız, ideal bir yay aracılığıyla duruş aşamasında zemine bağlı olarak düşünülen noktasal bir kütleden oluşur. Literatürde birçok araştırmacı, gerçekçi sistemlerde kaçınılmaz olan kritik fiziksel olayları ele almak için bu modele, sönümlenme ve tork ile harekete geçirme gibi eklemeler yapmıştır. SLIP şablonunda eksik olan bir diğer önemli kavram, üst gövdenin kapalı döngü sistem dinamikleri üzerindeki etkisidir. SLIP şablonu kütle merkezi davranışını etkin bir şekilde yakalayabilmesine rağmen, tam vücut stabilizasyonunu ve kontrolünü açıklamak için bir yapı sağlayamaz.

Bu tezin ilk bölümünde, vücuda bağlı yaylı kütle koşucuları için Merkezi Pivot Noktası (CPP) adı verilen yeni bir kontrol politikası sunuyoruz. Duruş aşamasında, CPP zemin reaksiyon kuvvetlerini kütle merkezinden geçecek şekilde yönlendirir ve bu kuvvetlerin vücutta yarattığı torku iptal eder. Böylelikle, CPP modeli hem vücu-

dun dönüş hem de öklid dinamiği için farklı kontrolcüler geliştirmeyi mümkün kılar. Bunlara ek olarak, CPP modelinin periyodik çözümlerinin özelliklerini ve kararlılığını analiz ediyoruz. Ardından, sistemin periyodik çözümlerini stabilize etmek adına, adım dinamiği için bir PD denetleyicisi ve yürüyüş seviyesinde apektan apekse olan ayrık dinamikler için de bir LQR (Linear Quadratic Regulator) geliştiriyoruz. Daha sonra önerilen kontrol şemasının çekim yöresini hesaplıyoruz ve modelin bozulmalar altında nasıl davrandığını gösteriyoruz.

Model dinamiklerinin matematiksel formunun kompakt olmasına rağmen, dinamiklerin doğrusal olmayan doğası nedeniyle dinamiklere niteleyici bir analitik ifade bulunamamaktadır. Tezin ikinci bölümünde, modelin duruş dinamiklerine sönümsüz ve simetrik olmayan yörüngeleri de kapsayacak şekilde bir analitik yaklaşım sunuyoruz. Yaklaşımımız radyal dinamikleri doğrusallaştırmak için basit bir şablon yerleştiren ve TSLIP'in analitik çözümü zor olan vücut dinamiklerinin üstesinden gelmek için açısız dinamikleri yaklaşıklayan radyal aktivasyona ve kısmi geri besleme ile doğrusallaştırma methoduna dayanır. Bir sonraki adımda, analitik yaklaşımımızın tahmin performansını farklı yerçekimi düzeltme yöntemlerini kullanarak simetrik olmayan yörüngeleri de kapsayan geniş bir yörünge kümesi için elde ediyoruz. Son olarak, genişletilmiş modeli periyodik çözümlerin özellikleri ve kararlılığı açısından analiz ediyoruz.

Elde ettiğimiz sonuçlar, önerilen modellerin ve ilgili kontrol politikasının, analitik yaklaşım çözümüyle birlikte insansı robotik sistemlerin tasarlanması ve kontrol edilmesinde faydalı olabileceğini göstermektedir.

Anahtar Kelimeler: bacaklı hareket, yay-kütle zıplayanı, periyodik çözüm, şablon model gömme, analitik yaklaşık çözümler

Dedicated to my family and my beloved husband

ACKNOWLEDGMENTS

I would like to state my deep appreciation and sincere gratitude to my supervisor Assist. Prof. M. Mert Ankaralı for his guidance, encouragement, and continuous support during my undergraduate and graduate studies. His knowledge and creativity helped me broaden my horizons in bio-inspired robotics throughout these five years. Also, I would like to thank Prof. Dr. Uluç Saranlı for inspiring me to study legged locomotion. I would like to express my gratitude for the privilege of working with him and learning from the best. I also thank Prof. Dr. Kemal Leblebicioğlu for letting me participate in his TÜBİTAK project, which was an excellent opportunity to improve my theoretical knowledge and engineering skills.

One of the most rewarding aspects of my M.S. study was the opportunity to work with the fantastic group of people in Autonomous Technologies for Robotic Locomotion and Sensing (ATLAS) Laboratory. I am very thankful to all members, particularly to my mentor, Osman Kaan Karagöz, one of the most original minds I know, for inspiring me with his expertise and endless sources of new ideas and for many interesting conversations on life and everything else. I would like to express my sincere thanks to Simay Atasoy for her constant moral support during my endeavors; she was always the first to be there for me and give me the courage to proceed. I am fortunate to have her as my close friend. Further thanks also go to JeanPiere Demir, S. Ulaş Bingöl, and Raha Shabani for their insight and help during my studies.

Some of my friends from the outside of the laboratory directly or indirectly contributed to completing my dissertation. My sincere thanks to my esteemed colleagues, Uğur Güngör, Emre Akgül and Serkan Tarçın, for their understanding and support. I also thank my friends Zeynep Suvacı and Büşra Sariarslan for always being there to listen, motivate and elate.

I appreciate the Scientific and Technological Research Council of Turkey (TÜBİTAK) for awarding me the 2210 M.S. scholarship and the project 1001-117E966. Most im-

portantly, I thank my beloved husband, Umur Gökmen, for his undying love, understanding, and support. He has always put up with me with patience when struggling with my dissertation study.

I would like to express my gratitude to my precious sister Gülseren Tamirci and my lovely niece Derin Tamirci for cheering me up at any time and in any circumstances.

Last but not least, I owe my loving thanks to my parents, my loving mother, Nermin Sever, and my caring father, Niyazi Sever. Without their love and faith in me, none of this would have been possible.

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGMENTS	x
TABLE OF CONTENTS	xii
LIST OF TABLES	xv
LIST OF FIGURES	xvi
LIST OF ABBREVIATIONS	xx
CHAPTERS	
1 INTRODUCTION	1
1.1 Motivation and Problem Definition	1
1.2 Contributions and Novelties	4
1.3 The Outline of the Thesis	6
2 TRUNK-SLIP MODEL AND DYNAMICS	7
2.1 Trunk-SLIP Model	7
2.2 Cartesian Dynamics	8
3 CENTRAL PIVOT POINT CONCEPT AND ANALYSIS	11
3.1 Central Pivot Point Concept	11
3.2 Model Analysis	11

3.3	Gait Controller	13
3.4	Analysis of the CPP Concept	14
3.4.1	The Fixed Points of CPP Concept	14
3.4.2	Stability Analysis and Basins of Attraction of the Gait Controller	19
4	MODEL EXTENSION	23
4.1	Extended Model Stance Dynamics	24
4.2	Template Embedding	25
4.2.1	Basic Template: Undamped Spring-Mass Model	25
4.2.2	Embedding Undamped Spring-Mass Model	27
5	ANALYTICAL APPROXIMATE MAPS FOR EXTENDED TSLIP MODEL	29
5.1	Approximate Analytic Solutions to Stance Trajectories	29
5.2	Times for Bottom and Liftoff Events	32
5.3	Compensation for the Effects of Gravity	34
5.4	Approximate Analytical Apex Return Map with Gravity Correction .	37
5.4.1	Descent Phase Map	38
5.4.2	Approximate Stance Phase Map	40
5.4.3	Ascent Phase Map	40
6	PREDICTIVE PERFORMANCE AND STABILITY ANALYSIS	43
6.1	Simulation Environment and Performance Criteria	43
6.2	Simulation Results	46
6.2.1	Predictive Performance of Virtual States	46
6.2.2	Stability Analysis of the Fixed Points	57
7	CONCLUSIONS AND FUTURE WORK	59

REFERENCES	63
A DERIVATION OF THE EQUATIONS OF MOTION	71
A.1 Cartesian Dynamics	72
A.2 Virtual Leg Dynamics	73
B DERIVATIONS FOR SYSTEM PARAMETERS AND TRANSFORMA- TION MAPS BETWEEN CARTESIAN AND VIRTUAL LEG COORDI- NATES	75
B.1 Derivation of Leg Angle	75
B.2 Derivation of Hip Torque	75
B.3 Derivations of Virtual Leg Touchdown Angle	77
B.4 Derivations of Transformation Maps	77
C DERIVATIONS OF ANALYTIC APPROXIMATION FOR THE VIRTUAL STANCE STATES	79
C.1 Derivation of the Approximate Virtual Leg Trajectory	79
C.2 Derivation of the Approximate Angular Trajectory	80

LIST OF TABLES

TABLES

Table 2.1 Notation used for cartesian dynamics throughout the dissertation.	9
Table 2.2 TSLIP Parameters and their values	10
Table 4.1 Notation used for virtual polar dynamics throughout the dissertation	24
Table 5.1 Notation for flight and stance return maps and transition between coordinates.	38
Table 6.1 Ranges of interest for initial conditions and control inputs used in simulation experiments	44
Table 6.2 Statistical analysis of prediction performance of our analytic ap- proximation procedure with no gravity correction and using Saranlı’s and Arslan’s gravity correction methods in liftoff position p_{lo} and velocity \dot{p}_{lo} , apex position p_a and height z_a , apex energy E_a and stance time duration t_s . μ and σ stands for the mean and the standard deviation and <i>max</i> gives the maximum value of the related percentage prediction error. Also, sim- ulations results with zero and nonzero body pitch rates are stated on the top and bottom parts of the table, respectively.	47

LIST OF FIGURES

FIGURES

Figure 1.1	The Trunk-Spring Loaded Inverted Pendulum (TSLIP) model. a) Morphological parameters, b) Central Pivot Point model, including forces and torques acting on the model, c) model extension for deriving the analytical approximation.	4
Figure 2.1	Locomotion phases for the Trunk-SLIP model (shaded areas) and transition events (at the boundaries).	8
Figure 3.1	A scheme of Poincaré map with a third dimensional Poincaré hyper-plane, $F(X)$. X_k stands for the status vector of the system.	12
Figure 3.2	The fixed point distribution of the CPP model when $\alpha_{td} = 109.8^\circ$. The red "x" represents the fixed point $\mathbf{X}^* = [1.2, 3.7096, 0, 0]^T$	15
Figure 3.3	Evaluation of $\ \mathbf{X}_k^s - \mathbf{X}_{k+1}^s\ $ on $z = 1.2m$	16
Figure 3.4	Change of apex state in one step for $\theta = 0^\circ$, $\theta = 5^\circ$ and $\theta = -5^\circ$	18
Figure 3.5	Variation of the fixed point subspace by the leg angle at touch-down, α_{td}	18
Figure 3.6	Maximum eigenvalues, λ_{max} of the Jacobi matrix calculated by using the fixed points given in 3.5.	19
Figure 3.7	Maximum eigenvalues, λ_{max} of the Jacobi matrix calculated by using the fixed points obtained when the system runs under the action of the controller.	20

Figure 3.8 The height z , COM forward velocity \dot{x} , pitch angle θ and pitch rate $\dot{\theta}$ of the system with an example running simulation with an initial point of $z = 1.4m$, $\dot{x} = 2.91m/s$, $\theta = 20^\circ$, $\dot{\theta} = 100^\circ/s$, aiming to reach the fixed point state $\mathbf{X}^* = [1.2, 3.7096, 0, 0]^T$. The dashed red lines show the desired nominal values which is the fixed point values. 21

Figure 3.9 The stable domain of attraction for the system under the gait level LQR controller. The green region implies the initial apex-state conditions from which the system converges to a stable fixed point which is $z^* = 1.2m$ and $\dot{x}^* = 3.7096m/s$. The dashed lines stand for example simulations which show that the progression of the apex states. Simulations are conducted for **a)** $\theta = 0^\circ$, $\dot{\theta} = 0^\circ/s$ **b)** $\theta = 20^\circ$, $\dot{\theta} = 0^\circ/s$ **c)** $\theta = 0^\circ$, $\dot{\theta} = 100^\circ/s$ 22

Figure 4.1 Spring-Mass Hopper model and states in polar coordinates. 23

Figure 4.2 Formation of virtual leg system and physical system parameters 25

Figure 4.3 Undamped Spring-Mass Model 26

Figure 5.1 Embedding performance on virtual leg length, $r^v(t)$ during stance. Blue dashed curve represents the experimentally found virtual leg length, $r_{experimental}^v$, while red dashed curve stands for the approximate analytic solution of the virtual leg length, $r_{approximated}^v$. Shaded pink region illustrates the stance phase and green horizontal lines denote the critical time events, from left to right touchdown, bottom and liftoff. Sinusoidal solution of virtual leg length has amplitude $\frac{\dot{r}_{td}^v}{\omega_0^v}$, frequency ω_0^v and offset r_0^v . Δr_{max}^v stands for the maximum virtual leg compression. 31

Figure 5.2 An illustration of apex return map between the apex states X_a and X_{a+1} . Supmaps are expressed as the descent map, $F_d := F_{a \rightarrow td}(X_a)$, stance map $F_s := F_{td \rightarrow lo_v}(X_{td}^v)$ and the ascent map $F_a := F_{lo_v \rightarrow a}(X_{lo_v}^c)$. Coordinate transformations are symbolized as $T_{c \rightarrow v}$ and $T_{v \rightarrow c}$ 37

Figure 6.1 Statistical results for the prediction performance of the proposed analytical approximation method in liftoff position p_{lo} and velocity \dot{p}_{lo} , apex position p_a and height z_a without gravity correction and using Saranlı’s and Arslan’s gravity correction methods. These results are obtained taking $\dot{\theta} = 0$ and including both symmetric and non-symmetric strides under zero damping. Hollow markers, filled markers and vertical bars illustrate the related percentage prediction error’s mean, maximum, and standard deviation, respectively. 48

Figure 6.2 A statistical representation for percentage prediction errors of **(a)** liftoff position p_{lo} and **(b)** velocity \dot{p}_{lo} as a function of the relative touchdown angle $\alpha_{td, rel}$. Lines represent mean values of the percentage prediction errors, while the standard deviations are represented by shaded and colored regions of the associated gravity correction method. 50

Figure 6.3 A statistical representation for percentage prediction errors of **(a)** apex position p_a and **(b)** height z_a as a function of the relative touchdown angle $\alpha_{td, rel}$. Lines represent mean values of the percentage prediction errors, while the standard deviations are represented by shaded and colored regions of the associated gravity correction method. 51

Figure 6.4 A statistical representation for percentage prediction errors of **(a)** liftoff velocity \dot{p}_{lo} and **(b)** apex position p_a as a function of the natural frequency ω_0^v . Lines represent mean values of the percentage prediction errors, while the standard deviations are represented by shaded and colored regions of the associated gravity correction method. 53

Figure 6.5 A performance comparison predicting **(a)** liftoff velocity \dot{p}_{lo} and **(b)** apex position p_a using three gravity correction methods. Colored areas indicate the correction method, which gives the minimum mean of the error percentage in that combination of natural frequency (ω_0^v) and relative touchdown angle $\alpha_{td, rel}$ 54

Figure 6.6	Percentage prediction errors regarding (a) the liftoff velocity and (b) apex position versus the virtual leg state, θ^v . θ_n^v denotes the natural virtual leg touchdown angle, corresponding the fixed point state $\mathbf{X}^* = [1.2301, 4.1611, 0, 0]^T$ and natural touchdown angle $\alpha_{td,n} \approx 113 \text{ deg}$. .	56
Figure 6.7	The fixed point subspace variation by the touchdown leg angle, α .	57
Figure 6.8	Maximum eigenvalues, λ_{max} of the Jacobi matrix calculated by using the fixed points given in Figure 6.8.	58
Figure B.1	Geometric properties of Trunk SLIP model with inserted virtual leg and CPP concept parameters	76

LIST OF ABBREVIATIONS

COM	Center of Mass
SLIP	Spring Loaded Inverted Pendulum
TSLIP	Trunk Spring Loaded Inverted Pendulum
LQR	Linear Quadratic Regulator
PD	Proportional-Derivative
2D	two-dimensional
3D	three-dimensional
EOM	Equations of Motion
ODE	Ordinary Differential Equation
IC	Initial Condition

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Gait patterns such as trotting, pronking, and galloping, which enable cursorial animals to navigate land, are omnipresent among dynamic legged locomotion behaviors. The superiorities of these animals, such as agility, efficient energy use, and robustness against disturbances, motivate robotics scientists to develop single or multilegged platforms that operate similarly [1–7]. Also, the superior mobility of legged animals compared to other terrestrial platforms is accompanied by dynamics that include nonlinear, complex, and hybrid elements [8–10]. To overcome these difficulties while designing a legged robotic system or explaining animal locomotion, Full and Koditschek proposed a hierarchical structure [11]. In this framework, while simple yet effective models called "templates" explain targeted behavior such as running and walking, parsimonious models called "anchors" account for real systems or more complex systems. Creating or finding policies on anchor models, such that the anchor behaves like a template establishes the connection between a template and an anchor. This structure enables the use of simpler models to explain animal behavior and to analyze and control legged robots [12–16].

The SLIP model was first introduced in [17], and since then, it has gained status as one of the most well-known, versatile templates to describe biomechanical phenomena [18] and to function as a control target for running behavior of legged robots [19–22]. Although its mathematical simplicity and its ability to define the behavior of COM dynamics for various legged platforms, it does not provide a framework on how to stabilize upper body dynamics, which is highly critical to be neglected in

humans and humanoid systems. Several studies introduced methods such as reactive PD controller [15], passive geometric stability [23], and gyro-stabilizer [24] in order to keep the upper body in balance during stable legged locomotion. However, when such methods are in action, it becomes difficult to track the center of mass trajectories defined by the SLIP model [25]. For this reason, instead of SLIP, more inclusive templates that include upper body dynamics may be a better choice for descriptive accuracy, robustness and locomotion efficiency.

Poulakakis and Grizzle [26] introduced the first template model that integrates the upper body dynamics into a SLIP-like model, the Asymmetric SLIP¹(ASLIP). Then, they proposed a hierarchical control structure that can potentially stabilize the body dynamics and the locomotion. At the first level, they apply a continuous control policy at the stance phase to maintain the body around the desired posture and create an invariant manifold on which SLIP-like dynamics can be applied. They used a gait level SLIP-based controller at the second level to regulate the locomotion.

A study by Bayir [29] showed that conditioning a model similar to T-SLIP with a PD controller to stabilize upper body dynamics yields unstable periodic solutions. They have concluded on the requirement of time-varying controller configurations in order to stabilize these periodic solutions.

Later, Maus et al. [30] introduced the Virtual Pivot Point concept using the same template model in [26], inspired by the experimental locomotion data of humans, dogs, and chickens. This model expresses running behavior, aiming to stabilize the body like a virtual stable pendulum employing a spring between the hip and the foot and a virtual pivot point created above the center of mass.

There are several methods in the literature to simplify processes such as locomotion stability analysis, controller design, and motion planning in legged robots with complex dynamics like the model we propose. For example, methods proposed in [31–36] rely on empirical data/intuitive research or numerical methods such as the fourth-order Runge–Kutta method and the NewMark- β method.

Although SLIP dynamics have a straightforward mathematical derivation, they have

¹This template is called Asymmetric SLIP (ASLIP) in [27], Trunk-SLIP (TSLIP) in [28]. We refer this template as TSLIP throughout this dissertation.

no exact analytical solution due to the non-integrable stance dynamics. Considering this situation, parameter analysis and control implementation become problematic [37] when using numerical integration-based methods. For instance, due to the absence of an exact analytical representation of a legged model, a parametric connection cannot be established between the system parameters and the model's performance. Also, when the numerical integration-based controllers [38–40] are implemented for complex legged platforms, their performance is hindered by execution time and the high computational cost of numerical methods.

Because of these limitations, approximate analytical solutions can be considered more suitable for mitigating the nonlinearities of complex model dynamics than numerical methods. Compared to numerical methods, such as interpolation-based methods described in [41], analytical approximation methods can also provide necessary input for designing high-performance controllers and analyzing gait stability. The analytic approximation method was first implemented by Schwind and Koditschek, where a closed-form approximation was derived using the mean-value theorem augmented with a Picard-type iteration [42]. Ghigliazza et al. [43] neglected the effects of gravity to obtain a simple analytic approximation for the stance dynamics of the SLIP; also, they applied a simple fixed-leg reset policy to demonstrate the existence of stable gaits. Geyer et al. [44] developed a straightforward analytic approximation to the stance dynamics of the SLIP model to predict symmetric COM trajectories using the assumption of conservative angular momentum.

On the other hand, Arslan et al. [45] and Saranlı et al. [39] designed gravity correction algorithms to model the nonconservative effects of angular momentum on the cases of nonsymmetric trajectories. Shahbazi et al. [46] proposed an analytical approximate solution to the otherwise nonintegrable double-stance dynamics of the bipedal-SLIP model. They also implemented their method in human-like gaits and gait transitions to investigate its practicality. Considering the gravitational effects, Yu et al. presented a perturbation-based technique to obtain an analytic approximation to the stance dynamics in symmetric and nonsymmetric trajectories of the SLIP [47]. In order to reduce the computation duration and increase the practicality of performing online control actions, Piovan and Byl introduced a partial feedback linearization-based method for actuator displacement to find an analytic approximation of SLIP

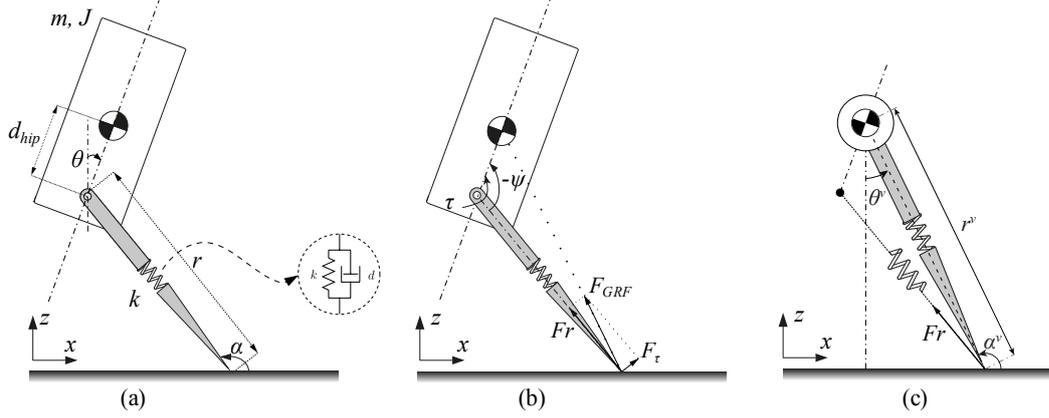


Figure 1.1: The Trunk-Spring Loaded Inverted Pendulum (TSLIP) model. **a)** Morphological parameters, **b)** Central Pivot Point model, including forces and torques acting on the model, **c)** model extension for deriving the analytical approximation.

dynamics [48].

