OPTIMAL CONTROL OF A HALF CIRCULAR COMPLIANT LEGGED MONOPOD

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

BY

YASEMİN ÖZKAN AYDIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2013

Approval of the thesis:

OPTIMAL CONTROL OF A HALF CIRCULAR COMPLIANT LEGGED MONOPOD

submitted by **YASEMİN ÖZKAN AYDIN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Gönül TURHAN SAYAN
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Kemal LEBLEBİCİOĞLU
Supervisor, **Electrical and Electronics Engineering Department, METU**

Assoc. Prof. Dr. Afşar SARANLI
Co-supervisor, **Electrical and Electronics Engineering Department, METU**

Examining Committee Members:

Prof. Dr. Ömer MORGÜL
Electrical and Electronics Engineering Dept., BİLKENT

Prof. Dr. Kemal LEBLEBİCİOĞLU
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Aydan ERKMEN
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Uluç SARANLI
Computer Engineering Department, METU

Assist. Prof. Dr. Yiğit YAZICIOĞLU
Mechanical Engineering Dept., METU

Assoc. Prof. Dr. Emre TUNA
Electrical and Electronics Engineering Dept., METU

Date:

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, Last Name: YASEMİN ÖZKAN AYDIN

Signature :

ABSTRACT

OPTIMAL CONTROL OF A HALF CIRCULAR COMPLIANT LEGGED MONOPOD

, YASEMİN ÖZKAN AYDIN
Ph.D., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Kemal LEBLEBİCİOĞLU
Co-Supervisor : Assoc. Prof. Dr. Afşar SARANLI

December 2013, 119 pages

Legged robots have complex architecture because of their nonlinear dynamics and unpredictable ground contact characteristics. They can be also dynamically stable and exhibit dynamically dexterous behaviors like running, jumping, flipping which require complex plant models that may sometimes be difficult to build. In this thesis, we focused on half circular compliant legged monopod that can be considered as a reduced-order dynamical model for the hexapod robot, called RHex.

The main objective of this thesis is the development of an algorithm based on optimal control theory that allows a half circular compliant legged monopod to run with a desired forward velocity or height starting from rest. The algorithm optimizes the foot attack angle and parameters of the PD controller while minimizing the error between desired and actual velocity. The effectiveness of the developed control strategy is verified by a variety of computer simulations. After obtaining a wide range of constant velocity running, we designed another energy-efficient controller which can effectively perform one of the dynamic behaviors, jumping motion. The jumping motion increases the maneuverability of the robot on the variety of rough terrain in a qualitatively satisfying manner. We analyze the performance of the our jumping control with a different jumping task, called triple jumping motivated by the exceptional performance of triple jumper athletes.

Keywords: Optimal Control, Legged Robots, Compliant Leg, Hexapod Running, RHex

ÖZ

YARIM DAİRE ESNEK TEK BACAKLI ROBOTUN EN İYİ DENETİMİ

, YASEMİN ÖZKAN AYDIN
Doktora, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Kemal LEBLEBİCİOĞLU
Ortak Tez Yöneticisi : Doç. Dr. Afşar SARANLI

Aralık 2013 , 119 sayfa

Bacaklı robotlar lineer olmayan dinamikleri ve tahmin edilemeyen yer temas karakteristikleri nedeniyle karmaşık bir yapıya sahiptirler. Bununla beraber dinamik olarak kararlı olup, koşma, sıçrama, takla atma gibi kapsamlı modelleme çalışması gerektiren davranışları gerçek-leştirebilirler. Bu tezde, altı bacaklı RHex robotunun basitleştirilmiş bir modeli olarak düşüne-bileceğimiz tek esnek yarım daire bacaklı robot üzerinde yoğunlaşacağız.

Bu tezin ana odak noktası optimal kontrol teorisini temel alan ve robotun istenilen yatay hızda koşmasını sağlayacak bir kontrol algoritmasının geliştirilmesidir. Bu algoritma robotun yere değmesi esnasındaki bacak açısını ve PD kontrolcünün parametrelerini en iyileştirmektedir. Geniş bir hız yelpazesinde gerekli olan kontrolcü tasarı-mından sonra atlama hareketini yapacak kontrolcü tasarımı yapılmıştır. Atlama harake-ti robotun engebeli arazilerde hareketini kolaytıran bir hareket çeşididir. Atlama denetimcisinin performansı üç adım atlama hareketini kullanarak analiz edilmiştir.

Anahtar Kelimeler: Optimal Kontrol, Bacaklı Robotlar, Esnek Bacak, RHex

To my family,

ACKNOWLEDGMENTS

I would like to express my very great appreciation to my PhD advisors, Prof. Dr. Kemal LEBLEBİCİOĞLU and Assoc. Prof. Dr. Afşar SARANLI, for supporting my research over the years. Their encouragements and helps made me feel confident to fulfill my desire and to overcome every difficulty. Besides my advisors, I would like to thank the rest of my thesis committee: Assoc. Prof. Dr. Uluç SARANLI, Assist. Prof. Dr. Yiğit YAZICIOĞLU, Prof. Dr. Aydan Erkmen and Prof. Dr. Ömer MORGÜL and Assist. Prof. Dr. Emre TUNA for their encouragement, insightful guidance and support on my research interests.

I would like to give special thanks to Prof. Daniel KODITSCHEK, for inviting me his group as a visiting scholar and supporting my research interest. I am particularly grateful for the assistance given by Dr. Haldun KOMŞUOĞLU while he was a post-doc in the KodLab group, and Dr. Kevin GALLOWAY, B. Deniz ILHAN, Anirudha MAJUMDAR and other KodLab members for their kind helps and supports.

I would like to thank TÜBİTAK (Scientific and Technological Research Council of Turkey) 2211 and 2214 Scholarship Programme for support and for funding my PhD. studies.

I wish to thank my best altruistic friend Reyhan ZENGİN, and the colleagues from my laboratories (Computer Vision, Rolab and ATLAS), for helping me get through the difficult times, and for all the emotional support, entertainment, and caring they provided.

I would like to express the profound gratitude from my deep heart to my beloved parents and my sister Kübra ÖZKAN, for their love and continuous support. I wish to thank to my precious husband Enes AYDIN, who has always supported me in any respect during the completion of the thesis. Lastly, and most importantly, as a new part of my life, I thank our son Ufuk AYDIN for being such a quite and good baby, making it possible to write and finish my thesis during his life in the mother's womb. This thesis is dedicated to my husband and son.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vi

ACKNOWLEDGMENTS viii

TABLE OF CONTENTS ix

LIST OF TABLES xiii

LIST OF FIGURES xv

CHAPTERS

- 1 INTRODUCTION 1
  - 1.1 Motivation and Contributions 5
  - 1.2 Organization of the Thesis 6
- 2 ONE LEGGED ROBOT MODEL 9
  - 2.1 Introduction 9
  - 2.2 Half Circular Compliant Leg Model 10
    - 2.2.1 Experimental Validation of the Leg Model 18
  - 2.3 Dynamic Equations of One Legged Robot 22
    - 2.3.1 Free Fall Simulation 26

2.4 Conclusion 28

3 OPTIMAL CONTROL OF LOCOMOTION 29

3.1 Introduction 29

3.2 Horizontal Velocity Control 29

3.3 Height Control 30

3.4 Results 32

3.4.1 Horizontal Velocity Control 32

3.4.2 Height Control 41

3.5 Stability Analysis 44

3.5.1 Apex Return Map 45

3.6 Conclusion 51

4 JUMPING CONTROL 53

4.1 Introduction 53

4.2 Jumping Control Problem 53

4.3 Results 55

4.4 Conclusion 58

5 OBSTACLE AVOIDANCE 59

5.1 Introduction 59

5.2 Problem Definition 60

5.2.1 Case I: Free Initial State 62

5.2.2 Case II: Fixed Initial State 64

5.3 Results 65

5.3.1 Results of Case I: Free Initial State 65

5.3.2 Results of Case II: Fixed Initial State 67

5.4 Sensitivity Analysis 71

5.4.1 Sensitivity to Initial Conditions 71

5.4.2 Sensitivity to Controller Parameters 77

5.5 Triple Jump Controller based on Sensitivity Analysis 78

5.6 Triple Jump Controller with a Safety Margin 80

5.7 Conclusion 87

6 HALF CIRCULAR LEG TEMPLATE FOR A HEXAPEDAL RUNNING 89

6.1 Introduction 89

6.2 Templates for Legged Locomotion 90

6.2.1 Spring Loaded Inverted Pendulum (SLIP) Template 90

6.2.2 Half Circular, Compliant Beam (CCB) Template . 91

6.3 Experimental Set-up and Data Collection 92

6.4 Fitting Models to Hexapod Running Data 97

6.4.1 Fitting Procedure 97

6.4.1.1 Stance Extraction 97

6.4.1.2 Data Processing 98

6.5 Fitting Prodecure 100

6.5.1 Fitting to Individual Stance Phase of One Gait . . . 100

6.5.2 Fitting to All Stance Phase of One Gait 101

6.6 Conclusion 104

7 CONCLUSIONS AND FUTURE WORK 105

REFERENCES 109

CURRICULUM VITAE 117

LIST OF TABLES

TABLES

Table 2.1 Table of variables used in (2.3) 11

Table 2.2 The leg, robot and environment parameters 17

Table 2.3 Event functions for stance and flight phase. 26

Table 3.1 Optimized K_p and K_d values for different ranges of $\dot{x}_{des}$ 30

Table 3.2 Parameter optimization results of height control 32

Table 3.3 Parameter optimization results of height control using the cost function (3.19) 49

Table 5.1 Sensitivity of triple jump trajectory to a perturbation in the initial states of the robot. The first column gives the percentage disturbance in the corresponding states and other three columns present the percentage of deviation from the desired final apex states. This results are obtained by jumping over obstacle with a h_{obs} = 0.5 (m) and w_{obs} = 0.1 (m). 73

Table 5.2 The effects of controller parameter variations in the final desired apex states of the triple jump trajectory. 77

Table 6.1 Edubot parameters 93

Table 6.2 Parameters of each gait 99

Table 6.3 The results of the first step of the system identification. 100

Table 6.4 The results of the first step of optimization. 102

Table 6.5 The mean, maximum and minimum leg radius of CCB model and leg length of SLIP model that are calculated by using optimum θ_{td} and fx_0 given in Table 6.4 103

Table 6.6 The optimization results of the last part. We fixed the leg parameters ($[EI, k]$ are given in Table 6.4 and $[l_{0_{mean}}, r_{mean}]$ are given in Table 6.5) and we re-optimized the control parameters of both models. 104

LIST OF FIGURES

FIGURES

Figure 1.1 The SensoRHex robot platform 4

Figure 2.1 The forces and torques at the cross-section of curved beam 11

Figure 2.2 Angles and distances used to calculate strain energy of curved beam. The ingtegration starts from loading point ($\gamma = 0$) and ends at fixed point ($\gamma = (\pi - \theta)$). Δx and Δy are the distances from loading point to crosssection. 12

Figure 2.3 The images of a half circular leg in the experimental setup that is used to confirm our circular leg assumption for different compression levels. The leg properties are the same as that used in this study. We fit a circle to the deflected leg only considering the leg part between the hip and ground contact point. 13

Figure 2.4 The radius of fitted circle vs. root mean square of error that is defined as sum of distances from the selected points on the leg to the fitted circle. 14

Figure 2.5 The confirmation of the circular leg assumption during dynamic running with a same leg used in this study. 15

Figure 2.6 Horizontal and vertical components of external force, $[F^s_x, F^s_y]$ and corresponding cross-sectional moments $[M^s_x, M^s_y]$ 16

Figure 2.7 The undeflected(dark) and deflected(light) leg with a leg angle θ. The leg is attached to the body from H. A and A' are the points with a same arc length on the undeflected and deflected leg, respectively. The deflection amount in the direction of external forces are, $\delta_x = x_2 - x_1$ and $\delta_y = y_1 - y_2$. 18

Figure 2.8 Top view of experimental setup (Presented with permission from [1]). Three markers are inserted to hip, loading point and at the end of leg. The cable is connected between force plate and loading point on the leg. (Left) Linear stage platform is in the initial position and no force is applied to leg. (Right) Linear stage platform has been moved 20 mm and leg is deflected. 19

Figure 2.9 Schematic diagram of the experimental setup (Presented with permission from [2]). The leg is rigidly attached to a linear stage platform from the hip. The leg is deflected by moving the stage on the direction indicated by the red arrow. 20

Figure 2.10 (a)-(b) Comparision of the radial force-deflection relation of Castigliano Model ($*$) with real data ($\square$) at the stiffness extremes, LSS0 and LSS4. The axis are defined in the reference frame attached to the point C (see Fig. 2.9) (c)-(d) Comparision of the deflection path of Castigliano Model ($*$) with real data ($\square$) at the stiffness extremes, LSS0 and LSS4. . . . 21

Figure 2.11 Percentage error (2.15) vs. radial deflection of the Castigliano Model at the stiffness extremes (a) LLS0 (b) LSS4. 22

Figure 2.12 External forces and torque acted on the robot and the leg 23

Figure 2.13 The free fall simulation of the C-leg robot. The robot is let free to fall under gravity with an initial condition of $[x = 0, \dot{x} = 0, y = 0.5\ m, \dot{y} = 0, \theta = 0, \dot{\theta} = 0]$ for ten successive jumping. The time step of ode45 is taken 0.001 sec. The energy plots are: (a) The kinetic and gravitational potential energy vs. time (b) The strain energy vs. time (c) The total mechanical energy vs. time plots of the robot 27

Figure 3.1 Stance, flight phase and step length of the robot 31

Figure 3.2 Optimization results of horizontal velocity control for different choices of desired velocity. 33

Figure 3.3 The first column presents kinetic and gravitational potential energy, second column is the strain energy and the last column is the total mechanical energy of the robot. 34

Figure 3.4 25 step running with a horizontal velocity 0.5 m/s 35

Figure 3.5 25 step running with a horizontal velocity 1.5 m/s 36

Figure 3.6 30 step running with a horizontal velocity of the first 15 steps 1.0 m/s, last 15 steps 1.5 m/s 37

Figure 3.7 30 step running with a horizontal velocity of the first 15 steps 1.0 m/s, last 15 steps 0.5 m/s 38

Figure 3.8 (a) The normalized leg radius vs. percentage of the total spring force to the its max. value and (b) The torque applied during the stance phase for the desired forward velocity from 0.2 – 2.0 m/s. 39

Figure 3.9 (a) The steady-state horizontal stiffness of the desired forward velocity from 0.2 – 2.0 m/s during the stance phase (b) The steady-state vertical stiffness of the desired forward velocity from 0.2 – 2.0 m/s during the stance phase (c) The total stiffness of the desired forward velocity from 0.2 – 2.0 m/s during the stance phase (d) The horizontal vs. vertical spring force of the desired forward velocity from 0.2 – 2.0 m/s during the stance phase 40

Figure 3.10 Accuracy of steady-state velocity control for different initial condition and desired velocity pairs. The color code corresponds to the values of the N-RMSE calculated by (3.8). 41

Figure 3.11 Accuracy of steady-state height control for different initial condition and desired height pairs. The color code corresponds to the values of the N-RMSE calculated by (3.9). 41

Figure 3.12 30 step running with a desired height (a) 0.15 m (b) 0.25 m (c) 0.35 m (d) 0.45 m and initial height 0.3 m 42

Figure 3.13 50 step running with a desired height (a) first 20 steps 0.5 m, then 15 steps 0.25 m, last 15 steps 0.4 m (b) first 20 steps 0.35 m, then 15 steps 0.5 m, last 15 steps 0.25 m (c) first 20 steps 0.5 m, then 15 steps 0.25 m, last 15 steps 0.4 m (d) first 20 steps 0.35 m, then 15 steps 0.5 m, last 15 steps 0.25 m 43

Figure 3.14 Poincare map 46

Figure 3.15 The position of the robot at stance and flight phase 46

Figure 3.16 (a) The fixed points of desired velocity 0.1 to 2 m/s with the predefined controller parameters. (b) The eigenvalues of Jacobian of desired velocity range 0.1-2 m/s for different percentage perturbation amount of initial states. 48

Figure 3.17 (a) The fixed points of desired height 0.15 to 0.5 m with the controller parameters given in Table 3.3 (b) The second eigenvalue of Jacobian of desired height range 0.15-0.5 m for different percentage perturbation amount of initial states. The first eigenvalue is unity for all desired height. 50

Figure 4.1 The apex height for the COM during flight found by using J_2 of (4.1) for different values of the optimization weight λ. Cell shades represent the optimal apex height (h^*_{apex}) at the last step. 55

Figure 4.2 The step length for the COM during flight found by using J_2 of (4.1) for different values of the optimization weight λ. Cell shades represent the optimal step length (L^*_{step}) at the last step. 55

Figure 4.3 Jumping height h^*_{apex} and length L^*_{step} for different values of the optimization weight λ for the one-shot jumping problem. Shades for each point represents (a) the take-off angle of the body at the last step, (b) the take-off velocity of the body at the last step. 56

Figure 5.1 Jumping over obstacles whose height is 0.3 m and width is 0.3 m. The red line represents the clearance of the robot. 61

Figure 5.2 The optimum take-off angle (a) and velocity (b) surfaces of the COM for a different obstacle size. 61

Figure 5.3 The phases of the free initial state problem 63

Figure 5.4 The preparation and triple jump phase of the second case. The obstacle is placed at $x = 5\ m$. The shaded area represents the initial state space. The robot is left from any point in the shaded area with a zero velocity. 64

Figure 5.5 (a) The optimum leg attack angle vs. initial position of the COM. (b) θ_{td} vs. K_d 66

Figure 5.6 (a) The optimum leg attack angle vs. initial position of the COM. (b) θ_{td} vs. K_d 66

Figure 5.7 The desired final state (a)$z_d = [3.7\,m, 1.5\,m/s, 0.2\,m, 0\,m/s]$ (b)$z_d = [4\,m, 1\,m/s, 0.3\,m, 0\,m/s]$. The blue rectangular part is the initial state space. 67

Figure 5.8 Optimum trajectory for the obstacle with a (a) height = 0.1 m, width = 0.1 (b) height = 0.1 m, width = 0.3 m 68

Figure 5.9 Optimum trajectory for the obstacle with a (a) height = 0.2 m, width = 0.3 (b) height = 0.3 m, width = 0.3 m 69

Figure 5.10 Optimum trajectory for the obstacle with a (a) height = 0.4 m, width = 0.1 (b) height = 0.4 m, width = 0.4 m 70

Figure 5.11 The undisturbed (solid line) and disturbed (dashed line) COM trajectory of the robot that jumps over obastacle with $h_{obs} = 0.5$(m) and $w_{obs} = 0.1$(m) using the triple jump controller. The disturbance amount equals to (a) 0.1% (b) 10% of the initial x position. 74

Figure 5.12 The undisturbed (solid line) and disturbed (dashed line) COM trajectory of the robot that jumps over obastacle with $h_{obs} = 0.5$(m) and $w_{obs} = 0.1$(m) using the triple jump controller. The disturbance amount equals to (a) 0.1% (b) 10% of the initial $\dot{x}$ position. 75

Figure 5.13 The undisturbed (solid line) and disturbed (dashed line) COM trajectory of the robot that jumps over obastacle with $h_{obs} = 0.5$(m) and $w_{obs} = 0.1$(m) using the triple jump controller. The disturbance amount equals to (a) 0.1% (b) 10% of the initial y position. 76

Figure 5.14 The optimum scale of the controller parameters that eliminate normalized 10% error in the initial x position of the robot for different obstacle size h_0 and w_0 scaled between $0.1 - 0.5$. The color scale represents the scale of (a) θ^*_{des1} (b) θ^*_{des3}. 79

Figure 5.15 The optimum scale of the controller parameters that eliminate 3% error in the initial forward velocity of the robot for different obstacle size h_0 and w_0 scaled between $0.1 - 0.5$. The color scale represents the scale of (a) θ^*_{des1} (b) θ^*_{des3}. 80

Figure 5.16 The optimum scale of the controller parameters that eliminate 10% error in the initial height of the robot for different obstacle size h_0 and w_0 scaled between $0.1 - 0.5$. The color scale represents the scale of (a) θ^*_{des1} (b) θ^*_{des3}. 80

Figure 5.17 The new representation of obstacle with a 10% safety margin. . . . 81

Figure 5.18 The COM trajectory of the robot that jumps over obstacle ($h_{obs} = 0.1$ (m) and $w_{obs} = 0.1$) (top) with a 10% safety margin (bottom) without the safety margin. 82

Figure 5.19 The COM trajectory of the robot that jumps over obstacle ($h_{obs} = 0.2$ (m) and $w_{obs} = 0.3$) (top) with a 10% safety margin (bottom) without the safety margin. 83

Figure 5.20 The COM trajectory of the robot that jumps over obstacle ($h_{obs} = 0.5$ (m) and $w_{obs} = 0.5$) (top) with a 10% safety margin (bottom) without the safety margin. 83

Figure 5.21 The optimum reference angle signals of triple jump controller without safety margin added to obstacle size. The height and width of the obstacle are scaled between 0.1-0.5 (m). The color scale represents the optimum reference angle signal of controller at the (left) first (middle) second (right) third step of triple jump. 84

Figure 5.22 The optimum reference angle signals of triple jump controller with a 10% safety margin added to obstacle size. The height and width of the obstacle are scaled between 0.1-0.5 (m). The color scale represents the optimum reference angle signal of controller at the (left) first (middle) second (right) third step of triple jump. 84

Figure 5.23 The optimum touchdown angle of triple jump controller without safety margin added to obstacle size. The height and width of the obstacle are scaled between 0.1-0.5 (m). The color scale represents the optimum touchdown angle of controller at the (left) first (middle) second (right) third step of triple jump. 85

Figure 5.24 The optimum touchdown angle of triple jump controller with a 10% safety margin added to obstacle size. The height and width of the obstacle are scaled between 0.1-0.5 (m). The color scale represents the optimum touchdown angle of controller at the (left) first (middle) second (right) third step of triple jump. 85

Figure 5.25 The optimum starting conditions of triple jump phase without safety margin added to obstacle size. The height and width of the obstacle are scaled between 0.1-0.5 (m). The color scale represents the optimum (left) initial forward velocity (middle) initial height (right) the initial starting x position of triple jump phase. 86

Figure 5.26 The optimum starting conditions of triple jump phase with a 10% safety margin added to obstacle size. The height and width of the obstacle are scaled between 0.1-0.5 (m). The color scale represents the optimum (left) initial forward velocity (middle) initial height (right) the initial starting x position of triple jump phase. 86

Figure 6.1 Edubot [3] 90

Figure 6.2 The SLIP Model 91

Figure 6.3 The CCB Model 91

Figure 6.4 (a)The Buehler two speed profile is defined by 4 parameters (b) Side view of the robot with a desired leg angle of the fligth and stance phase $[\theta_{td}, \theta_{lo}, t_s, t_p]$ [4]. 93

Figure 6.5 Sagittal plane movement of the robot in the world coordinate frame 97

Figure 6.6 The stance phases of one run (top), mean of the actual $\dot{\theta}_{act}$, target angular velocity $\dot{\theta}_t$ and shifted target angular velocity $\dot{\theta}_{st}$ of the left (middle) and right tripod (bottom). 98

Figure 6.7 The controller parameters $[\phi_{lo}, \phi_{td}, t_s]$ of (a) SLIP model (b) CCB template obtained in the second step 101

Figure 6.8 Fitting errors of (a) SLIP template (b) CCB template 101

Figure 6.9 The geometrical stance extraction procedure of SLIP and CCB models. The dashed part of the COM trajectory is eliminated since the height of the COM in this part does not satisfy the geometrical constraint given in (6.14) 103

CHAPTER 1

INTRODUCTION

Mobile robots are widely used for military, industrial and medical applications, as well as space exploration, education and entertainment. In this thesis, we concentrated on land based mobile robots that can be categorized into three main classes: wheeled, tracked and legged robots.

Biologists explain the nonexistence of the wheel-like locomotion in nature with an unsuitability of the biological process to produce bio-wheels [5]. They also claim that the wheel-like locomotion requires more energy to activate muscles compared to other forms of locomotion [6]. Wheels are good at locomoting in a structured environment such as roads and railways, whose infrastructures and material properties are adjusted to obtain smooth, fast and energy efficient movement [7]. Despite the efficiency of wheeled locomotion in the structured environments, it is not the most suitable type of locomotion in the environment where the vertical and horizontal irregularities of the terrain block the movement of the wheels. [8] classified the limitations of the wheeled locomotion into three categories; surface compliance, roughness of terrain and size of the obstacle. On the uneven and soft ground or when the mass of the vehicle is increased, the rolling resistance increases [9]. The maximum height of the obstacle that a wheeled vehicle overcomes is depent on the radius of the wheels [10]. If the height of the obstacle is higher than half of the wheel radius, the vehicle does not propel its body over it.

In nature, animals use the rolling motion as a second form of a locomotion to escape from their enemies [11] or adapt to environemental changes [12]. There are several species such as caterpillars [13], wheel spiders, stomatopod [11, 12], salamanders

[14] etc. that exhibit self-propelled rolling motion by converting their bodies into a wheel [15, 16].

Monopod robots have been studied by many researchers since they are simpler but still exhibit relevant hopping gaits. Raibert and his colleagues designed a hopping robot that was a significant step for legged robotic systems [17]. Their 2D model consisted of a body and a compliant leg with two actuated joints. Raibert extended this mechanism to a 3D hopper as well [18]. Many other researchers developed different types of one legged robots, with one, two or three link legs actuated by hydraulic or pneumatic actuators after Raibert's work [19, 20, 21, 22]. Differently from the previous studies, Zeglin built a planar bow-legged hopper that used a curved leg profile to take advantage of passive energy storing property of elastic materials [23].

The first systematic study on the effect of semicircular feet on passive dynamic walking was performed by Mcgeer [24]. In this study, he showed that humans normally produce an effective foot curvature with a radius equal to 30% of leg length and that with a circular shaped foot, we can capture the motion of center of pressure effectively. This study also revealed that, as the arc radius of feet increased, the step-to-step transition [25, 26] cost decreased [24]. In addition, a study conducted by Adamczyk et al. [27] analyzed the effect of varying foot curvature on the mechanical and metabolic cost of biped walking and it was shown that rolling feet reduces step-to-step transition costs. Another study on the effects of circularly shaped feet on the locomotion of a 3D passive dynamic walker demonstrated that circular feet increases the walking speed and stability of gaits [28]. In summary, with a semi-circular feet, one can increase the efficiency, reliability and speed of dynamic bipedal locomotion [29].