1.2 Contributions and Novelties

In this study, we present a new control policy we have developed, which we call the Central Pivot Point (CPP), and examine its stability for the body attached spring-mass runner. CPP, in the stance phase, directs ground reaction forces through COM and cancels the torque created by these forces on the body of the template model represented in 1.1(a). Thus, it is possible to develop different controllers for both the body's rotational and euclidean dynamics by using the CPP model.

In addition, the existence of the periodic solutions of such systems are vital as it allows them to be controlled with the least possible effort (such as energy, torque) without opposing the conditions dictated by the natural dynamics of the system. At the same time, it makes it possible to develop controllers on these trajectories since the presence of periodic solutions allows for closed-form expressions of the system's trajectories. Therefore, we systematically investigated the existence and characteristics of the periodic solutions of the CPP-controlled model in this paper. Then, we stabilize the periodic solutions of the CPP by developing a PD controller for the pitch dynamics and LQR (Linear Quadratic Regulator) for euclidean dynamics. Later, in

order to show the model's behavior under disturbances, we compute the domain of attraction of the proposed control scheme. This part of the thesis is also available in [49] as a previously published work.

In general, the SLIP model is an accurate option to use as a control target when analyzing complex-morphological models like ours since there already exists many analytical approximations for SLIP in the literature [39, 42, 44, 45]. However, it is possible to use even simpler models in terms of dynamics and morphology. Rather than embedding the SLIP template and using its approximation, we used CPP as a control target, and in order to simplify the model dynamics, we consider the existence of a virtual leg between the toe and the COM as seen in Figure 1.1 part (c). In line with this assumption, we embedded an undamped spring-mass model with ordinary and linear stance dynamics to handle the strong nonlinearities in our model's virtual radial dynamics. Then, initially, we obtain an approximation using the assumption of conservation of angular momentum for angular dynamics. Later, we applied the gravity correction methods available in the literature to our model since the constant angular momentum assumption is not applicable for non-symmetric motions. Overall, the final contribution in this thesis is obtaining a precise closed-form analytic approximation for the stance phase of the proposed underactuated spring-mass runner via radial actuation and partial feedback linearization. Keeping in mind that utilizing an analytic approximation is a tradeoff between the estimation accuracy and the complexity in formulation, we revealed whether it makes sense to exert effort on embedding the SLIP model rather than embedding a more straightforward template via the prediction accuracy results of our analytic approximation approach. These results have confirmed not only the validity of this policy in various parameter combinations and operating conditions but also the analytic approximation method being a computationally efficient tool for locomotion control of the complex legged models. Once again, we conducted a comprehensive and systematic study on the existence and stability of the periodic solutions of the undamped spring-mass embedded TSLIP.

1.3 The Outline of the Thesis

The thesis' organization is divided into two main parts. The first part is as follows: Chapter 2 represents the TSLIP model, its dynamical formulation in cartesian coordinates, and transition between gait phases.

Chapter 3 focuses on the CPP concept, necessary tools to analyze such systems, and the gait controller to stabilize the model. Also, in this chapter, we report the periodic solutions of the CPP concept, these solutions' dependence on control parameters, the stability analysis of periodic solutions, and how the model performs under the gait controller.

The second part of the thesis starts with Chapter 4 which gives an explanation of the extended TSLIP model and its dynamics in a new polar coordinate system which we called virtual polar coordinates. Later in this chapter, we report the template embedding as a control policy on the model's radial dynamics. Afterwards, we describe embedding an analytically linear model to the radial component of the TSLIP.

In Chapter 5, an analytic approximation for the stance dynamics is given by compensating the nonlinear effects of gravity on angular momentum, for handling non-symmetric TSLIP trajectories. Also, we provide a comparison between the gravity correction methods of [39,45] on our approximation scheme. Towards the end of this chapter, the overall apex return map of the extended model is provided.

Chapter 6 accounts for the single stride prediction performance analysis of TSLIP's apex return map. This chapter begins by clarifying the simulation environment and performance criteria requirements. Subsequently, we continue with the simulation results, including predictive performance analysis and the stability analysis of the fixed points. We demonstrate the performance improvement by using a large set of initial conditions and control parameters, including the natural frequency.

Finally, in Chapter 7, we finalize the thesis with a review of our study by summarizing the main takeaways and possible future work.

CHAPTER 2

TRUNK-SLIP MODEL AND DYNAMICS

2.1 Trunk-SLIP Model

Trunk-SLIP (TSLIP) model, presented in Fig. 1.1(a), is an extended version of the SLIP model to account for robotic platforms where upper dynamics are not negligible. The model is composed of a body with mass m and inertia J , a compliant mass-less leg with spring constant k and a torque action at the hip joint. This hip joint does not coincide with the body's center of mass (COM). Table 2.1 presents the definitions of the variables used throughout the dissertation and Table 2.2 gives the physical parameters' values which are adopted from [28].

During the gait for each step, the TSLIP model alternates between two distinct dynamical phases, flight and stance. The flight phase is divided into two sub-phases, descent and ascent, while the stance phase has three critical events, touchdown, bottom, and liftoff. Figure 2.1 represents the transitions between these phases and events.

The moment, t_{td} , when the leg comes into contact with the ground corresponds to the "*touchdown event*." That means the touchdown event function is defined as the toe position coinciding with the ground, $z_{hip} - r_0 \sin(\alpha) = 0$. When the leg reaches its maximum compression and the spring's potential energy reaches its peak value, the "*bottom event*" occurs. $\dot{r}^v = 0$ at $t = t_b$.

The instant t_{lo} , when the toe takes off from the ground, and the leg hits its maximum extension at the end of the stance phase, is called the "*liftoff event*". Zero vertical ground reaction force in the stance phase, $F_{GRF_z} = 0$ is an indication of the liftoff event.

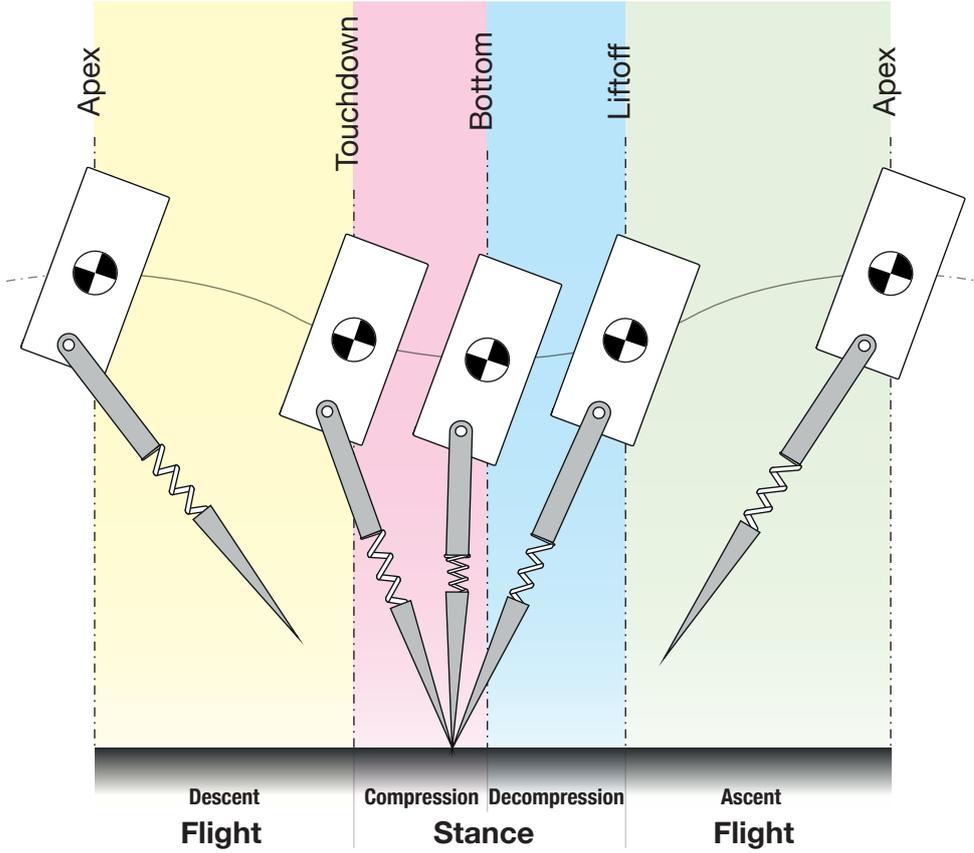


Figure 2.1: Locomotion phases for the Trunk-SLIP model (shaded areas) and transition events (at the boundaries).

There is also one more important event of the flight phase called the "*apex event*" where the trunk reaches its maximum height, z_a . Apex event occurs when the vertical velocity of the trunk crosses zero in flight phase, $\dot{z}_a = 0$.

2.2 Cartesian Dynamics

For both of the flight and stance phases, a typical representation of TSLIP's generalized coordinates is defined by the COM position in cartesian coordinates and the body's pitch angle. The generalized coordinates in cartesian coordinate frame can be formed as

$$q^c = [x, z, \theta]^T \subseteq \mathbb{R}^2 \times S.$$

Table 2.1: Notation used for cartesian dynamics throughout the dissertation.

States and Control Inputs	
q^c	Generalized state variables in cartesian coordinates
X^c	Body state vector in cartesian coordinates
x, z	Horizontal and vertical body COM position
\dot{x}, \dot{z}	Horizontal and vertical body COM velocity
x_{hip}, z_{hip}	Horizontal and vertical body hip position
r, θ, α	Leg length, body angle and leg angle with respect to the ground
$\dot{r}, \dot{\theta}$	Leg compression and body swing rate
p_θ	Angular momentum
$z_a, \dot{x}_a, \theta_a, \dot{\theta}_a$	Apex height, velocity, body angle and body swing rate
$r_{td}, \theta_{td}, \alpha_{td}, t_{td}$	Touchdown leg length, body angle, leg angle and time
$\dot{r}_{td}, \dot{\theta}_{td}$	Touchdown velocities in polar coordinates
K_p, K_d	Proportional and derivative gain
Kinematic and Dynamic System Parameters	
m, J, g	Body mass, moment of inertia and gravitational acceleration
d_{hip}	Distance between hip and the COM
r_0, k, d	Physical leg rest length, leg stiffness, active damping

During the flight phase, the motion of the COM, which is managed only by the gravitational forces, follows a ballistic trajectory. This trajectory is defined by the dynamics

$$\ddot{\mathbf{q}}^c = \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix}, \quad (2.1)$$

during the stance phase the toe remains stationary on the ground and the dynamics of the stance phase take the form

$$\ddot{\mathbf{q}}^c = \begin{bmatrix} 1/m & 0 & 0 \\ 0 & 1/m & 0 \\ 0 & 0 & 1/J \end{bmatrix} \left(D_q \psi \tau + D_q r F_r - \begin{bmatrix} 0 \\ mg \\ 0 \end{bmatrix} \right), \quad (2.2)$$

Table 2.2: TSLIP Parameters and their values

Physical Parameter	Value (unit)
m	80 (kg)
J	5 (kgm^2)
g	9.8 (m/s^2)
d_{hip}	0.1 m
r_0	1 m
k	20 kN/m

where F_r is the leg force governing by

$$F_r = -k(r - r_0) - d\dot{r}. \quad (2.3)$$

F_r and the hip torque, τ contribute to the accelerations through the Jacobians $D_q\psi$ and $D_q r$. $D_q\psi$ and $D_q r$ are calculated by taking the derivative of body-leg angle, ψ and leg length, r with respect to state variables, q^c , respectively (see Appendix A.1 for details). Equation 2.2 yields non-integrable stance dynamics as in the SLIP model [50].

Before the stance phase, the controller fixes the value of the damping coefficient, d . When the system is on the desired periodic solution, the nominal value of the damping is 0; also we tune d to stabilize and control the fixed points.

CHAPTER 3

CENTRAL PIVOT POINT CONCEPT AND ANALYSIS

3.1 Central Pivot Point Concept

The main idea behind the CPP concept is torque application from the hip to prevent the formation of any torque on the body. The hip torque makes the CPP available by directing the ground reaction forces to the COM throughout the whole movement during the stance phase, as illustrated in Figure 1.1(b). It is calculated as a function of F_r as

$$\tau_m = F_r r \tan(\gamma) . \quad (3.1)$$

where γ B.6, denotes the angle between the physical and the virtual leg as indicated in Figure 4.2 (derivation details of γ are provided in Appendix B). Under the effect of τ_m , the pitch dynamics of the TSLIP model are separated from the other dynamics. In this way, we can apply separate controllers to stabilize the system. We use a PD controller to control the pitch dynamics, while we use an LQR controller for other dynamics.

3.2 Model Analysis

In the analysis of dynamics and the evaluation of a dynamical system, the Poincaré Map Method demonstrated in Figure 3.1 is a convenient principle due to the periodic nature of the behavior. This method is based on a hyper-plane crossing the periodic trajectories of the system whose state-space model is defined by n generalized co-

ordinates. Thus the flow of the periodic system is expressed independently in time with a reduction in the generalized coordinates by one dimension. This hyper-plane that intersects the orbits is called the Poincaré Section. We define the k^{th} intersection of the system state vector with this hyper-plane as \mathbf{X}_k . Then the Poincaré Map can be represented as $F(\mathbf{X}_k) = \mathbf{X}_{k+1}$ using the connection between two successive intersections.

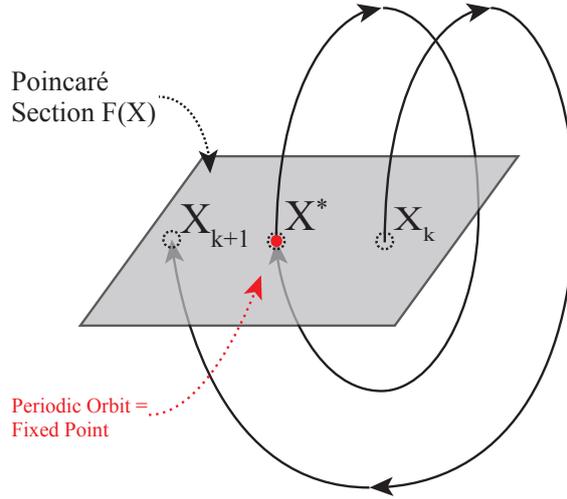


Figure 3.1: A scheme of Poincaré map with a third dimensional Poincaré hyper-plane, $F(X)$. X_k stands for the status vector of the system.

The solutions that consistently cross the Poincaré Section from the exact point are called the periodic orbits, and they are formulated as $F(\mathbf{X}^*) = \mathbf{X}^*$. In this expression, the point indicated by \mathbf{X}^* is called a fixed point. The fixed points provide insight into the local stability of the system, which is determined by the eigenvalues of the Jacobian matrix constructed by using these fixed points.

In the CPP Model, the hyper-plane specified by $\dot{z} = 0$ and $\ddot{x} < 0$ conditions can be selected as the Poincaré Section. These conditions define the highest position of the system on the vertical axis in the flight phase. Since the dynamics of the system is invariant along the horizontal axis, on this hyper-plane, the system status vector is defined $\mathbf{X}_k = [z, \dot{x}, \theta, \dot{\theta}]^T$ with four coordinates. However, $\dot{\theta} = 0$ is required for the existence of any fixed point. Therefore, the fixed point subspace is in 3 dimensional space, $\mathbf{X}_k^s = [z, \dot{x}, \theta, 0]^T$.

As stated earlier, the local stability of the fixed points can be analyzed by computing eigenvalues of the Jacobian matrix, $A = \frac{\partial P}{\partial \mathbf{X}}(\mathbf{X}^*)$. If the maximum eigenvalue of A is less than unity, the model is stable.

3.3 Gait Controller

Since τ_m separates the pitch dynamics and euclidean dynamics, we can develop a separate controller architecture for both dynamics. For pitch dynamics we form a PD law with high gain in the form of

$$\tau_{pd} = \left(K_p(\theta^* - \theta) - K_d\dot{\theta} \right).$$

Then the resulting hip torque becomes

$$\tau = \tau_m + \tau_{pd}. \quad (3.2)$$

If the pitch angle is at a desired orientation, all ground reaction forces pass through the COM. Otherwise, PD law tries to rotate the body to the desired orientation.

To stabilize other degrees of freedom (z and \dot{x}), we developed an event-based controller using LQR [51] and we chose the leg angle at touchdown, α_{td} , and damping ratio, d , as controller parameters. α_{td} is well known and widely used control input for the SLIP model [43, 52, 53] and damping controller is effectively used in a hopper to inject energy into the system or remove energy from the system [54]. In the flight phase, LQR calculates the necessary α_{td} and d to stabilize the model.

With these control inputs, the Poincaré Map becomes $F(\mathbf{X}_k, \mathbf{u}_k) = \mathbf{X}_{k+1}$ where $\mathbf{u}_k = [\alpha_{td}, d]^T$. The linearization of F around \mathbf{X}_k yields

$$\Delta \mathbf{X}_{k+1} = A \Delta \mathbf{X}_k + B \Delta \mathbf{u}_k. \quad (3.3)$$

where $\Delta \mathbf{X}_k = \mathbf{X}_k - \mathbf{X}^*$, $\Delta \mathbf{u}_k = \mathbf{u}_k - \mathbf{u}^*$ and $B = \frac{\partial P}{\partial \mathbf{X}}(\mathbf{u}^*)$. We assume that high gain PD law fixes the error in θ and $\dot{\theta}$. We do not consider the effects of pitch dynamics on z and \dot{x} , and we do not try to stabilize pitch dynamics using LQR. Hence, we use

the sub-matrices of A and B , which correspond to Δz and $\Delta \dot{x}$ while constructing the LQR problem. Let us define $A[1, 2; 1, 2]$ is a $[2 \times 2]$ matrix obtained from the first two columns and the first two rows of A , let $B[1, 2; 1, 2]$ is a $[2 \times 2]$ matrix obtained from the first two columns and the first two rows of B , then $A_S = A[1, 2; 1, 2]$ and $B_S = B[1, 2; 1, 2]$ are the matrices under investigation.

The cost function of infinite-horizon LQR controller as follows

$$J = \sum_{k=1}^{\infty} \mathbf{X}_k^T Q \mathbf{X}_k, \quad (3.4)$$

where Q is a positive definite matrix. Since CPP is a conceptual model, we only consider the state weights; we do not include inputs to the J . We construct the optimal state feedback K , as

$$K = (B_S^T P B_S)^{-1} B_S^T P A_S, \quad (3.5)$$

where P is the solution of the following discrete Riccati Algebraic Equation

$$P = Q + A_S^T (P - P B_S (B_S^T P B_S)^{-1} B_S^T P) A_S. \quad (3.6)$$

Finally, \mathbf{u}_k in Eq. (3.3) is calculated as follows

$$\mathbf{u}_k = - \begin{pmatrix} K \\ 0 \end{pmatrix} \Delta \mathbf{X}_k. \quad (3.7)$$

3.4 Analysis of the CPP Concept

3.4.1 The Fixed Points of CPP Concept

As stated earlier, an exact analytic solution of (2.2) can not be found. Therefore, we use numerical integration methods to obtain the CPP's fixed point subspace. Particularly, to compute Poincaré Map, F , we integrate the dynamics of the model from

apex to apex using MATLAB "ode45," a variable time-step solver based on 4th order Runge-Kutta integrator formula, the Dormand-Prince pair [55]. Due to the round-off errors introduced by the numerical integration, identifying the fixed points transforms into a search problem. This search problem can be expressed as the set

$$S_{\mathbf{X}^*} := \{\mathbf{X}_k^s \mid M > \|\mathbf{X}_k^s - \mathbf{X}_{k+1}^s\|\}. \quad (3.8)$$

$S_{\mathbf{X}^*}$ stands for the subspace of the fixed points, while M is a predefined threshold that specifies which apex state is a fixed point. Also, we should compatibly define the tolerance values for numerical integration and optimization algorithm and the threshold value M .

We use a built-in optimization algorithm called "fminunc" of MATLAB to specify $S_{\mathbf{X}^*}$. In order to cover 3D state-space, $\mathbf{X}_k^s = [z, \dot{x}, \theta, 0]^T$, this algorithm is started from various initial states. The main objective of the optimization is to minimize $\arg \min_{\mathbf{X}_k^s} \|\mathbf{X}_k^s - \mathbf{X}_{k+1}^s\|$ for every point in this 3D state-space. When the output points \mathbf{X}_k^s of the optimization algorithm satisfies the constraint $M > \|\mathbf{X}_k^s - \mathbf{X}_{k+1}^s\|$, they are considered as fixed points defined in the set $S_{\mathbf{X}^*}$.

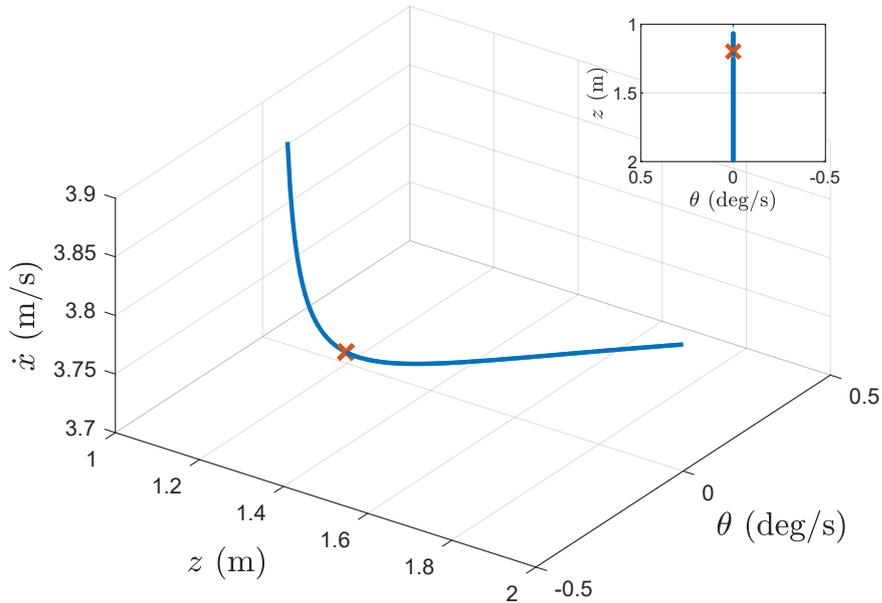


Figure 3.2: The fixed point distribution of the CPP model when $\alpha_{td} = 109.8^\circ$. The red "x" represents the fixed point $\mathbf{X}^* = [1.2, 3.7096, 0, 0]^T$.

Since we look for the fixed points of the CPP model, all the control parameters should be at their nominal value, $d = 0$, $K_p = 0$ and $K_d = 0$, and we selected $\alpha_{td} = 109.8^\circ$ for this example. For a given M , the resulting $S_{\mathbf{X}^*}$ is a curve on the plane that is defined by $\theta = 0^\circ$, see 3.2. This shows that the fixed point subspace is one-dimensional and lies on the plane defined by $\theta = 0^\circ$ for $\alpha_{td} = 109.8^\circ$.