The advantages of wheeled locomotion and rolling feet have resulted in the evaluation of the latest legs used on RHex robots [30, 31]. Various approaches have been proposed to analyse the dynamical motion of RHex with half circular legs [4, 32, 33]. Many of these have used the *Spring-Loaded Inverted Pendulum* (SLIP) model as a template to model the motion of the *center of mass* (COM) of the robot. However, limited research effort has been put to model the integration between the compliance of the leg with its half circular shape and the resulting effect on the locomotion of the hopper. The pseudo-rigid-body (PRB) model [34] has been used to characterize the deflection path of the loading point on the leg under static [1] and dynamic load [35]. In this model, the deflection behavior of the curved leg was represented with two rigid links connected by a torsional spring from the effective center of rotation (characteristic pivot) [36]. However, the selection of the position of characteristic pivot and torsional spring constant proved to be difficult since the position of the contact point changes during locomotion.

In [23], Sayginer considered the rolling kinematics of the semi-circular leg and have also used Castigliano's Theorem [37, 38, 39] to model the deflection behaviour of the leg under loading. As explained in the experimental study performed by Aydin et al. [2], Castigliano's Model captures the spatial compliance of the variable stiffness circular leg more accurately than a PRB model under static loads. Extending on the results of [2], we have implemented the leg model in a dynamic mode of operation. In the present study, we improve on the model in [40] by lifting the assumption on the force direction which was therein assumed to point in the direction of the hip (the connection of the leg with the rotary motor).

In robotics, optimal control techniques, using torque and energy related objective function, have been widely used to obtain stable, energy efficient and robust locomotion on a variety of environment. A considerable amount of literature has been published on trajectory optimization of legged robots in irregular terrain. Iida and Tedrake optimized the motor control parameters of the underactuated two segment leg monopod to locomote in uneven terrain [41]. Vermuelen et al. constructed polynomial functions for the flight and stance phases and optimized the locomotion parameters to perform hopping motion in complex environment [42]. Also, a stair hopping trajectory of the monopod with a flat foot was obtained by minimizing actuation energy in hopping cycle [43].

The application of optimization to biped walking have gathered attention of many researchers. Optimal control problems considering minimum energy consumption as an objective were defined to optimize the leg stiffness of the biped robots for a stable, cyclic gait [44, 45]. Chevallereau and Aoustin defined a polynomial function of joint variables and optimized the coefficients of this function while maximizing forward velocity and minimizing torque and consumed energy to obtain cyclic gaits for a biped without actuated ankles [46]. Mombaur et al. described two level optimization problem and attempted to minimize energy in the low level while optimizing the parameters of biped robot model in the upper level to realize stable periodic gaits [47]. A spline based parametric optimization technique was used to minimize the joint torque to optimize walking phases including the transition configurations of seven link planar biped [48]. From a different point of view, Tedrake accomplised a stable limit cycle gait for a 3D passive dynamic walker by minimizing the eigenvalues of its return map [49].

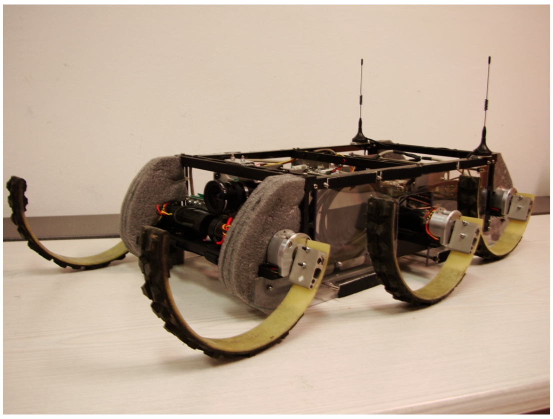


Figure 1.1: The SensoRHex robot platform

In this study, we optimized the controller parameters and touchdown angle of the half circular legged monopod whose leg morphology is similar to the RHex family of hexapods. Previous studies were performed on the optimization of different gait patterns of RHex. Weingarten et al. described the implementation of the gait tuning process to obtain an efficient, stable gait using two different specific resistance based cost functions [50]. Also, [51] investigated the effect of the leg stiffness on the dynamics of hexapod robot using reduces order dynamic model and they optimized the controller parameters and the leg stiffness to find most stable gaits. Different from the gait optimization approaches, Saranli et al. defined a constraint optimal control problem to maximize energy injection during the self-righting maneuvers of RHex [52].

1.1 Motivation and Contributions

The performance of the wheeled robots in the structured environment and the legged robots in the rough terrain results in a combination of these two types of locomotion. The hexapedal mobile robot RHex is one of the robots that exploits wheel-type leg morphology to locomote in a wide variety of environmental conditions. Obtaining the mathematical model of a hexapedal robot plays a critical role in understanding parts of dynamic locomotion. For this purpose, simplified model called *templates* are used to describe the dynamics of the hexapedal locomotive behavior [4, 32, 53, 54]. One of the most significant model that is used widely in modeling the center of mass (COM) motion of multilegged animals [55] is the passive spring-loaded inverted pendulum (SLIP) [56].

An overriding goal throughout this dissertation has been to present a thorough dynamical model of the half circular complaint legged monopod that considers rolling motion, elasticity and geometry of the leg and ground friction. This model is significantly different from the previous model [57, 58] that uses same methodology to obtain force-deflection relation of the half circular compliant leg but restricts the direction of the spring force. In this thesis, we improve the model in [57] by removing the assumption on the spring force direction which was therein assumed to point in the direction of the hip joint. An alternative approach to solve circular compliant beam force characteristics is the pseudo rigid body model. This model is sufficiently accurate especially in the presence of static loading conditions, but in the dynamic running, it is challenging to obtain principal model parameters since the contact point of the leg changes.

So far, there has been little discussion about the control of a monopod with a circular leg. This thesis aims to contribute to the study of dynamically running half circular compliant legged monopod, by proposing a control strategy based on numerical optimization techniques. We describe a simple solution to generate reference trajectories for different types of locomotion such as running with a desired forward velocity and jumping across or over a desired height or length.

As far as we know, the only model that is used to capture the dynamics of hexapedal locomotion is SLIP model. In this thesis, we also propose a new template model that can be used to characterize the dynamics of hexapedal running with a half circular compliant leg. We used experimental COM trajectory of the hexapedal robot to show that a simple, energetically conservative monopod with a half circular elastic leg can produce a similar trajectory provided by RHex type of robots.

1.2 Organization of the Thesis

This thesis is organized as follows: Chapter 2 outlines our methodology to obtain the dynamic equations of the half circular complaint legged monopod. First, we obtained force-deflection relationship of a circular compliant beam under the external forces using Castigliano's Theorem. We experimentally validated this relationship and adapted this solution approach to our system and obtained unknown external forces applied to C-leg as a function of state variables. The combined translational and rotational dynamics of the robot are obtained by Newton-Euler method. The motion of the body is analyzed under the constraint of rolling without slipping

Chapter 3 develops an algorithm based on optimal control theory that allows a half circular compliant (HCC) legged monopod to locomote with (1) a desired forward velocity and (2) a desired height starting from zero initial velocity. We investigated the stability of the gait when the reference forward velocity or height changed using Apex Return Map analysis. To apply the return map analysis to our system first we have obtained fixed points of the system and a transition function that calculates the dependency of two subsequent apex height.

A new jumping control method is subsequently described in Chapter 4. The controller parameters and touchdown angle of the leg are optimized using the cost function that is a convex function of a jumping height and distance. We show the relation between initial velocity, take-off angle and jumping performance.

In the Chapter 5, we have proposed control strategies to achieve rough terrain locomotion. Inspiring from the performance of long and high jump athletes, we focused on jumping over obstacle which has height and width. We have identified the principles that govern optimum speed and leg angle, for the take-off phase of high and long jumping with a compliant half circular legged monopod. Based on sensitivity analysis, we have investigated the effects of disturbances on the output of controlled system. We have determined the sensitivity of triple jump trajectory to variations in the initial conditions and controller parameters of the system. Using the results of sensitivity analysis, we have tried to find suitable sets of controller parameters that eliminate the effects of variations in initial conditions. We also modify the trajectory of the robot taking into account the positioning error of the obstacle by adding safety margin around to it.

Chapter 6 analyzes the dynamic motion of the hexapod robot, called Edubot, using two different modeling approach using the time trajectory of a full body configuration and leg states. We propose a low dimensional circular compliant beam (*CCB*) template as a candidate template model for the hexapedal running and we compare the performance of our candidate template with the performance of widely used Spring Loaded Inverted Pendulum (*SLIP*) template.

The last chapter summarizes and discusses the results of the research.

CHAPTER 2

ONE LEGGED ROBOT MODEL

2.1 Introduction

Obtaining dynamic modeling of robot platforms is an important topic for the design, simulation, and control of robots. Different techniques have been proposed and used by the robotics community. Lagrange and Newton-Euler are two classical dynamical modeling methods which are commonly used [59].

The Lagrange approach is based on the energy principles. It formulates the forces as a combination of kinetic and potential energies of the system to derive the closed-form equations and it eliminates the internal joint reaction forces and moments. This method can be useful for multiple degree of freedom systems if there is no constraints on motion. The Newton-Euler method takes into account the external and constraint forces and relates them with the motion of a system. It constructs free-body diagram considering unknown reaction forces and moments between interconnected elements. It writes individual equations of motion for each body and solves them in a numeric and recursive way. In this thesis, we have chosen Newton-Euler method to model dynamics of the robot since the energy relation of compliant leg is not a simple function of state variables and it is very challenging to get dynamic equations with Lagrange approach.

This section describes our effort for modeling the monopod dynamics while considering geometric and elastic properties of the leg. When the robot stands on the ground, the deflection of contact point on C-shaped leg due to the reaction force applied by the ground should be found in order to characterize the dynamical behavior of the robot. Until now, different modeling approaches have been used to describe the function of compliant property of the leg during the locomotion of the robot. The very common approach is to use *Spring Loaded Inverted Pendulum* (SLIP) model, a simple spring-mass template [4, 53, 56]. This model consists of a massless springy leg that is connected to a point mass. The exchange between kinetic and potential energy of SLIP model can predict the dynamic properties of a variety of living organisms that have different leg number and skeleton type. Although the SLIP model can capture the main characteristics of the dynamic locomotion of humans and animals [55], it can sometimes oversimplifies the nonlinear characteristics of the complex mechanisms.

In our previous study, we presented closed form equations for a variable stiffness leg using a well-known energy based deflection analysis method for elastic beams, Castigliano's Theorem [37, 38, 39]. We compared the results of Castigliano's method with a simple closed form approach, called the PRB model [34] and experimentally showed that Castigliano's method estimates the deflection of the curved beam for different stiffness settings more accurately than the PRB model [2].

In the rest of this thesis, we consider this curved leg model as a basis for our spring force calculation and employ the same methodology used in [2]. The spring forces obtained in the next section are implemented in the dynamic equations of the motion in stance phase that are obtained by the Newton-Euler method, which is based on the dynamic equations of the isolated bodies. The leg and robot parameters used in the Section 2.2 and 2.3 are given in Table 2.2.

2.2 Half Circular Compliant Leg Model

The force acting on a body causes some strain energy if the force cannot rotate or translate the body. Due to stiffness of the leg, this strain energy will results in some deflection on the leg. That is, a point on the leg will deflect by a distance in a direction not necessarily parallel to the force applied. According to Castigliano's Theorem [39], the amount of deflection is the derivative of the total strain energy stored in the elastic element, with respect to a force parallel to the direction in which the deflection is calculated.

Instead of dealing with forces and deflections in arbitrary directions, the problem

Table 2.1: Table of variables used in (2.3)

Variable	Definition	Unit
A	cross-section area	m^2
E	young modulus	GPa
G	shear modulus	GPa
e	eccentricity	m
r	radius of curvature	m
C	correction factor	unit-less

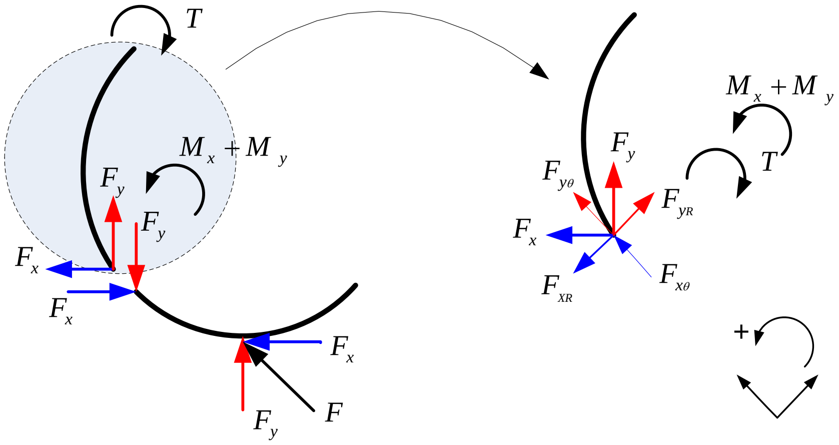


Figure 2.1: The forces and torques at the cross-section of curved beam

is divided into sub-problems in which the x- y components of the external forces (they can be considered reaction forces) are handled. First of all, the reaction force F applied by the ground is decomposed into its components F_x and F_y; then these forces are moved to cross-section. The shear force, normal force and bending moment occurring at a cross section due to the external force F can be expressed as

$$\begin{aligned} F_\theta &= F_{x\theta} + F_{y\theta}, \\ F_R &= F_{yR} - F_{xR}, \qquad (2.1) \\ M &= M_x + M_y, \end{aligned}$$

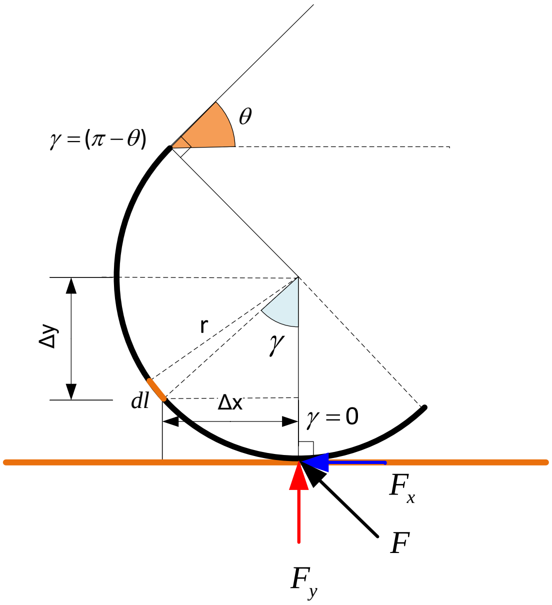


Figure 2.2: Angles and distances used to calculate strain energy of curved beam. The ingtegration starts from loading point ($\gamma = 0$) and ends at fixed point ($\gamma = (\pi - \theta)$). Δx and Δy are the distances from loading point to crosssection.

where

$$\begin{aligned}
F_{xR} &= F_x \sin(\gamma),\\
F_{yR} &= F_y \cos(\gamma),\\
F_{x\theta} &= F_x \cos(\gamma),\\
F_{y\theta} &= F_y \sin(\gamma),\\
M_x &= -F_x(r-\Delta y) = -F_x r(1-\cos(\gamma)),\\
M_y &= F_y \Delta x = F_y r \sin(\gamma).
\end{aligned} \tag{2.2}$$

The forces and deflection components at the cross section can be seen in Fig. 2.1. The integral of these shear force, normal force and bending moments at a cross section from loading point to hip point gives the three strain energy terms caused by these forces. The total strain energy of a circular curved beam then can be written as

$$U = r(\int \frac{M^2\,d\gamma}{2AEe} + \int \frac{F_\theta^2\,d\gamma}{2AE} + \int \frac{CF_R^2\,d\gamma}{2AG}) - \int \frac{MF_\theta\,d\gamma}{AE}, \tag{2.3}$$

where U is the total strain energy, and M, F_θ, and F_R are the total bending moment, tangential and radial forces, respectively, created by external forces at the cross section of the curved beam [39]. The remaining variables used in (2.3) are defined in Table 6.1.

As depicted in Fig. 2.2, there is a relation between the arc length of the curved beam and the variable of integration

$$dl = r\, d\gamma, \tag{2.4}$$

where l is the arc length of the curved beam. The integration interval starts from the loading point and ends at the fixed end of the curved beam. In general, if the radius of the curved beam is at least ten times larger than its thickness, the strain energy expression (2.3) can be approximated by

$$U \approx \int \frac{M^2 r\, d\gamma}{2EI_{cs}}, \tag{2.5}$$

where $I_{cs}(m^4)$ is the second moment of area of beam cross-section [39].

The partial derivative of the total strain energy, U of (2.5) with respect to any external force F_i gives the deflection, δ_i, of the loading point in the direction of the F_i to yield

$$\delta_i = \frac{\partial U}{\partial F_i}. \tag{2.6}$$

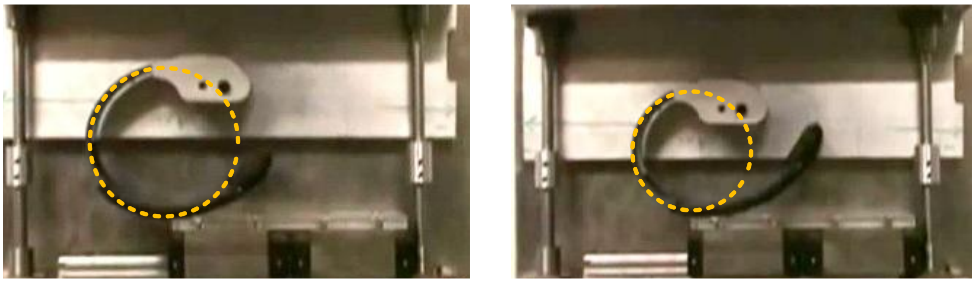


Figure 2.3: The images of a half circular leg in the experimental setup that is used to confirm our circular leg assumption for different compression levels. The leg properties are the same as that used in this study. We fit a circle to the deflected leg only considering the leg part between the hip and ground contact point.

We now apply this theorem to our system. The leg, robot and environment parameters used in this and next chapter are given in Table 2.2 Since there is no damping in the model, the ground reaction force exerted by the leg during stance phase was assumed to be equivalent to spring force F^s. In this study, we also use a circular leg assumption,

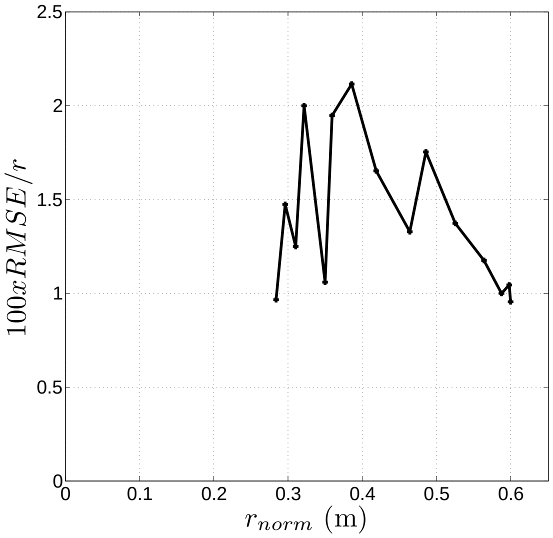


Figure 2.4: The radius of fitted circle vs. root mean square of error that is defined as sum of distances from the selected points on the leg to the fitted circle.

i.e., that the leg preserves its circular shape after deflection, and that the deflected leg radius, r is equal to the radius of a circular arc that passes from the ground contact point G and hip point H with an angle θ. We confirmed our circular leg assumption with an experimental setup shown in Fig. 2.3. In this experiment, the leg whose properties are the same as that used in this study was fixed the moving platform from hip and we applied force to obtain different amount of compression. We fitted a circle to the points selected on a compressed part of the leg and calculated the root mean square of distances from the points to the fitted circle. In Fig. 2.4, we give the percentage error vs. the radius of fitted circle. The errors are smaller than 2.2% for all deflection amount. Actually, the whole shape of the leg is closer to an elliptic arc, but the shape of the leg part where the force is applied (from hip to contact point) is closer to circular arc, which supports our circularity assumption. We also confirm this assumption while dynamic locomotion (see Fig. 2.5) Consequently, the strain energy of the compliant half circular leg of Fig. 2.6 can be written as

$$U \;=\; \frac{r}{2EI_{cs}} \int_0^{(\pi-\theta)} (M_x^s + M_y^s)^2 \, \mathrm{d}\gamma, \tag{2.7}$$

where r_u is the undeflected leg radius, and M_x^s and M_y^s are moments at the cross section caused by the horizontal and vertical components of the spring force, F_x^s and

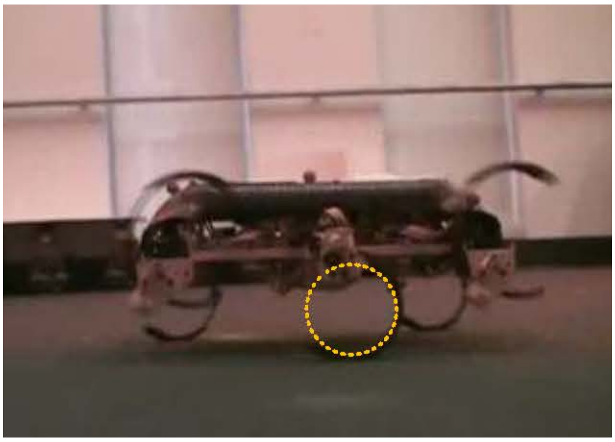


Figure 2.5: The confirmation of the circular leg assumption during dynamic running with a same leg used in this study.

F^s_y, respectively, and can be written as

$$\begin{aligned} M^s_x &= -F^s_x r(1-\cos\gamma) \\ M^s_y &= F^s_y r\sin\gamma. \end{aligned} \tag{2.8}$$

Lets define the upper integral limit of (2.7) as

$$\zeta \triangleq (\pi-\theta). \tag{2.9}$$

The total strain energy and deflection amount in the direction of external forces can then be written as

$$\begin{aligned} U &= \frac{r^3}{8EI}\Big(-6F^s_xF^s_y + 6F^{s2}_x\zeta + 2F^{s2}_y\zeta + 8F^s_xF^s_y\cos(\zeta) \\ &\quad - 2F^s_xF^s_y\cos(2\zeta) - 8F^{s2}_x\sin(\zeta) + (F^{s2}_x - F^{s2}_y)\sin(2\zeta)\Big), \\ \delta_x &= \frac{\partial U}{\partial F^s_x} = \frac{r^3}{4EI}(6\zeta - 8\sin(\zeta) + \sin(2\zeta))F^s_x \\ &\quad + \frac{r^3}{4EI}(-3 + 4\cos(\zeta) - \cos(2\zeta))F^s_y, \\ \delta_y &= \frac{\partial U}{\partial F^s_y} = +\frac{r^3}{4EI}(-3 + 4\cos(\zeta) - \cos(2\zeta))F^s_x \\ &\quad + \frac{r^3}{4EI}(2\zeta - \sin(2\zeta))F^s_y. \end{aligned} \tag{2.10}$$

If we write (2.10) in matrix form, we obtain the displacement of the loading point as

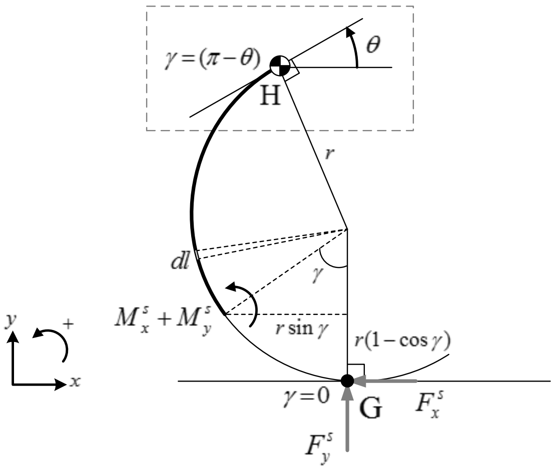


Figure 2.6: Horizontal and vertical components of external force, $[F_x^s, F_y^s]$ and corresponding cross-sectional moments $[M_x^s, M_y^s]$

a linear function of external forces as

$$\begin{pmatrix}\delta_x\\ \delta_y\end{pmatrix} = \begin{pmatrix}C\end{pmatrix}\begin{pmatrix}F_x^s\\ F_y^s\end{pmatrix}, \tag{2.11}$$

where C is the compliance matrix and its elements are

$$\begin{aligned} C_{11} &= \frac{r^3}{4EI}(6\zeta - 8\sin(\zeta) + \sin(2\zeta)),\\ C_{12} &= \frac{r^3}{4EI}(-3 + 4\cos(\zeta) - \cos(2\zeta)),\\ C_{21} &= \frac{r^3}{4EI}(-3 + 4\cos(\zeta) - \cos(2\zeta)),\\ C_{22} &= \frac{r^3}{4EI}(2\zeta - \sin(2\zeta)). \end{aligned} \tag{2.12}$$

In order to calculate forces acting on the monopod body through the compliant legs, we first need to obtain the deflection of the loading point. For this purpose, the undeflected ground contact point is found with the same leg angle θ, assuming that this undeflected leg touches the ground from the point on which external forces F_x^s and F_y^s are applied, i.e. $HA = HA'$ (see Fig. 2.7). The difference between the positions of the loading point on the deflected and undeflected legs gives the deflection amount in the

Table 2.2: The leg, robot and environment parameters

Leg Parameters	E	Young modulus	$9.5810^9\ Pa$
	m_{leg}	mass of the leg	$0.005\ kg$
	r_u	rest leg radius	$0.06\ m$
	b_{leg}	cross-section width	$0.025\ m$
	h_{leg}	cross-section height	$0.0035\ m$
	I_{cs}	second moment of area	$\frac{b_{leg}h_{leg}^3}{12}$
	I_{leg}	moment of inertia about hip	$2m_{leg}r_u^2$
Robot Parameters	m_b	mass of the body	$1.5\ kg$
	b_b	width of the body	$0.15\ m$
	h_b	height of the body	$0.06\ m$
	I_b	moment of inertia	$\frac{1}{12}m_b(b_b^2 + h_b^2)$
	g	gravity	$9.8\ m/s^2$

direction of external forces and can be written as

$$\begin{aligned} \delta_x &= r\sin(\theta) - r_u\sin(\theta) + r_u\sin(\zeta\frac{r}{r_u} + \theta) \\ \delta_y &= r_u\cos(\theta) - r_u\cos(\zeta\frac{r}{r_u} + \theta) - r\cos(\theta) - r. \end{aligned} \tag{2.13}$$

The forces can then be calculated from (2.11) as

$$\begin{pmatrix} F_x^s \\ F_y^s \end{pmatrix} = \begin{pmatrix} K \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix}, \tag{2.14}$$

where the stiffness matrix K, is the inverse of the compliance matrix, C of (2.11).