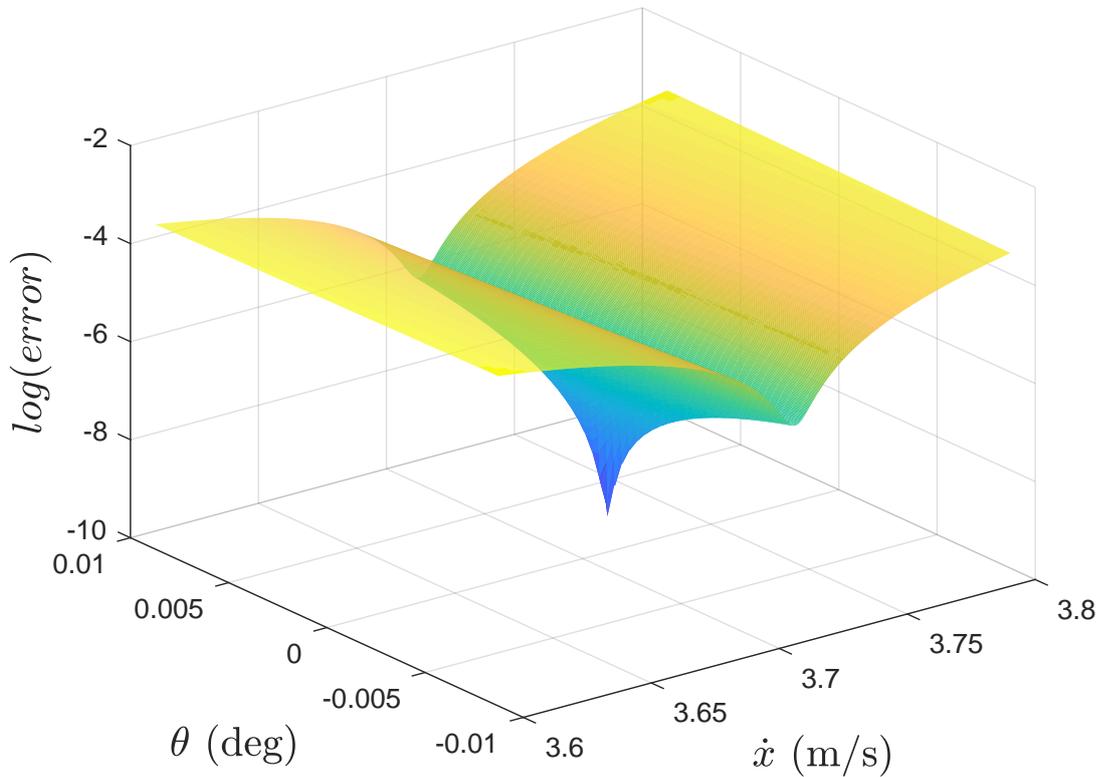


Figure 3.3: Evaluation of $\|\mathbf{X}_k^s - \mathbf{X}_{k+1}^s\|$ on $z = 1.2m$

To corroborate our claim, we computed a set of simulations. For instance, we chose the plane defined by $z = 1.2m$ which intersects the curve in 3.2 at the red cross point "x". Then, we evaluated $error = \|\mathbf{X}_k^s - \mathbf{X}_{k+1}^s\|$ for the whole plane and provide the corresponding $error$ results on 3.3. The intersection of the fixed point subspace and the plane defined by $z = 1.2m$ is the only minimum on this plane. The existence of a single minimum is the validation of our claim when $\alpha_{td} = 109.8^\circ$. Moreover, we simulate the system in a more extensive range to validate the claim holds for $91^\circ < \alpha_{td} < 120^\circ$.

We give an example of how the states of a fixed point evolve from apex to apex on 3.4. We started the simulation from $\mathbf{X}^* = [1.2, 3.7096, 0, 0]^T$ which is shown by

the red " x " in 3.2 and plotted its states in 3.4. As expected, all the states returned to their default values. Since all ground reaction forces pass through the center of mass, $\ddot{\theta}$ is 0, consequently $\dot{\theta}$ and θ are conserved.

Also, we investigated the apex states that are not on $\theta = 0^\circ$ plane in 3.4. When the body leans forward, the system gains energy by increasing its speed at the apex state while conserving its height. On the other hand, it loses its energy when the body leans backward. These results also show why the fixed point subspace is located on $\theta = 0^\circ$.

In addition investigating the fixed points for different α_{td} , we extended the fixed point search space to the interval defined by

$$\begin{aligned} 0.8 < z < 2 \\ 0 < \dot{x} < 7.5 \\ \theta &= 0^\circ \\ \dot{\theta} &= 0^\circ \\ 91^\circ < \alpha_{td} < 120^\circ \end{aligned}$$

3.5 represents the distribution of fixed points depending on α_{td} . For a given α_{td} at almost all heights, z , the fixed points exist. Along with the change in α_{td} , it is observed that the curve of fixed point space shifts in the direction of \dot{x} . This shows that we can cover a large portion of (z, \dot{x}) space by changing α .

We evaluated the stability of the fixed points on 3.5. The results are on 3.6. As seen, there exists a small stable region of fixed points on the upper left corner. However, most of them are unstable; hence a controller action is needed to stabilize the system and to create a basin of attraction for a desired fixed point.

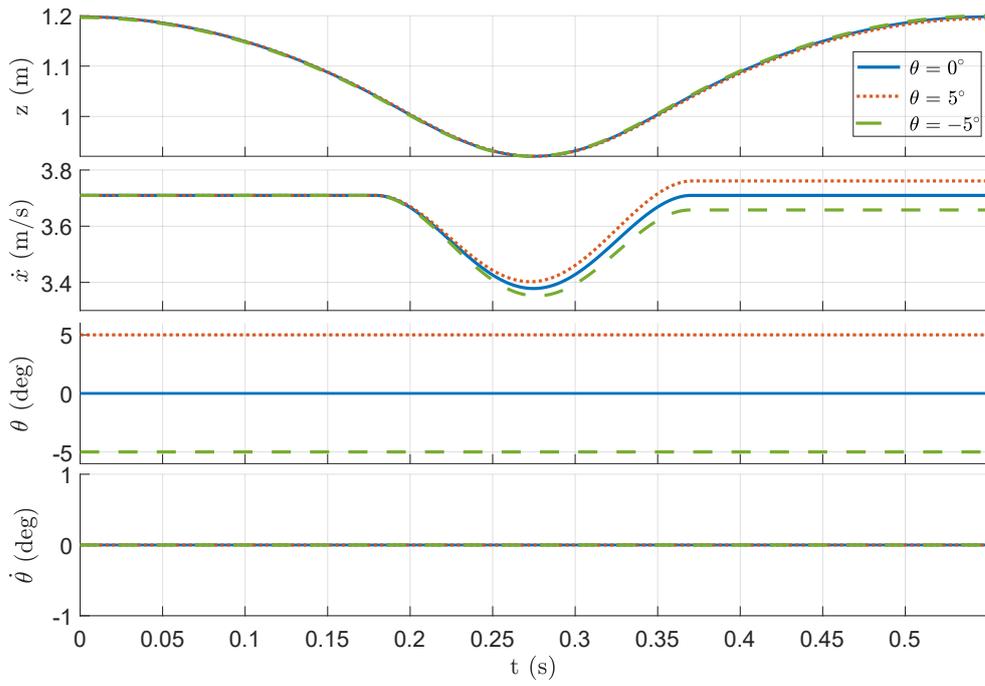


Figure 3.4: Change of apex state in one step for $\theta = 0^\circ$, $\theta = 5^\circ$ and $\theta = -5^\circ$

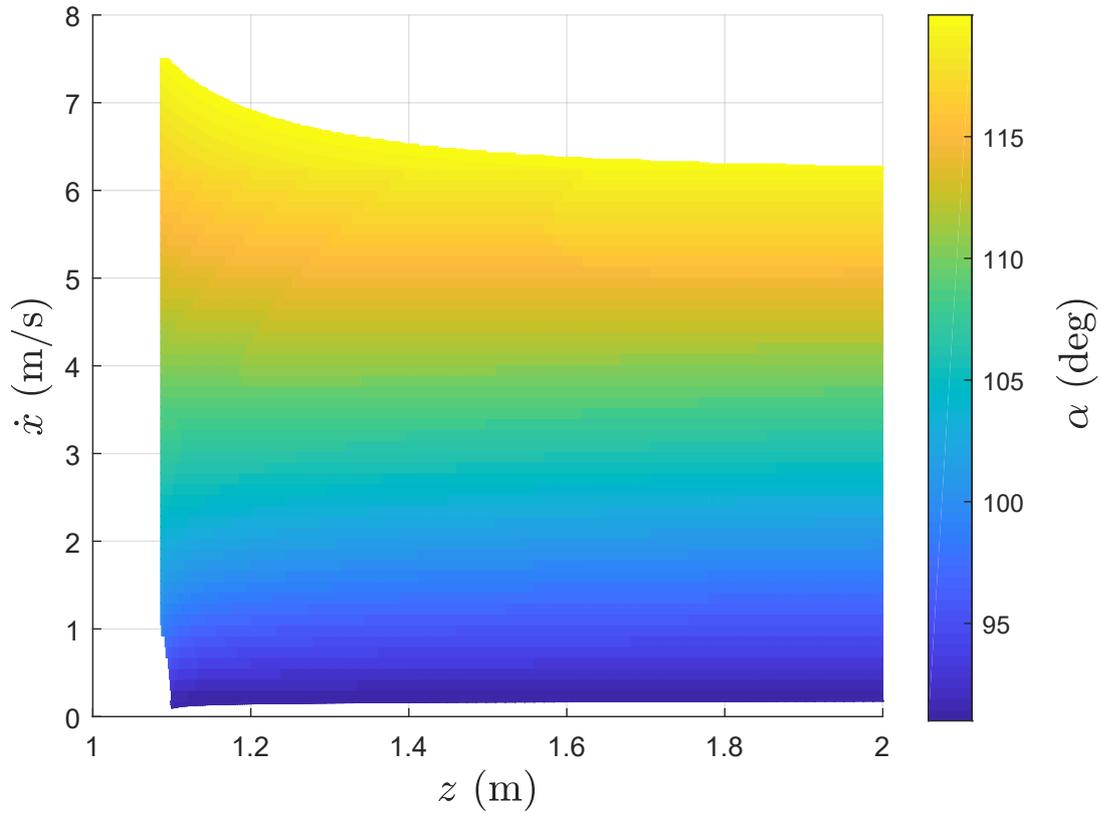


Figure 3.5: Variation of the fixed point subspace by the leg angle at touchdown, α_{td} .

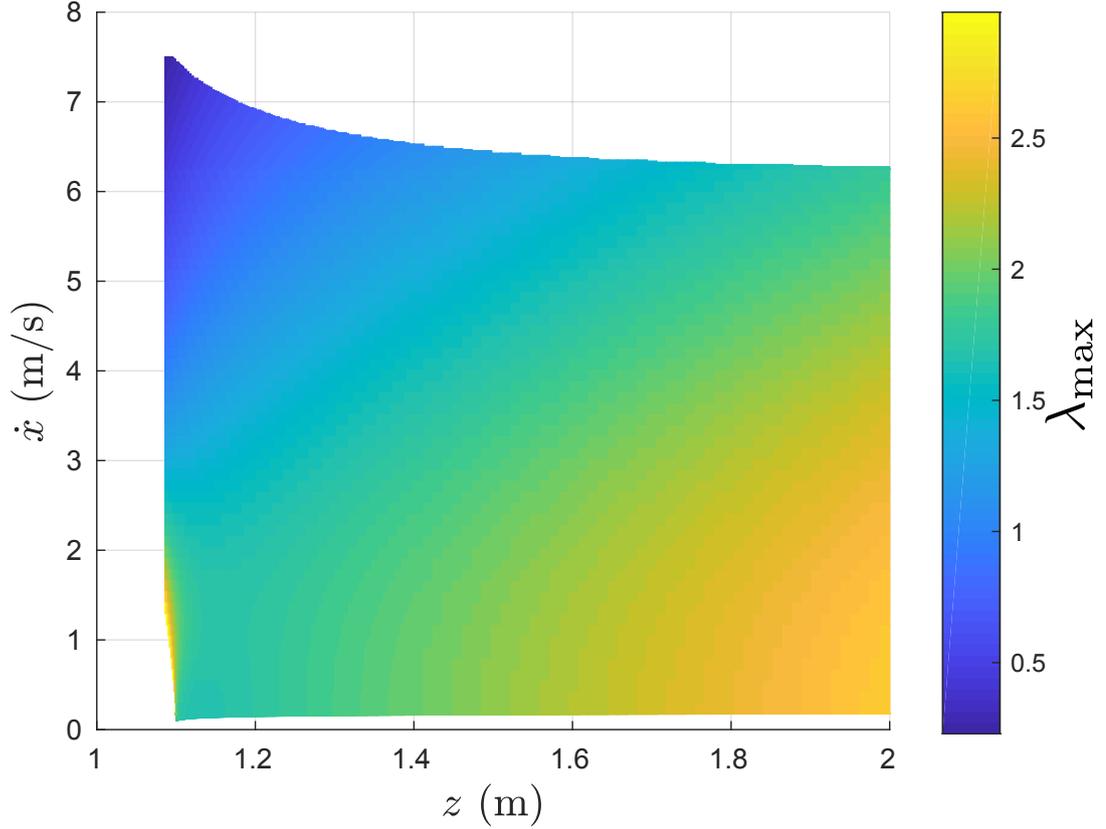


Figure 3.6: Maximum eigenvalues, λ_{max} of the Jacobi matrix calculated by using the fixed points given in 3.5.

3.4.2 Stability Analysis and Basins of Attraction of the Gait Controller

After investigating the fixed point existence of the system, we implemented the high gain PD controller to control the pitch dynamics and the LQR controller to regulate the height and the forward velocity by actively stabilizing the running behavior. As stated earlier, the stability characteristics of the system with no controller is given in 3.6, and only a small portion of the eigenvalues are less than unity, meaning that the most of the fixed points are unstable. As shown in 3.7, all the eigenvalues can be made less than one by using the PD controller and LQR to achieve a stable gait. Moreover, as seen in the example run in 3.8, the control structure leads to a stable periodic solution. In this running experiment, initial condition is given as $\mathbf{X}_0 = [1.4m, 2.91m/s, 20^\circ, 100^\circ/s]^T$. 3.8 indicates that the gait of the model converges to a periodic solution with minimal steady-state error (less than 1%), indicating that the

combination of the PD controller with the LQR controller accomplishes the stability of locomotion in approximately less than ten steps.

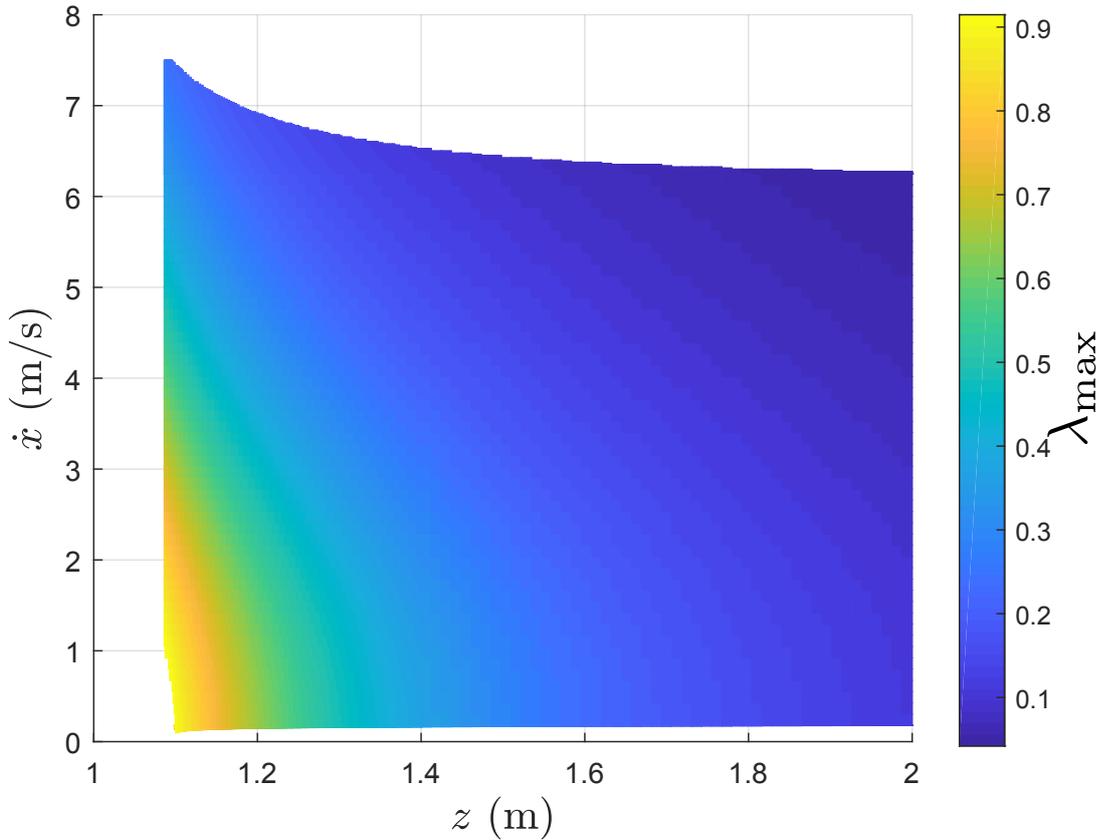


Figure 3.7: Maximum eigenvalues, λ_{max} of the Jacobi matrix calculated by using the fixed points obtained when the system runs under the action of the controller.

We conducted simulations to characterize the basins of attraction of our model and to define the stability properties of the gait controller. In these simulations, we operated the system by providing different initial conditions and checked whether it converged to a pre-determined fixed point, precisely $\mathbf{X}^* = [1.2m, 3.7096m/s, 0^\circ, 0^\circ/s]^T$. We considered the gait is stable if the system ran for 30 seconds, and the deviation from the fixed point at the last apex state was less than 1% of the fixed point.

In Figure 3.9, the variation in the stable basin of attraction of the system for different initial conditions is given. First, we analyzed the domain of attraction by keeping the pitch angle and the pitch rate as zero and sweeping the initial speed and height in an extensive range. Then investigated the behavior of the system by applying pitch angle $\theta = 20^\circ$ and pitch rate $\dot{\theta} = 100^\circ/s$, separately. The shaded green region in

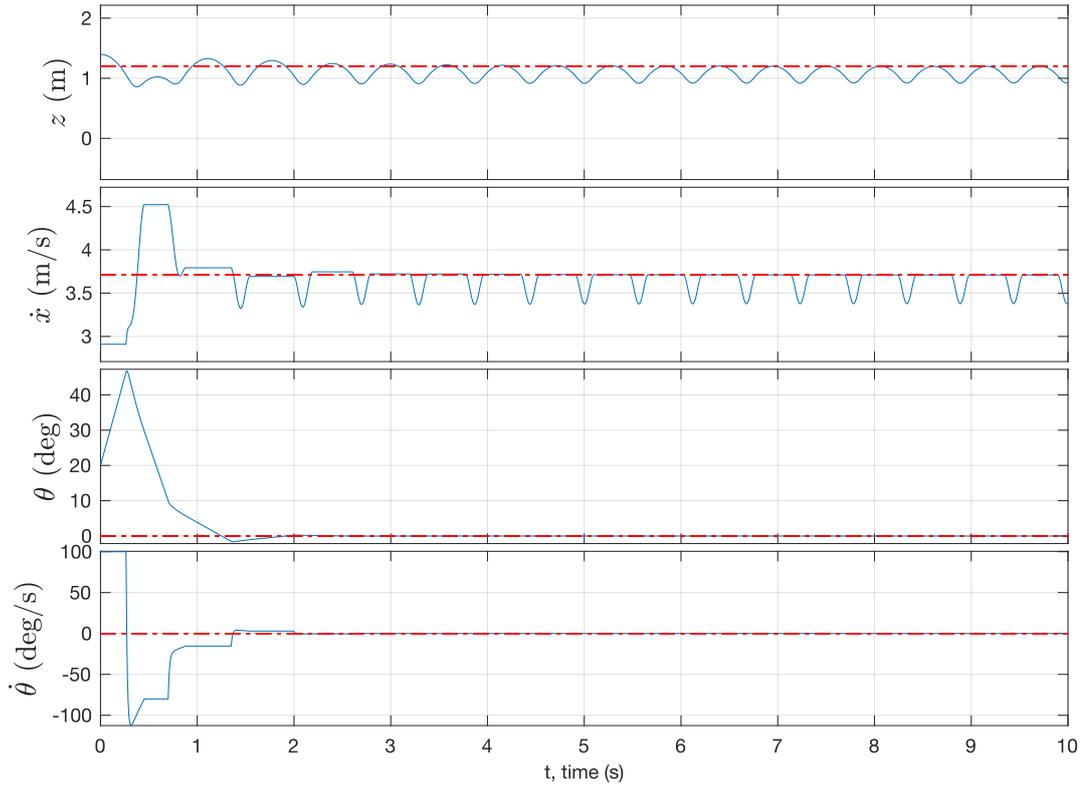


Figure 3.8: The height z , COM forward velocity \dot{x} , pitch angle θ and pitch rate $\dot{\theta}$ of the system with an example running simulation with an initial point of $z = 1.4m$, $\dot{x} = 2.91m/s$, $\theta = 20^\circ$, $\dot{\theta} = 100^\circ/s$, aiming to reach the fixed point state $\mathbf{X}^* = [1.2, 3.7096, 0, 0]^T$. The dashed red lines show the desired nominal values which is the fixed point values.

3.9, shifted in the direction of increasing \dot{x} while the shaded area remains almost the same if we add disturbance on pitch angle or the pitch rate. There is no torque limitation; therefore, stable locomotion can be attained with our controller even when we initiated the system at radical disturbances.

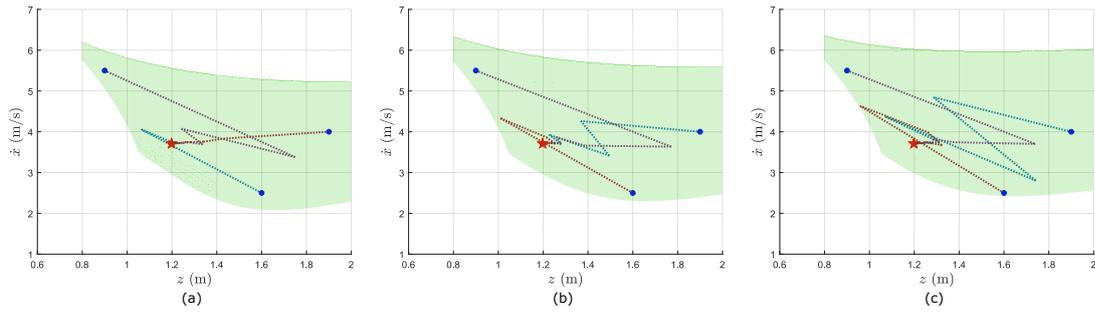


Figure 3.9: The stable domain of attraction for the system under the gait level LQR controller. The green region implies the initial apex-state conditions from which the system converges to a stable fixed point which is $z^* = 1.2m$ and $\dot{x}^* = 3.7096m/s$. The dashed lines stand for example simulations which show that the progression of the apex states. Simulations are conducted for **a)** $\theta = 0^\circ$, $\dot{\theta} = 0^\circ/s$ **b)** $\theta = 20^\circ$, $\dot{\theta} = 0^\circ/s$ **c)** $\theta = 0^\circ$, $\dot{\theta} = 100^\circ/s$.