The equations obtained above are used to evaluate deflection of the half-circle legs of RHex type robot, and any other robots that have curved leg, in the direction of the external forces. According to material used to design the leg, the strain energy of the system has different eccentricity, young modulus and shear modulus values which are the coefficients in the derived equations.

In this section, the effect of forces and moments applied on the curved leg is analyzed in terms of stiffness behavior and change in leg geometry. Castigliano's Theorem for curved members is used for calculating the deflection of the leg under the effect of external forces. The calculated deflection values will be used to improve the studies made before, and also to do the dynamic analysis more accurately.

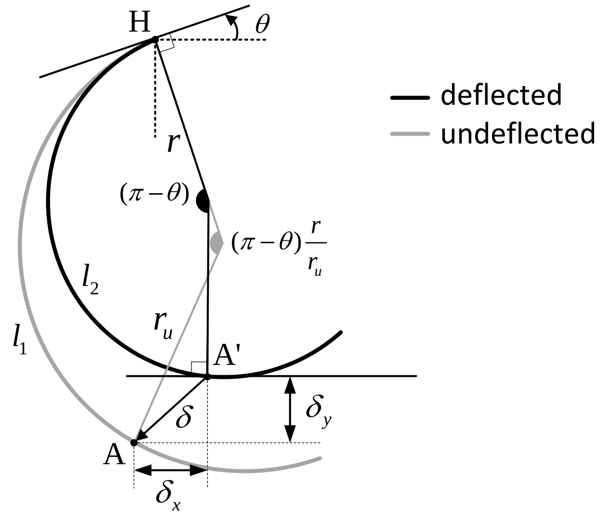


Figure 2.7: The undeflected(dark) and deflected(light) leg with a leg angle θ. The leg is attached to the body from H. A and A' are the points with a same arc length on the undeflected and deflected leg, respectively. The deflection amount in the direction of external forces are, $\delta_x = x_2 - x_1$ and $\delta_y = y_1 - y_2$.

2.2.1 Experimental Validation of the Leg Model

In the previous study [2], we experimentally investigated the accuracy of the our leg model that uses Castigliano's Theorem to characterize the deflection behavior of the compliant, half circular, tunable stiffness robot leg. The leg stiffness adjustment mechanism was designed to understand the role of leg compliance in a dynamic running [60, 61].

To observe the deflection behavior of C-leg under the static load and to validate the deflection results predicted by Castigliano's Theorem, we used the experimental setup shown in Fig. 2.8. The tunable stiffness half circular leg was anchored to the platform of a linear stage from its hip. The effective stiffness of the leg was adjusted by changing the position of a compliant slider. The AMTI HE 6x6 multi axis force plate and the MICOS linear stage were mounted on aluminum base plate, vertically and horizontally, respectively. The linear stage can move maximum 20 mm at 10 mm/s. An Optotrak 3020 Motion capture system recorded the deflection path of the leg at a sampling rate of 200 Hz by tracking the position of the three markers mounted rigidly to

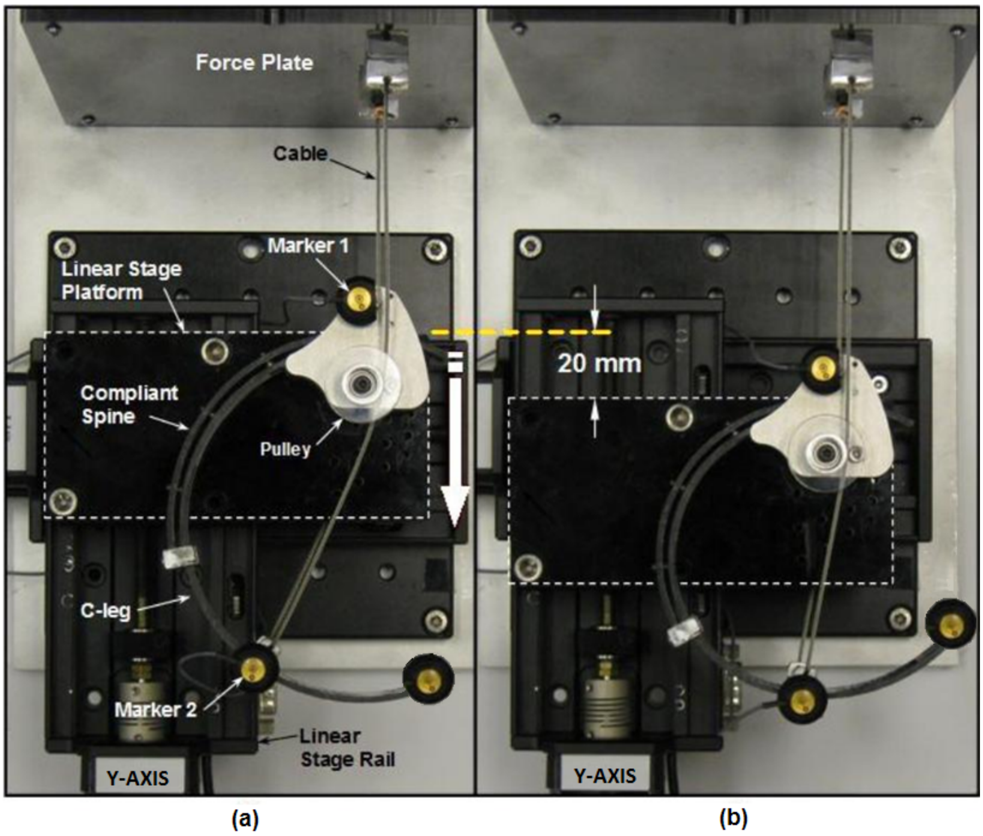


Figure 2.8: Top view of experimental setup (Presented with permission from [1]). Three markers are inserted to hip, loading point and at the end of leg. The cable is connected between force plate and loading point on the leg. (Left) Linear stage platform is in the initial position and no force is applied to leg. (Right) Linear stage platform has been moved 20 mm and leg is deflected.

the leg. A flexible steel cable was connected between force plate and loading point (Marker 2) passing from the pulley that was anchored to the hip. This pulley keeps the cable normal to force plate and decreases the dimension of force measurement data to one. Although the position of the loading point is constant, the pulley provides rolling contact motion when the linear stage moves along the y-direction. The force plate is capable of measuring maximum 10 pounds with a 12-bit resolution and 200 Hz. sampling rate. The reaction forces at the loading point and leg deflection path camera data were collected for the leg stiffness settings (LSSs) 0 to 4.

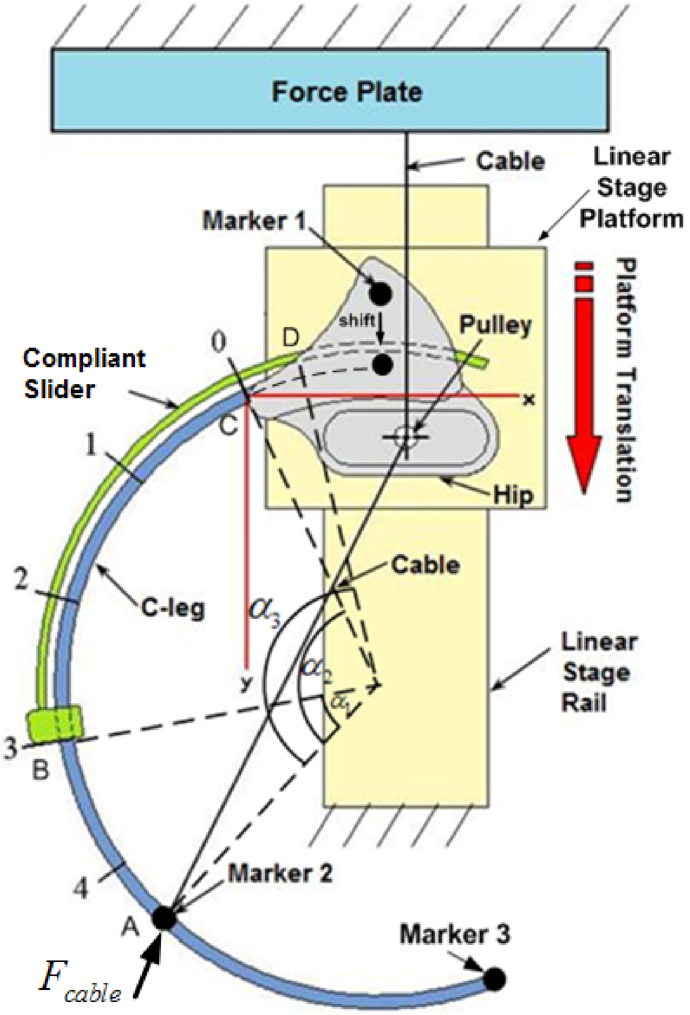
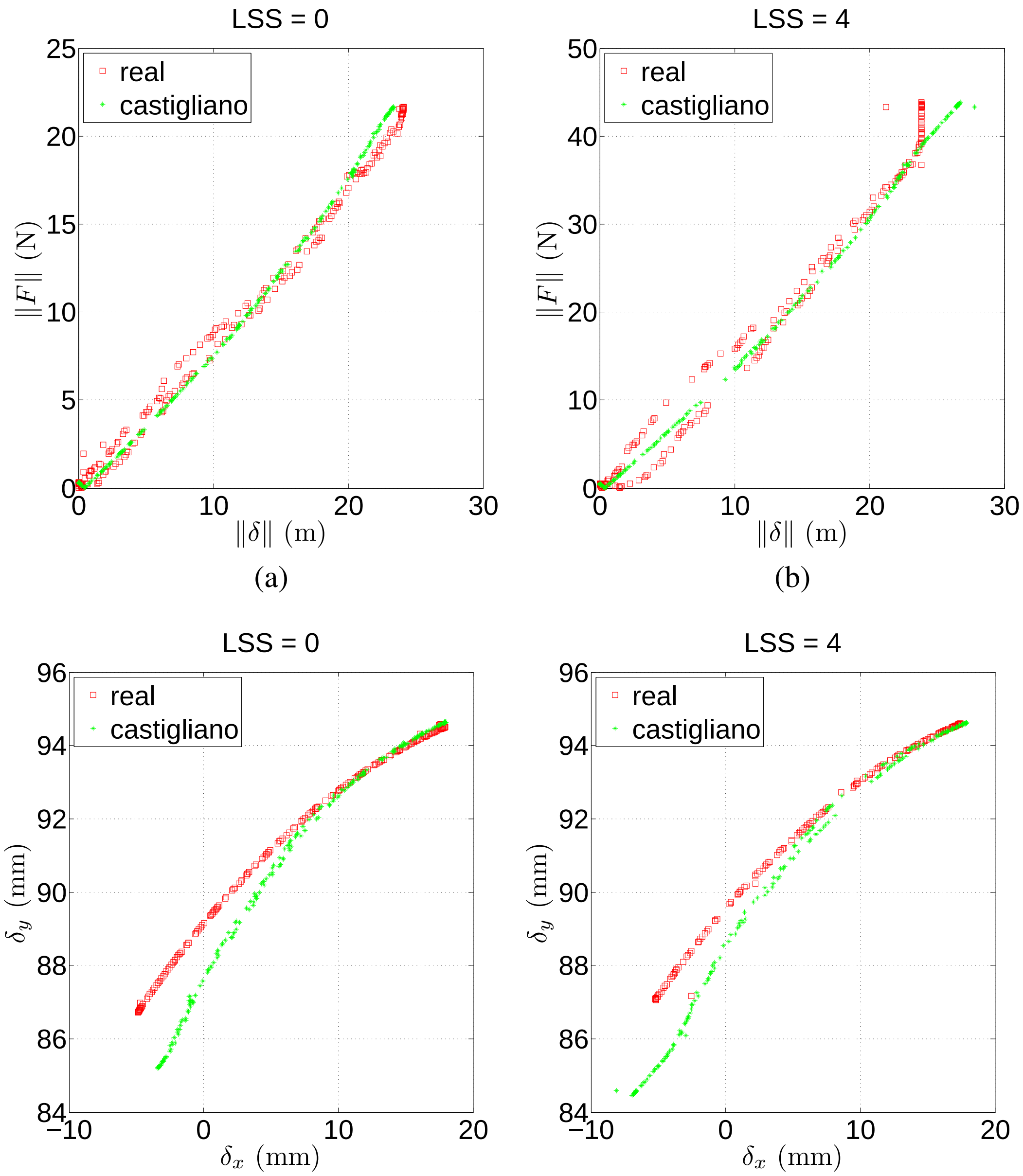


Figure 2.9: Schematic diagram of the experimental setup (Presented with permission from [2]). The leg is rigidly attached to a linear stage platform from the hip. The leg is deflected by moving the stage on the direction indicated by the red arrow.



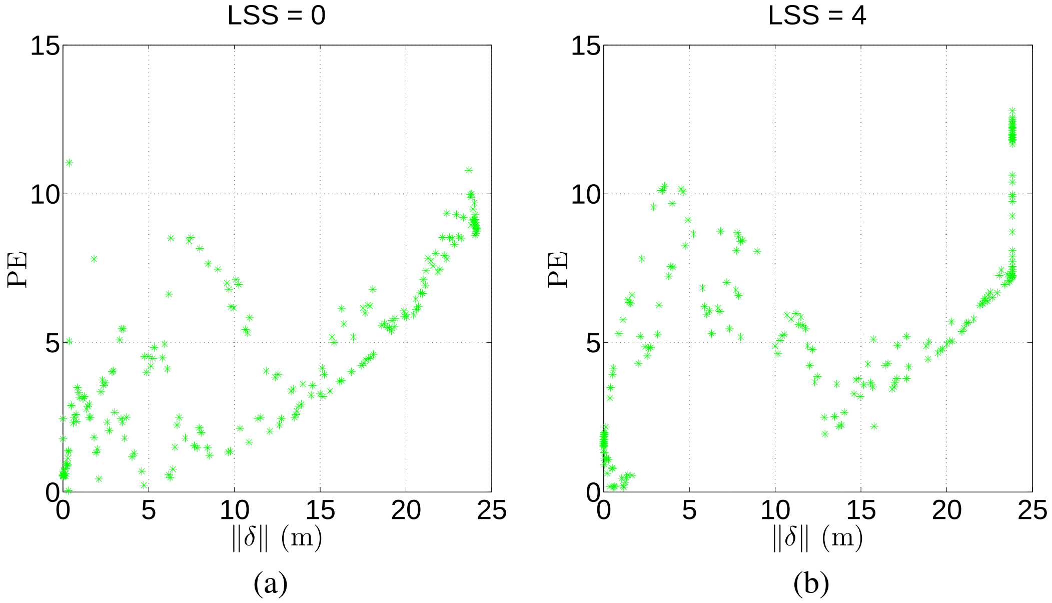
(a) (b) (c) (d)

Figure 2.10: (a)-(b) Comparision of the radial force-deflection relation of Castigliano Model ($*$) with real data (□) at the stiffness extremes, LSS0 and LSS4. The axis are defined in the reference frame attached to the point C (see Fig. 2.9) (c)-(d) Comparision of the deflection path of Castigliano Model ($*$) with real data (□) at the stiffness extremes, LSS0 and LSS4.

In Fig. 2.10, we present the actual and predicted deflection path of loading point (Marker 2). We define the corresponding percentage errors as follows:

$$PE(i) = \frac{||q(i)_{act} - q(i)_{est}||_2}{d_{max}} 100\% \tag{2.15}$$

from $i = 1 : n$ where q is the x-y position of loading point in the coordinate frame that is attached to point C (see Fig. 2.9), n is the number of samples and d_{max} is the magnitude of actual maximum deflection. Since the model used in this thesis is not a tunable stiffness leg, we only consider the two extreme stiffness setting where $LSS = 0$ and $LSS = 4$. As seen from the Fig. 2.11 the error between the actual and estimated deflection path of loading point is less than 15%. This simple, computationally efficient model captures the elastic behavior of a half circular, compliant leg accurately and can be used to capture the dynamics of a running hexapod with a C-leg.



(a) (b)

Figure 2.11: Percentage error (2.15) vs. radial deflection of the Castigliano Model at the stiffness extremes (a) LLS0 (b) LSS4.

2.3 Dynamic Equations of One Legged Robot

In this section, we introduce the dynamical model of the planar half circular compliant legged monopod in rolling contact. The dynamic equations are derived by the Newton-Euler formulation since the expression of leg elastic potential energy as a function of the generalized coordinate is not straightforward . The robot we consider consists of a planar rigid body with a massless compliant leg (there is no inertial force on the leg) attached to the center of mass (see Fig. 2.12). The compliance of the leg allows storage of energy during compression (i.e. when $r < r_u$).

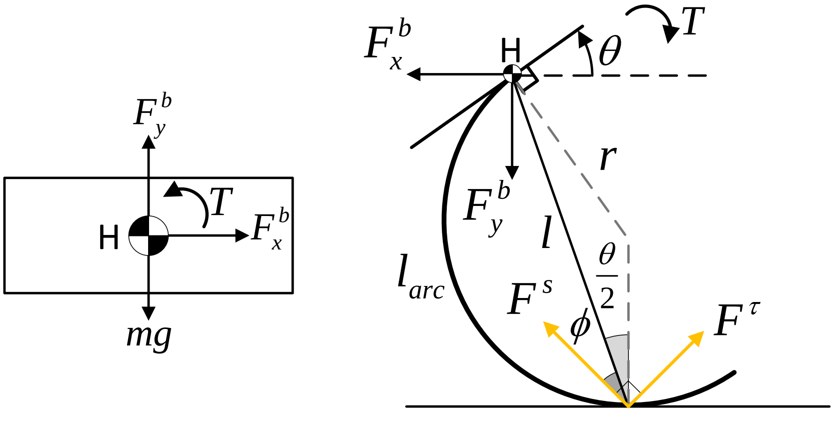


Figure 2.12: External forces and torque acted on the robot and the leg

The robot has two phases of motion; stance and flight. The weight force is applied at the center of mass and the body is supported by the leg spring force F^s. During the stance phase, the leg always touches the ground and the hip torque is controlled by a PD controller. The radius of the curvature of leg changes with the leg angle θ due to stiffness of the leg. In the flight phase, since the legs are not in contact with ground leg spring forces are equal to zero and the robot follows an unactuated projectile motion. In dynamic equations, we ignore the body pitching assuming that the body is a point mass with an infinite rotary inertia. Although, this may initially seem to be a strong assumption, our rationale is that monopod models are often used as embedded control targets for more complex legged robots such as the six legged robot presented in Fig. 1.1. The presence of multiple legs restrict the body to remain close to horizontal. This behavior can be approximated in a simplified model by use of infinite inertia.

The half circular compliant leg with no damping will be assumed to roll without slipping on the ground surface. The associated no slip constraint is that the relative velocity between the contact point on the leg and ground should be equal to zero [62]. To simplify the calculation, the contact point can be treated as a stationary point at every instant during continuous motion. We define a constraint force F^τ to enforce the no-slip condition, assuming that it is perpendicular to the spring force F^s. Note that this constraint force has no effect on the deflection.

Fig. 2.12 shows external forces and the hip torque acting on the robot. In this figure, F^b_x and F^b_y are the reaction forces transmitted to the monopod body, T is the torque applied from the hip point, l is the distance between hip and contact point, and ϕ is the spring angle that is defined between l and F^s as

$$\phi = \frac{(\pi - \theta)}{2} - \tan(\frac{F^s_y}{F^s_x}). \tag{2.16}$$

The states of the robot are defined as $[x, \dot{x}, y, \dot{y}, \theta, \dot{\theta}]$. According to the free body diagram in Fig. 2.12, the equations of motion with the constraint force during the stance phase can be written as

$$\begin{aligned} m_b\, \ddot{x} - F^b_x &= 0 \qquad (2.17)\\ m_b\, \ddot{y} - F^b_y + m_b g &= 0\\ I_{leg}\, \ddot{\theta} + T + |\vec{F^s}|\; l\; sin\phi - |\vec{F^\tau}|\; l\; cos\phi &= 0\\ F^s_x + F^b_x - F^\tau_x &= 0\\ -F^s_y + F^b_y - F^\tau_y &= 0\\ \ddot{x_c} &= 0, \end{aligned}$$

where we have

$$\begin{aligned} [F^s_x \quad F^s_y] &= |F^s|[sin(\phi + \frac{\theta}{2}) \quad cos(\phi + \frac{\theta}{2})]\\ [F^\tau_x \quad F^\tau_y] &= |F^\tau|[cos(\phi + \frac{\theta}{2}) \quad sin(\phi + \frac{\theta}{2})]. \end{aligned}$$

We calculate the leg radius r assuming that the leg preserves its circular shape after deflection. We fit a circle which has an angle θ, passing from the hip point and is tangent to the ground. This radius can be calculated from the arc length of the leg starting from the fixed end of the leg to the contact point as

$$r = \frac{l_{arc}}{(\pi - \theta)}. \tag{2.18}$$

The position of the contact point with respect to x is then equal to

$$x_c \;=\; x + \frac{l_{arc}}{(\pi - \theta)}\, sin\theta. \tag{2.19}$$

If we differentiate x_c with respect to time twice, keeping the arc length l_{arc} constant, we can capture the no-slip constraint as

$$\begin{aligned} \ddot{x}_c &= \ddot{x} + \frac{l_{arc}\ddot{\theta}}{(\pi-\theta)^2}(sin\theta + cos\theta(\pi-\theta)) \\ &+ \frac{2l_{arc}\dot{\theta}^2}{(\pi-\theta)^3}(sin\theta + cos\theta(\pi-\theta) - sin\theta(\pi-\theta)^2). \end{aligned} \tag{2.20}$$

We now write the dynamic and constraint equations in a matrix form as

$$\begin{bmatrix} m_b & 0 & 0 & -1 & 0 & 0 \\ 0 & m_b & 0 & 0 & -1 & 0 \\ 0 & 0 & I_{leg} & 0 & 0 & -l_{arc}cos\phi \\ 0 & 0 & 0 & 1 & 0 & -cos\frac{\theta}{2}+\phi \\ 0 & 0 & 0 & 0 & 1 & -sin\frac{\theta}{2}+\phi \\ 1 & 0 & C_1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \\ F^b_x \\ F^b_y \\ F^\tau \end{bmatrix} = \begin{bmatrix} 0 \\ -m_b g \\ -T - F^s l_{arc} sin\phi \\ -F^s_x \\ F^s_y \\ C_2 \end{bmatrix}, \tag{2.21}$$

where

$$\begin{aligned} C_1 &:= \frac{l_{arc}}{(\pi-\theta)^2}(sin\theta + cos\theta(\pi-\theta)) \\ C_2 &:= -2l_{arc}\frac{\dot{\theta}^2}{(\pi-\theta)^3}(sin\theta + cos\theta(\pi-\theta) - sin(\theta)(\pi-\theta)^2). \end{aligned}$$

Note that the last row of (2.21) is the constraint equation for the no-slip case. In the flight phase, the ground reaction forces F_x and F_y acting on the robot are equal to zero and the body follows a ballistic trajectory. Only the gravitational force acts on the body. Therefore, the equations of motion during flight are

$$\begin{aligned} \ddot{x} &= 0 \\ \ddot{y} &= -g \\ \ddot{\theta} &= \frac{T}{I_{leg}}. \end{aligned} \tag{2.22}$$

The phase transition conditions of the robot are given in Table 2.3. From the Fig. 2.7 it can be seen that, when the radius of the leg equals to undeflected leg radius, the height of the COM is $r_u(1 + cos(\theta))$. If we use $r = r_u$ in Eq. 9, we obtain $\delta_x = 0$ and $\delta_y = 0$ means that both horizontal and vertical component of the ground reaction force are zero. We used this geometric constraint to detect touchdown and liftoff events without using force data.

Fig. 2.13 demonstrates the energy change of robot during simulation. Since the system does not have any damping, the mechanical energy is conserved at each step.

Table 2.3: Event functions for stance and flight phase.

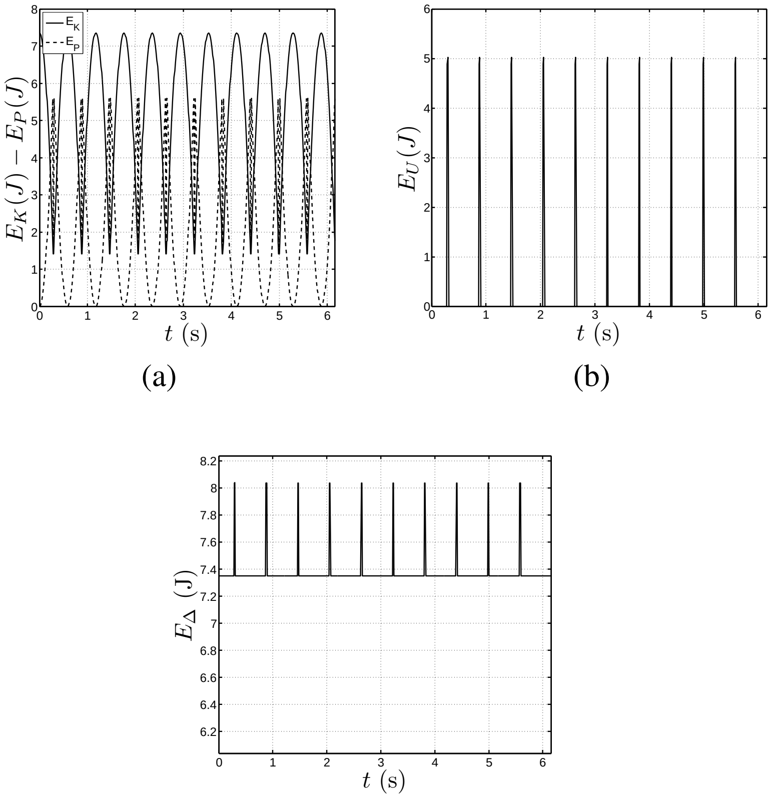
Event	Equation
Stance to Flight	$y = (\cos\theta + 1)r_u$
Flight to Stance	$y = (\cos\theta_{td} + 1)r_u$

2.3.1 Free Fall Simulation

In this simulation, we desire to verify the dynamic equations obtained in the previous section. An important way of validating model accuracy is to consider "no-torque" simulation and observe the total energy of the system to be constant. For this purpose, we performed a free-fall simulation that was carried out for the set of initial conditions in which all state variables except initial height of the robot were set to zero and no control input was applied during the simulation. The physical parameters of the robot's body and leg, which closely match the physical properties of Edubot, are given in Table 2.2.

The robot was left from $y = 0.5$m with a zero leg angle, θ, and it is allowed to exhibit ten successive jumps. When the robot strikes the ground, its leg starts to deflect. The amount of deflection equals the difference between deformed and undeformed coordinates (see (2.14)) of the contact point on leg. The coordinate of contact point is calculated from circle geometry. The problem is finding a circle which passes from hip and also tangent to leg from hip and ground contact point. It can be easily seen from (2.14) when the radius of the leg equals to undeflected leg radius the deflection of the contact point and the corresponding spring forces are equivalent to zero, which is also satisfied with the event functions that are given in Table 2.3. If the deflection amount is greater than zero, the unknown ground reaction forces in the dynamic equations are calculated by the deflection matrix given in (2.14). Considering the constraint equations, the position and velocity of the COM, the angle and angular velocity of the leg are obtained from integration of the accelerations in (2.21) at each time step. We used MATLAB built-in function ode45 that numerically solve differential equations with a variable step Runge-Kutta Method and we printed results with a time step $\Delta T = 0.001$.

Fig. 2.13a and Fig. 2.13b show the kinetic and potential energy variations during the simulation. Since the system has no damping, and we do not apply any torque during the stance phase, the total mechanical energy is conserved (see Fig. 2.13c) validates our model.



(a) (b)

(c)

Figure 2.13: The free fall simulation of the C-leg robot. The robot is let free to fall under gravity with an initial condition of $[x = 0, \dot{x} = 0, y = 0.5\ m, \dot{y} = 0, \theta = 0, \dot{\theta} = 0]$ for ten successive jumping. The time step of ode45 is taken 0.001 sec. The energy plots are: (a) The kinetic and gravitational potential energy vs. time (b) The strain energy vs. time (c) The total mechanical energy vs. time plots of the robot

2.4 Conclusion

This chapter demonstrates the dynamical equations of the monopod with a half circular compliant leg under the assumption of rolling without slipping. We used Castigliano's Theorem to obtain force-deflection relation of general compliant curved member and we integrated this solution approach into our dynamic model to calculate unknown external forces applied to the body. Although the leg model in this section is intended to be used for RHex type of robots, it can be applicable to any compliant curved structures other than half circular one. Throughout the thesis, we will use dynamic equations obtained in this section.

CHAPTER 3

OPTIMAL CONTROL OF LOCOMOTION

3.1 Introduction

The main objective of this section is the development of an algorithm based on optimal control theory that allows a half circular compliant (HCC) legged monopod to run with a desired forward velocity or height starting from zero initial state. The control algorithm chooses the most suitable touchdown angle and reference signal of the PD controller for each step of the robot, while considering the dynamic stability of the system.

3.2 Horizontal Velocity Control

First we consider the problem where the horizontal velocity of robot is regulated about a certain desired value. The controller used for this purpose is motivated by Raibert's three part controller [17].

The one legged robot system has only a single input applied from the hip actuator. During the stance phase, we use a Proportional-Derivative (PD) controller to apply a torque to the hip as

$$T = -K_p\,(\theta - \theta_{des}) - K_d\,\dot{\theta}\,, \tag{3.1}$$

where T is the actuator torque generated at the hip, $\dot{\theta}$ is the angular velocity of the leg, θ_{des} is desired leg angle, and K_p and K_d are PD controller gains, respectively.

As explained in [17], the selection of footholds is critical since it directly affects the acceleration of the body. To stabilize the horizontal velocity of the robot, we optimize the touchdown angle and reference signal of the controller during the stance phase.

Firstly, we manually tuned K_p and K_d such that the robot exhibits smooth and stable running. Table 3.1 summarizes the K_p and K_d values we obtained through the manual tuning for different values of θ_{td}.

Our aim is to bring the horizontal velocity of the robot to the desired velocity. Taking the touchdown and liftoff conditions of Table 2.3 into account, the dynamic equations of the system, (2.17) and (2.22), were solved so that the horizontal velocity of the COM is equal to desired velocity. In other words, we find the control reference θ_{des} and touchdown angle θ_{td} that minimize the cost functional

$$J_1 = \sum_{i=1}^{n} (\dot{x}_{avg,i} - \dot{x}_{des})^2 + 10 \sum_{i=n-9}^{n} |\dot{x}_{avg,i-1} - \dot{x}_{avg,i}| \,, \tag{3.2}$$

subject to (2.17) and (2.22), where $\dot{x}_{avg,i}$ is the mean horizontal velocity during the flight phase of the i^{th} stride and n is the total number of strides. The first term in this functional attracts towards the desired velocity while the second term promotes stability and convergence. where $\dot{x}_{avg,i}$ is the mean horizontal velocity of flight phase of ith step and n is the total step number. Note that, while walking or running, we apply same constant angle and velocity reference at each step. To prevent undesirable solutions and to improve the feasibility of the solution, we used the penalty function below

$$P(\theta) = \begin{cases} 0 & \text{if } \theta \leq 65^o \\ +\infty & \text{if } \theta > 65^o \end{cases} . \tag{3.3}$$

Table 3.1: Optimized K_p and K_d values for different ranges of $\dot{x}_{des}$

$\dot{x}_{des}$(m/s)	0.1-0.3	0.4-1.0	1.1-1.3	1.4-1.8	1.9-2.0
K_p	100	200	300	335	380
K_d	7	8	9	13	13

3.3 Height Control

The aim of the second problem is to keep height of the robot around the desired value. The controller used for this purpose is motivated by Raibert's three part controller [17].

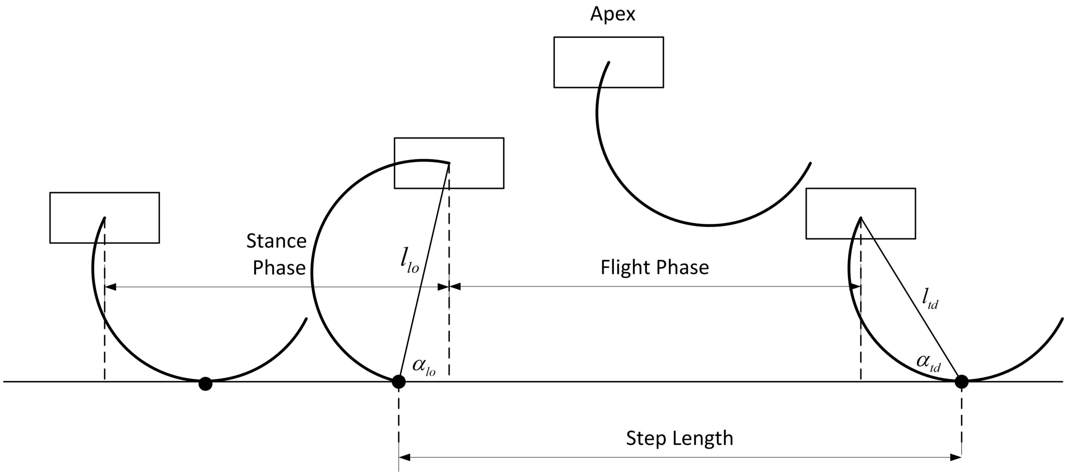


Figure 3.1: Stance, flight phase and step length of the robot

Similar to previous problem, during the stance phase, we use a Proportional-Derivative (PD) controller to apply a torque to the hip as

$$T = -K_p(\theta - \theta_{des}) - K_d\,\dot{\theta}, \tag{3.4}$$

where T is the actuator torque generated at the hip, θ is the actual leg angle, $\dot{\theta}$ is the actual angular velocity of the leg, θ_{des} is the desired leg angle, and K_p and K_d are PD controller gains, respectively.

As explained in [17], the selection of footholds is critical since it directly affects the acceleration of the body. To stabilize the height of the robot, we optimize the touchdown angle θ_{td} and the parameters of controller $[\theta_{des},\ K_p,\ K_d]$ during the stance phase. The optimal control problem can be defined as

$$\begin{aligned} \min J_2 = \sum_{i=1}^{n} (y_{apex,i} - y_{apex,des})^2 \\ \text{subject to (2.17) and (2.22),} \end{aligned} \tag{3.5}$$

where $y_{apex,i}$ is the height of the COM at ith step and n is the total step number.

Table 3.2: Parameter optimization results of height control

	Results of Optimization			
y_{des} (m)	θ_{des} (rad)	θ_{td} (deg)	K_p (Nm/rad)	K_d (Nm/(rad/s))
0.15	-0.5446	34.1277	33,4355	0,4152
0.20	-0.5424	36.9876	32,3226	0,4000
0.25	-0.5303	38.3130	31,9095	0,3829
0.30	-0.5312	39.0934	31,5695	0,3759
0.35	-0.5271	39.6882	31,5902	0,3702
0.40	-0.5266	40.1411	32.4941	0,3667
0.45	-0.5307	40.4255	33.9229	0,3621
0.50	-0.5342	40.6669	34.7490	0,3611

3.4 Results

3.4.1 Horizontal Velocity Control

Fig. 3.2 presents the solution of the optimization problem. We scaled the desired velocity from 0.1 m/s to 2 m/s with an increment 0.1 m/s. Initial states were manually chosen as $x = 0m$, $\dot{x} = 0m/s$, $y = 0.25m$, $\dot{y} = 0m/s$, ensuring that they remained within the domain of attraction of the steady-state locomotion.

The robot runs fifty steps and the cost function in (3.2) is minimized using MATLAB optimization toolbox built-in function *fminsearch* starting at an initial estimate $[\theta_{des}, \theta_{td}] = [0.2rad, 10^o]$.The function *fminsearch* uses Nelder-Mead Simplex direct search method [63] to find the minimum of a multidimensional unconstrained function. The method generates an n-dimensional simplex (where n is the length of the optimization space) and, usually, at the each iteration eliminate one of the vertices which gives the worst function value and continues until the user-defined tolerance values are satisfied. This method does not use gradients and finds the local optimum of the function very quickly, but it can not converge to global minimum if the initial estimate of the problem is distant from the global minimum. As seen from Fig. 3.2, when we increase the desired forward velocity the touchdown angle increases and these changes the vertical force applied to the body. Since the body exhibits projectile motion in flight phase, the change in the vertical force effects directly the duration

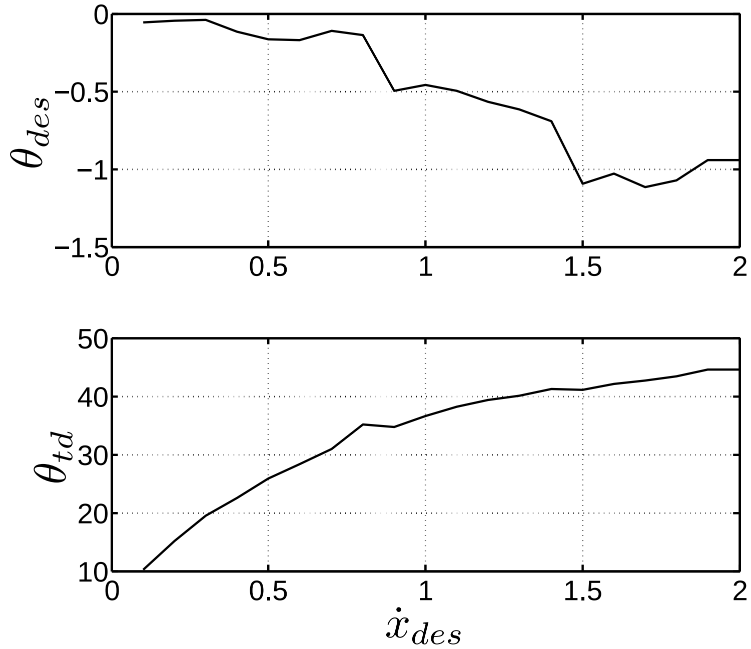


Figure 3.2: Optimization results of horizontal velocity control for different choices of desired velocity.

of flight phase. Similarly, the desired leg angle decreases as the $\dot{x}_{des}$ increases. By using the values $[\theta_{des}, \theta_{td}]$ obtained in section 3.2, one can find the sequence of footholds and corresponding thrust for the desired running profile.

The kinetic energy of the robot during the stance phase can be written as

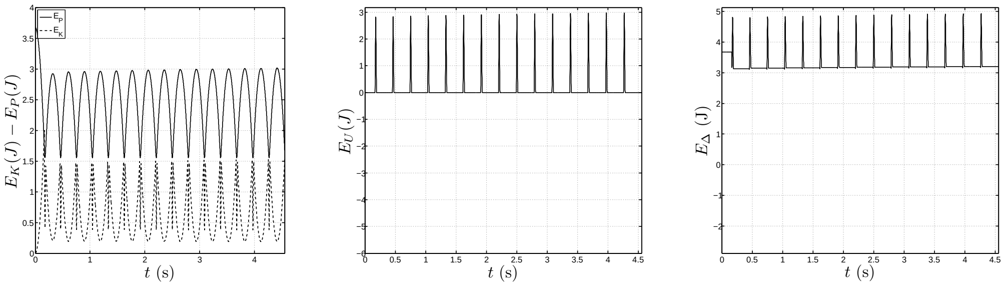
$$E_K = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) + \frac{1}{2}I_{leg}\dot{\theta}^2. \tag{3.6}$$

In flight we ignore the rotational kinetic energy of the leg. The total potential energy of the robot is

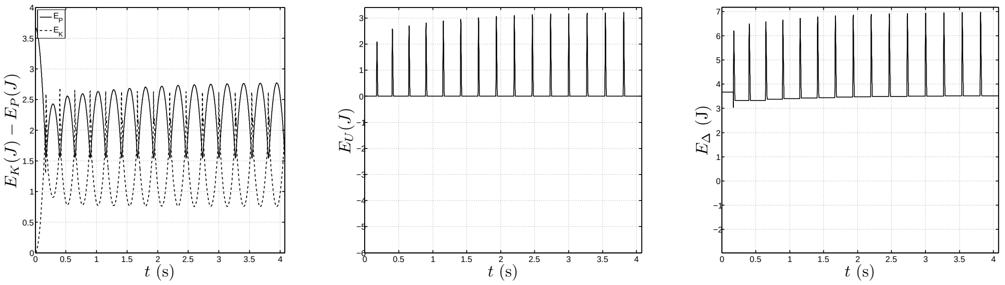
$$E_K = mgy + U_{strain} \tag{3.7}$$

where U_{strain} is given in (2.10).

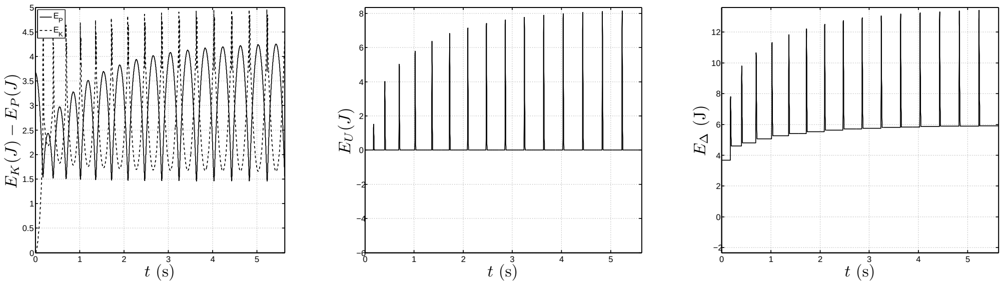
Fig. 3.3 compares the energy exchange vs. time corresponding to about ten consecutive steps of a typical running with desired forward speeds (0.5, 1.0, 1.5, 2.0 m/s). The first column is the kinetic and gravitational potential energy, the second column is the elastic strain energy that is stored in the leg due to the deformation and the last column is the total mechanical energy that is sum of kinetic energy and potential energies.



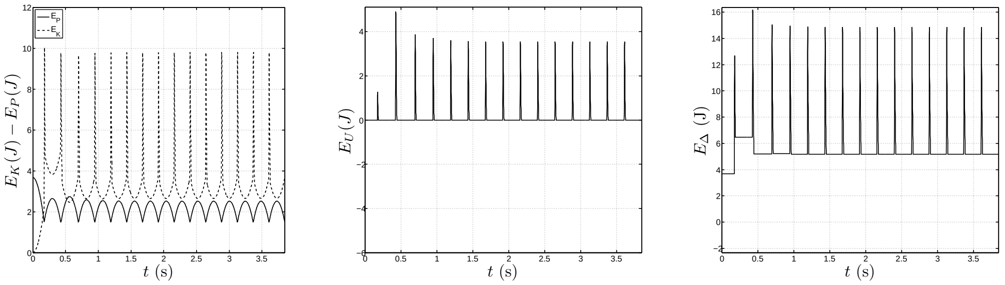
(a) $\dot{x}_{des}$ = 0.5 m/s (b) $\dot{x}_{des}$ = 0.5 m/s (c) $\dot{x}_{des}$ = 0.5 m/s



(d) $\dot{x}_{des}$ = 1.0 m/s (e) $\dot{x}_{des}$ = 1.0 m/s (f) $\dot{x}_{des}$ = 1.0 m/s



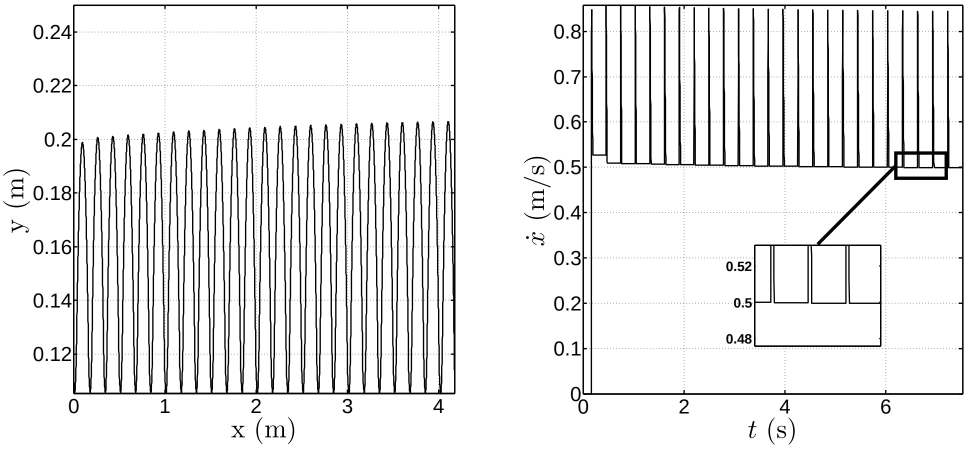
(g) $\dot{x}_{des}$ = 1.5 m/s (h) $\dot{x}_{des}$ = 1.5 m/s (i) $\dot{x}_{des}$ = 1.5 m/s



(j) $\dot{x}_{des}$ = 2.0 m/s (k) $\dot{x}_{des}$ = 2.0 m/s (l) $\dot{x}_{des}$ = 2.0 m/s

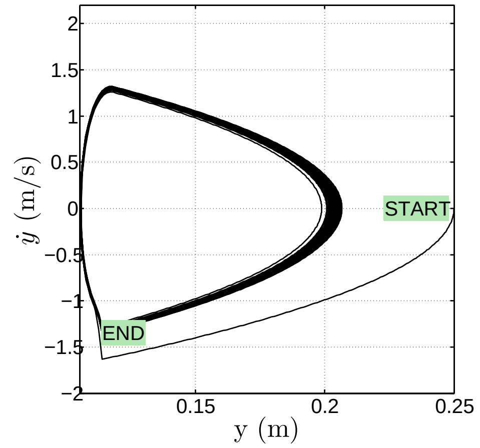
Figure 3.3: The first column presents kinetic and gravitational potential energy, second column is the strain energy and the last column is the total mechanical energy of the robot.

In our model no damping is employed and no impact is considered when the leg collides with the ground since the leg is compliant, the only way that we can dissipate energy is to use of actuator dynamics (see Fig. 3.8). As seen from the last column of Fig. 3.3, for all running speeds the total mechanical energy level stabilizes at equilibrium. During the stance phase of the slow forward speed running, while the kinetic and gravitational potential energy decreases the elastic strain energy increases, which means that the leg stores most of its energy and acts as brakes. This is not valid for the high speed running where both the kinetic and elastic strain energy increase as the potential energy decreases during the leg contacts with ground. The findings of this study is in agreement with previous study [64].



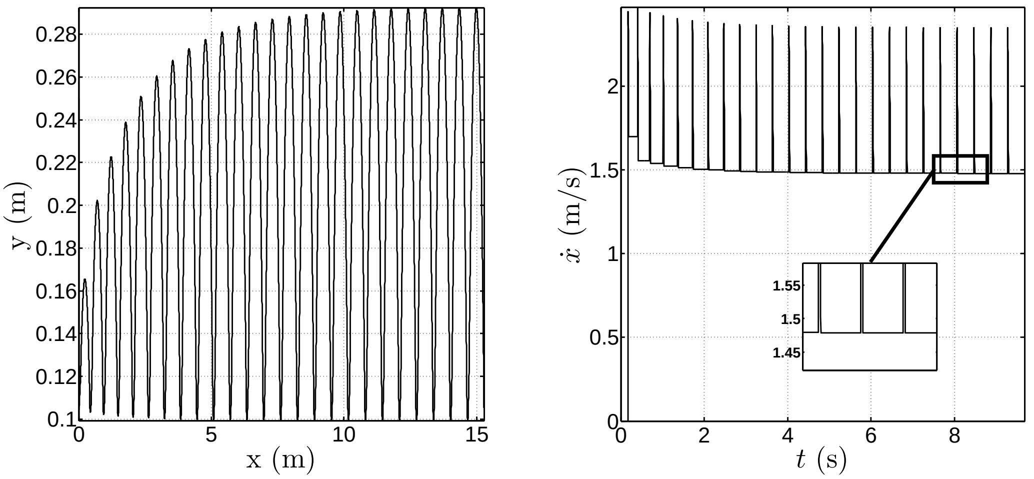
(a) y vs. x plot of COM

(b) $\dot{x}$ vs. time

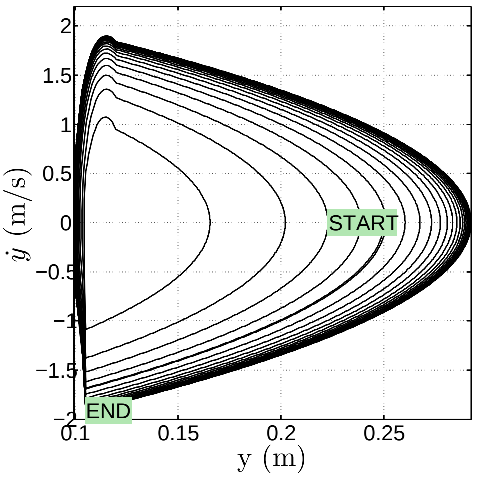


(c) phase plot

Figure 3.4: 25 step running with a horizontal velocity 0.5 m/s

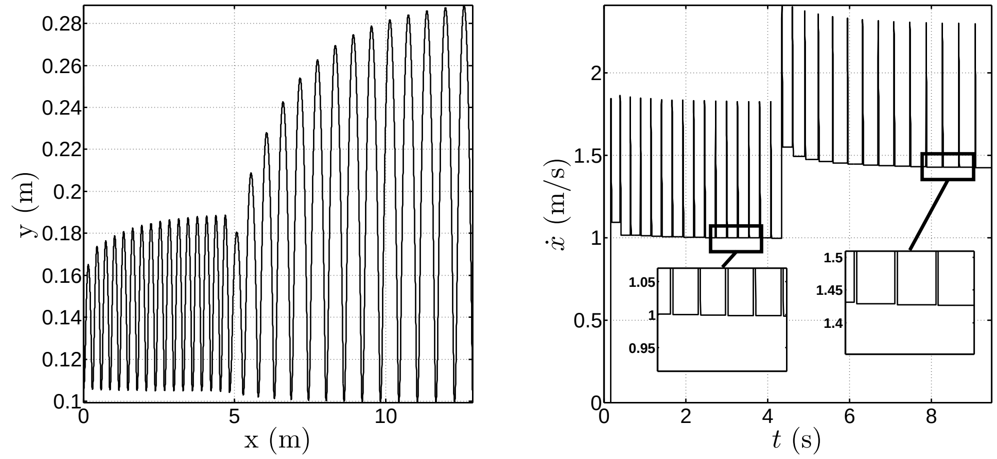


(a) y vs. x plot of COM (b) $\dot{x}$ vs. time



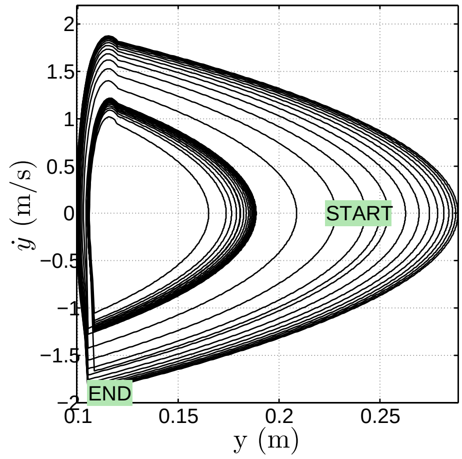
(c) phase plot

Figure 3.5: 25 step running with a horizontal velocity 1.5 m/s



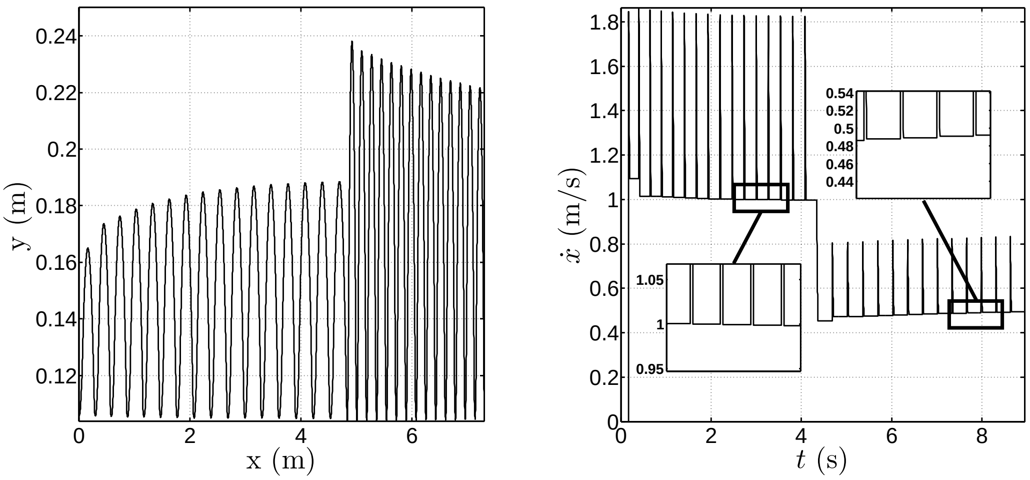
(a) y vs. x plot of COM

(b) $\dot{x}$ vs. time



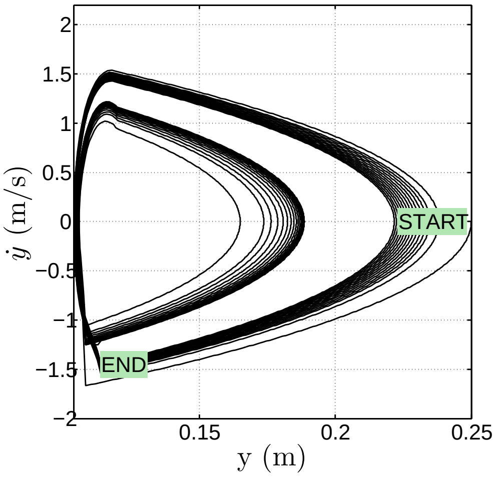
(c) phase plot

Figure 3.6: 30 step running with a horizontal velocity of the first 15 steps 1.0 m/s, last 15 steps 1.5 m/s