CHAPTER 4

MODEL EXTENSION

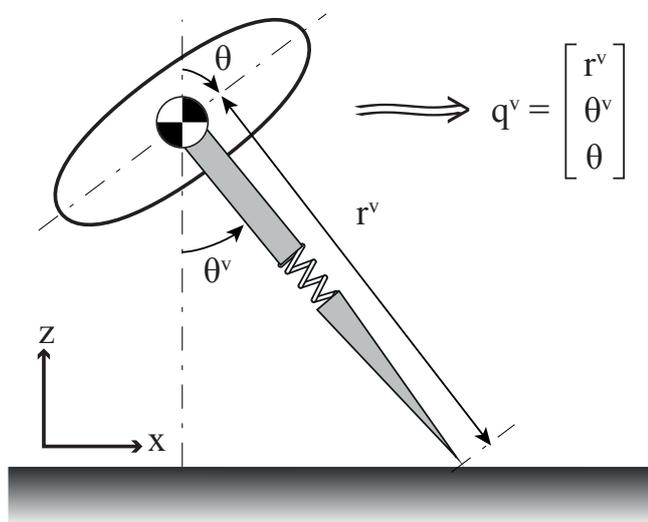


Figure 4.1: Spring-Mass Hopper model and states in polar coordinates.

The model illustrated in 4.1 is called the Spring-Mass hopper with a rigid body and a mass-less, fully passive leg. We aim to establish a physical association between our model and this model to have a sufficiently simple form of EOM to clarify the presentation of the stance dynamics. Firstly, we defined a virtual leg connecting the toe to its COM. Similar to the physical leg, the virtual leg is mass-less and has a linear spring with spring constant k^v . Also, in this virtual system, we consider that the model has a point mass at COM rather than a trunk, as shown in Figure 4.2 That means, eventually an extension for our model which can be called a virtual TSLIP model is formed.

Table 4.1: Notation used for virtual polar dynamics throughout the dissertation

States and Control Inputs	
q^v	Generalized state variables in virtual polar coordinates
X^v	Body state vector in virtual polar coordinates
r^v, θ^v	Virtual leg length and virtual leg angle
r_{td}^v, θ_{td}^v	Touchdown virtual leg length and leg angle
$\dot{r}_{td}^v, \dot{\theta}_{td}^v$	Touchdown virtual polar velocities
r_b^v, θ_b^v, t_b	Bottom virtual leg length, leg angle and time
$r_{lo}^v, \theta_{lo}^v, t_{lo}$	Liftoff virtual leg length, leg angle and time
$\dot{r}_{lo}^v, \dot{\theta}_{lo}^v$	Liftoff virtual polar velocities
Kinematic and Dynamic System Parameters	
r_0^v, k^v	Virtual leg rest length and virtual leg stiffness

4.1 Extended Model Stance Dynamics

Using polar coordinates is a simple and convenient option to obtain the stance dynamics. The definitions given in Table 4.1 are used for the virtual leg and the point mass in our derivation of the EOM in polar coordinates for the extended model. We defined the virtual generalized coordinates of the model as $q^v = [r^v, \theta^v, \theta]^T$.

The stance dynamics of our model in polar virtual leg coordinates concerning the toe location take the form 4.1. However in this case, the Jacobian (column) matrices (explained in detail in Appendix A.2), $D_{q^v}\psi$ and $D_{q^v}r$ are the derivatives of ψ and r with respect to the new state variables, q^v , respectively.

$$\ddot{q}^v = \begin{bmatrix} 1/m & 0 & 0 \\ 0 & 1/m & 0 \\ 0 & 0 & 1/J \end{bmatrix} \left(D_{q^v}\psi \tau + D_{q^v}r F_r - \begin{bmatrix} 0 \\ mg \\ 0 \end{bmatrix} \right) \quad (4.1)$$

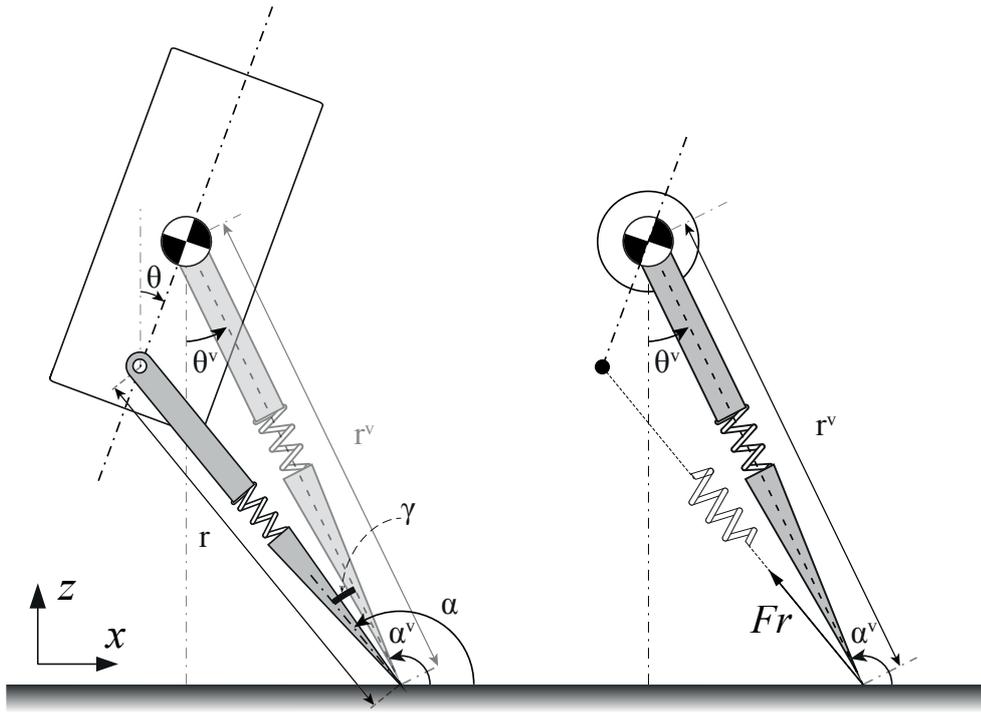


Figure 4.2: Formation of virtual leg system and physical system parameters

4.2 Template Embedding

Embedding a simple template model to our system, which has more complex and nonlinear dynamics, provide us to have fundamental linear dynamics by reducing them to a lower degree of freedom and using the control objectives designed for the template. In order to take advantage of this, we designed an embedding strategy based on the simplification of the model. Furthermore, this strategy helped its COM dynamics imitate an ideal template model's dynamical behavior and gait.

4.2.1 Basic Template: Undamped Spring-Mass Model

Including the gravity terms makes the stance dynamics problem of SLIP nonintegrable. In order to mitigate this effect, we can choose to modify the radial dynamics to not include gravity and angular momentum terms. However, this modification is still not enough for this model to be considered as an equivalent to the classical cen-

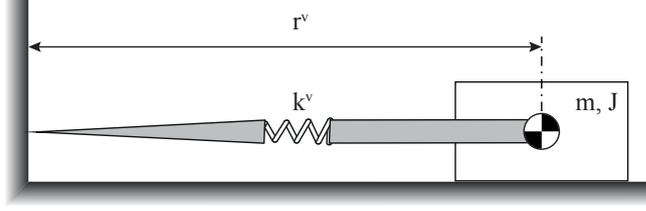


Figure 4.3: Undamped Spring-Mass Model

tral force problem. We also set the damping ratio of the embedded model as zero to overcome the effect of the damping losses on the system trajectories of the TSLIP model.

With the clarifications of the assumptions, we simplified the SLIP model by cancelling the nonlinear terms, namely the angular momentum and gravity terms, in the definition of radial stance dynamics yielding a simple undamped spring-mass model. Therefore the radial component of undamped model dynamics becomes as in the expression, (4.2), while the touchdown angle dynamics remain the same as the original SLIP model.

Figure 4.3 illustrates the lossless Spring-Mass model to be embedded. It consists of a 1-DOF planar rigid body with mass m , and a mass-less, compliant leg with a radial spring, k^v and rests length r_0^v . This system oscillates between r_{tb}^v (virtual leg length at bottom) and r_0^v .

$$\ddot{\mathbf{q}}^v = \begin{bmatrix} \ddot{r}^{v*} \\ \ddot{\theta}^v \end{bmatrix} = \begin{bmatrix} \frac{k^v}{m}(r_0^v - r^v) \\ \frac{g \sin(\theta^v) - 2r^v \dot{\theta}^v}{r^v} \end{bmatrix} \quad (4.2)$$

Here \ddot{r}^{v*} means the *desired* leg acceleration. Notice that the radial component of Equation 4.2 can be expressed in the standard form of 2^{nd} order ordinary differential equations (given in 4.3) in terms of the damping ratio, ξ , natural frequency, ω_0^v and the forcing function F^v

$$\ddot{r}^{v*} + 2\xi\omega_0^v + (\omega_0^v)^2 r_0^v = F^v \quad (4.3)$$

Since our model is undamped ($\xi = 0$) and by defining

$$\begin{aligned}\omega_0^v &= \sqrt{\frac{k^v}{m}} \\ F^v &= \frac{k^v}{m} r_0^v\end{aligned}\tag{4.4}$$

radial component of (4.2) reduces to,

$$\ddot{r}^{v*} + (\omega_0^v)^2 r_0^v = F^v\tag{4.5}$$

4.2.2 Embedding Undamped Spring-Mass Model

Our embedding policy is based on enforcing the COM dynamics close to that of the lossless Spring-Mass model during the stance phases, (4.6). This policy is similar to forcing the COM to follow a pre-specified target trajectory. To this end, we find a reasonable leg force, $F_{embedding}$, (4.8) which equalizes the radial leg acceleration of the COM and that of the embedded model.

$$\ddot{r}^{v*} - \ddot{r}^v = 0\tag{4.6}$$

Radial ground reaction force, F_r is defined as,

$$F_r = -k(r - r_0) - d\dot{r} + F_{embedding}.\tag{4.7}$$

By solving Equation 4.6 for $F_{embedding}$ yields

$$\begin{aligned}F_{embedding} &= \frac{1}{r} \left(r^v \left(d d_{hip} \sin \Phi \dot{\Phi} + \left(m(\dot{\theta}^v)^2 + k^v - 2k \right) (d_{hip} \cos \Phi - r^v) \right) + \right. \\ &\quad \left. k \left(-r_{td} r + d_{hip}^2 - (r^v)^2 \right) + (dr^v + gm \cos \theta^v + kr_0^v)(r^v + d_{hip} \cos \Phi) \right)\end{aligned}\tag{4.8}$$

where $\Phi := \theta + \theta^v$.

CHAPTER 5

ANALYTICAL APPROXIMATE MAPS FOR EXTENDED TSLIP MODEL

In addition to the embedding scheme, the non-integrable and highly nonlinear nature of the spring-mass hopper stance dynamics [50] requires an approximation. For this case, the approximate analytical stance motion models for SLIP-like platforms were considered as the best alternative, [42, 45, 56, 57]. As a matter of fact, these maps are also useful for the analysis of gait stability, motion planning and controller design for these type of platforms. The design of controllers for such platforms were based on data obtained from nature such as a diverse selection of animal footages, and numerical solutions of the problem, until the usage of approximate stance maps became widespread in the field of legged locomotion. A design example of a real-time dead-beat controller making use of observed gaits by interpolation can be found in [41]. Such examples serve as records to the laborious nature of this process of designing controllers. Compared to this example, several controllers making use of approximate stance maps in their design process [20, 58] have demonstrated much lower computational cost, along with the added trait of applicability to real-time control.

5.1 Approximate Analytic Solutions to Stance Trajectories

The fundamental principle of approximate stance maps is to predict liftoff states when the touchdown states are available. Notice that, for the stance period we can derive an analytic expression for the virtual leg length trajectory by embedding a much simpler model. Now, our objective is to obtain analytic approximation for the virtual leg angle trajectory.

After embedding the undamped TSLIP model with neglected gravity and angular mo-

momentum terms, an analytical solution to the radial motion $r^v(t)$ can easily be found. Note that the damping ratio of the 2^{nd} order ODEs given in 4.5 is zero ($\xi^v = 0$) since the system is lossless; therefore, it has always two purely imaginary roots, $s = \pm j\omega_0^v$. Then we can express $r^v(t)$ and $\dot{r}^v(t)$ as a combination of sinusoidal waves.

$$r^v(t) = A + B\cos(\omega_0^v t) + C\sin(\omega_0^v t) \quad (5.1)$$

$$\dot{r}^v(t) = \omega_0^v \left(-B\sin(\omega_0^v t) + C\cos(\omega_0^v t) \right) \quad (5.2)$$

where A , B and C are determined by the definition of dynamics of the undamped system 4.5 and the virtual touchdown states r_{td}^v , \dot{r}_{td}^v (derivation details in Appendix C.1). Also we assume a boundary condition at the instant of touchdown, $r_{td}^v = r_0^v$ and note that we take the touchdown time, $t_{td} = 0$ for the stance phase derivations.

Consequently, we have virtual leg length and velocity expressions as

$$r^v(t) = r_0^v + \frac{\dot{r}_{td}^v}{\omega_0^v} \sin(\omega_0^v t) \quad (5.3)$$

$$\dot{r}^v(t) = \dot{r}_{td}^v \cos(\omega_0^v t) \quad (5.4)$$

As illustrated in Figure 5.1, the solution of the virtual leg length $r^v(t)$ is a sinusoidal function with amplitude $\frac{\dot{r}_{td}^v}{\omega_0^v}$, frequency ω_0^v and offset r_0^v . Knowing that the virtual spring is compressed (or at rest for the touchdown state) during the stance phase, we only take the piece of the solution where $r^v(t) \leq r_0^v$ into consideration. In addition, $\Delta r_{max}^v = r_0^v - \frac{\dot{r}_{td}^v}{\omega_0^v}$ defines the maximum spring compression during the stance. Note that maximum relative spring compression is small with $\frac{\Delta r_{max}^v}{r_0^v} = 1 - \frac{\dot{r}_{td}^v}{r_0^v \omega_0^v} \ll 1$. Based on this inference, we can conclude that the relative radial spring compression remains relatively small with $\frac{r_0^v - r^v}{r_0^v} \ll 1$.

In order to find an analytic approximation to the angular trajectories, we used a method that is based on Geyer's methods proposed in [44]. This method relies on two essential assumptions. The first assumption supports that the virtual leg travels around a small angular span close to the vertical during the stance phase. This leads

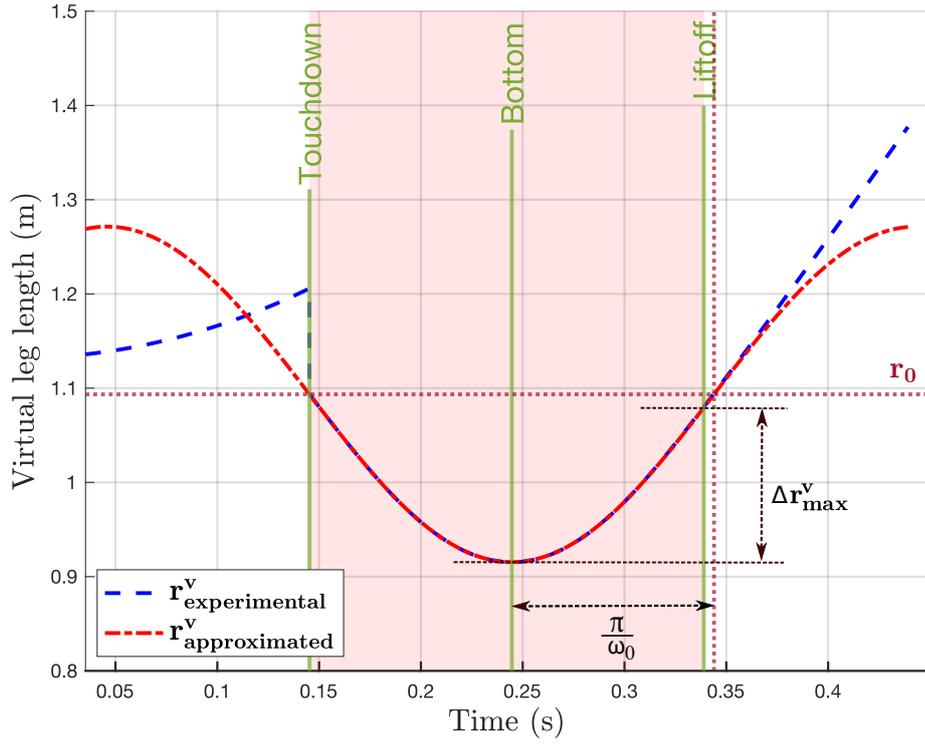


Figure 5.1: Embedding performance on virtual leg length, $r^v(t)$ during stance. Blue dashed curve represents the experimentally found virtual leg length, $r_{experimental}^v$, while red dashed curve stands for the approximate analytic solution of the virtual leg length, $r_{approximated}^v$. Shaded pink region illustrates the stance phase and green horizontal lines denote the critical time events, from left to right touchdown, bottom and liftoff. Sinusoidal solution of virtual leg length has amplitude $\frac{\dot{r}_{td}^v}{\omega_0^v}$, frequency ω_0^v and offset r_0^v . Δr_{max}^v stands for the maximum virtual leg compression.

the linearization of the gravity around $\theta^v = 0$ and consequently the conservation of the angular momentum during stance such that $p_{\theta^v} \approx p_{\theta^v_{td}}$. Moreover, in the virtual dynamics we assumed that the body is a point mass located at the COM stated as in the Chapter 4, this leads $p_{\theta^v} = m(r^v)^2\dot{\theta}^v$. However, the angular momentum can vary due to gravity towards the end of the stance phase.

Under the constant angular momentum assumption, the angular motion of the embedded model can be written as

$$\dot{\theta}^v = \frac{p_{\theta^v}}{m(r^v)^2} \quad (5.5)$$

Having sufficiently small relative spring compression allows approximating the term $\frac{1}{(r^v)^2}$ in Equation 5.5 by using Taylor Series expansion around the virtual rest length, r_0^v .

$$\frac{1}{(r^v)^2} \Big|_{r^v=r_0^v} = \frac{1}{(r_0^v)^2} - \frac{2}{(r_0^v)^3}(r^v - r_0^v) + O(r^v, r_0^v) \quad (5.6)$$

Hence the analytical solutions for the angular trajectory and the its rate of change are determined as in Equation 5.8 and 5.7. Derivation details are given in Appendix C.2.

$$\dot{\theta}^v(t) = \omega - \frac{2\omega}{\omega_0^v r_0^v} \dot{r}_{td}^v \sin(\omega_0^v t) \quad (5.7)$$

$$\theta^v(t) = \theta_{td}^v + \omega t + \frac{2\omega}{(\omega_0^v)^2 r_0^v} \dot{r}_{td}^v (\cos(\omega_0^v t) - 1) \quad (5.8)$$

where $\omega = \frac{p_{\theta^v}}{m(r_0^v)^2}$. Also, again $t_{td} = 0$ for the stance map.

5.2 Times for Bottom and Liftoff Events

While computing the stance dynamics, the touchdown time is assumed to be zero to simplify the derivations. When the virtual leg spring reaches maximum compression, the bottom event occurs. Hence the bottom event function is defined as $\dot{r}^v(t_b) = 0$. If we find the radial leg velocity at t_b using 5.4, we have

$$t_b = \frac{\pi}{2\omega_0^v} \quad (5.9)$$

For the ascending lossless TSLIP, liftoff occurs if the virtual leg length is equal to the virtual liftoff length, $r^v(t_{lo}) = r_{lo}^v$. t_{lo} can easily be computed analytically if r_{lo}^v is known. However, in our model, it is not known. For this reason, we use another identification of the liftoff event regarding the radial ground reaction force on the toe.

At liftoff, F_r crosses zero and starts to become negative. The related condition is given in the expression 5.10.

$$F_r = 0, \quad t = t_{lo} \quad (5.10)$$

Besides, we constructed our liftoff event assumption based on the virtual radial acceleration, that virtual liftoff radial acceleration should be equal to its liftoff value at liftoff instant. That means, $\ddot{r}^v(t_{lo}) = \ddot{r}_{lo}^v$. \ddot{r}^v can be expressed in terms of F_r as

$$\ddot{r}^v = -g\cos\theta^v + r^v(\dot{\theta}^v)^2 + F_r h(q^v) \quad (5.11)$$

where $h(q^v)$ is a function of the virtual states and by using the condition defined in 5.10, \ddot{r}^v becomes at t_{lo}

$$\ddot{r}^v(t_{lo}) = -g\cos\theta^v(t_{lo}) + r^v(t_{lo})(\dot{\theta}^v(t_{lo}))^2 \quad (5.12)$$

Once the value of $\ddot{r}^v(t_{lo})$ is determined, we only need to equate this result to the embedded radial dynamics given in 4.5 at t_{lo} , then solve this for t_{lo} .

Although it is not applicable to find exact analytical solutions to this set of equations, numerical methods are possible owing to the simple, one-dimensional nature of the problem. However, we integrate fair assumptions of liftoff times to achieve solutions that are true to the analytical nature of our approximations. To this end, we follow an iterative solution procedure. First, we made an explicit, initial choice for the liftoff time t_{lo}^i and then step by step calculate $\ddot{r}^v(t_{lo}^i)$.

An appropriate estimation for the liftoff time is $t_{lo}^i = 2t_b$ by accepting the r^v trajectory is symmetric during the stance phase. Hence, after the first iteration, the virtual state variables become

$$\begin{aligned}
r^v(t_{lo}^i) &= r_0^v \\
\theta^v(t_{lo}^i) &= \theta_{td}^v + \frac{\omega(t_{lo}^i)}{\omega_0^v} \left(\pi - 4 \frac{\dot{r}_{td}^v}{r_0^v \omega_0^v} \right) \\
\dot{\theta}^v(t_{lo}^i) &= \omega(t_{lo}^i)
\end{aligned} \tag{5.13}$$

where $\omega(t_{lo}^i) = \dot{\theta}_{td}^v$ due to the symmetric virtual leg length trajectory assumption. After rearranging and substituting liftoff state variables 5.13 in 5.12, we have

$$\begin{aligned}
\ddot{r}^v(t_{lo}^i) &= \ddot{r}_{lo}^v \\
&= -g \cos \left(\theta_{td}^v + \frac{\dot{\theta}_{td}^v}{\omega_0^v} \left(\pi - 4 \frac{\dot{r}_{td}^v}{r_0^v \omega_0^v} \right) \right) + r_0^v (\dot{\theta}_{td}^v)^2
\end{aligned} \tag{5.14}$$

Finally, solving $\ddot{r}_{lo}^v - \ddot{r}^v(t_{lo}) = 0$ for t_{lo} gives the analytic approximation for the liftoff time ($t_{lo} > 0$)

$$t_{lo} = \frac{\pi + \arcsin \left(\frac{\ddot{r}_{lo}^v}{\omega_0^v \dot{r}_{td}^v} \right)}{\omega_0^v} \tag{5.15}$$

5.3 Compensation for the Effects of Gravity

In Sections 5.1 and 5.2, we obtained analytic approximations to the stance map of the undamped TSLIP and decided times for critical events. We benefit from the constant angular momentum hypothesis throughout the stance period in our assumptions. However, as mentioned earlier, the angular momentum is not conservative due to the gravitational effects. These nonlinear effects of gravity arise either from the asymmetric features of the system or the environment. For instance, Saranlı [56] and Arslan [45] stated that legged robots running on rough terrain can have non-symmetric trajectories. In our model, we assume that the hopper runs on a sagittal plane; for that reason, it is not the environment that causes imbalances in the trajectories but the asymmetric structure and morphology of the model itself.

In this section, we apply a gravity correction to compensate for the angular momentum variation as an erroneous consequence of our assumptions and extend the domain

of validity of our analytic approximation method regarding the asymmetric trajectories.

Throughout the stance phase, the instantaneous angular momentum around the toe, $p_{\theta^v}(t)$ can be defined as

$$p_{\theta^v}(t) = p_{\theta^v}(t_{td}) + \int_{t_{td}}^t \tau(n) dn \quad (5.16)$$

$$\tau(t) = mgr^v(t) \sin(\theta^v(t))$$

where $\tau(t)$ and $p_{\theta^v}(t_{td})$ stand for the torque regarding the gravitational acceleration around the toe and the angular momentum at touchdown, respectively.

The exact calculation of the integral in Equation 5.16 is complicated even by utilizing the analytic approximations we made. Therefore, we review two methods proposed by Saranlı [56], and Arslan [45] to model the total effect of gravity on p_{θ^v} during the stance.

In the first method, [56], the corrected angular momentum, \hat{p}_{θ^v} is thought as a constant average value which is computed from the touchdown to the liftoff.

$$\hat{p}_{\theta^v} = \frac{1}{t_{lo} - t_{td}} \int_{t_{td}}^{t_{lo}} p_{\theta^v}(n) dn \quad (5.17)$$

Also a new approximation, which is based on taking the average of the integrand, $\tau(t)$ at the integral limits (touchdown and liftoff value of $\tau(t)$), is introduced.

$$\begin{aligned} \tau(t) &\approx \hat{\tau}(t) \\ &:= \frac{\tau(t_{td}) + \tau(t_{lo})}{2} \end{aligned} \quad (5.18)$$

Hence, using the equations 5.17 and 5.18, the corrected angular momentum yield

$$\hat{p}_{\theta^v} = p_{\theta^v}^{td} + mg \frac{t_{lo} - t_{td}}{4} \left(r_{lo}^v \sin(\theta_{lo}^v) + r_{td}^v \sin(\theta_{td}^v) \right) \quad (5.19)$$

where $p_{\theta^v}^{td} = m(r_{td}^v)^2 \dot{\theta}_{td}$.

The proposed method by Arslan et al. [45], is based on n-point discretization of the integral in 5.16. Referring this, $\hat{p}_{\theta^v}(t)$ can be rewritten as

$$\hat{p}_{\theta^v}(t) = p_{\theta^v}^{td} + (t - t_{td}) \left(\frac{1}{n} \sum_{i=0}^n mgr^v[i] \sin(\theta^v[i]) \right) \quad (5.20)$$

The discretized angular momentum can be computed for any stance state interval (t_i, t_f) denoting the initial and final times by approximating an average virtual leg length, $r_{av}^v(t_i, t_f)$. Choosing (t_i, t_f) such that $t_{td} \leq t_i < t_f \leq t_{lo}$ yields an update in Equation 5.3 as

$$r_{av}^v(t_i, t_f) = \frac{1}{t_f - t_i} \int_{t_i}^{t_f} r_0^v + \frac{\dot{r}_{td}^v}{\omega_0^v} \sin(\omega_0^v t) dt. \quad (5.21)$$

Then choosing the initial and the final states as the touchdown and liftoff states results in a new angular momentum term, 5.22 for the apex return map.

$$\hat{p}_{\theta^v} = p_{\theta^v}^{td} + mg \frac{t_{lo} - t_{td}}{2} r_{av}^v(t_{td}, t_{lo}) \left(\sin(\theta_{lo}^v) + \sin(\theta_{td}^v) \right) \quad (5.22)$$

Finally, by using these approximation schemes, we can update the angular momentum related derivations for example the expressions of the angular velocity, ω . Hence, in terms of \hat{p}_{θ^v} , the constant angular velocity, $\hat{\omega}$ is

$$\hat{\omega} = \frac{\hat{p}_{\theta^v}}{m(r_0^v)^2} \quad (5.23)$$

and all the instances of $\hat{\omega}$ is substituted in the angular trajectory equations, 5.8 and 5.7.

Note that our approach for computing the corrected angular momentum, \hat{p}_{θ^v} is an iterative process similar to the method introduced in [42]. Initially, calculation of \hat{p}_{θ^v} requires a preliminary estimate regarding the liftoff states. We determine the initial liftoff states by using the uncompensated stance map, which takes the angular momentum as $p_{\theta^v} \approx p_{\theta_{td}^v}$. Compensating the effects of gravity by using the approaches mentioned earlier is the second step for our trajectory approximation.

5.4 Approximate Analytical Apex Return Map with Gravity Correction

In general, *apex return map* (Figure 5.2) is an efficient abstract concept that facilitates understanding and characterizing the transient nature of the locomotion [59]. Also, using these maps simplifies the stability analysis and control of the periodic trajectories.

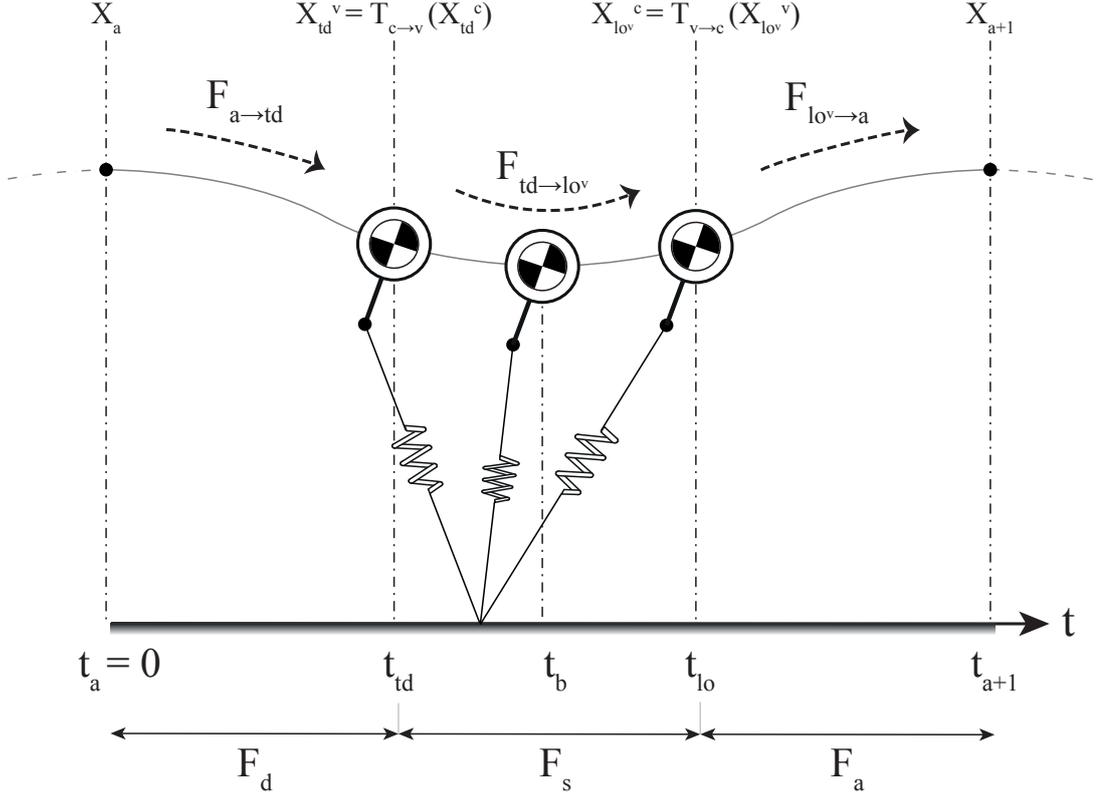


Figure 5.2: An illustration of apex return map between the apex states X_a and X_{a+1} . Submaps are expressed as the descent map, $F_d := F_{a \rightarrow td}(X_a)$, stance map $F_s := F_{td \rightarrow lo^v}(X_{td}^v)$ and the ascent map $F_a := F_{lo^v \rightarrow a}(X_{lo^v}^c)$. Coordinate transformations are symbolized as $T_{c \rightarrow v}$ and $T_{v \rightarrow c}$.

Our model is a hybrid system with stance and flight phases. Therefore, it is appropriate to divide the apex return map F , into three submaps. The notation given in Table 5.1 for the descent, stance, and ascent return maps are decided as F_d , F_s , and F_a , respectively. Thus, the complete apex return map can be standardized as $F := F_d \circ T_{c \rightarrow v} \circ F_s \circ T_{v \rightarrow c} \circ F_d$ where $T_{c \rightarrow v}$ and $T_{v \rightarrow c}$ denote the transformation maps

between the cartesian and the virtual polar coordinates.

Since the descent period represents the travel from the apex to the touchdown, the descent map can be reorganized as $F_d := F_{a \rightarrow td}(X_a)$. In a similar manner, $F_s := F_{td \rightarrow lo_v}(X_{td}^v)$ is the mapping from touchdown states to liftoff states in virtual coordinates, and $F_a := F_{lo_v \rightarrow a}(X_{lo_v}^c)$ is the mapping from liftoff states in virtual coordinates to apex states.

Table 5.1: Notation for flight and stance return maps and transition between coordinates.

Transformation and Mapping functions	
$T_{c \rightarrow v}(X^c)$	From cartesian to virtual polar coordinate transformation
$T_{v \rightarrow c}(X^v)$	From virtual polar to cartesian coordinate transformation
$F_{a \rightarrow td}(X_a^c)$	Apex to touchdown map
$F_{td \rightarrow lo_v}(X_{td}^v)$	Mapping from touchdown to liftoff states in virtual coordinates
$F_{lo_v \rightarrow a}(X_{lo_v}^c)$	Mapping from liftoff in virtual coordinates to apex states

Deciding return maps for descent and ascent phases is straightforward since, during the flight phase, the body tracks a ballistic trajectory. On the contrary, the exact return maps are not available for the stance phase, likewise the explicit analytical solutions. Consequently, we construct an approximate return map for the stance phase.

5.4.1 Descent Phase Map

It is more natural to solve the flight dynamics in the cartesian coordinates. Hence, for a given initial apex state, touchdown states in cartesian coordinates can be obtained by simply using the projectile motion formulas. We define the apex state vector as $X_a = [x_a, \dot{x}_a, z_a, \dot{z}_a, \theta_a, \dot{\theta}_a]^T$ and touchdown state vector as $X_{td}^c = [x_{td}, \dot{x}_{td}, z_{td}, \dot{z}_{td}, \theta_{td}, \dot{\theta}_{td}]^T$. The apex to touchdown return map is derived as in 5.24.

$$X_{td}^c = F_{a \rightarrow td}(X_a) = \begin{bmatrix} x_a + \dot{x}_a t_{td} \\ \dot{x}_a \\ z_a + \dot{z}_a t_{td} - \frac{1}{2} g t_{td}^2 \\ \dot{z}_a - g t_{td} \\ \theta_a + \dot{\theta}_a t_{td} \\ \dot{\theta}_a \end{bmatrix} \quad (5.24)$$

Note that, in our $F_{a \rightarrow td}$ derivations, we take $t_{td} \neq 0$. t_{td} is calculated according to the touchdown condition of

$$z_{hip}(t_{td}) - r_0 \sin(\alpha_{td}) = 0 \quad (5.25)$$

where $z_{hip}(t_{td})$ denotes the height of the hip point at touchdown. Defining the initial state vector as $X_0 = [x_0, \dot{x}_0, z_0, \dot{z}_0, \theta_0, \dot{\theta}_0]^T$ (initial state can be any flight state not necessary to be an apex state) and the initial time as $t_0 = 0$, by using the model geometry and the flight dynamics we can rewrite 5.25 as

$$z_0 - \dot{z}_0 t_{td} - \frac{1}{2} g t_{td}^2 - d_{hip} \cos(\theta_0 + \dot{\theta}_0 t_{td}) - r_0 \sin(\alpha_{td}) = 0 \quad (5.26)$$

Unfortunately, the exact transient solution of Equation 5.26 can not be expressed analytically. We can either solve it numerically by transforming it to an optimization problem or utilizing the main principle of CPP that assures no torque formation on the trunk (details given in chapter 3). As a result the angular movement of the trunk is restrained during the locomotion with $\dot{\theta} = 0$. Even when $\dot{\theta}_0 \neq 0$, it will eventually be cancelled after a single stride due to the hip torque, τ_m . Due to these applications, we can say that t_{td} is quasi-analytic and becomes

$$t_{td} = \frac{\dot{z}_0 + \sqrt{\dot{z}_0^2 + 2g \left(z_0 - d_{hip} \cos(\theta_0) - r_0 \sin(\alpha_{td}) \right)}}{g} \quad (5.27)$$

5.4.2 Approximate Stance Phase Map

As pointed out earlier, we derive stance trajectories in polar coordinates for simplicity. In descent return map, we obtained the touchdown states in cartesian coordinates, now we need to convert X_{td}^c into X_{td}^v . Using the transformation map between cartesian and virtual leg states, $T_{c \rightarrow v}(q^c)$ (appendix B), we can reorganize the liftoff trajectories in a compact form by combining 5.3 and 5.8 with liftoff time, 5.15 and applying gravity correction introduced by Saranlı.

Hence, reassuming the start time of the stance map is $t_{td} = 0$, the resulting approximate stance return map becomes

$$X_{lo}^v = F_{td \rightarrow lo^v}(X_{td}^v) = \begin{bmatrix} r_0^v + \frac{\dot{r}_{td}^v}{\omega_0^v} \sin(\omega_0^v t_{lo}) \\ \dot{r}_{td}^v \cos(\omega_0^v t_{lo}) \\ \theta_{td}^v + \hat{\omega} t_{lo} + \frac{2\hat{\omega}}{(\omega_0^v)^2} \frac{\dot{r}_{td}^v}{r_0^v} (\cos(\omega_0^v t_{lo}) - 1) \\ \hat{\omega} - \frac{2\hat{\omega}}{\omega_0^v r_0^v} \dot{r}_{td}^v \sin(\omega_0^v t_{lo}) \\ \theta_{td} + \dot{\theta}_{td} t_{lo} \\ \dot{\theta}_{td} \end{bmatrix} \quad (5.28)$$

where, toe is stationary on a flat surface with zero height. Note, even applying gravity correction, the proposed analytic approximation has a straightforward analytical form that we can effortlessly design dynamical locomotion controllers.

5.4.3 Ascent Phase Map

As stated previously for the descent map, we find it useful to express flight dynamics in cartesian coordinates. Under the coordinate transformation, $T_{v \rightarrow c}(X_{lo^v}^v)$, liftoff states in the cartesian coordinates are found as $X_{lo^c}^c$ (details given in the appendix B). Hence, finally, the resulting liftoff to apex map turns into

$$X_{a+1} = F_{l_{ov} \rightarrow a}(X_{l_{ov}}^c) = \begin{bmatrix} x_{l_o} + \dot{x}_{l_o}(t_{a+1} - t_{l_o}) \\ \dot{x}_{l_o} \\ z_{l_o} + \dot{z}_{l_o}(t_{a+1} - t_{l_o}) - \frac{1}{2}g(t_{a+1} - t_{l_o})^2 \\ \dot{z}_{l_o} - g(t_{a+1} - t_{l_o}) \\ \theta_{l_o} + \dot{\theta}_{l_o}(t_{a+1} - t_{l_o}) \\ \dot{\theta}_{l_o} \end{bmatrix} \quad (5.29)$$

where $t_{a+1} = \frac{\dot{z}_{l_o}}{g}$ denotes the time when the next apex event occurs.

CHAPTER 6

PREDICTIVE PERFORMANCE AND STABILITY ANALYSIS

6.1 Simulation Environment and Performance Criteria

Throughout this chapter, we carry out a comprehensive performance analysis of the analytically approximated apex return map of TSLIP in single stride predictive performance. We used normalized percentage prediction errors PE as a performance evaluation indicator for critical state variables. The definition of PE is adopted from [39] and is given as

$$PE := 100 \frac{\|X_{num} - X_{approx}\|_2}{\|X_{num}\|_2} \quad (6.1)$$

where X denotes the state variable of interest while the the subscripts num and $approx$ stand for results obtained using numerical integration and analytic approximation methods, respectively.

We use a sufficiently large set of initial conditions and control inputs collected in Table 6.1 to investigate approximations. In other words, we perform the simulations for every five-dimensional set, whose dimensions are the initial states and control inputs. These dimensions are decided as the apex height z_a , the apex velocity \dot{x}_a , the body orientation and rotational velocity at apex θ_a and $\dot{\theta}_a$, the virtual spring constant k^v , and the “relative touchdown angle” $\alpha_{td,rel}$ which is defined in [56] as

$$\alpha_{td,rel} = \alpha_{td} - \alpha_{td,n} \quad (6.2)$$

where $\alpha_{td,n}$ indicates the neutral touchdown angle of the fixed point of the apex return

Table 6.1: Ranges of interest for initial conditions and control inputs used in simulation experiments

Initial Condition Ranges						
Dimension	z_a	\dot{x}_a	θ_a	$\dot{\theta}_a$	$\alpha_{td,rel}$	k^v
Range	[1.1, 1.5]	[0, 8]	[-10, 10]	[0, 15]	[-0.4, 0.2]	[8, 32]
Unit	m	m/s	$^\circ$	$^\circ/s$	rad	kN/m

map. Also, note that changing k^v affects the natural frequency of the system due to the relation stated in 4.4.

In order to be consistent with the existing work on robotics and bio-mechanics, we choose the intervals for these dimensions as in Table 6.1 for our simulations. An example for choosing appropriate ranges is Arampatzis’s research on human running [60]. This research carried out on humans with an average mass of 80 kg and average leg length of 1 m has proved that the leg stiffness should be roughly [10, 50] kN/m while humans run at speeds in the range of [2.5, 6.5] m/s .

Further examples based on experimental biology evidence are the analyses of human running experiments conducted either by Müller et al. [61] and Lee [62]. According to their studies, the maximum leg compression is circa 5 – 11 % of the rest length and touchdown leg angle, $\alpha_{td} \in [109, 114]^\circ$. Another example can be given in the field of legged robots such as the RHex hexapod, [21]. The RHex, whose mass is around 10 kg , leg length is 0.25 m , has compliant legs, each with a stiffness of approximately 2000 N/m .

Note that we do not introduce damping as a control parameter since even in the presence of damping, its effect on system trajectories will disappear after a single stride due to embedding the template model, which is lossless with $\hat{d} = 0$.

There are some critical requirements that the solutions for the trajectories should satisfy to maintain the error percentage inside meaningful bounds. In circumstances of not satisfying these conditions, some error situations arise. These error conditions are listed in [56] as

- either a trajectory never leaves the ground $\dot{z}_{lo} < 0$ or prevents foot protraction $z_a < 1$. That means the rest length of the leg is greater than the next apex height causing no restriction on the touchdown angle of the next apex state.
- the leg compression exceeds the maximum allowable limit which is set to be 25 % of the rest length.

We exclude the trajectories that encounter either of these error conditions in our computations of percentage prediction errors. Besides, we calculate a numerical solution of the TSLIP dynamics for every single-step simulation and take it as a nominal-reference result. This reference is called *the ground truth* and found using MATLAB "ode45". In addition, we estimate the approximate apex states based on the methods introduced in Chapter 5 and utilize the error criteria formulation, 6.1 *PE* to investigate and compare estimation performances.

Essentially, if the approximated apex state is in the valid range, we can deduce the performance across multiple strides using the information from a single stride since the prediction errors increase cumulatively over each stride.

6.2 Simulation Results

6.2.1 Predictive Performance of Virtual States

This section clarifies and compares the predictive performance of our analytic approximation approach with and without gravity correction methods proposed by Saranlı [56] and Arslan [45]. We investigate percentage prediction errors in the liftoff velocity and position, $\dot{p}_{lo} := [\dot{x}_{lo}, \dot{z}_{lo}]^T$ and $p_{lo} := [x_{lo}, z_{lo}]^T$, the apex position, $p_a := [x_a, z_a]^T$ and height, z_a , apex energy, E_a and stance time duration, t_s . Note that, to guarantee a reasonable normalization which is valid for different types of gaits (such as gaits with zero apex velocities), we consider the liftoff velocity instead of the apex velocity.

For example, according to the formulation 6.1, the percentage prediction errors of liftoff velocity and apex positions can be defined as

$$PE_{lo_{vel}} = 100 \frac{\|[\dot{x}_{lo}, \dot{z}_{lo}] - [\hat{\dot{x}}_{lo}, \hat{\dot{z}}_{lo}]\|_2}{\|[\dot{x}_{lo}, \dot{z}_{lo}]\|_2} \quad (6.3)$$

$$PE_{a_{pos}} = 100 \frac{\|[x_a, z_a] - [\hat{x}_a, \hat{z}_a]\|_2}{\|[x_a, z_a]\|_2} \quad (6.4)$$

$$(6.5)$$

Note that, " $\hat{\cdot}$ " notation is used for the estimated states. By numerical integration, a single stride of the TSLIP model outputs $[\dot{x}_{lo}, \dot{z}_{lo}]$ and $[x_a, z_a]$, while $[\hat{\dot{x}}_{lo}, \hat{\dot{z}}_{lo}]$ and $[\hat{x}_a, \hat{z}_a]$ are found by model approximation and they represent estimated liftoff and apex states.

Using various initial conditions and parameters in the ranges shown in Table 6.1, we conducted 5,250,000 simulations in total; 3,741,315 of them are accepted as valid. Using data obtained from these valid simulations for two gravity correction methods and no gravity correction case, we provide the mean μ , standard deviation σ , and maximum value *max* of percentage prediction errors in Figure 6.1. Also, in Table 6.2 the corresponding numerical values are given in detail.

Table 6.2: Statistical analysis of prediction performance of our analytic approximation procedure with no gravity correction and using Saranlı's and Arslan's gravity correction methods in liftoff position p_{lo} and velocity \dot{p}_{lo} , apex position p_a and height z_a , apex energy E_a and stance time duration t_s . μ and σ stands for the mean and the standard deviation and *max* gives the maximum value of the related percentage prediction error. Also, simulations results with zero and nonzero body pitch rates are stated on the top and bottom parts of the table, respectively.