(a) y vs. x plot of COM

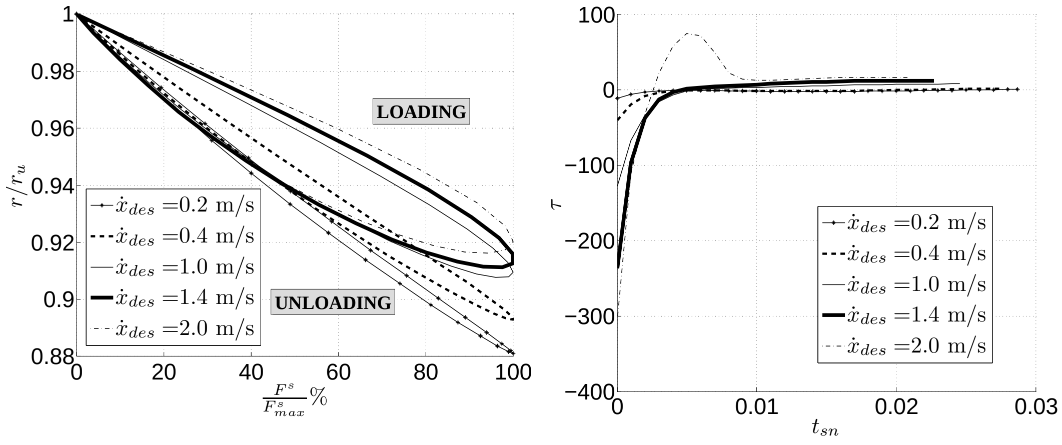
(b) $\dot{x}$ vs. time



(c) phase plot

Figure 3.7: 30 step running with a horizontal velocity of the first 15 steps 1.0 m/s, last 15 steps 0.5 m/s

Fig. 3.4 and Fig. 3.5 show 25 steps running simulations using parameters $[\theta_{des}, \theta_{td}]$ obtained in section 3.2. As seen from these figures, the horizontal velocity of the robot converges to the desired velocity after about five steps. If the initial energy of the system is not around the total energy at steady-state, then the robot attempts to gain energy by compressing its leg in first two or three steps. Our results show that even though we desire to control the horizontal velocity, we also stabilize the steady-state hopping height of the robot.



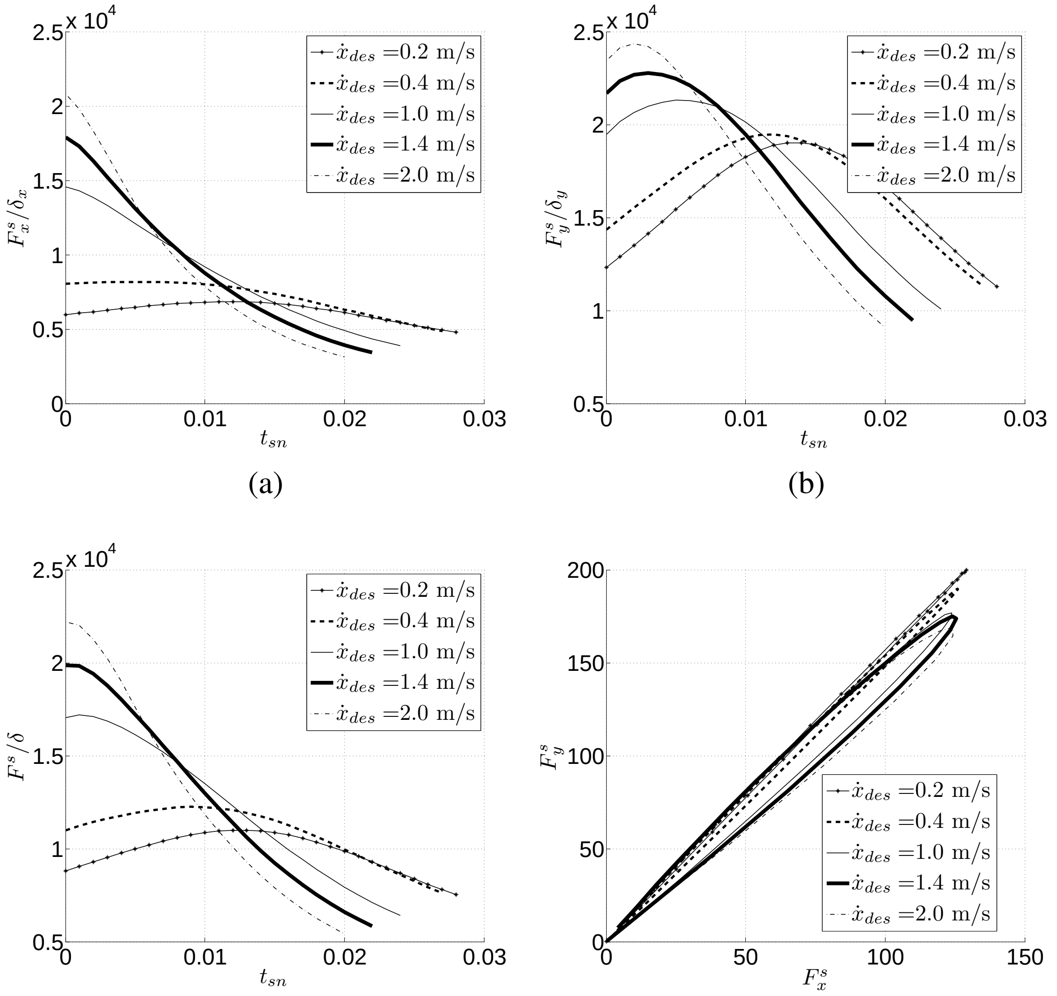
(a) (b)

Figure 3.8: (a) The normalized leg radius vs. percentage of the total spring force to the its max. value and (b) The torque applied during the stance phase for the desired forward velocity from 0.2 – 2.0 m/s.

Fig. 3.6 and Fig. 3.7 show two different velocity profile running results. The robot runs 30 steps, in the first 15 steps, it runs with a different velocity than last 15 steps, i.e. the touchdown angle and controller parameters are changed after 15 steps. In order to investigate the steady-state error of the second part of running, we simulate the system with different velocity pairs. We first increased the desired velocity halfway through the running and then vice versa. Fig. 3.10 shows the desired velocities and deviations from the desired velocity with a percentage error measure defined as

$$N\text{-}RMSE := \frac{\sqrt{\frac{1}{15}\sum_{i=15}^{30}(\dot{x}_{2_{avg,i}} - \dot{x}_{2_{des}})^2}}{\dot{x}_{2_{des}}}, \tag{3.8}$$

where $\dot{x}_{2_{avg,i}}$ is the average horizontal flight velocity during the i^{th} step. This measure quantifies how far the final steady-state velocity is from the desired velocity across different initial conditions.



(a) (b)

(c) (d)

Figure 3.9: (a) The steady-state horizontal stiffness of the desired forward velocity from 0.2 – 2.0 m/s during the stance phase (b) The steady-state vertical stiffness of the desired forward velocity from 0.2 – 2.0 m/s during the stance phase (c) The total stiffness of the desired forward velocity from 0.2 – 2.0 m/s during the stance phase (d) The horizontal vs. vertical spring force of the desired forward velocity from 0.2 – 2.0 m/s during the stance phase

Fig. 3.8 shows the relation between the normalized instantaneous leg radius (r/r_u) and the total spring force $\|F^s\|$ for a different forward running speed. As depicted in the figure, the loading and unloading curve characteristics imply that absorbed energy of the leg is smaller than the released energy during the stance phase. The difference between these two energies is the energy lost resulted by nonlinear characteristic of the compliant circular leg. The energy lost increases by the forward speed of the robot. For a small forward speed the force-deflection relation is almost linear.

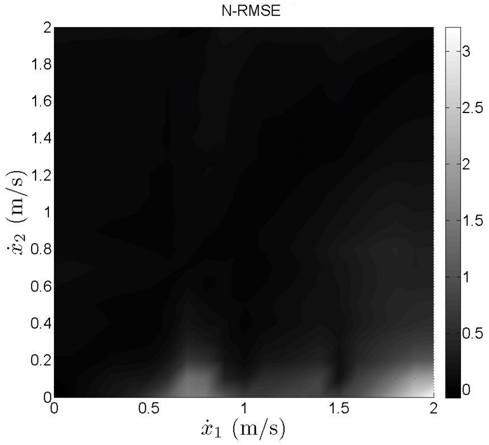


Figure 3.10: Accuracy of steady-state velocity control for different initial condition and desired velocity pairs. The color code corresponds to the values of the N-RMSE calculated by (3.8).

3.4.2 Height Control

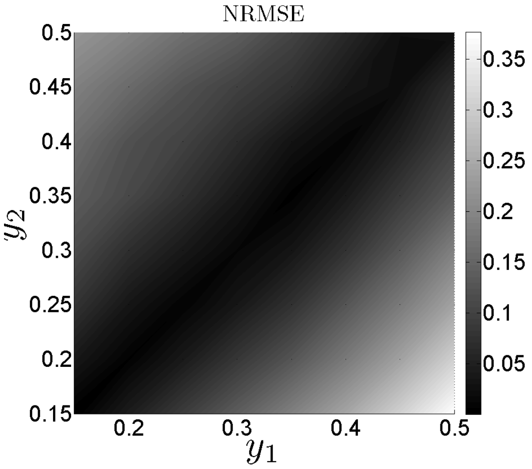
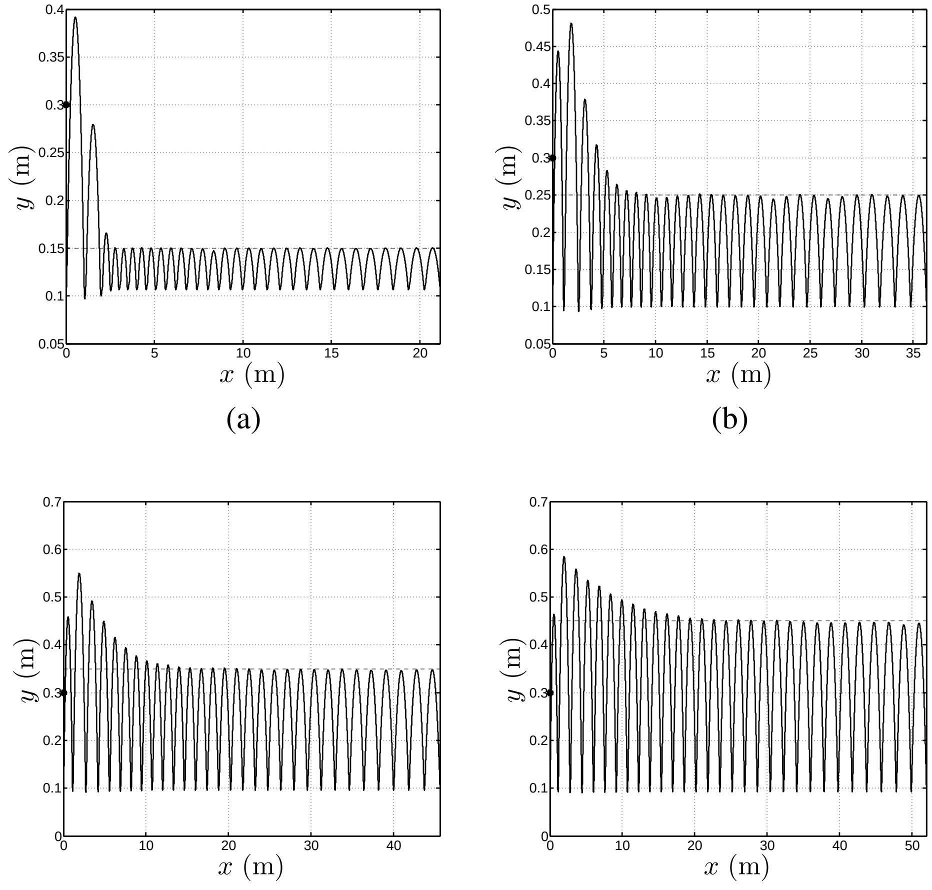


Figure 3.11: Accuracy of steady-state height control for different initial condition and desired height pairs. The color code corresponds to the values of the N-RMSE calculated by (3.9).

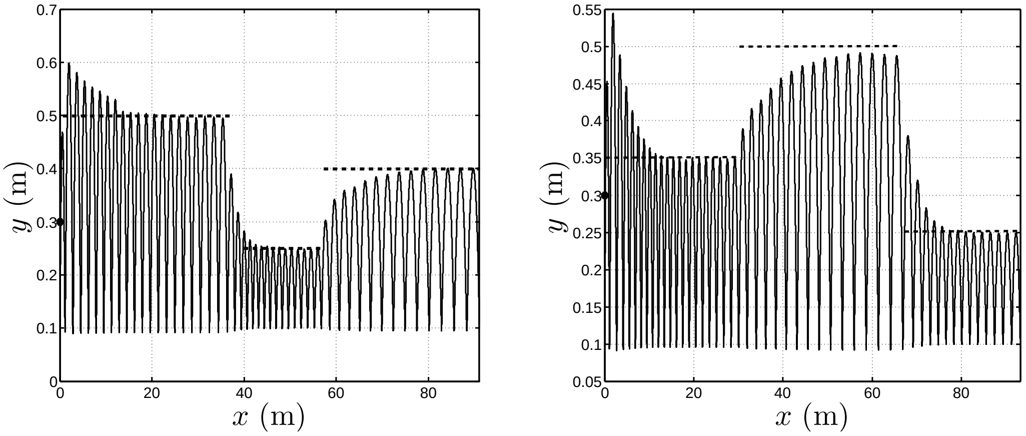
Fig. 3.12 shows the height control results for a different desired hopping height. The robot initially left from $y = 0.3\ m$ with a zero velocity . We run the robot 30 steps with the controller parameters given in Table 3.2 that were obtained by minimization of the objective function (3.19). Although the thrust is not applied vertically, the height of the robot converges to desired height and the robot presents stable motion. It takes about five or six steps to arrive at equilibrium. When the desired height is close to initial height of the robot it takes two or three steps to converge to the desired height.



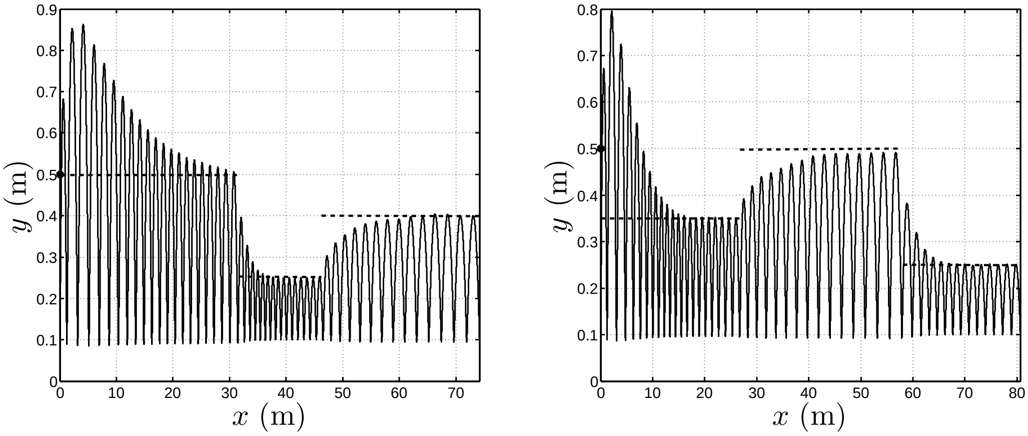
(a) (b)

(c) (d)

Figure 3.12: 30 step running with a desired height (a) 0.15 m (b) 0.25 m (c) 0.35 m (d) 0.45 m and initial height 0.3 m



(a) initial height of COM is 0.3 m (b) initial height of COM is 0.3 m



(c) initial height of COM is 0.5 m (d) initial height of COM is 0.5 m

Figure 3.13: 50 step running with a desired height (a) first 20 steps 0.5 m, then 15 steps 0.25 m, last 15 steps 0.4 m (b) first 20 steps 0.35 m, then 15 steps 0.5 m, last 15 steps 0.25 m (c) first 20 steps 0.5 m, then 15 steps 0.25 m, last 15 steps 0.4 m (d) first 20 steps 0.35 m, then 15 steps 0.5 m, last 15 steps 0.25 m

We repeated the above simulations with an initial height 0.5 *m* to realize the disturbance effect of the initial conditions. As the difference between initial and desired height increase, the oscillation amount of the COM increases (see Fig. 3.13c and Fig. 3.13d). The same rule is valid for the desired height that changes during the running. In Fig. 3.13, we applied different thrust by changing the controller parameters and touchdown angle of the leg after 15 and 30 steps. It requires again five or six steps for the hopping amplitude to stabilize the desired value. Similar to previous section, we investigated the steady-state error of the second part of running when the target height of the robot is changed after the halfway of running. We first increased the desired height halfway through the running and then vice versa. Fig. 3.11 shows the desired height and deviations from the desired height with a percentage error measure defined as

$$N\text{-}RMSE := \frac{\sqrt{\frac{1}{15}\sum_{i=15}^{30}(y_{apex,i} - y_{2_{des}})^2}}{y_{2_{des}}}, \tag{3.9}$$

where $y_{2_{apex,i}}$ is the apex height of the robot at the i^{th} step. This measure quantifies how far the final steady-state height is from the desired height across different initial conditions. As seen from the Table 3.2, the reference leg angle and K_d value are almost constant for different desired hopping height. The proportional constant of the controller K_p and the touchdown angle θ_{td} are dominant parameters that change the dynamics of the locomotion.

3.5 Stability Analysis

Designing a controller which achieves dynamically stable and self-balancing gaits for one legged robots is important since this type of gaits are more energy efficient and faster than statically ones [17, 18]. In this section we examine the stability of a monopod hopper motion on a limit cycle. The conventional stability definitions such as stability around the equilibrium point may not be applicable to analyze the locomotion stability of monopod robots since it is very hard to find an equilibrium point where the system states can stay forever. [65] suggests an approximate solution for the dynamics of the simple SLIP model which includes gravitational effects. They investigate periodicity of the motion with an one dimensional apex return map. Similar return map analysis are performed using analytic map of two successive touchdown states or two successive apex states. With tuned controller parameters there exists stable periodic gaits in the controlled SLIP model.

In the rest of this section, we apply the *Poincare Map* or called *Return Map* analysis to the our system. We begin with defining some useful theorems for the analysis of periodic motions of dynamical systems.

Definition 3.5.1 ***Limit Cycle*** *is a closed trajectory whose neighboring trajectories spiral either toward or away from it as time goes to infinity [66].*

Definition 3.5.2 ***A limit cycle is stable*** *if all neighboring trajectories approach to limit cycle.*

Consider the flow defined by differential equation

$$\dot{x} = F(x) \tag{3.10}$$

where $F : D \rightarrow R^n$ is a continuously differentiable map. Poincare Map discretizes the continuous time nth order autonomous systems by an $(n-1)$ order discrete-time system [67] and gives iterative relation rather than a differential relation between the two successive intersection points. It is useful for analyzing high-order systems since the dimension of the problem is reduced. For example, in the three dimensional state space, we draw a surface, S, that cuts through the limit cycle as shown in Fig. 3.14. Let e is the point where the periodic orbit of the nth order system intersects the $(n-1)th$ order manifold. The trajectory starting from e will pass the same point in T seconds, where T is the period of periodic orbit (see Fig. 3.14). The trajectories starting on S whose initial states are very close to e intersect S in approximately in T seconds in the vicinity of e. The Poincare Map $P : U \rightarrow S$ is defined

$$P(x) = \Phi(\omega, x) \tag{3.11}$$

where $\Phi(\omega, x)$ is the solution of dynamic equations in Eq. 3.10 with a initial state of x and $t = 0$ and $\omega(t, x)$ is the time it takes for the trajectory starting at x to first return to S.

3.5.1 Apex Return Map

In this section we analyze the stability of the C-leg locomotion by defining a Poincare Section at the apex point, where the robot reaches its maximum height. We use numerical simulations to show that our horizontal velocity controller achieves stable

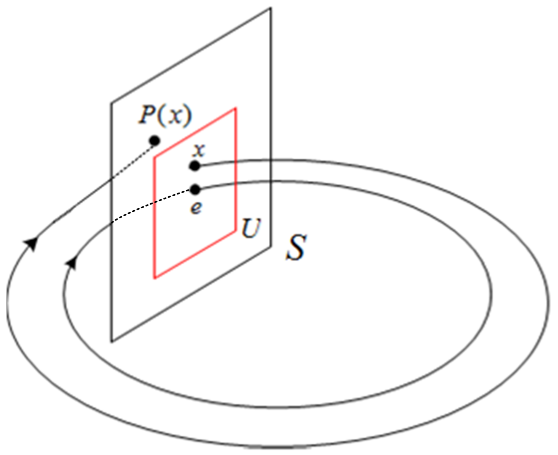


Figure 3.14: Poincare map

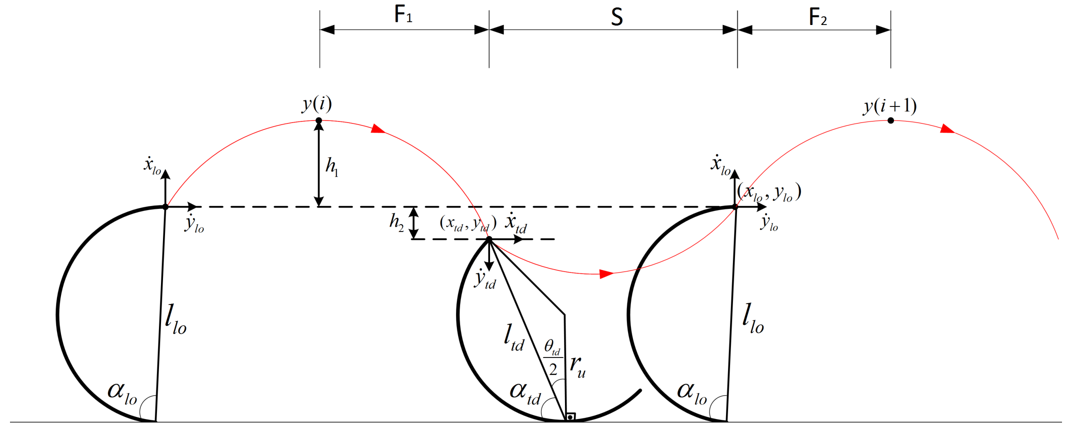


Figure 3.15: The position of the robot at stance and flight phase

locomotion for the forward speed range 0.1 – 2.0 m/s. To apply the return map analysis to our system first we need to obtain a transition function that calculates the dependency of two subsequent apex heights $y(i)$ and $y(i+1)$. During the flight phase the horizontal velocity is constant and the vertical velocity is zero at apex. Since the vertical velocity at apex is zero, the dimension of the map is decreased by one. The apex-to-apex map, P, starts with flight phase and then continues with stance phase and reaches the next apex with an other flight phase. The transition can be considered as a combination of three subsequent maps that consist of continuous differential equations (2.21) and (2.22) and discrete impact events (Table 2.3) and can be written as $P = F_2 \circ S \circ F_1$. In this map F_1 is the apex to touchdown, S is the touchdown to lift-off and F_2 is the lift-off to next apex dynamics of the robot (see Fig. 3.15).

The Poincare Map P, or termed the "stride function" by Mcgeer [24], determines the states of the next apex as a function of previous apex states. During stance, the C-leg locomotion is non-integrable since the gravity forces are not neglected. As a result, we derive the apex return map numerically through simulation. The gait is called stable if all eigenvalues of the linearized apex return map around the fixed point have an absolute value smaller than one [68, 69]. The fixed points of the map **P**,

$$\mathbf{z}_{i+1} = \mathbf{P}(\mathbf{z}_i) \tag{3.12}$$

where $\mathbf{z} = [\dot{x}, y]$ and $\mathbf{P} = [P_1, P_2]$, correspond to period-1 oscillatory motion on a limit cycle, i.e. the system returns to the same states after one step. We can also write the above equation as

$$\begin{aligned} \dot{x}_{n+1} &= P_1(\dot{x}_n, y_n) \\ y_{n+1} &= P_2(\dot{x}_n, y_n) \qquad n = 1, 2, ... \end{aligned} \tag{3.13}$$

To find the Jacobian numerically, we perturbed the initial condition in a small neighborhood of the fixed point and recorded the next apex states. Let a fixed point **p** and the near by point $\bar{\mathbf{p}}$ be defined as

$$\mathbf{p} = \begin{bmatrix} \dot{x}^* \\ y^* \end{bmatrix} \quad \text{and} \quad \bar{\mathbf{p}} = \begin{bmatrix} \dot{x}^* + \Delta\dot{x} \\ y^* + \Delta y \end{bmatrix} \tag{3.14}$$

where Δx and Δy are sufficiently small. Then

$$\mathbf{P}(\bar{\mathbf{p}}) = \begin{bmatrix} P_1(\dot{x}^* + \Delta\dot{x}, y^* + \Delta y) \\ P_2(\dot{x}^* + \Delta\dot{x}, y^* + \Delta y) \end{bmatrix}. \tag{3.15}$$

Using Taylor series expansion we obtain

$$\mathbf{P}(\bar{\mathbf{p}}) = \mathbf{p} + D\mathbf{P}(\mathbf{p}) \begin{bmatrix} \Delta\dot{x} \\ \Delta y \end{bmatrix} + O(\epsilon^2) \tag{3.16}$$

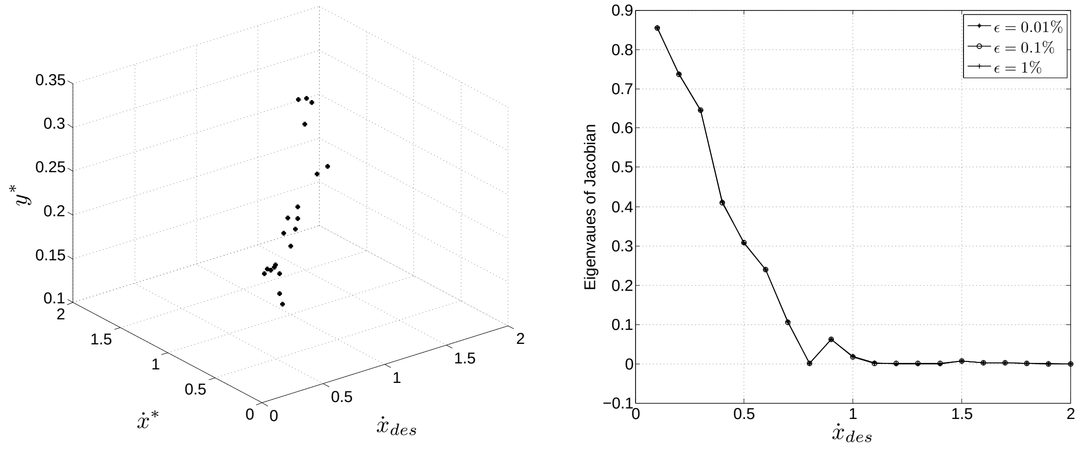
The Jacobian matrices $D\mathbf{P}(\mathbf{p})$ can be written as

$$D\mathbf{P}(\mathbf{p}) = \begin{bmatrix} \dfrac{\partial P_1(\dot{x}^*, y^*)}{\partial \dot{x}} & \dfrac{\partial P_1(\dot{x}^*, y^*)}{\partial y} \\ \dfrac{\partial P_2(\dot{x}^*, y^*)}{\partial \dot{x}} & \dfrac{\partial P_2(\dot{x}^*, y^*)}{\partial y} \end{bmatrix}. \tag{3.17}$$

To find the periodic gait cycles, we used the steady-state results obtained in Chp.3.2. We started with a set of initial guess close to the steady-state solution of constant velocity running results, and we tried to minimize the cost function

$$C = \|\mathbf{z}_{n+1} - \mathbf{P}(\mathbf{z}_n)\| \tag{3.18}$$

using Nelder-Mead optimization algorithm. The stopping criterion is chosen $C < 10^{-6}$. The fixed points of constant horizontal velocity running for 20 different speed goals are given in Fig. 3.16a.



(a) (b)

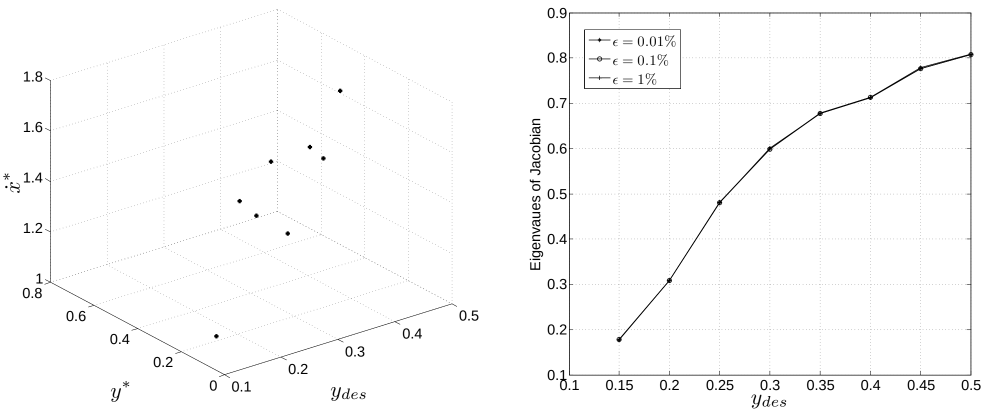
Figure 3.16: (a) The fixed points of desired velocity 0.1 to 2 m/s with the predefined controller parameters. (b) The eigenvalues of Jacobian of desired velocity range 0.1-2 m/s for different percentage perturbation amount of initial states.

The local stability of the periodic orbit is determined by the eigenvalues of the linearized Jacobian of Poincare map, $D\mathbf{P}(\mathbf{p})$, at fixed points. If all the eigenvalues (λ_1, λ_2) of $D\mathbf{P}(\mathbf{p})$ have magnitudes strictly less than unity, then the corresponding periodic gait is asymptotically stable [68, 69]. If the magnitude of one of the eigenvalues is bigger than one, the system is unstable. In our case, the Jacobian evaluated at fixed points has the unity eigenvalue that is independent from the control input. The trivial eigenvalue does not affect the stability, so we determined the asymptotic behavior by the second eigenvalue which lies inside the unit circle for all range of forward velocity and depends strongly on the controller used. The fixed points are attracting in the y direction and neutrally stable in the $\dot{x}$ direction. This means that a small perturbation of $\dot{x}$ from the equilibrium puts the system on a periodic orbit that stays near the equilibrium. But the velocity does not return to that equilibrium. Since the other eigenvalue is less than unity, we can conclude that our forward velocity controller makes the locomotion partially asymptotically stable for typical speed range $0.1 - 2.0$ m/s. [70]. The Fig. 3.16b also shows that when the velocity of the robot increases, the stability of locomotion on the y direction also increases.

We have also examined the stability of locomotion obtained using controller parameters (Table 3.2) that regulates the amplitude of the hopping motion by specifying the torque applied to the leg on each hop. We could not find any fixed points with this controller parameters, thus to investigate the effect of parameter changes on the stability we tried a new cost function that optimizes both initial apex states and controller parameters. The new cost function does not consider only the difference between desired and actual apex height at steady-state but also considers the forward speed of the robot.

Table 3.3: Parameter optimization results of height control using the cost function (3.19)

	Results of Optimization					
y_{des} (m)	$\dot{x}^*$ (m/s)	y^* (m)	θ_{des} (rad)	θ_{td} (deg)	K_p (Nm/rad)	K_d (Nm/(rad/s))
0.15	1.0427	0.1666	-0.5372	9.5466	32.0325	0.4095
0.20	1.5304	0.1915	-0.5580	10.9758	31.6396	0.4171
0.25	1.4132	0.2449	-0.5120	14.0358	31.4496	0.3795
0.30	1.5617	0.3087	-0.5829	17.6926	33.5734	0.4586
0.35	1.2182	0.3608	-0.6268	20.6767	32.2163	0.4626
0.40	1.5137	0.3904	-0.5094	22.3730	31.7641	0.3714
0.45	1.4013	0.4588	-0.5278	26.2922	30.7907	0.3630
0.50	1.6103	0.5121	-0.5345	29.3431	31.1045	0.3684



(a) (b)

Figure 3.17: (a) The fixed points of desired height 0.15 to 0.5 m with the controller parameters given in Table 3.3 (b) The second eigenvalue of Jacobian of desired height range 0.15-0.5 m for different percentage perturbation amount of initial states. The first eigenvalue is unity for all desired height.

To stabilize the height of the robot, we optimize the touchdown angle θ_{td} and the parameters of controller $[\theta_{des},\ K_p,\ K_d]$ during the stance phase. We define the new cost function as follows

$$\begin{aligned} \min J_3 &= ||\mathbf{z}_1 - \mathbf{P}(\mathbf{z}_0)|| \\ &\text{subject to (2.17) and (2.22),} \end{aligned} \tag{3.19}$$

where $z_{0,1} = [\dot{x}_{0,1}, y_{0,1}]$ are the apex states of the consecutive apex states and $\mathbf{P}$ is the apex return map defined in (3.12). To obtain fixed points and desired hopping height, we have minimized the cost function starting with an initial estimate of $\mathbf{z}_0 = [1\ m/s, y_{des}\ m]$ using the nonlinear optimization function *fminsearch* of MATLAB's Optimization Toolbox. We obtained fixed points for desired hopping heights starting with different guesses of initial states. Minimizing this cost function not only gives the controller parameters that exhibit smooth hopping motion with a desired height, but also provides optimum starting hopping conditions for our system. We analyzed the stability of the system around the fixed points obtained by optimization procedure. Similar to the velocity control part, one of the eigenvalues of controlled system is always unity. The variation of the second eigenvalue for different desired height is given in Fig. 3.17b. The eigenvalues of the system under the height control gets close to unity as the desired height increases.

3.6 Conclusion

In this chapter, we designed a controller to obtain desired running velocity and height of the robot. Although the structure of the controller is simple, we can achieve stable running performance by changing touchdown angle of the leg and controller reference signals. The constant reference signal of the PD controller can be replaced by a nonlinear function. The performance of the controller should be investigated for other leg design and system parameters.

The findings of this study indicates that although we have a drastically different leg morphology, to increase horizontal velocity of the robot, the touchdown angle should be increased as in shown in other biomedical studies [71, 72]. Future work includes the analysis of transition phase between two velocities and effects of initial states on the dynamics to obtain more accurate and robust controller.

CHAPTER 4

JUMPING CONTROL

4.1 Introduction

Although wheeled robots are capable of locomote on even ground with a considerable speeds, they can not perform well in the environment that has obstacles or other uneven structures. The legs give the robot maneuverability to climb body size objects, and increase the ability of negotiation in uneven terrain. When the legged robot encounters an obstacle higher than its own size, the jumping or hopping behavior are the one of the ways to overcome this situation using a minimal amount of energy.

Jumping is a complex behavior involving the coordination of whole body part movements to execute rapid locomotion. Biological systems use different control strategies to perform desired jumping tasks. There are several robots that are designed to perform jumping like motions. Some of them are focused on continuous hopping behavior [17, 18, 73, 74], while the others only consider stand by jumping [75, 76, 77]. The common property of these jumping robots is to have elastic leg structures.

In this section we describe an optimal control approach for exploring the jumping behavior of compliant legged monopod that can locomote in an unstructured environment. We introduce basic jumping controller structure that regulates the parameters of PD controller and the leg attack angle while minimizing the convex function of jumping height and distance.

4.2 Jumping Control Problem

The scenario we focused on in this section is running a number of steps with a constant velocity, then maximize the step length or the height of the robot for the last step. We used the weighted cost function (4.1) inspired from athletes performing long/high jump. These jumping motions include a preparation phase consisting of run up and adjustment of vertical height for an ideal take-off [78]. The initial velocity is important since at the take-off moment, the body momentum is converted to the upward and forward motion [79].

By taking into account physical limitations of the robot associated with motor speed limitations, maximum leg deflection and leg's kinematic reachability range, the optimal control problem defined as follows: find the control references $[\theta_{des}, \dot{\theta}_{des}]$ and touchdown angle θ_{td} that minimize the cost functional

$$\begin{aligned} \min J_3 = \; -\lambda \, L_{step} - (1-\lambda) \, y_{apex} \\ \text{subject to (2.17) and (2.22),} \end{aligned} \tag{4.1}$$

where L_{step} is the step length, h_{apex} is the vertical position of COM at the last apex (See Fig.3.1) and λ is a weighting coefficient. During steady-state running, we define the step length as the distance between the initial and final positions of the ground contact point of the leg on the flight phase as

$$L_{step} = \dot{x}_{avg} T_f + l_{lo} \cos \alpha_{lo} + l_{td} \cos \alpha_{td}, \tag{4.2}$$

where $\dot{x}_{avg}$ is the mean horizontal running speed in flight phase, T_f is the duration of flight phase, l_{lo}, l_{td} virtual leg length defined between hip point and ground contact point of the leg at the lift-off (*lo*) and touch-down (*td*) moments, respectively. Equations for virtual leg angles, α_{lo} and α_{td}, as a function of the leg angle θ are given as

$$\begin{aligned} \alpha_{lo} &= \frac{\pi}{2} - \theta_{lo} \\ \alpha_{td} &= \frac{\pi - \theta_{td}}{2}, \end{aligned} \tag{4.3}$$

where θ_{lo} and θ_{td} are the leg angle at the lift-off and touchdown moments, respectively.

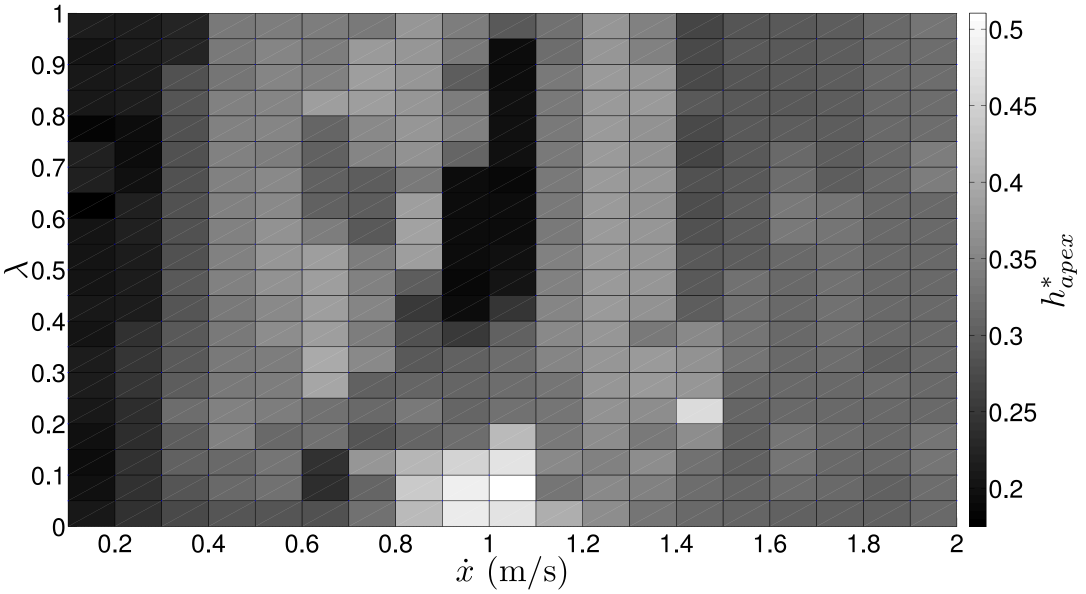


Figure 4.1: The apex height for the COM during flight found by using J_2 of (4.1) for different values of the optimization weight λ. Cell shades represent the optimal apex height (h^*_{apex}) at the last step.

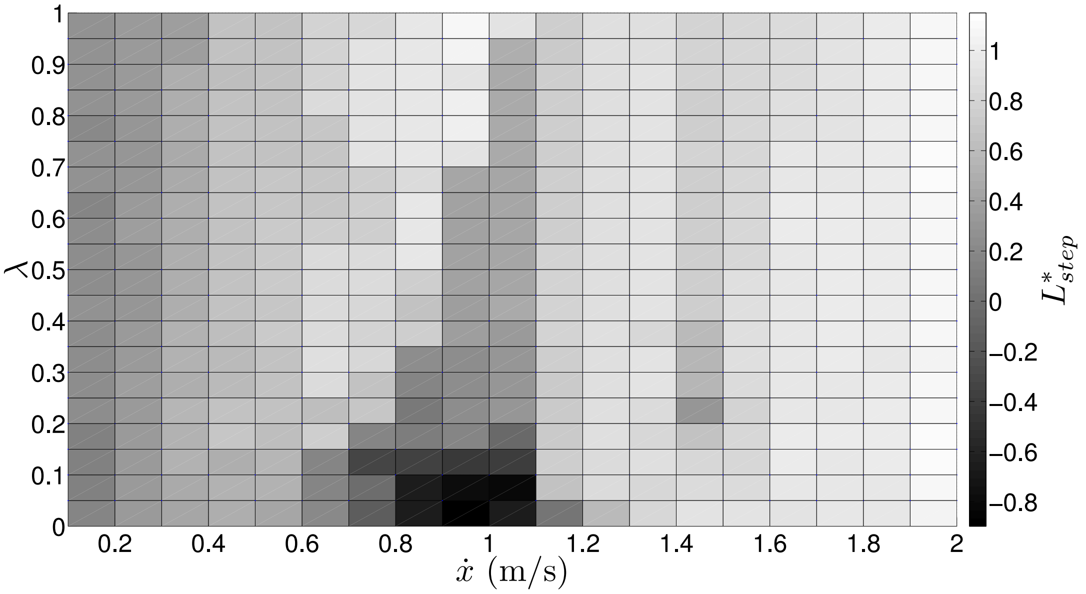
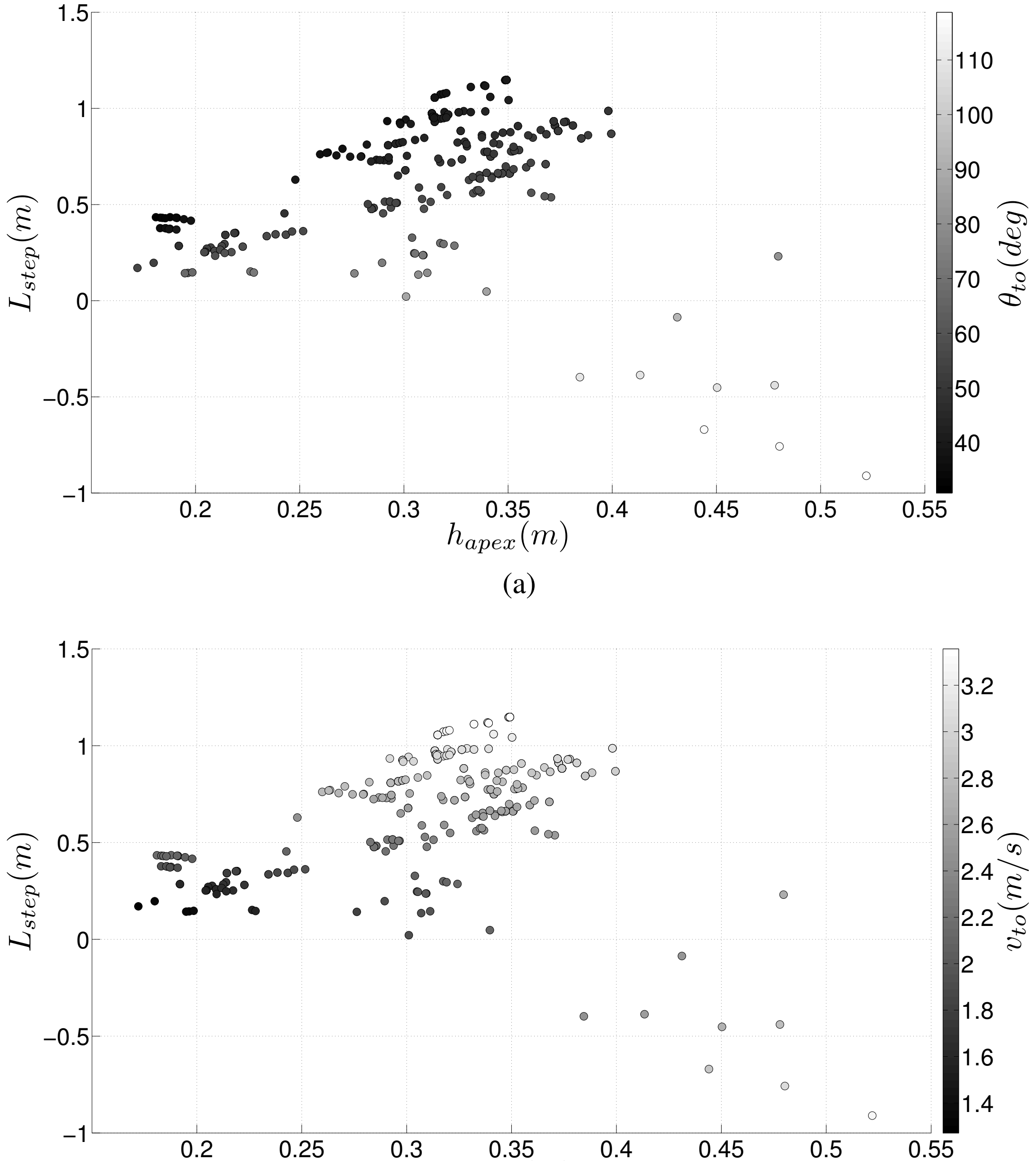


Figure 4.2: The step length for the COM during flight found by using J_2 of (4.1) for different values of the optimization weight λ. Cell shades represent the optimal step length (L^*_{step}) at the last step.



(a)

(b)

Figure 4.3: Jumping height h^*_{apex} and length L^*_{step} for different values of the optimization weight λ for the one-shot jumping problem. Shades for each point represents (a) the take-off angle of the body at the last step, (b) the take-off velocity of the body at the last step.

We give the results for the problem running with a constant speed and maximize the jumping height or distance at the last step. We used final states of the ten step constant velocity running as an initial condition for the jumping control problem. Fig. 4.1 and Fig. 4.2 give the maximum apex height and step length obtained using different weight value (λ) in (4.1). We changed the λ from 0 to 1 and we solved the optimal control problem (4.1) described in section 4.2 for different horizontal velocity using MATLAB optimization toolbox built-in function *fminsearch* starting at an initial estimate $[\theta_{des}, \theta_{td}] = [0.2rad, 30^o]$. The structure of the leg controller is identical to the one used in Section 3.2.

As seen from the figures, for the velocities smaller than 1 m/s, changing λ of (4.1) does not have significant effect on the jumping performance. When the initial horizontal velocity is bigger than 1 m/s, the small λ values increase the optimal apex height, as expected (see Fig. 4.1). Note that, the maximum apex height is obtained when the horizontal velocity is 2 m/s and λ is 0.1. This means that we need to run at a high velocity to get appropriate initial conditions for the high jump and to consider the jumping distance. As the initial horizontal velocity increases, the λ should be chosen close to 1 to get maximum jumping distance. A possible explanation for this might be that, when the horizontal velocity is between 1 m/s and 1.5 m/s, to maximize the jumping distance, the robot attempts to increase its vertical velocity at the last step to get a long flight duration using λ value about 0.6. As the horizontal velocity increases, the influence of flight duration on the jumping decreases.

Typically, the maximum horizontal distance in projectile motion is obtained when the take-off angle of the body is 45^o and the maximum vertical distance is obtained when the take-off angle is 90^o. To check the performance of our jumping controller, we looked at the take-off angle (Fig. 4.3a) and velocity (Fig. 4.3b) of the body at the last step for different horizontal velocity. Fig. 4.3a shows that the maximum jumping distances are achieved with a take-off angle about 50^o and the maximum apex heights are obtained when the take-off angle is about 90^o, which is close to ideal case. It can also be seen from Fig. 4.3b that the take-off velocity has a strong effect on jumping height and distance as expected.

4.4 Conclusion

This study underlined the importance of touchdown angle and controller parameters selection on the dynamics of half circular compliant legged monopod. The combination of running with constant speed and maximizing jumping height or distance at the last step problems can be extended to the design of auto-pilot for the real time rough terrain locomotion. Since our robot model does not include any sensor to detect its environment, the dimension of the obstacle and the map of the terrain must be given previously to control the selection of suitable footholds. The weights in (4.1) can be adjusted according to height and width of the obstacles. In the next chapter we will discuss the obstacle avoidance problem.

Future studies, which considers ellipsoid shape of the deflected leg will need to be performed to get more realistic results. We hope that our research will be valuable in understanding the dynamic behavior of the RHex type robots.

CHAPTER 5

OBSTACLE AVOIDANCE

5.1 Introduction

Although traditional ground robot locomotion techniques traverse obstacles of a similar order of magnitude as their size well, they could not cope with obstacles which are two times larger their size. The most effective way of traveling safely over different types of terrain with obstacles would be to fly over it [80]. Similar to flying motion, jumping over obstacles is an another way to traverse rough terrain especially for the legged locomotion.

The well known kinematic equations that used to analyze motions containing flight phase are

$$h_{apex} = \frac{(v\ sin\theta)^2}{2g} \tag{5.1}$$

and

$$r_{flight} = v^2 \frac{sin(2\theta)}{g} \tag{5.2}$$

where h_{apex} is the maximum height and r_{flight} is the maximum range of ballistic projectiles, v is the take-off velocity, θ is the take-off angle and g is the gravitational force. The greater the projectile angle, the smaller the horizontal velocity but the greater the time in the air. Thus if the take-off and landing height is same, it is clear from (5.2) that the maximum horizontal range is achieved when the take-off angle is 45. Likewise, from (5.1) in order to maximize the height of a jump, the ideal take-off angle is 90^o.

In this chapter, we have proposed control strategies to achieve rough terrain locomotion with a compliant half circular legged monopod. We have firstly focused on jumping over obstacle which has height and width. We have inspired from the performance of long and high jump athletes. [79] identified the principles that govern optimum speed and leg angle, for the take-off phase of high and long jumping. He concluded that a high jumper should run up at a moderate speed and set down the foot from take-off board with a leg angle about 45^o. A long jumper should run up as fast as possible. With a slow run up speeds, high jumpers control their take-off and obtain the complicated movements for the aerial phase. A fast run-up increases the horizontal component of velocity at take-off, but shortens the duration of ground contact and reduces the vertical impulse. The horizontal component of velocity at take-off is more important in long jumping than in high jumping, so a faster run-up is desirable in long jumping.