	$\dot{\theta} = 0$					
	No Gravity Correction		Saranlı's Method		Arslan's Method	
	$\mu \pm \sigma$	max	$\mu \pm \sigma$	max	$\mu \pm \sigma$	max
p_{lo}	2.00 \pm 1.76	30.15	1.71 \pm 1.82	30.00	2.99 \pm 2.46	30.89
\dot{p}_{lo}	6.58 \pm 4.79	46.11	3.19 \pm 2.14	32.32	3.26 \pm 2.50	40.82
p_a	4.33 \pm 3.63	33.63	2.31 \pm 1.96	33.72	3.24 \pm 2.60	33.28
z_a	1.92 \pm 2.54	55.58	1.60 \pm 2.47	55.22	2.07 \pm 2.75	55.33
E_a	2.02 \pm 1.48	10.80	1.36 \pm 1.16	12.81	1.08 \pm 1.07	14.62
t_s	0.67 \pm 0.53	9.16	0.67 \pm 0.53	9.16	0.67 \pm 0.53	9.16
	$\dot{\theta} \neq 0$					
	No Gravity Correction		Saranlı's Method		Arslan's Method	
	$\mu \pm \sigma$	max	$\mu \pm \sigma$	max	$\mu \pm \sigma$	max
p_{lo}	2.09 \pm 1.76	22.36	1.73 \pm 1.84	19.41	2.95 \pm 2.46	19.61
\dot{p}_{lo}	6.86 \pm 4.67	39.40	3.33 \pm 2.26	23.59	3.24 \pm 2.51	30.15
p_a	4.70 \pm 3.66	33.30	2.55 \pm 2.32	34.08	3.32 \pm 2.79	34.15
z_a	2.18 \pm 2.99	40.04	1.79 \pm 2.93	41.41	2.22 \pm 3.18	41.54
E_a	2.08 \pm 1.45	10.19	1.37 \pm 1.12	11.58	1.03 \pm 1.02	12.84
t_s	0.76 \pm 0.59	6.72	0.76 \pm 0.59	6.72	0.76 \pm 0.59	6.72

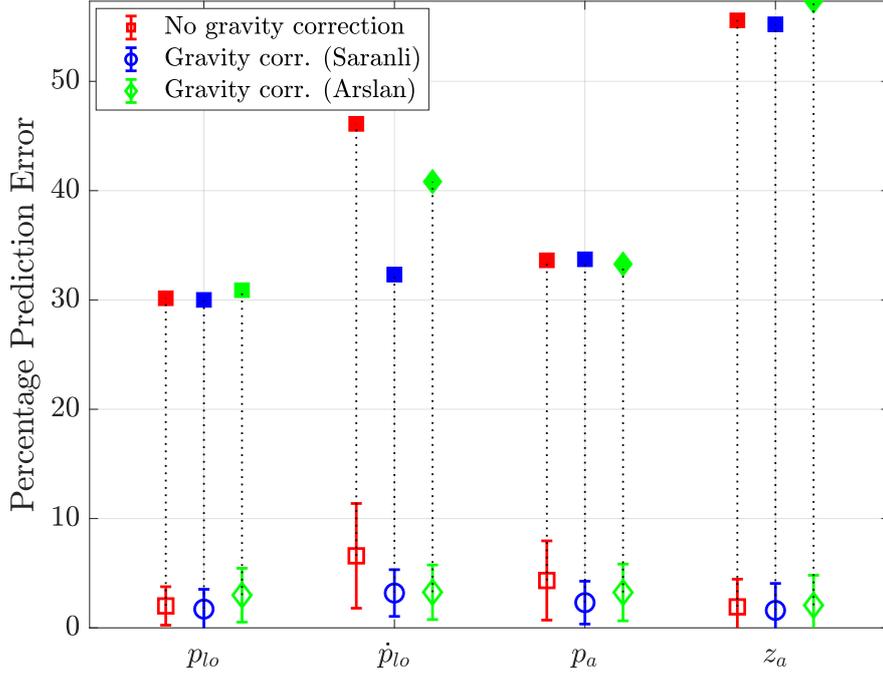


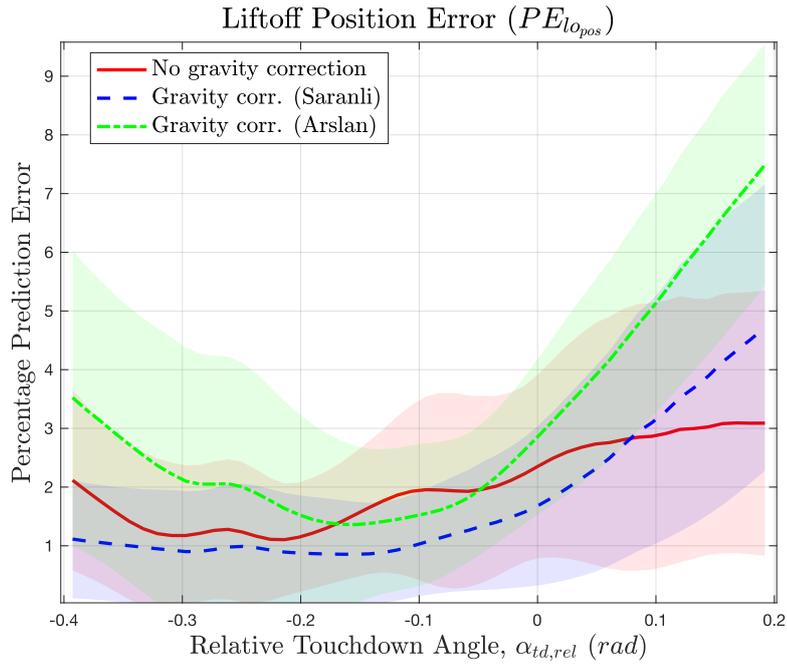
Figure 6.1: Statistical results for the prediction performance of the proposed analytical approximation method in liftoff position p_{lo} and velocity \dot{p}_{lo} , apex position p_a and height z_a without gravity correction and using Saranlı’s and Arslan’s gravity correction methods. These results are obtained taking $\dot{\theta} = 0$ and including both symmetric and non-symmetric strides under zero damping. Hollow markers, filled markers and vertical bars illustrate the related percentage prediction error’s mean, maximum, and standard deviation, respectively.

We can use these statistical results to confirm that gravity correction methods can be employed enhance our approximation scheme’s performance, especially in predicting velocity elements as expected. This is possible due to changing gravity affecting primarily the angular momentum and inevitably the liftoff velocity.

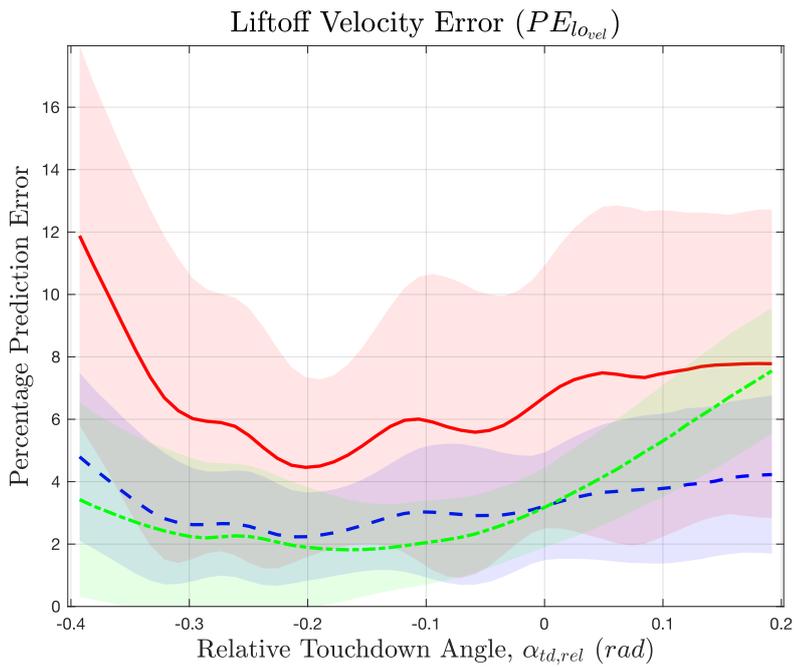
On the other hand, gravity correction does not affect the stance duration since the liftoff time is calculated before the correction takes place, unless we perform another iteration for t_{lo} with the corrected value of $\omega(t_{lo})$. However, performing another iteration is not necessary since the obtained results without gravity correction already keep the stance time prediction errors sufficiently small, around 0.3 – 0.4%.

In order to demonstrate the influence of the analytical approximation scheme on trajectories obtained around fixed points, we analyze the predictive performances as a function of $\alpha_{td, rel}$ as illustrated in Figures 6.2 and 6.3. These figures affirm that correcting gravity reduces the estimation error percentages not only for the fixed point trajectories but also the non-symmetric trajectories. In addition, initially, setting the pitch rate as $\dot{\theta}_0 \neq 0$, causes a deterioration in the prediction of the subsequent apex state. Therefore, for the following simulations we selected the pitch rate as zero since it will eventually converge to zero due to the gait controller, which tries to stabilize the pitch angle.

When we consider the mean values of the prediction errors concerning all simulations, applying gravity correction on angular momentum improves the predictive performance. As stated in Table 6.2, with gravity correction, the average predictive errors remain below 4%. As observed from Figures 6.2 and 6.3, the gravity correction approaches proposed by Saranlı [56] and Arslan [45] are comparable in terms of prediction performance. However, Saranlı’s method performs slightly better than Arslan’s since the method introduced by Saranlı gives a consistent performance profile and stands stable even for extreme touchdown angles. In contrast, Arslan’s approach deviates and gives high prediction errors, especially for non-symmetric trajectories with major touchdown angles. As expected, both algorithms give accurate prediction results and minimum percentage errors around the symmetric trajectories, including the span of relative touchdown angle $\alpha_{rel, td} \in [-0.2, 0.05]$ rad. Also, Saranlı’s algorithm provides more accurate predictions for some asymmetrical trajectory regions far outside the mentioned touchdown angle range. In addition, the touchdown angles outside this range correspond to very abrupt transitions in motion, but they can also be ignored by a suitable controller conscious of current approaches’ limitations. Therefore, the gravity correction scheme can be selected considering the environment and operating requirements of the robot.

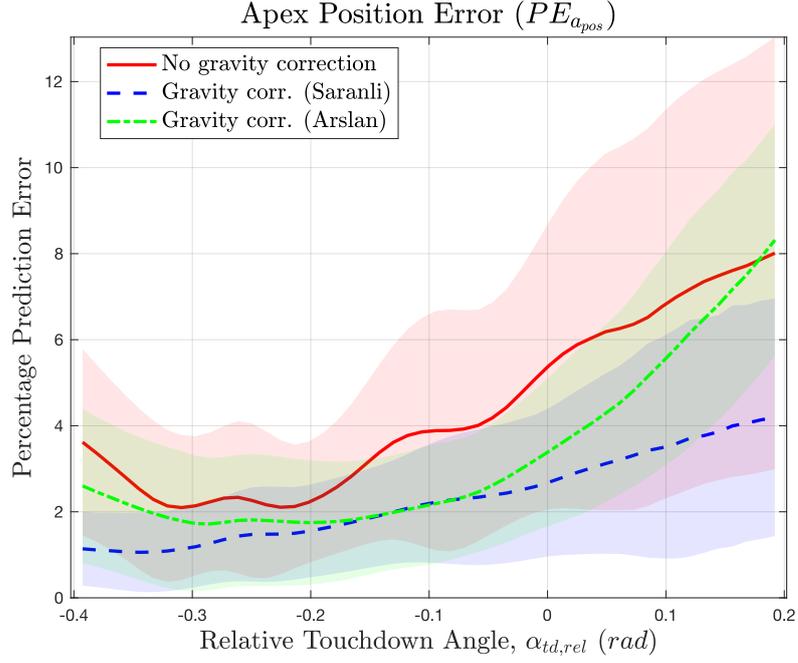


(a)

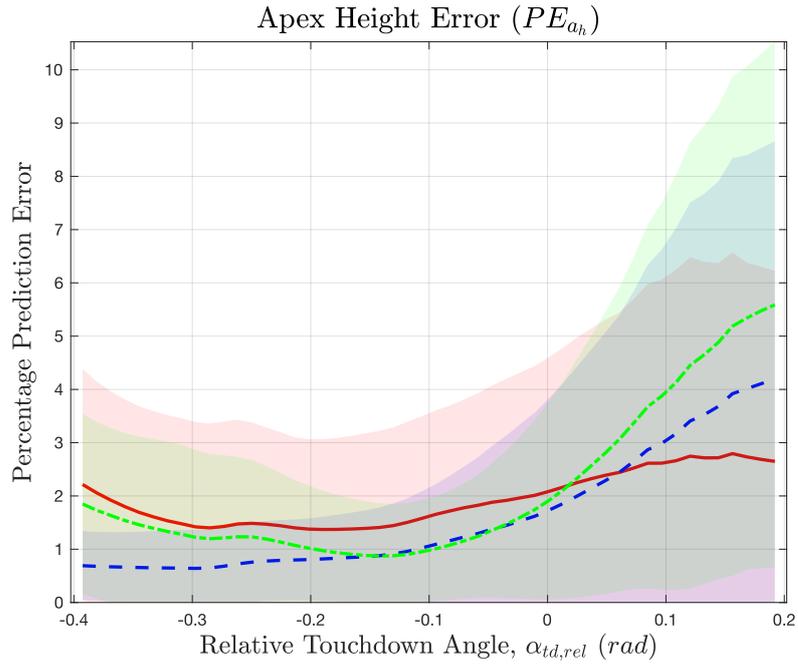


(b)

Figure 6.2: A statistical representation for percentage prediction errors of **(a)** liftoff position p_{l_o} and **(b)** velocity \dot{p}_{l_o} as a function of the relative touchdown angle $\alpha_{td,rel}$. Lines represent mean values of the percentage prediction errors, while the standard deviations are represented by shaded and colored regions of the associated gravity correction method.



(a)

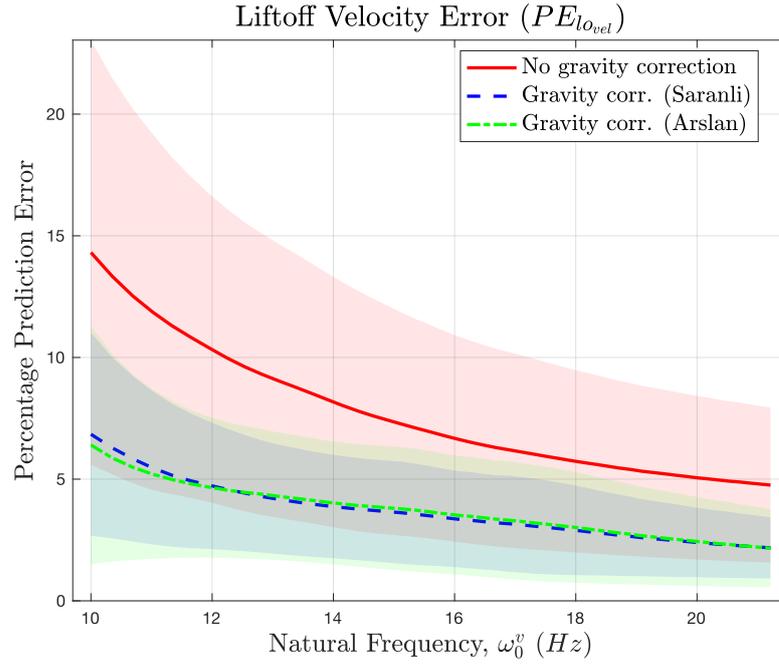


(b)

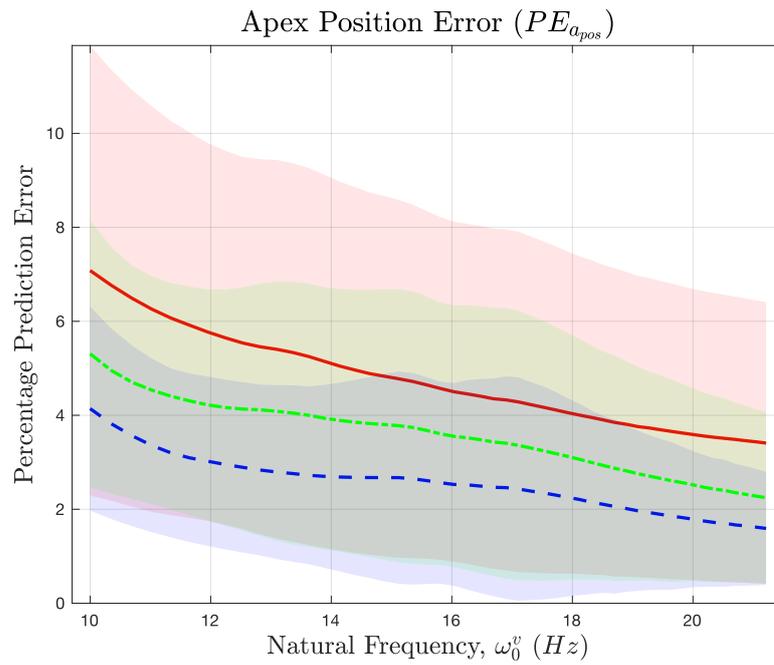
Figure 6.3: A statistical representation for percentage prediction errors of **(a)** apex position p_a and **(b)** height z_a as a function of the relative touchdown angle $\alpha_{td,rel}$. Lines represent mean values of the percentage prediction errors, while the standard deviations are represented by shaded and colored regions of the associated gravity correction method.

Having a monotonically decreasing PE graph (Figure 6.4) with increasing natural frequency ω_0^v provides great convenience when designing advanced control algorithms, as well as having significant benefits when performing the locomotion stability analysis. Since ω_0^v is a function of the virtual stiffness, we can achieve adequate prediction results by adjusting the physical spring parameters. For instance, we have better estimation performance when the leg stiffness is high, which leads to looser legs. Accordingly, a flexible leg causing relatively large spring compression reduces the prediction accuracy. In addition, as seen in this Figure, even though Saranlı's and Arslan's methods perform almost identical for velocity components with the varying natural frequency, they become distinct in terms of prediction accuracy of position components.

In order to bring additional clarification on the parameter dependence of prediction errors, we observe the PE distribution of important stance variables over the control inputs (natural frequency and relative touchdown angle). Figure 6.5 demonstrate the regions in the control input space where the different gravity correction methods produce the minimum percentage estimation error. For varying $(\alpha_{rel,td}, \omega_0^v)$, the corresponding gravity correction methods are comparable for predicting liftoff velocity, but both of them generally give lower prediction errors than no correction case gives. Nevertheless, for a small region where the natural frequency is low and the touchdown angle is close to its nominal value, ignoring gravity correction is appropriate in terms of the accuracy of apex position prediction. Also, Arslan's method for compensating gravitational effects performs better for the proximity of nominal touchdown angle for predicting apex position, while Saranlı's method provides more reasonable results for a broader region.



(a)



(b)

Figure 6.4: A statistical representation for percentage prediction errors of **(a)** liftoff velocity \dot{p}_{lo} and **(b)** apex position p_a as a function of the natural frequency ω_0^v . Lines represent mean values of the percentage prediction errors, while the standard deviations are represented by shaded and colored regions of the associated gravity correction method.

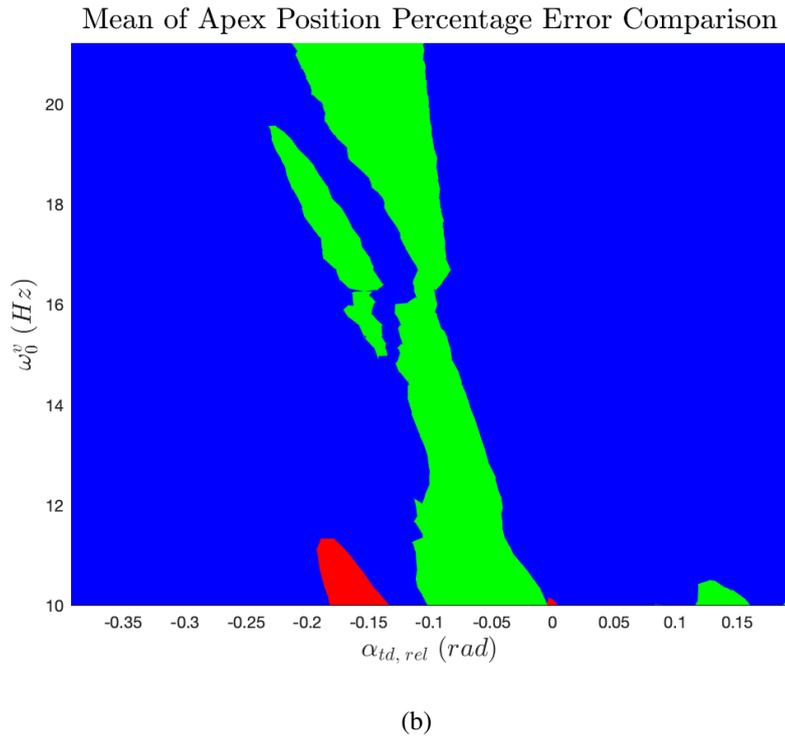
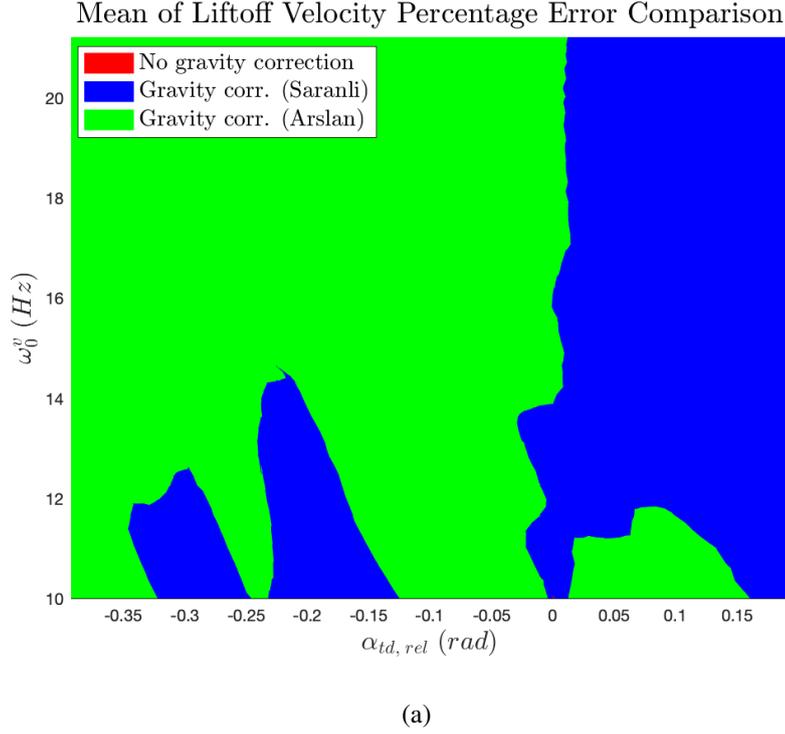
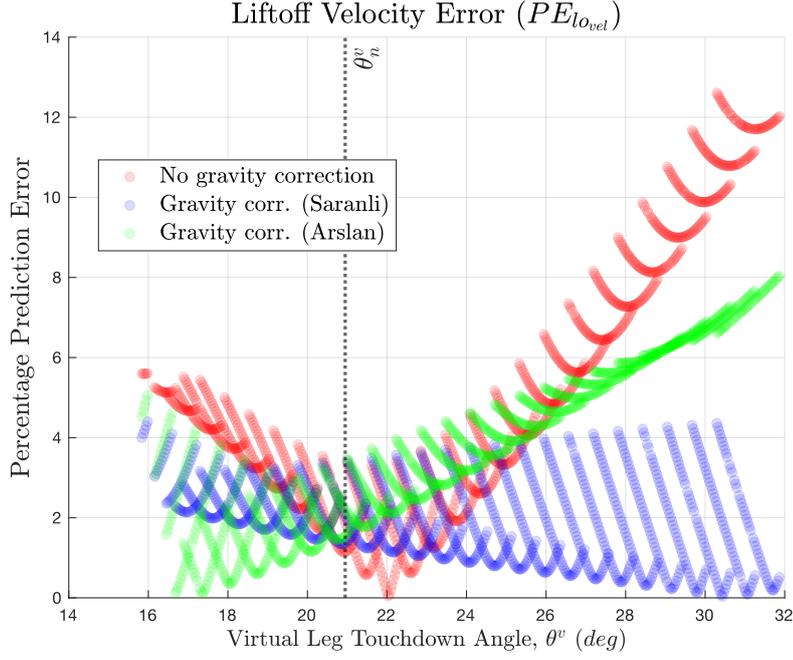


Figure 6.5: A performance comparison predicting (a) liftoff velocity \dot{p}_{lo} and (b) apex position p_a using three gravity correction methods. Colored areas indicate the correction method, which gives the minimum mean of the error percentage in that combination of natural frequency (ω_0^v) and relative touchdown angle $\alpha_{td,rel}$.

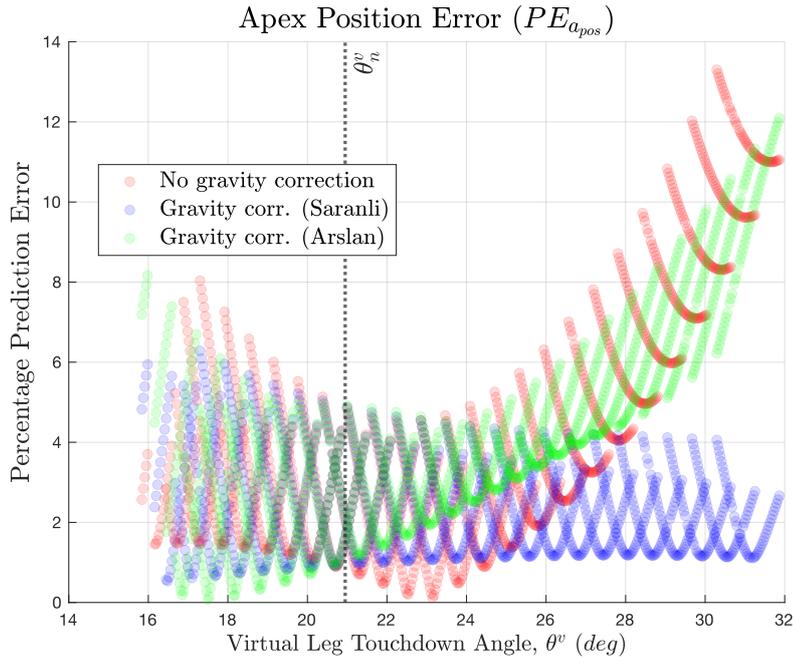
Our final observation on the model's predictive performance is, its relation to the virtual polar states, particularly to the virtual touchdown angle θ^v . We can reduce the dimension of the system by one, by expressing θ^v as an analytical combination of physical touchdown angle and the body pitch angle (for details, see appendix, section B.3). Therefore, the influence of two control parameters can be fused into one parameter.

To observe the effects of θ^v on prediction performance, we found a reference θ_n^v value regarding a periodic solution of the system. Then, we start the system from this periodic solution and disturb it around this reference value by sweeping θ_n^v in a range defined for α and θ . As an indication of virtual states' influence on the prediction, the distribution of liftoff velocity prediction error versus the virtual touchdown angle θ^v graph is provided in Figure 6.6a.

α dominates the general distribution of the PE vs. θ^v graphs, while the minor parabolic distributions represent the θ dependence. Also, each short parabolic curve denotes the θ distribution while α is kept constant. The general distributions of "no gravity correction" graphs given in 6.6 imply that the more reasonable prediction performance is achieved with θ^v values closer to the nominal value, as expected. Diverging from this nominal value in both ways, decreasing and increasing, reduces the prediction accuracy. That signifies the prediction performance with no gravity correction is susceptible under the significant changes of the virtual leg swept. On the contrary, compensating gravitational effects result in more stable percentage prediction graphs with varying θ^v .



(a)



(b)

Figure 6.6: Percentage prediction errors regarding **(a)** the liftoff velocity and **(b)** apex position versus the virtual leg state, θ^v . θ_n^v denotes the natural virtual leg touchdown angle, corresponding the fixed point state $\mathbf{X}^* = [1.2301, 4.1611, 0, 0]^T$ and natural touchdown angle $\alpha_{td,n} \approx 113 \text{ deg}$.

6.2.2 Stability Analysis of the Fixed Points

Surveying open-loop stability characteristics of legged locomotion models can lead to the reasons why they are widely used by legged runners and also provide experimental verifications explicitly [63]. In this section of the thesis, we isolate the fixed points of the undamped spring-mass model embedded TSLIP and characterize the properties of the corresponding periodic solutions by providing the results of systematically performed simulations. In these simulations, we followed the same procedure described in Section 3.4.1. We set the controller parameters to their nominal values $K_p = 0$ and $K_d = 0$ to find the fixed points properly by running the optimization algorithm.

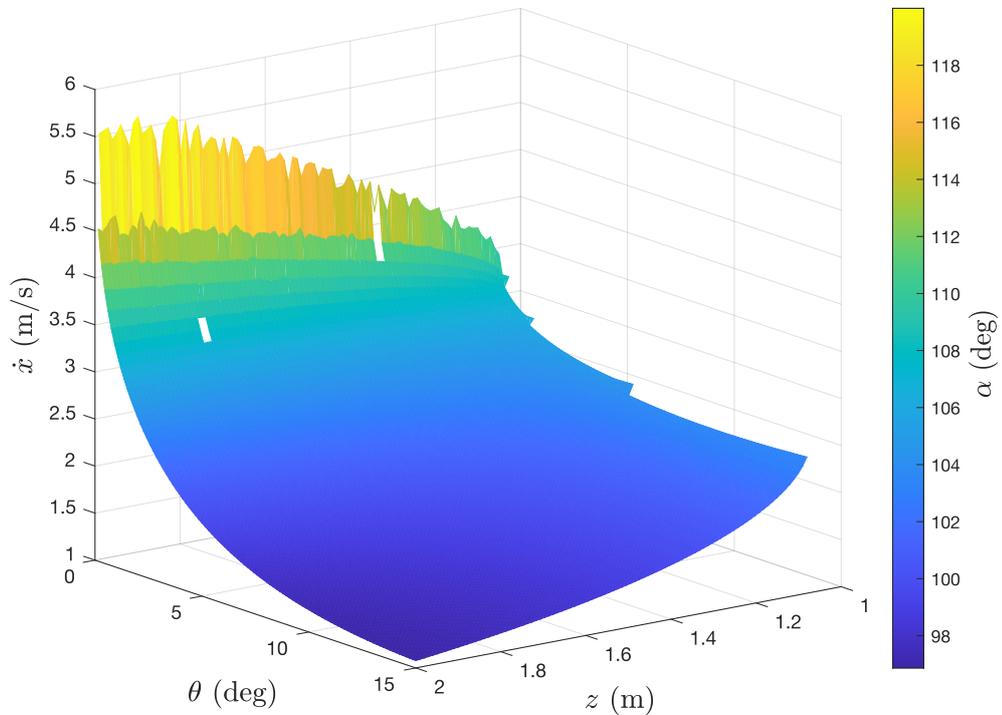


Figure 6.7: The fixed point subspace variation by the touchdown leg angle, α .

The colored illustrations of the fixed-point manifolds provided in this section help visualizing the fixed-point characteristics. 6.7 represents the distribution of fixed points in a broad region determined by control parameters including α . The significant difference between the eigenvalue analysis of "the CPP integrated TSLIP model" and "the undamped spring-mass model mass embedded TSLIP model" is the existence

of fixed points for different body lean angles. In the CPP case, fixed points are found only when the lean angle is zero. Embedding the undamped spring-mass model broadens the fixed point state-space and enhances the maneuverability of the system, such as operating at various state conditions, including forward speed, height, and pitch angle.

Investigating the behavior of eigenvalues calculated from Jacobian matrices is highly used to describe the stability of the fixed points. Fixed points that are represented by maximum eigenvalues λ_{max} below unity are stable. By using this fact, we analyze the stability of the fixed points illustrated in Figure 6.7 and provide a distribution of the associated maximum eigenvalues in Figure 6.8. As observed, the entire set of the fixed points is unstable with minimum value of maximum eigenvalue set, $|\lambda_{max}| \approx 1.0553$. Due to the unstable behavior of fixed points, a controller action is needed to stabilize the system and operate it on the desired trajectories.

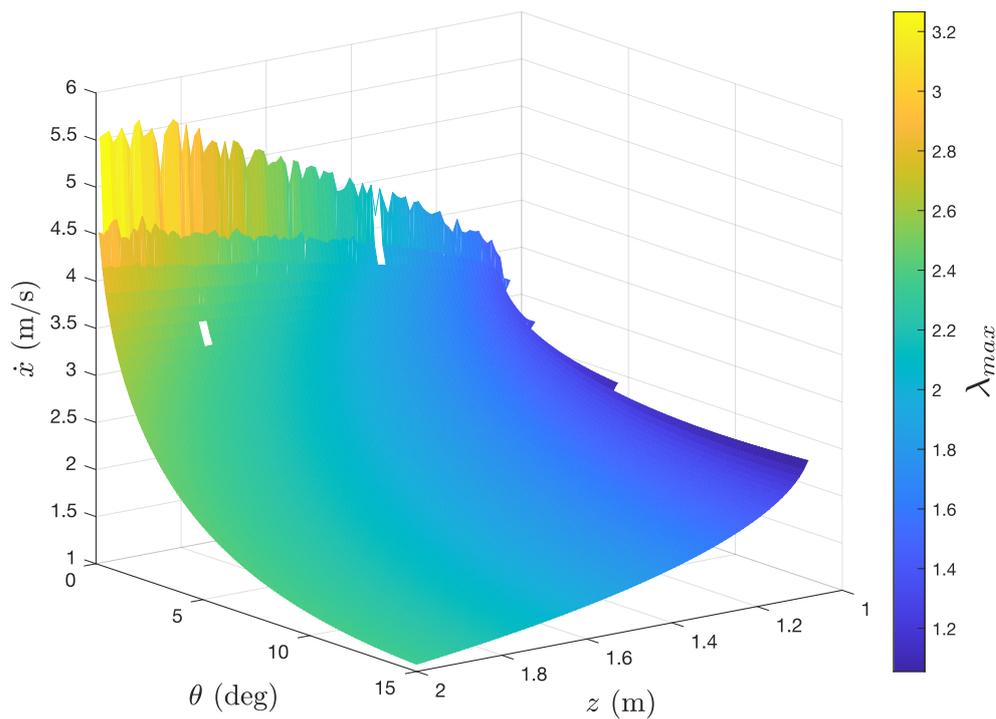


Figure 6.8: Maximum eigenvalues, λ_{max} of the Jacobi matrix calculated by using the fixed points given in Figure 6.8.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

This thesis aims to encapsulate our work on a specialized field of legged robotics. We started from the description of a body attached underactuated spring-mass model called the Trunk-SLIP (TSLIP) and analyzed its dynamics. A novel scheme for the analysis of this model, called Central Pivot Point (CPP) which is used to stabilize the body dynamics was presented, and the analytical approximation of TSLIP's stance map was generated.

In the first part of this thesis, we presented the CPP concept for controlling the TSLIP model, including pitch dynamics. We systematically analyzed the system's fixed points and showed that the majority of the fixed points of the model are unstable. Using the fact that CPP separates the pitch dynamics and euclidean dynamics, we developed a high gain PD controller for pitch dynamics and an LQR for euclidean dynamics. We demonstrated that the proposed controller structure could stabilize all the fixed points of the model. Also, we tested the performance of the gait controller under disturbances. We illustrated the existence and stability of periodic solutions with a large domain of attraction through many simulation experiments and established that our control structure could reject even radical disturbances to states. Overall, we showed the effectiveness and robustness of the gait controller on the TSLIP template.

Many methods of the existing literature use numerical integration and iteration-based methods to explain the stance phase dynamics of the legged platforms. Also, the available methods concentrate on the symmetrical strides of legged platforms' conservative versions during steady-state locomotion. Such constraints significantly decrease the applicability of these platforms' design and control on uneven ground, where steps can deviate from periodicity. Therefore, in the second part of the the-

sis, to avoid such methods' dynamical and practical complexity, we proposed a radial actuation on partial feedback linearization-based method to derive an approximate analytical solution for the stance map of the TSLIP model. Comprising solely of fundamental formulations, the developed closed-form approximation could analytically and accurately describe the non-integrable trajectories of our model's dynamics. To verify the performance and applicability of our proposal on the approximate map, we ran comprehensive simulations that cover an extensive range of parameter combinations and operating conditions. The evaluation of the simulations' performance showed that our method of approximation could provide precise estimations for the critical states of the apex return map and stance trajectories of the TSLIP. Also, the approximate apex return map enables parametric synthesis and smooth operation with high-performance control algorithms such as model predictive control and deadbeat control. In addition to approximation, we applied two different gravity correction methods available in the literature to compensate for the gravitational effects on angular momentum. We compared the results with or without gravity correction stacked on our analytic approximation method. The correction on gravity further enhanced the estimation performance of the proposed analytic approximation, particularly in the cases of non-symmetric trajectories. For almost all valid simulations, percentage prediction errors remain below 4%, even in extreme conditions.

Finally, we investigated the existence of the extended model's fixed points and their dependency concerning the state variables and control parameters by using numerical methods. We revealed that most of the fixed points of the model are unstable, which is the first reason that points out the necessity of a successful gait-level running controller. Also, we can use our analytic approximation results to investigate stability characteristics of our legged locomotion model by generalizing the early practices of Poincare methods [43, 44, 64–66], since it is challenging to analyze the stability of the model, which suffers from non-symmetric trajectories, in the absence of an exact analytic apex return map [40]. Moreover, we rely on our proposal for the extended model's analytical representation to be utilized in designing and analyzing dexterous locomotion controllers on irregular terrain. Therefore, we plan to construct a high-performance controller to stabilize all the fixed points.

In some practical systems, it might also be the case that the center of mass is lower

than the hip point. Although we have concentrated on the TSLIP configuration, our methodology can should be applicable to the aforementioned case as the subject of future research. Finally, embedding the TSLIP model can physically implement the locomotion control scheme to more complex legged platforms. To this end, our long-range plan involves using the CPP concept and analytic approximation of TSLIP as a control target for humanoid systems in a simulation environment.

REFERENCES

- [1] M. Ahmadi and M. Buehler, “Controlled passive dynamic running experiments with the arl-monopod ii,” *IEEE Transactions on Robotics*, vol. 22, 2006.
- [2] K. Sreenath, H. W. Park, I. Poulakakis, and J. W. Grizzle, “A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel,” *International Journal of Robotics Research*, vol. 30, 2011.
- [3] W. C. Martin, A. Wu, and H. Geyer, “Robust spring mass model running for a physical bipedal robot,” vol. 2015-June, 2015.
- [4] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, “Efficient bipedal robots based on passive-dynamic walkers,” *Science*, vol. 307, 2005.
- [5] M. Hutter, C. Gehring, M. A. Höpflinger, M. Blösch, and R. Siegwart, “Toward combining speed, efficiency, versatility, and robustness in an autonomous quadruped,” 2014.
- [6] H. W. Park, P. M. Wensing, and S. Kim, “High-speed bounding with the mit cheetah 2: Control design and experiments,” *International Journal of Robotics Research*, vol. 36, 2017.
- [7] C. Semini, V. Barasuol, T. Boaventura, M. Frigerio, M. Focchi, D. G. Caldwell, and J. Buchli, “Towards versatile legged robots through active impedance control,” *International Journal of Robotics Research*, vol. 34, 2015.
- [8] T. Kubow and R. Full, “The role of the mechanical system in control: a hypothesis of self-stabilization in hexapedal runners,” *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 354, no. 1385, pp. 849–861, 1999.
- [9] M. H. Dickinson, C. T. Farley, R. J. Full, M. Koehl, R. Kram, and S. Lehman, “How animals move: an integrative view,” *science*, vol. 288, no. 5463, pp. 100–106, 2000.

- [10] C. Li, T. Zhang, and D. I. Goldman, “A terradynamics of legged locomotion on granular media,” *science*, vol. 339, no. 6126, pp. 1408–1412, 2013.
- [11] R. J. Full and D. E. Koditschek, “Templates and anchors: neuromechanical hypotheses of legged locomotion on land,” *Journal of experimental biology*, vol. 202, no. 23, pp. 3325–3332, 1999.
- [12] M. Ahmadi and M. Buehler, “The arl monopod ii running robot: Control and energetics,” in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 3, pp. 1689–1694, IEEE, 1999.
- [13] A. Hereid, S. Kolathaya, M. S. Jones, J. Van Why, J. W. Hurst, and A. D. Ames, “Dynamic multi-domain bipedal walking with arias through slip based human-inspired control,” in *Proceedings of the 17th international conference on Hybrid systems: computation and control*, pp. 263–272, ACM, 2014.
- [14] U. Saranlı and D. E. Koditschek, “Template based control of hexapedal running,” in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 1, pp. 1374–1379, IEEE, 2003.
- [15] M. M. Ankaralı and U. Saranlı, “Control of underactuated planar pronking through an embedded spring-mass hopper template,” *Autonomous Robots*, vol. 30, no. 2, pp. 217–231, 2011.
- [16] W.-C. Lu, M.-Y. Yu, and P.-C. Lin, “Clock-torqued rolling slip model and its application to variable-speed running in a hexapod robot,” *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1643–1650, 2018.
- [17] R. Blickhan, “The spring-mass model for running and hopping,” *Journal of Biomechanics*, vol. 22, 1989.
- [18] T. A. McMahon and G. C. Cheng, “The mechanics of running: How does stiffness couple with speed?,” *Journal of Biomechanics*, vol. 23, pp. 65–78, 1990. International Society of Biomechanics.
- [19] B. Brown and G. Zeglin, “The bow leg hopping robot,” vol. 1, 1998.
- [20] W. J. Schwind, *Spring loaded inverted pendulum running: A plant model*. PhD thesis, 1998.

- [21] U. Saranlı, M. Buehler, and D. E. Koditschek, “Rhex: A simple and highly mobile hexapod robot,” *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 616–631, 2001.
- [22] P. Holmes, R. J. Full, D. Koditschek, and J. Guckenheimer, “The dynamics of legged locomotion: Models, analyses, and challenges,” *SIAM review*, vol. 48, no. 2, pp. 207–304, 2006.
- [23] B. Brown and G. Zeglin, “The bow leg hopping robot,” in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, vol. 1, pp. 781–786, IEEE, 1998.
- [24] N. Shemer and A. Degani, “Analytical control parameters of the swing leg retraction method using an instantaneous slip model,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4065–4070, IEEE, 2014.
- [25] W. C. Martin, A. Wu, and H. Geyer, “Experimental evaluation of deadbeat running on the atrias biped,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1085–1092, 2017.
- [26] I. Poulakakis and J. W. Grizzle, “The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper,” *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1779–1793, 2009.
- [27] I. Poulakakis and J. Grizzle, “Formal embedding of the spring loaded inverted pendulum in an asymmetric hopper,” in *2007 European Control Conference (ECC)*, pp. 3159–3166, IEEE, 2007.
- [28] M. A. Sharbafi, C. Maufroy, M. N. Ahmadabadi, M. J. Yazdanpanah, and A. Seyfarth, “Robust hopping based on virtual pendulum posture control,” *Bioinspiration & biomimetics*, vol. 8, no. 3, p. 036002, 2013.
- [29] G. Bayır, *Identification and stability analysis of periodic motions for a planar legged runner with a rigid body and a compliant leg*. PhD thesis, Bilkent University, 2013.

- [30] H.-M. Maus, S. 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, p. 70, 2010.
- [31] J. Pratt and G. Pratt, “Intuitive control of a planar bipedal walking robot,” vol. 3, 1998.
- [32] M. H. Raibert, *Legged robots that balance*. MIT press, 1986.
- [33] A. Sato and M. Buehler, “A planar hopping robot with one actuator: Design, simulation, and experimental results,” vol. 4, 2004.
- [34] J. G. Karssen, M. Haberland, M. Wisse, and S. Kim, “The optimal swing-leg retraction rate for running,” 2011.
- [35] B. Miller, J. Schmitt, and J. E. Clark, “Quantifying disturbance rejection of slip-like running systems,” vol. 31, 2012.
- [36] J. Schmitt and J. Clark, “Modeling posture-dependent leg actuation in sagittal plane locomotion,” *Bioinspiration and Biomimetics*, vol. 4, 2009.
- [37] H. Yu, H. Gao, and Z. Deng, “Toward a unified approximate analytical representation for spatially running spring-loaded inverted pendulum model,” *IEEE Transactions on Robotics*, vol. 37, 2021.
- [38] A. Wu and H. Geyer, “The 3-d spring-mass model reveals a time-based deadbeat control for highly robust running and steering in uncertain environments,” *IEEE Transactions on Robotics*, vol. 29, 2013.
- [39] U. Saranlı, Ö. Arslan, M. M. Ankaralı, and Ö. Morgül, “Approximate analytic solutions to non-symmetric stance trajectories of the passive spring-loaded inverted pendulum with damping,” *Nonlinear Dynamics*, vol. 62, no. 4, pp. 729–742, 2010.
- [40] 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.
- [41] S. G. Carver, *Control of a spring-mass hopper*. PhD thesis, Cornell University, 2003.