5.2 Problem Definition

The obstacle is placed at a desired distance from the robot's initial position, in our case it is 5 meters. Dimensions and roughness property of the obstacle must be given before solving the optimal control problem since the robot does not have any sensor units. We decompose the problem into smaller optimization tasks. First we find the optimum take-off angle, velocity and take-off distance (it is defined as the horizontal distance between the robot center of mass and obstacle when take-off) which consume minimum energy to jump over obstacle. If the air resistance is ignored, an object projected for horizontal distance continuous to travel horizontally until it is stopped by external force which is usually provided by ground. The longer an object is in the air with a horizontal velocity component, the longer it can continue to move in the horizontal direction. So, if the obstacle does not have height, i.e. it is a hole, we need to maximize the vertical component of velocity to increase flight time and the horizontal component of velocity to increase jumping distance. Thus, for all take-off velocity, we need to utilize take-off angle 45^o to maximize the jumping distance if the take-off and landing height are same [81, 82]. On the other hand, if the obstacle has height, we attempt to jump over obstacle without striking it. The main purpose is to consume minimum energy while jump over obstacle, i.e. use the path that has minimum take-off energy. One of the example jumping path with a minimum energy is given in Fig. 5.1.

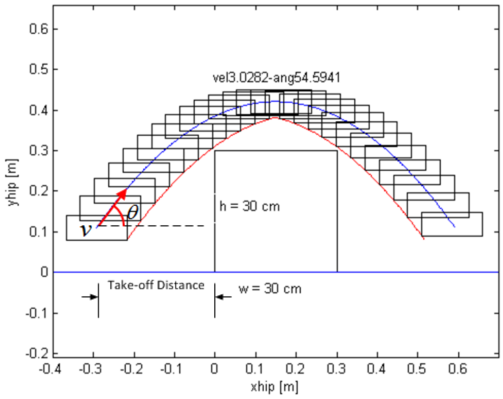
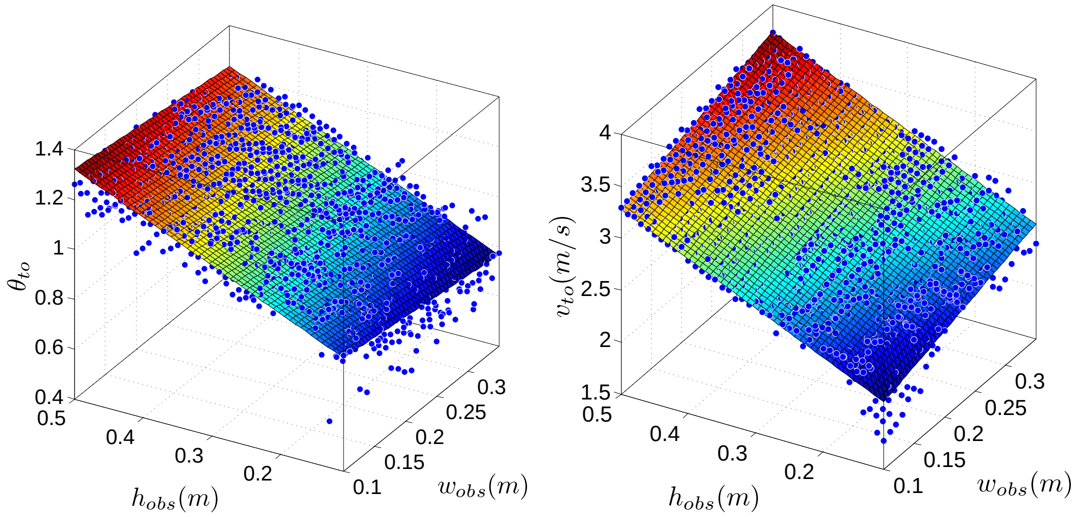


Figure 5.1: Jumping over obstacles whose height is 0.3 m and width is 0.3 m. The red line represents the clearance of the robot.



(a) (b)

Figure 5.2: The optimum take-off angle (a) and velocity (b) surfaces of the COM for a different obstacle size.

For different obstacle size (height is scaled between 0.1m and 0.5m and width is scaled between 0.1m-0.35m) we find the optimum take-off angle and velocity of the COM that result in a trajectory that is safe and consumes minimum energy. We fit a surface to the points defined by obstacle size, take-off velocity and angle of the COM.

The equation of the surfaces that are parametrized by obstacle size, take-off velocity and angle of the body can be written as

$$\theta_{to}(w_{obs}, h_{obs}) = 0.7557 - 0.3486\ w_{obs} + 1.204\ h_{obs} \tag{5.3}$$

$$v_{to}(w_{obs}, h_{obs}) = 1.64 + 1.85\ w_{obs} + 2.904\ h_{obs}. \tag{5.4}$$

Fig. 5.2 shows the corresponding surfaces. There is a significant correlation between obstacle size and take-off angle and velocity of the body [83]. The maximum take-off angle is about 75^o and it is required to jump over the obstacle whose $w_{obs} = 0.15\ m$ and $h_{obs} = 0.5\ m$. The maximum take-off velocity is about 3.7 m/s and it is required to jump over the obstacle whose $w_{obs} = 0.35\ m$ and $h_{obs} = 0.5\ m$.

5.2.1 Case I: Free Initial State

After obtaining approximate take-off angle, velocity and take-off distance for the desired jumping motion, we calculate the starting point of the triple jumping. We have used triple jumping techniques as in athletes to attain the greatest possible horizontal and vertical distance with a minimum energy. A triple jumper should run up with an average speeds during the last 5 m before the take-off, and to maintain as much of this speed as possible during steps of triple jump; the hop, the step and the jump [84].

According to distance between robot and obstacle we first run s steps with the fastest constant horizontal velocity which is close to horizontal take-off velocity. In this step we used the controller parameters and touch down angle obtained in Chapter 3. The constant speed step number is important since robot gains energy which facilitates the preparation phase of jumping. The constant running step number s is adjusted manually, but it should be considered as an optimization variable. After running s steps with a constant velocity, we go into 3-step preparation phase. This phase is important since we bring the robot to the desired initial states for the triple jumping. The robot must be able to orientate itself prior to jumping in this phase. The main problem is that we can not be able to find appropriate controller input to bring the system desired states all the time. We need to investigate the sensitivity of the triple jump controller part to the initial state variations. This analysis will be done in Section 5.4.

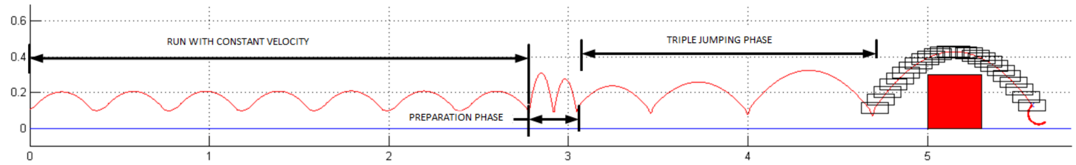


Figure 5.3: The phases of the free initial state problem

In the triple jump problem we optimize the touchdown angle and PD controller reference signal of each step. Note that, these control parameters are not same for all of the last three steps. These parameters are found so that the take-off velocity and angle are equal to desired one.

If we summarize the jumping procedure;

1. Put the obstacle to the desired position in the workspace
2. Give the height and width of the obstacle, also define the obstacle type, i.e. hole or hill
3. First find the safe takeoff velocity, angle and x position of the body to jump over the given obstacle. The objective function is the minimization of the consumed energy during jumping over the obstacle.
4. After obtaining desired takeoff velocity, angle and x position of the body to jump over the obstacle, calculate the distance between robot's initial position and desired take off position. This distance is then used to calculate the distance of run-up phase.
5. Run with a constant speed until the preparation phase start.
6. Find desired control input to bring system from constant speed running conditions to triple jumping starting condition.
7. Find control parameters and suitable touch down angle of triple jump phase.
8. Perform jumping.

5.2.2 Case II: Fixed Initial State

In the second case, we fixed the initial condition of the triple jumping phase and designed a controller which brings the system from any initial configuration to final desired configuration that is initial state of the triple jump. The desired final state is chosen so that if the robot starts its triple jumping motion from this state, it can jump over considerable amount of obstacles that have different size.

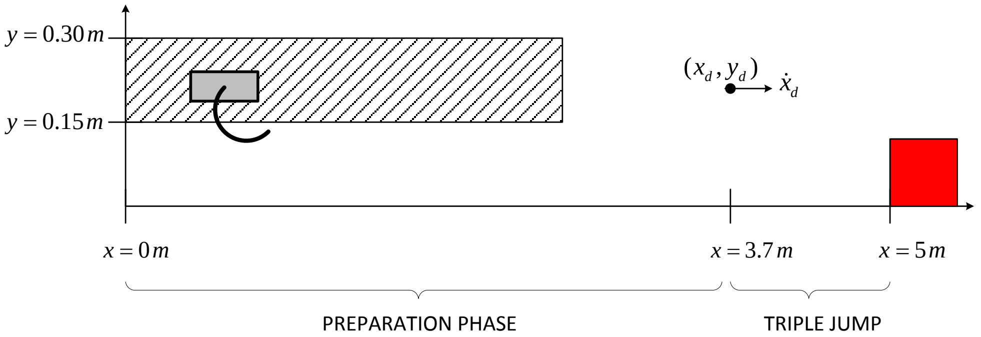


Figure 5.4: The preparation and triple jump phase of the second case. The obstacle is placed at $x = 5$ m. The shaded area represents the initial state space. The robot is left from any point in the shaded area with a zero velocity.

Similar to the first problem, the obstacle is placed to $x = 5$ m. We choose two desired initial state for the triple jumping phase; $\mathbf{z}_1 = [x, \dot{x}, y, \dot{y}] = [3.7\,m, 1.5\,m/s, 0.2\,m, 0\,m/s]$ and $\mathbf{z}_2 = [x, \dot{x}, y, \dot{y}] = [4\ m, 1\ m/s, 0.3\ m, 0\ m/s]$, that are the state of the robot at the last apex of the preparation phase. The initial states of preparation phase are defined as

$$\begin{aligned} 0\,m \le x \le 2\,m, \quad \dot{x} = 0 \\ 0.15\,m \le y \le 0.3\,m, \quad \dot{y} = 0. \end{aligned} \tag{5.5}$$

Fig. 5.4 presents the workspace of the fixed initial state problem. The robot is left from the any point in the shaded area with a zero horizontal and vertical velocities. The optimal control algorithm tries to find leg angle and controller parameters that bring the system near to desired final state.

The control input of the system is similar to defined in Section 3. Different from the previous velocity control problem we also optimize the K_p and K_d gains of PD controller. The control problem can be defined as follows

$$\begin{aligned} \min J_4 = \; & \|z - z_d\| \\ \text{subject to } & (2.17) \text{ and } (2.22), \end{aligned} \tag{5.6}$$

where z is the final apex state of the preparation phase and z_d is the desired initial state of the triple jump.

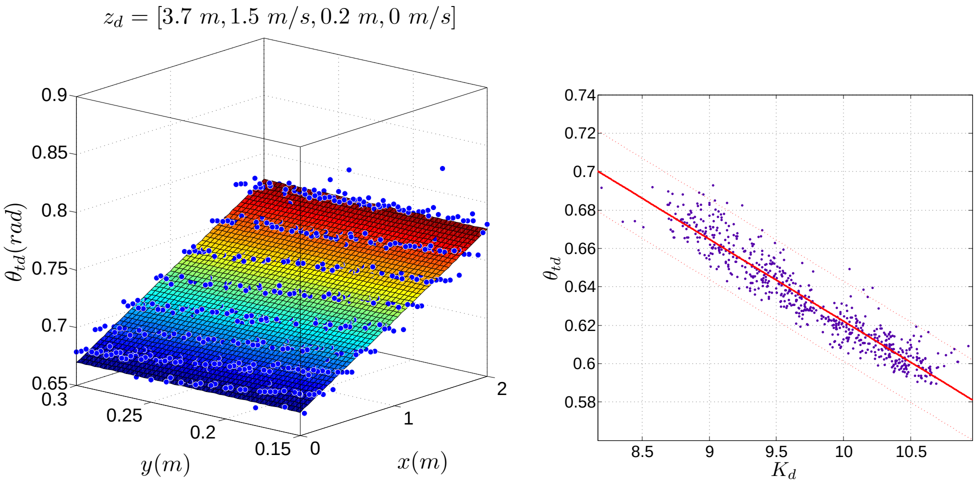
5.3 Results

5.3.1 Results of Case I: Free Initial State

We control the step length and jumping height by controlling horizontal running speed and running height. Different from the jumping over obstacle with height, if we have hole obstacle we need only maximize the step length while minimizing the energy consumed during jumping. The vertical velocity at the take off adjusts the duration of the flight phase and it is somehow important to jump over hole. But vertical take-off velocity is not as important as in jumping over hill type obstacles as in long jump running athletes who increase their horizontal velocity and use optimum vertical velocity to get desired flight time.

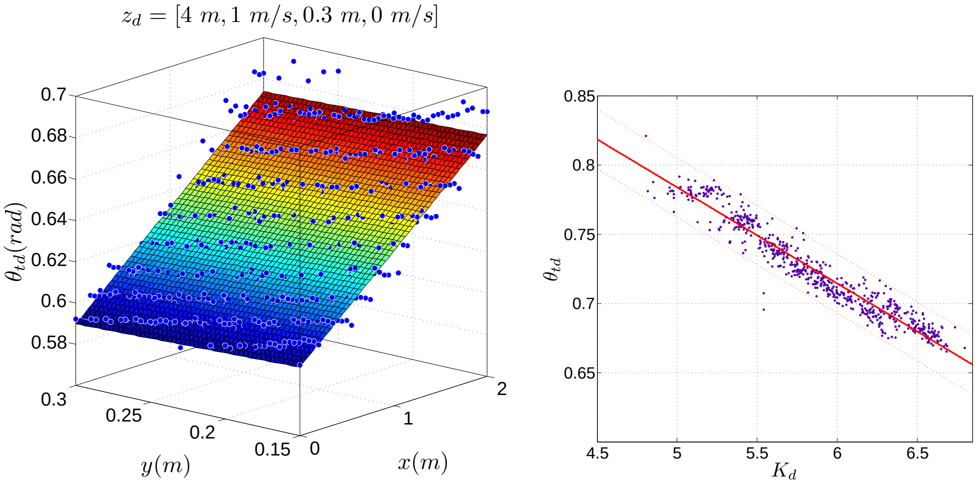
Fig. 5.8 - Fig. 5.10 show the jumping performance of the robot for the different sized obstacles. As seen from the Fig. 5.7a, if the size of the obstacle is small compared to robot size, the robot does not change its initial velocity and touch down angle at the last steps. As the height of the obstacle increases, the robot compresses its leg to gain energy which is required at the take-off. At the last three steps jump energy should be stored in the leg, ready for instantaneous release when take-off. The preparation phase is more remarkable when the height and width of the obstacle increased at the same time. The step number of run-up phase is chosen so that the robot reach the desired take off phase with a desired take-off velocity. If the obstacle width is larger than its height, the robot attempt to increase its horizontal velocity in last 5 steps as in the long jumpers. The step number of preparation phase can be adjusted to get more

accurate take-off performance.



(a) (b)

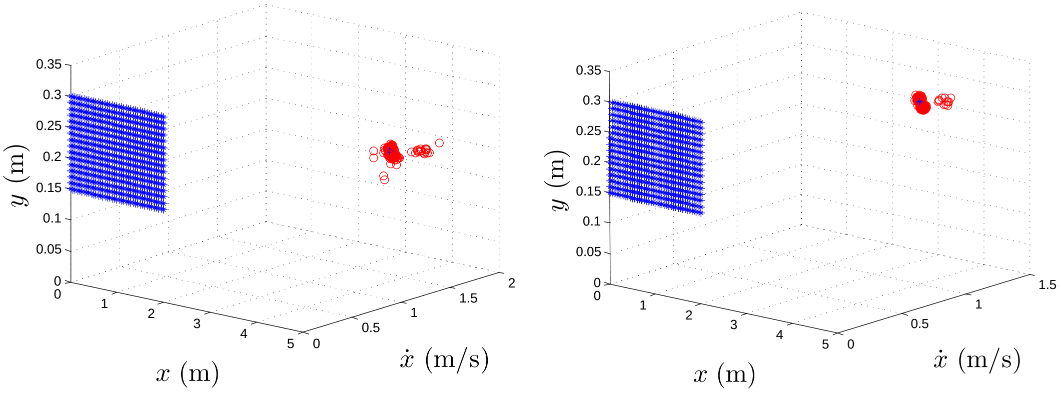
Figure 5.5: (a) The optimum leg attack angle vs. initial position of the COM. (b) θ_{td} vs. K_d



(a) (b)

Figure 5.6: (a) The optimum leg attack angle vs. initial position of the COM. (b) θ_{td} vs. K_d

5.3.2 Results of Case II: Fixed Initial State

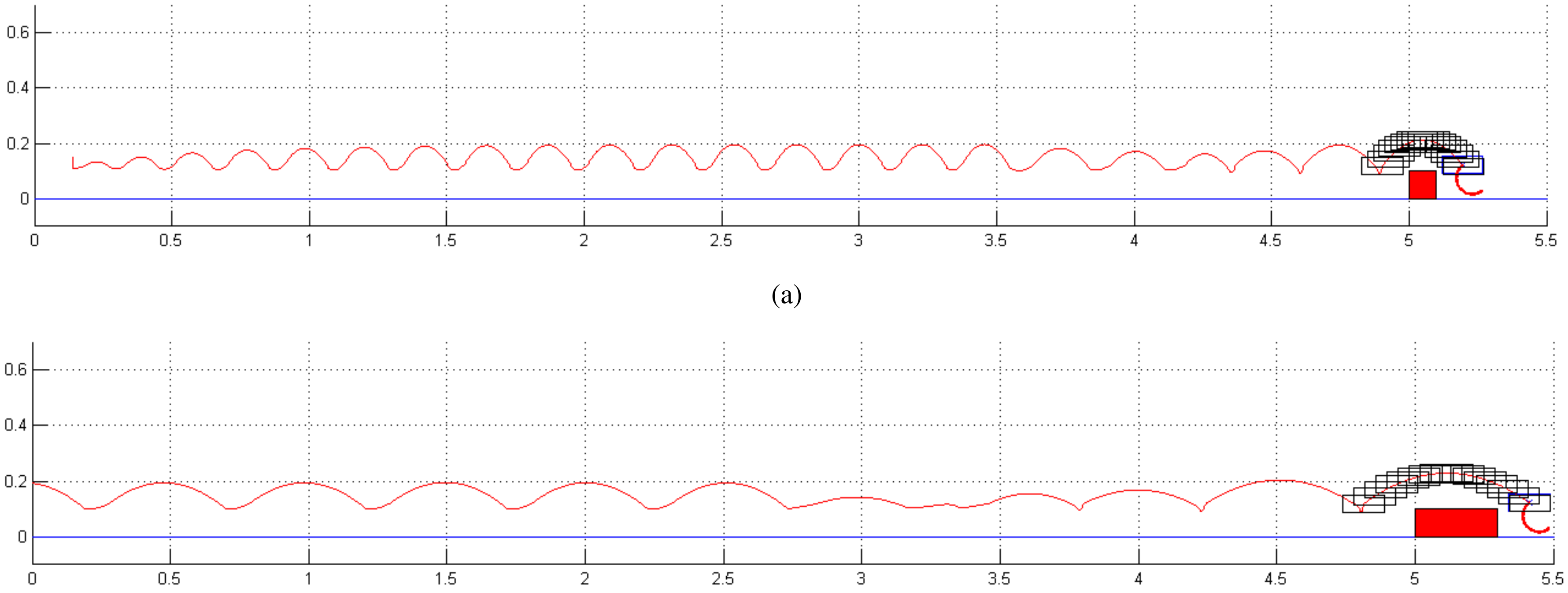


(a) (b)

Figure 5.7: The desired final state (a)$z_d = [3.7\ m, 1.5\ m/s, 0.2\ m, 0\ m/s]$ (b)$z_d = [4\ m, 1\ m/s, 0.3\ m, 0\ m/s]$. The blue rectangular part is the initial state space.

Fig. 5.7a and Fig. 5.7b shows the resulting final state of the robot in preparation phase. The rectangular area represents initial state space where the robot starts its locomotion from rest. According to distance between initial and final COM position of the robot the optimal touchdown angle and other controller parameters are changed. We fit a surface to understand the relation between the desired touchdown angle of the robot and initial position of COM. The surface equation for z_{d1} and z_{d2} can be written as

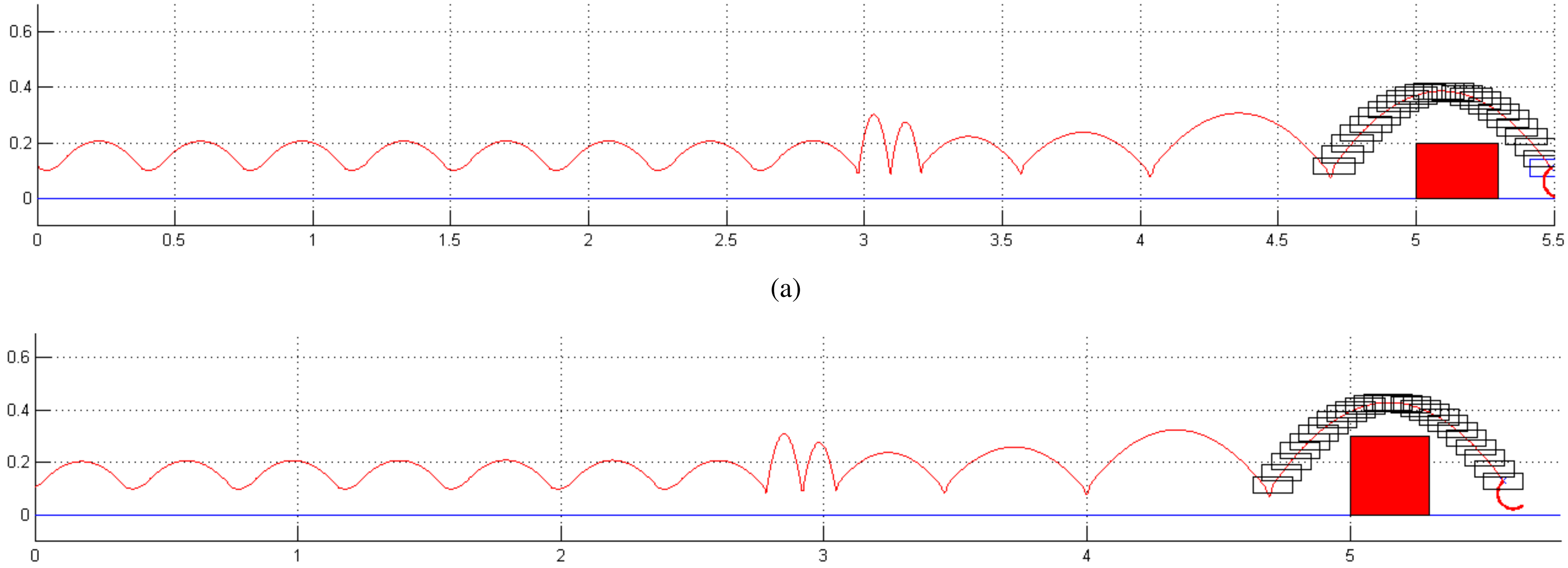
$$\begin{aligned} \theta_{td}(x, y) &= 0.6697 + 0.05369x + 0.001192y && \text{for } z_{d1} \qquad (5.7) \\ \theta_{td}(x, y) &= 0.596 + 0.04197x - 0.02079y && \text{for } z_{d2} \end{aligned}$$



(a)

(b)

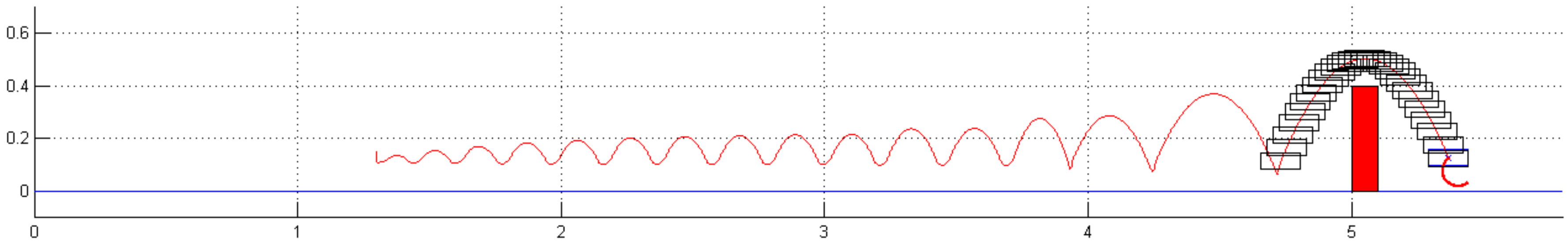
Figure 5.8: Optimum trajectory for the obstacle with a (a) height = 0.1 m, width = 0.1 (b) height = 0.1 m, width = 0.3 m



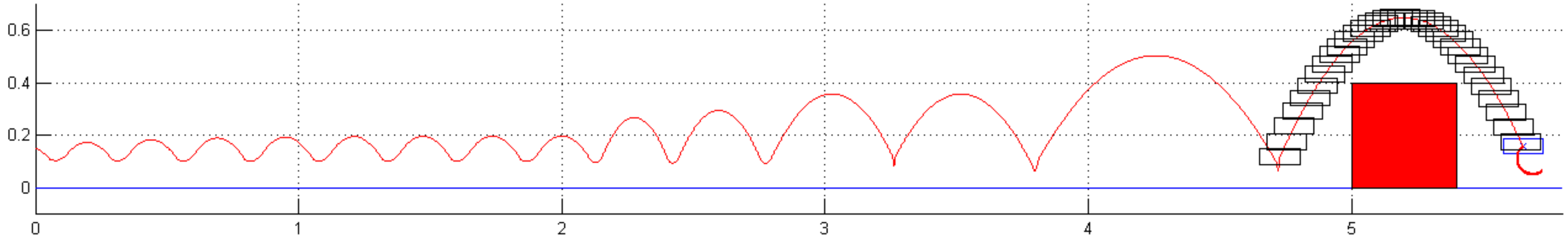
(a)

(b)

Figure 5.9: Optimum trajectory for the obstacle with a (a) height = 0.2 m, width = 0.3 (b) height = 0.3 m, width = 0.3 m



(a)



(b)

Figure 5.10: Optimum trajectory for the obstacle with a (a) height = 0.4 m, width = 0.1 (b) height = 0.4 m, width = 0.4 m

5.4 Sensitivity Analysis

Analyzing the effects of disturbances on the output of the controlled system is necessary before implementing the controller on the real robot platform. The sensitivity analysis approach is based upon linearizing the system around a nominal trajectory rather than around an equilibrium point and try to interpret the variations around that trajectory.