- [42] W. J. Schwind and D. E. Koditschek, “Approximating the stance map of a 2-dof monopod runner,” *Journal of Nonlinear Science*, vol. 10, 2000.
- [43] R. M. Ghigliazza, R. Altendorfer, P. Holmes, and D. Koditschek, “A simply stabilized running model,” *SIAM review*, vol. 47, no. 3, pp. 519–549, 2005.
- [44] H. Geyer, A. Seyfarth, and R. Blickhan, “Spring-mass running: simple approximate solution and application to gait stability,” *Journal of Theoretical Biology*, vol. 232, no. 3, pp. 315–328, 2005.
- [45] Ö. Arslan, U. Saranlı, and Ö. Morgül, “An approximate stance map of the spring mass hopper with gravity correction for nonsymmetric locomotions,” in *2009 IEEE International Conference on Robotics and Automation*, pp. 2388–2393, 2009.
- [46] M. Shahbazi, R. Babuska, and G. A. Lopes, “Unified modeling and control of walking and running on the spring-loaded inverted pendulum,” *IEEE Transactions on Robotics*, vol. 32, 2016.
- [47] H. Yu, M. Li, and H. Cai, “Approximating the stance map of the slip runner based on perturbation approach,” 2012.
- [48] G. Piovan and K. Byl, “Approximation and control of the slip model dynamics via partial feedback linearization and two-element leg actuation strategy,” *IEEE Transactions on Robotics*, vol. 32, 2016.
- [49] O. K. Karagöz, I. Sever, U. Saranlı, and M. M. Ankaralı, “Analysis and control of a body-attached spring-mass runner based on central pivot point approach,” vol. 2020-July, 2020.
- [50] P. Holmes, “Poincaré, celestial mechanics, dynamical-systems theory and “chaos”,” *Physics Reports*, vol. 193, no. 3, pp. 137–163, 1990.
- [51] G. C. Goodwin, S. F. Graebe, M. E. Salgado, *et al.*, *Control system design*. Upper Saddle River, NJ: Prentice Hall., 2001.
- [52] Ö. Arslan and U. Saranlı, “Reactive planning and control of planar spring–mass running on rough terrain,” *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 567–579, 2012.

- [53] T. Cnops, Z. Gan, and C. D. Remy, “The basin of attraction for running robots: Fractals, multistep trajectories, and the choice of control,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1586–1591, IEEE, 2015.
- [54] G. Seçer and U. Saranlı, “Control of planar spring–mass running through virtual tuning of radial leg damping,” *IEEE Transactions on Robotics*, no. 99, pp. 1–14, 2018.
- [55] J. R. Dormand and P. J. Prince, “A family of embedded runge-kutta formulae,” *Journal of Computational and Applied Mathematics*, vol. 6, 1980.
- [56] U. Saranlı, Ö. Arslan, M. M. Ankaralı, and Ö. Morgül, “Approximate analytic solutions to non-symmetric stance trajectories of the passive spring-loaded inverted pendulum with damping,” *Nonlinear Dynamics*, vol. 62, 2010.
- [57] H. Hamzaçebi and Ö. Morgül, “On the periodic gait stability of a multi-actuated spring-mass hopper model via partial feedback linearization,” *Nonlinear Dynamics*, vol. 88, pp. 1237–1256, 4 2017.
- [58] U. Saranlı, *Dynamic locomotion with a hexapod robot*. PhD thesis, University of Michigan, Ann Arbor, 2002.
- [59] R. Altendorfer, D. E. Koditschek, and P. Holmes, “Stability analysis of legged locomotion models by symmetry-factored return maps,” vol. 23, 2004.
- [60] A. Arampatzis, G.-P. Brüggemann, and V. Metzler, “The effect of speed on leg stiffness and joint kinetics in human running.,” *Journal of Biomechanics*, vol. 32, pp. 1349–1353, 1999.
- [61] R. Müller, M. Ernst, and R. Blickhan, “Leg adjustments during running across visible and camouflaged incidental changes in ground level,” *Journal of Experimental Biology*, vol. 215, 2012.
- [62] C. R. Lee and C. T. Farley, “Determinants of the center of mass trajectory in human walking and running,” *Journal of Experimental Biology*, vol. 201, 1998.
- [63] D. L. Jindrich and R. J. Full, “Many-legged maneuverability: Dynamics of turning in hexapods,” *Journal of Experimental Biology*, vol. 202, 1999.

- [64] J. Schmitt, “A simple stabilizing control for sagittal plane locomotion,” *Journal of Computational and Nonlinear Dynamics*, vol. 1, 2006.
- [65] S. G. Carver, N. J. Cowan, and J. M. Guckenheimer, “Lateral stability of the spring-mass hopper suggests a two-step control strategy for running,” *Chaos*, vol. 19, 2009.
- [66] H. Geyer, A. Seyfarth, and R. Blickhan, “Compliant leg behaviour explains basic dynamics of walking and running,” *Proceedings of the Royal Society B: Biological Sciences*, vol. 273, 2006.

Appendix A

DERIVATION OF THE EQUATIONS OF MOTION

We used the Lagrangian Method to derive the equations of motion of the system. The Lagrangian, L is expressed as a combination of kinetic energy, K and potential energy, P due to gravity, see Equation A.1. The procedure for finding the EOM is as follows

$$\begin{aligned}K &= \frac{1}{2}m(\dot{x}^2 + \dot{z}^2) + J\dot{\theta}^2 \\P &= mgz \\L &\equiv K - P\end{aligned}\tag{A.1}$$

$$\frac{d}{dt} \left(\frac{\delta L}{\delta \dot{q}_i} \right) - \frac{\delta L}{\delta q_i} = f(F_r, \tau), \quad i = 1, 2, 3.\tag{A.2}$$

Function $f(F_r, \tau)$ is basically the matrix multiplication of the generalized force matrix, $\begin{bmatrix} D_q r & D_q \psi \end{bmatrix}$ with the externally applied forces and torques $\begin{bmatrix} F_r \\ \tau \end{bmatrix}$ to the system.

$$\begin{aligned}f(F_r, \tau) &= \begin{bmatrix} D_q r & D_q \psi \end{bmatrix} \begin{bmatrix} F_r \\ \tau \end{bmatrix} \\ &= D_q r F_r + D_q \psi \tau\end{aligned}\tag{A.3}$$

where $D_q r = \frac{\delta r}{\delta q_i}$ and $D_q \psi = \frac{\delta \psi}{\delta q_i}$.

A.1 Cartesian Dynamics

By using the Euler-Lagrange Equation, (A.2) in cartesian coordinates, the left hand side, LHS_{EOM}^c and right hand side, RHS_{EOM}^c of the EOM expression 2.2 yield

$$LHS_{EOM}^c = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J \end{bmatrix} \ddot{q}^v + \begin{bmatrix} 0 \\ mg \\ 0 \end{bmatrix} \quad (A.4)$$

$$RHS_{EOM}^c = D_{q^c} r F_r + D_{q^c} \psi \tau$$

To determine $D_{q^c} r$ and $D_{q^c} \psi$, we need the expressions for r and ψ in terms of the cartesian state variables.

$$\begin{aligned} r &= \sqrt{x_{hip}^2 + z_{hip}^2} \\ \psi &= \alpha + \theta + \frac{\pi}{2} \end{aligned} \quad (A.5)$$

$$\begin{aligned} x_{hip} &= x - d_{hip} \sin(\theta) \\ z_{hip} &= z - d_{hip} \cos(\theta) \\ \alpha &= \text{atan2}(z_{hip}, x_{hip}) \end{aligned} \quad (A.6)$$

where (x_{hip}, z_{hip}) and α stand for the hip position and the leg angle with respect to the ground. After substituting A.6 in expression, A.5 and taking q_i^c derivative of r and ψ , $D_{q^c} r$ and $D_{q^c} \psi$ become

$$\begin{aligned} D_{q^c} r &= \begin{bmatrix} \frac{-d_{hip} \sin(\theta) + x}{\sqrt{(-d_{hip} \sin(\theta) + x)^2 + (-d_{hip} \cos(\theta) + z)^2}} \\ \frac{-d_{hip} \cos(\theta) + z}{\sqrt{(-d_{hip} \sin(\theta) + x)^2 + (-d_{hip} \cos(\theta) + z)^2}} \\ \frac{d_{hip} (-\cos(\theta)x + \sin(\theta)z)}{\sqrt{(-d_{hip} \sin(\theta) + x)^2 + (-d_{hip} \cos(\theta) + z)^2}} \end{bmatrix} \\ D_{q^c} \psi &= \begin{bmatrix} \frac{d_{hip} \cos(\theta) - z}{(-d_{hip} \sin(\theta) + x)^2 + (-d_{hip} \cos(\theta) + z)^2} \\ \frac{-d_{hip} \sin(\theta) + x}{(-d_{hip} \sin(\theta) + x)^2 + (-d_{hip} \cos(\theta) + z)^2} \\ 1 + \frac{d_{hip} (-d_{hip} + \sin(\theta)x + \cos(\theta)z)}{d_{hip}^2 - 2d_{hip} \sin(\theta)x + x^2 - 2d_{hip} \cos(\theta)z + z^2} \end{bmatrix} \cdot \end{aligned} \quad (A.7)$$

A.2 Virtual Leg Dynamics

By using the Euler-Lagrange Equation, (A.2), in the reference of the virtual leg coordinates, the left hand side, LHS_{EOM}^v and right hand side, RHS_{EOM}^v of the EOM expression 2.2 are found as

$$LHS_{EOM}^v = \begin{bmatrix} m \left(g \cos(\theta^v) - r^v (\dot{\theta}^v)^2 + \ddot{r}^v \right) \\ m r^v \left(-g \sin(\theta^v) + 2\dot{r}^v \dot{\theta}^v + r^v \ddot{\theta}^v \right) \\ J\ddot{\theta} \end{bmatrix} \quad (A.8)$$

$$RHS_{EOM}^v = D_{q^v} r F_r + D_{q^v} \psi \tau$$

In virtual leg reference frame, we need expressions for r and ψ in terms of the virtual state variables in order to determine $D_{q^v} r$ and $D_{q^v} \psi$. For this end, cartesian body states are mapped with a transition, $T_{c \rightarrow v}$ to the virtual states.

$$q^c = T_{c \rightarrow v}(q^v) := \begin{bmatrix} x \\ z \\ \theta \end{bmatrix} = \begin{bmatrix} -r^v \sin(\theta^v) \\ r^v \cos(\theta^v) \\ \theta \end{bmatrix} \quad (A.9)$$

Recalling the equations of A.5, A.6 and A.9, r and ψ in terms of q^v variables are defined as stated in Equation A.10. This expressions can also be verified using '*The Cosine Theorem*' and some geometric features (details are given in Appendix B).

$$r = \sqrt{d_{hip}^2 - 2d_{hip}r^v \cos(\theta + \theta^v) + (r^v)^2} \quad (A.10)$$

$$\psi = \frac{\pi}{2} + \text{atan2}(-d_{hip} \cos(\theta) + \cos(\theta^v)r^v, -d_{hip} \sin(\theta) - r^v \sin(\theta^v)) + \theta$$

By taking q_i^v derivative of r and ψ , the Jacobian matrices, $D_{q^v} r$ and $D_{q^v} \psi$ are defined as in Equation A.11.

$$\begin{aligned}
D_{q^v} r &= \begin{bmatrix} \frac{-d_{hip} \cos(\theta + \theta^v) + r^v}{\sqrt{d_{hip}^2 - 2d_{hip} \cos(\theta + \theta^v) r^v + (r^v)^2}} \\ (d_{hip} r^v \text{Sin}[\theta + \theta^v]) \\ \frac{-d_{hip} \cos(\theta + \theta^v) + r^v}{\sqrt{d_{hip}^2 - 2d_{hip} \cos(\theta + \theta^v) r^v + (r^v)^2}} \\ (d_{hip} r^v \text{Sin}[\theta + \theta^v]) \\ \frac{-d_{hip} \cos(\theta + \theta^v) + r^v}{\sqrt{d_{hip}^2 - 2d_{hip} \cos(\theta + \theta^v) r^v + (r^v)^2}} \end{bmatrix} \\
D_{q^v} \psi &= \begin{bmatrix} -((d_{hip} \text{Sin}[\theta + \theta^v])) \\ (d_{hip}^2 - 2d_{hip} \cos(\theta + \theta^v) r^v + (r^v)^2) \\ \frac{r^v (-d_{hip} \cos(\theta + \theta^v) + r^v)}{(d_{hip}^2 - 2d_{hip} \cos(\theta + \theta^v) r^v + (r^v)^2)} \\ \frac{r^v (-d_{hip} \cos(\theta + \theta^v) + r^v)}{(d_{hip}^2 - 2d_{hip} \cos(\theta + \theta^v) r^v + (r^v)^2)} \\ \frac{r^v (-d_{hip} \cos(\theta + \theta^v) + r^v)}{(d_{hip}^2 - 2d_{hip} \cos(\theta + \theta^v) r^v + (r^v)^2)} \end{bmatrix}.
\end{aligned} \tag{A.11}$$

Consequently, the virtual leg dynamics can be expressed by

$$\begin{aligned}
\ddot{q}^v &:= \begin{bmatrix} \ddot{r}^v \\ \ddot{\theta}^v \\ \ddot{\theta} \end{bmatrix} \\
&= \begin{bmatrix} -g \cos(\theta^v) + r^v \dot{\theta}^v{}^2 + \frac{k(r^v)^2 + k(d_{hip}^2 - r_0 \sqrt{d_{hip}^2 - 2d_{hip} \cos(\theta + \theta^v) r^v + (r^v)^2}) - dd_{hip} \cos(\theta + \theta^v) r^v + r^v (-2d_{hip} k \cos(\theta + \theta^v) + d^v + dd_{hip} \text{Sin}[\theta + \theta^v] (\dot{\theta} + \dot{\theta}^v))}{m(d_{hip} \cos(\theta + \theta^v) - r^v)} \\ \frac{g \sin(\theta^v) - 2r^v \dot{\theta}^v}{r^v} \\ 0 \end{bmatrix}
\end{aligned} \tag{A.12}$$

Appendix B

DERIVATIONS FOR SYSTEM PARAMETERS AND TRANSFORMATION MAPS BETWEEN CARTESIAN AND VIRTUAL LEG COORDINATES

B.1 Derivation of Leg Angle

The relation between the hip states and the COM states is given in Equation A.6. Further declaration on hip position and horizontal leg angle in terms of the virtual leg states is

$$\begin{aligned}x_{hip} &= -r^v \sin(\theta^v) - d_{hip} \sin(\theta) \\z_{hip} &= r^v \cos(\theta^v) - d_{hip} \cos(\theta) \\ \alpha &= \text{atan2}(z_{hip}, x_{hip})\end{aligned}$$

Hence, the leg angle with respect to the ground, α becomes

$$\alpha = \text{atan2}(r^v \cos(\theta^v) - d_{hip} \cos(\theta), -r^v \sin(\theta^v) - d_{hip} \sin(\theta)) \quad (\text{B.1})$$

B.2 Derivation of Hip Torque

Recalling 3.1, the relation between the hip torque and the leg force is determined in terms of the tangent of γ . We can identify γ by using trigonometric laws, namely *Sine Rule* and *Cosine Rule*, applied on the triangle formed by hip point, COM and toe position as illustrated in Figure B.1.

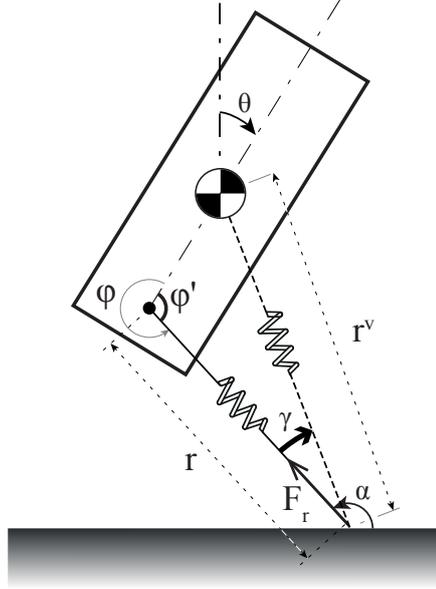


Figure B.1: Geometric properties of Trunk SLIP model with inserted virtual leg and CPP concept parameters

Sine and Cosine rules give

$$\frac{r^v}{\sin(\psi')} = \frac{d_{hip}}{\sin(\gamma)} \quad (\text{B.2})$$

$$d_{hip}^2 = (r^v)^2 + r^2 - 2rr^v \cos(\gamma) \quad (\text{B.3})$$

where r^v and ψ' is defined as

$$r^v = \sqrt{r^2 + d_{hip}^2 - 2rd_{hip}\cos(\psi')} \quad (\text{B.4})$$

$$\psi' = 2\pi - \psi \quad (\text{B.5})$$

Substituting r^v and ψ' into B.2 and simplifying the result leads

$$\gamma = \arctan\left(\frac{d_{hip} \sin(\psi)}{r - d_{hip} \cos(\psi)}\right) \quad (\text{B.6})$$

Consequently, the hip torque becomes

$$\tau_m = F_r r \frac{d_{hip} \sin(\psi)}{r - d_{hip} \cos(\psi)}. \quad (\text{B.7})$$

B.3 Derivations of Virtual Leg Touchdown Angle

It is possible to derive θ^v at touchdown as a function of the touchdown leg angle α , body pitch angle θ , and other morphological system parameters.

Using the model's geometry, θ^v can be expressed as

$$\theta^v(\alpha, \theta) = \alpha - \gamma(\psi(\alpha, \theta)) - \frac{\pi}{2} \quad (\text{B.8})$$

where γ is a function of ψ (Equation A.5) and it can be expressed as in Equation B.6. Therefore we can implicitly define γ by α and θ .

B.4 Derivations of Transformation Maps

In these derivations, we assume a flat surface with zero height and toe is located at the origin of the global cartesian coordinate frame. Recalling geometry of the virtual leg, the relation between the cartesian body states and the virtual leg length and angle can easily be obtained as follows

$$\begin{aligned} r^v &= \sqrt{x^2 + z^2} \\ \theta^v &= \text{atan2}(-x, z) \end{aligned} \quad (\text{B.9})$$

Taking time derivatives of the parameters in B.9 gives

$$\begin{aligned} \dot{r}^v &= \frac{x\dot{x} + z\dot{z}}{\sqrt{x^2 + z^2}} \\ \dot{\theta}^v &= \frac{x\dot{z} - z\dot{x}}{x^2 + z^2} \end{aligned} \quad (\text{B.10})$$

Note θ is a common state for both the virtual leg and the cartesian states. Thus, by arranging B.9 and B.10, we can write down the transition map, $T_{c \rightarrow v}(X^c)$

$$X^v = T_{c \rightarrow v}(X^c) = \begin{bmatrix} \sqrt{x^2 + z^2} \\ \frac{x\dot{x} + z\dot{z}}{\sqrt{x^2 + z^2}} \\ \text{atan2}(-x, z) \\ \frac{x\dot{z} - z\dot{x}}{x^2 + z^2} \\ \theta \\ \dot{\theta} \end{bmatrix} \quad (\text{B.11})$$

On the other hand, if we write down COM location in polar coordinates, we have $T_{v \rightarrow c}(X^v)$.

$$X^c = T_{v \rightarrow c}(X^v) = \begin{bmatrix} -r^v \sin(\theta^v) \\ -\dot{r}^v \sin(\theta^v) - r^v \cos(\theta^v) \\ r^v \cos(\theta^v) \\ \dot{r}^v \cos(\theta^v) - r^v \sin(\theta^v) \\ \theta \\ \dot{\theta} \end{bmatrix} \quad (\text{B.12})$$

Appendix C

DERIVATIONS OF ANALYTIC APPROXIMATION FOR THE VIRTUAL STANCE STATES

C.1 Derivation of the Approximate Virtual Leg Trajectory

In section 5.1, the transient solution for the undamped system leg length dynamics are given in Equation 5.1 in terms of the coefficients, A , B and C . Coefficient A can be found by using the derivations of \ddot{r}^v and r^v as

$$A = \frac{F^v}{(\omega_0^v)^2}$$

B and C are determined by the virtual touchdown states r_{td}^v and \dot{r}_{td}^v . Also we take the touchdown time, $t_{td} = 0$ for the stance phase derivations. In addition, we assume a boundary condition at the touchdown moment, $r_{td}^v = r_0^v$. That results

$$\begin{aligned} B &:= r_0^v - \frac{F^v}{(\omega_0^v)^2} \\ C &:= \frac{\dot{r}_{td}^v}{\omega_0^v} \end{aligned} \tag{C.1}$$

Note that $\frac{F^v}{(\omega_0^v)^2} = r_0^v$ due to the definition of F^v . Thus the coefficients become $A = r_0^v$ and $B = 0$ yielding virtual leg length trajectories of TSLIP during the stance phase as in the expression

$$\begin{aligned} r^v(t) &= r_0^v + \frac{\dot{r}_{td}^v}{\omega_0^v} \sin(\omega_0^v t) \\ \dot{r}^v(t) &= \dot{r}_{td}^v \cos(\omega_0^v t) \end{aligned}$$

r_{td}^v and \dot{r}_{td}^v can easily be determined by using the initial conditions of the system and the apex return map.

C.2 Derivation of the Approximate Angular Trajectory

Recall from Section 5.1, that the rate of change of the virtual leg angle is expressed in terms of the constant angular momentum and the virtual leg length as

$$\dot{\theta}^v = \frac{p_{\theta^v}}{m(r^v)^2}. \quad (\text{C.2})$$

Applying the Taylor Series Expansion for the term $\frac{1}{(r^v)^2}$ around r_0^v and substituting the result in Equation C.2 gives

$$\frac{1}{(r^v)^2} \Big|_{r^v=r_0^v} = \frac{1}{(r_0^v)^2} - \frac{2}{(r_0^v)^3}(r^v - r_0^v) + O(r^v, r_0^v) \quad (\text{C.3})$$

$$\begin{aligned} \dot{\theta}^v &= \frac{p_{\theta^v}}{m(r_0^v)^2} \left(3 - 2\frac{r^v}{r_0^v} \right) \\ &= \omega \left(3 - 2\frac{r^v}{r_0^v} \right) \end{aligned} \quad (\text{C.4})$$

Then, recalling the solution of the virtual leg length $r^v(t)$ in 5.3, $\dot{\theta}^v$ is computed as

$$\dot{\theta}^v = \omega \left(1 - \frac{2}{\omega_0^v r_0^v} \dot{r}_{td}^v \sin(\omega_0^v t) \right) \quad (\text{C.5})$$

Integrating $\dot{\theta}^v$ when the touchdown time is taken as zero, $t_{td} = 0$

$$\int_{\theta_{td}^v}^{\theta^v} d\theta^v = \int_0^t \omega \left(1 - \frac{2}{\omega_0^v r_0^v} \dot{r}_{td}^v \sin(\omega_0^v t) \right) dt \quad (\text{C.6})$$

Using the Integration by Parts rule, we have an approximate analytic model for the virtual leg angle as

$$\theta^v = \theta_{td}^v + \omega t + \frac{2\omega}{(\omega_0^v)^2 r_0^v} \dot{r}_{td}^v (\cos(\omega_0^v t) - 1) \quad (\text{C.7})$$