There are different types of disturbances that should be considered when designing controller structure. First and foremost, the small and large disturbances in the initial conditions of the system may result in considerable impact on the output especially if the controller is open loop. The stability and accuracy of the controller should be analyzed when the trajectory of the dynamic system does not start from the desired initial point, but from its neighborhood. Secondly, the parameters of the system such as physical properties of robot platform and environment can be time variant and most of time they can be determined with a certain accuracy.

In this section, we test the robustness of our triple jumping controller in the presence of initial state and controller parameter disturbances and we try to develop a metric for those variations and relate it to the performance of triple jumping controller.

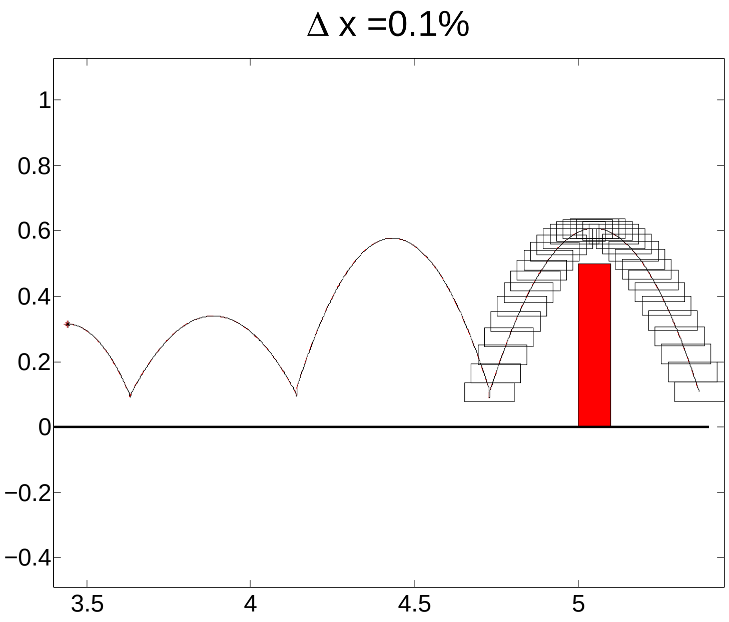
5.4.1 Sensitivity to Initial Conditions

To determine the sensitivity of the triple jump trajectory in the case of time invariant initial state perturbations, we choose the trajectory of robot that jumps over obstacle whose height is 0.5 (m) and width is 0.1 (m). We have added disturbance to the initial states of the robot with an increment $\pm 0.1\%, 1\%, 10\%$ and we calculated the deviation of final states from the desired nominal values. The results are given in Table 5.1. Although we only give the results of jumping over one obstacle, the relation of the initial and final states are similar for all jumping conditions. The disturbance of initial x state affects only the final x position of the robot, while $\dot{x}$ changes both x_f and $\dot{x}_f$. The final error at x and $\dot{x}$ are linearly dependent on the initial perturbation amount. The final height of the robot is independent from the perturbation at the initial x and $\dot{x}$. On the other hand, the disturbance on the initial height changes all of the final states and the final variations are not linear functions of perturbation amount. When the initial height is less than its nominal value, the amount of the energy that brings the robot to desired trajectory should be increased.

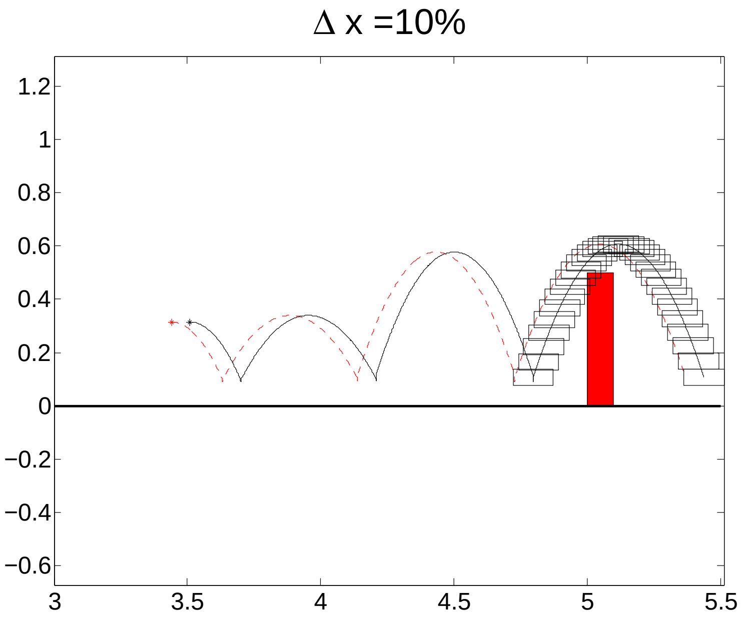
Fig. 5.11 - Fig. 5.13 show the perturbed and optimal unperturbed trajectories of the robot with respect to positive extreme disturbances 0.1% and 10% given in Table 5.1. It can be seen from the figures that the small errors in the initial states do not affect the jumping performance so much. If the offset in initial condition gets larger, the final trajectory lies off and the robot does not perform the desired jumping task.

Table 5.1: Sensitivity of triple jump trajectory to a perturbation in the initial states of the robot. The first column gives the percentage disturbance in the corresponding states and other three columns present the percentage of deviation from the desired final apex states. This results are obtained by jumping over obstacle with a $h_{obs} = 0.5$ (m) and $w_{obs} = 0.1$ (m).

	Error %		
$\Delta x_i\%$	Δx_f (m)	$\Delta \dot{x}_f$ (m/s)	Δy_f (m)
0.1	0.0136	0	0
1	0.1363	0	0
10	1.3629	0	0
-10	-1.3629	0	0
-1	-0.1363	0	0
-0.1	-0.0136	0	0
$\Delta \dot{x}_i\%$	Δx_f (m)	$\Delta \dot{x}_f$ (m/s)	Δy_f (m)
0.1	0.0283	0.0902	0
1	0.2829	0.9018	0
10	2.8294	9.0181	0
-10	-2.8294	-9.0181	0
-1	-0.2829	-0.9018	0
-0.1	-0.0283	-0.0902	0
$\Delta y_i\%$	Δx_f (m)	$\Delta \dot{x}_f$ (m/s)	Δy_f (m)
0.1	0.0125	-0.0390	0.4497
1	0.1359	-0.0209	0.4741
10	1.5853	-0.0065	6.0270
-10	-1.7261	-0.0444	-6.1449
-1	-0.1693	-0.0329	-0.3808
-0.1	-0.0427	-0.0328	-0.1493

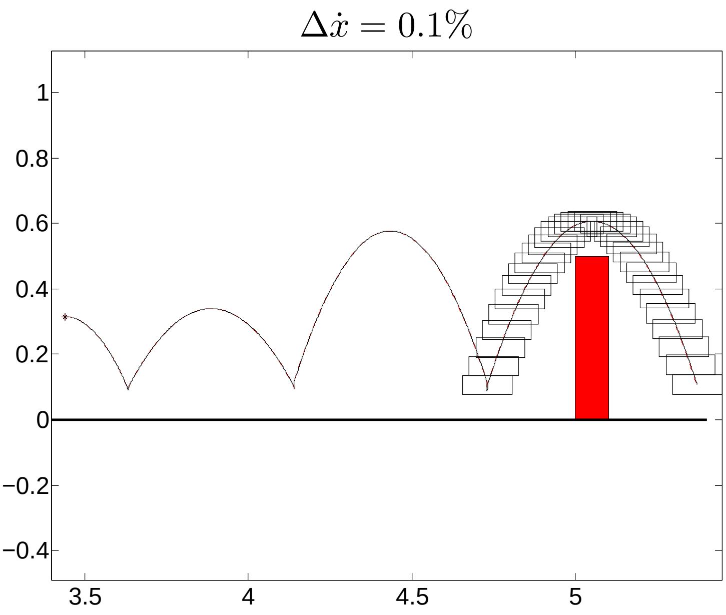


(a)

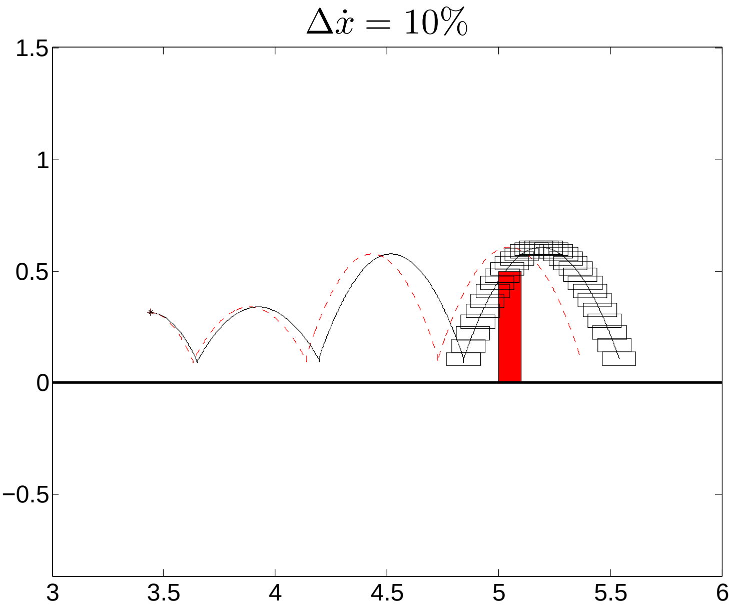


(b)

Figure 5.11: The undisturbed (solid line) and disturbed (dashed line) COM trajectory of the robot that jumps over obastacle with $h_{obs} = 0.5$(m) and $w_{obs} = 0.1$(m) using the triple jump controller. The disturbance amount equals to (a) 0.1% (b) 10% of the initial x position.

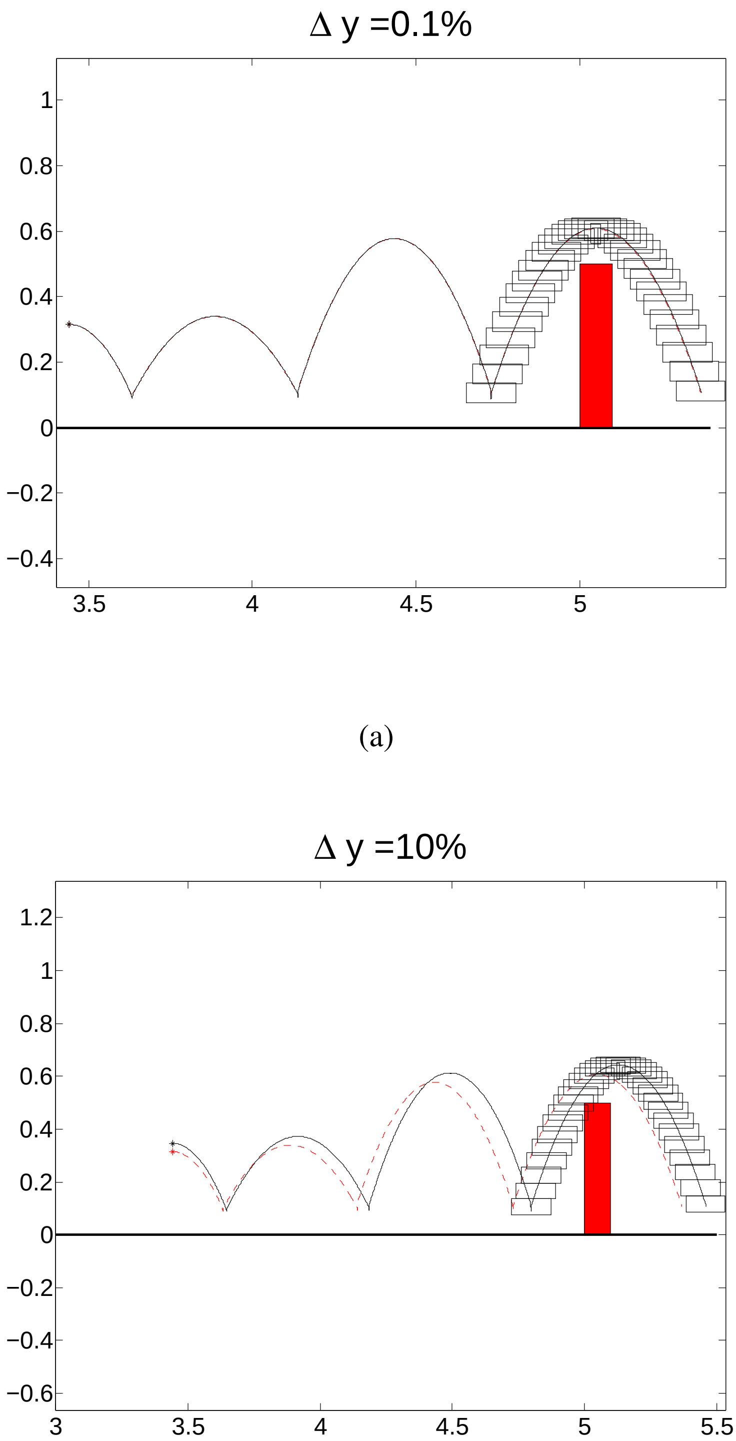


(a)



(b)

Figure 5.12: The undisturbed (solid line) and disturbed (dashed line) COM trajectory of the robot that jumps over obastacle with $h_{obs} = 0.5$(m) and $w_{obs} = 0.1$(m) using the triple jump controller. The disturbance amount equals to (a) 0.1% (b) 10% of the initial $\dot{x}$ position.



(a)

(b)

Figure 5.13: The undisturbed (solid line) and disturbed (dashed line) COM trajectory of the robot that jumps over obastacle with $h_{obs} = 0.5$(m) and $w_{obs} = 0.1$(m) using the triple jump controller. The disturbance amount equals to (a) 0.1% (b) 10% of the initial y position.

Table 5.2: The effects of controller parameter variations in the final desired apex states of the triple jump trajectory.

Parameter Change: 10 %						Final Error		
θ_{des1}	θ_{td1}	θ_{des2}	θ_{td2}	θ_{des3}	θ_{td3}	x_f	$\dot{x}_f$	y_f
↑	-	-	-	-	-	↓	↓	↓
-	↑	-	-	-	-	↑	-	↑
-	-	↑	-	-	-	-	-	-
-	-	-	↑	-	-	-	-	-
-	-	-	-	↑	-	↓	↓	↑
-	-	-	-	-	↑	↑	↑	↑

5.4.2 Sensitivity to Controller Parameters

Trajectory sensitivity analysis provides valuable insights into understanding the influence of parameter variations on the dynamic behavior of the robot. Properties which are not obvious from the actual system response are often evident in the sensitivities. In this section similar to previous part, we have analyzed the sensitivity of trajectory to control parameters. We are interested in how uncertainty in control parameters propagate to uncertainty in the final state of the robot.

We have increased the nominal values of controller parameters and we have performed triple jump for different size of obstacles. We see that the most significant variations are obtained when the reference angle of the first and last step are changed. When we increase the reference angle of first or last step, the final x position and velocity decreases. The increase in the touchdown angle of first or last step increases the final x position of the robot as expected. The final forward velocity raises with the positive change of touchdown angle at the last step. Final height of the robot only decreases with an increase in the reference angle of the first step. The impact of the parameter variations in the second step of triple jump is almost negligible. We summarized the results in Table 5.2. When we decrease the control parameters all direction of deviations at the final state reverse.

5.5 Triple Jump Controller based on Sensitivity Analysis

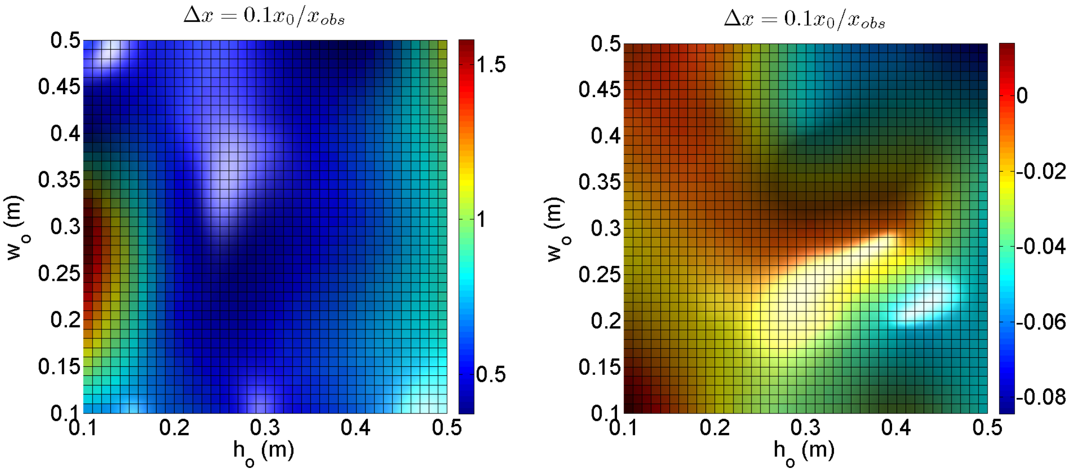
In the two previous sections we have investigated the sensitivity of triple jump trajectory to variations in the initial conditions and controller parameters of the robot. In this section, we try to nullify the effect of disturbances in the initial conditions of the robot by suitable choice of control parameters θ_{des1} and θ_{des2}. For this purpose, we define a new cost function that optimizes the scale of the controller parameters, c_1 and c_2, by minimizing the trajectory error at the last step of triple jump. The new scaled controller parameters are $\theta_{des1new} = \theta_{des1}(1 + c_1)$ and $\theta_{des3new} = \theta_{des3}(1 + c_2)$. The other controller parameters $[\theta_{td1}, \theta_{des2}, \theta_{td2}, \theta_{td3}]$ are fixed to their nominal values that are obtained in Section 5.3.1. The cost function defined as

$$\begin{aligned} \min J &= \|\mathbf{z}_{3disturbed} - \mathbf{z}_{3nominal}\| \\ &\text{subject to (2.17) and (2.22),} \end{aligned} \tag{5.8}$$

where $\mathbf{z}_3 = [x_f, \dot{x}_f, y_f]$ is the COM trajectory of the robot at the last step.

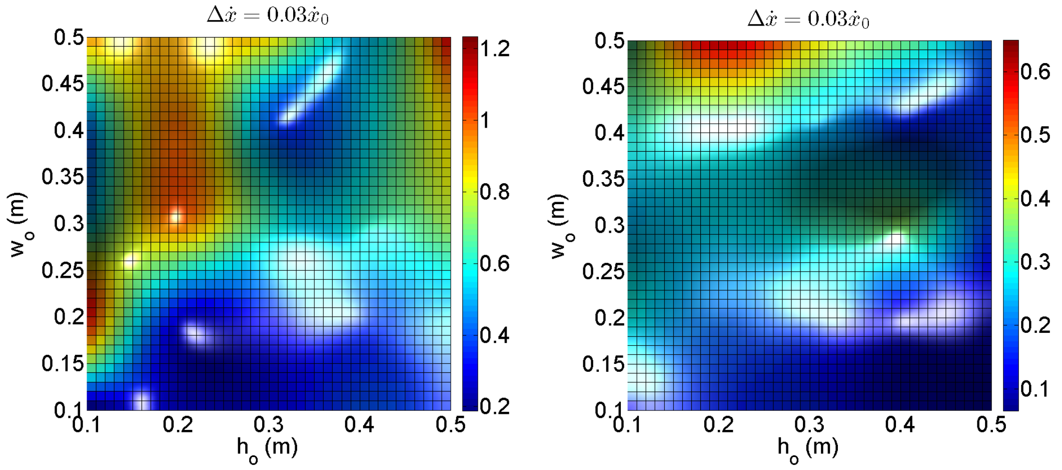
Fig. 5.14 - Fig. 5.16 show the parameter optimization results of the triple jump controller for various obstacle sizes. First, we investigated the effect of varying the initial x position of the robot. We add 10% error normalized with x position of the obstacle to nominal value of x and we minimize the cost function given in (5.8). The rest of the controller parameters are set to their nominal values. The optimum scale of the controller parameters that eliminate normalized 10% error in the initial x position of the robot for different obstacle sizes h_0 and w_0 scaled between $0.1 - 0.5$ are given in Fig. 5.14. Since adding positive disturbance to initial x position only affects the final x position of the robot, we obtained increased θ_{des1} for all obstacle size. There is a small drop in θ_{des3} to satisfy optimum jumping height at the last step. Secondly, we disturbed the initial $\dot{x}$, which results in increase of final x position and forward velocity of the robot. Both θ_{des1} and θ_{des3} increases to decrease the final state error. As seen from the Fig. 5.15, when the height of the robot is less than its width to not collide with an obstacle the forward velocity and x position must be adjusted carefully. Finally, we added positive 10% error to initial height of the robot and we obtained controller scales given in Fig. 5.16. We observed a large increment in θ_{des1}

and a small decline in θ_{des3} to decrease the final height error. Starting triple jump with a height higher than its nominal value adds the system extra energy especially when the height of the robot is less than its width. The energy must be removed to obtain energy efficient jumping motion. As seen from the Fig. 5.16, we obtained higher scales factor for the obstacles whose height is less than 0.3 m. Our results show that having tuned the parameters of controller based on the sensitivity analysis, we can compensate errors in initial states by tuning only two parameters of the controller not all of them.



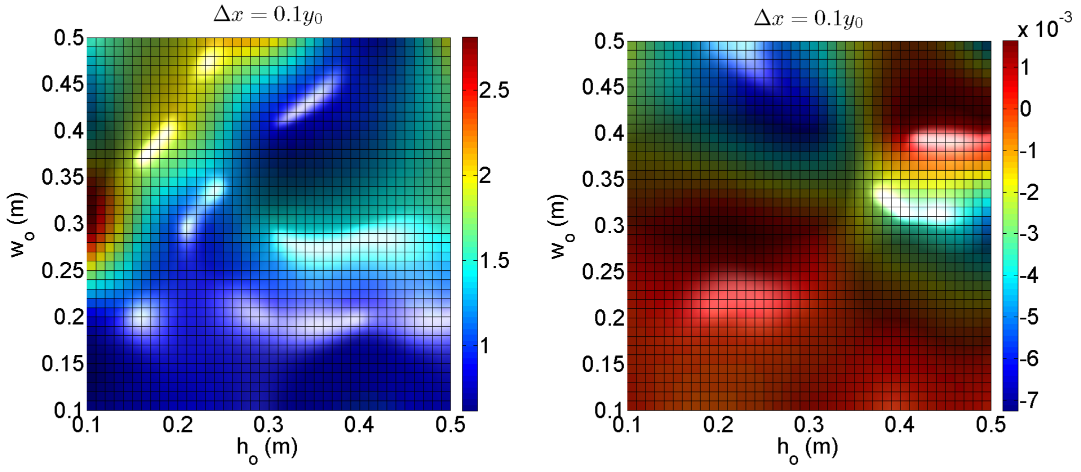
(a) scale of θ^*_{des1} (b) scale of θ^*_{des3}

Figure 5.14: The optimum scale of the controller parameters that eliminate normalized 10% error in the initial x position of the robot for different obstacle size h_0 and w_0 scaled between $0.1 - 0.5$. The color scale represents the scale of (a) θ^*_{des1} (b) θ^*_{des3}.



(a) scale of θ^{*}_{des1} (b) scale of θ^{*}_{des3}

Figure 5.15: The optimum scale of the controller parameters that eliminate 3% error in the initial forward velocity of the robot for different obstacle size h_0 and w_0 scaled between $0.1 - 0.5$. The color scale represents the scale of (a) θ^{*}_{des1} (b) θ^{*}_{des3}.



(a) scale of θ^{*}_{des1} (b) scale of θ^{*}_{des3}

Figure 5.16: The optimum scale of the controller parameters that eliminate 10% error in the initial height of the robot for different obstacle size h_0 and w_0 scaled between $0.1 - 0.5$. The color scale represents the scale of (a) θ^{*}_{des1} (b) θ^{*}_{des3}.

5.6 Triple Jump Controller with a Safety Margin

The sensitivity analysis of the robot states that we require very accurate initial conditions and controller parameters for the reliable jumping motion without colliding with obstacles. Although we only consider uncertainty in the states and control parameters of the robot, there are other uncertainties such as position of obstacle, random disturbances at the states of robot while in motion, the sensor noises etc.

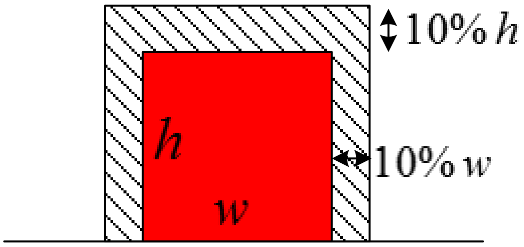


Figure 5.17: The new representation of obstacle with a 10% safety margin.

In this section, to avoid collisions with obstacle, we modify the trajectory of robot taking into account the positioning accuracy. We define an area, called safety margin, around an obstacle that the robot can jump over it safely. The new representation of obstacle with a safety margin is given in Fig. 5.17. Then we re-optimized the reference signals and touchdown leg angles as well as initial states $[x, \dot{x}, y]$ of triple jump by considering the new size of obstacle.

Fig. 5.18-5.20 show the COM trajectory of the robot that jumps over different size of obstacles without and with 10% safety margin added to obstacle size. The size of the safety margin is selected intuitively, it needs careful study and analysis of the way the source of uncertainties and their effects at the jumping performance.

Fig. 5.21-5.24 compare the variations of controller parameters and initial conditions of the triple jump phase when we add safety margin. Fig. 5.21 and Fig. 5.22 present the variation of optimum reference angle of controller at each step. θ_{des1} is almost similar for both case, where θ_{des2} incerases when the height of the robot is less than 0.3 m and θ_{des3} decreases when the width of the robot is less than 0.3 m.

Fig. 5.23-5.24 present the changes of optimum touchdown angles of robot at each step. θ_{td1} increases when the height of the robot is greater than 0.3 m, θ_{td2} increases when the both height and width of the robot is greater than 0.3 m and θ_{td3} increases when the height of the robot is greater than its width.

Fig. 5.25-5.26 show the changes of optimum states of robot at the beginning of triple jump. The initial velocity of the robot increases for all obstacle size as expected. When obstacle height is bigger than its width the initial height increases. We obtained decline in x_{runup} since the robot needs more distance to store energy before jump over obstacle.

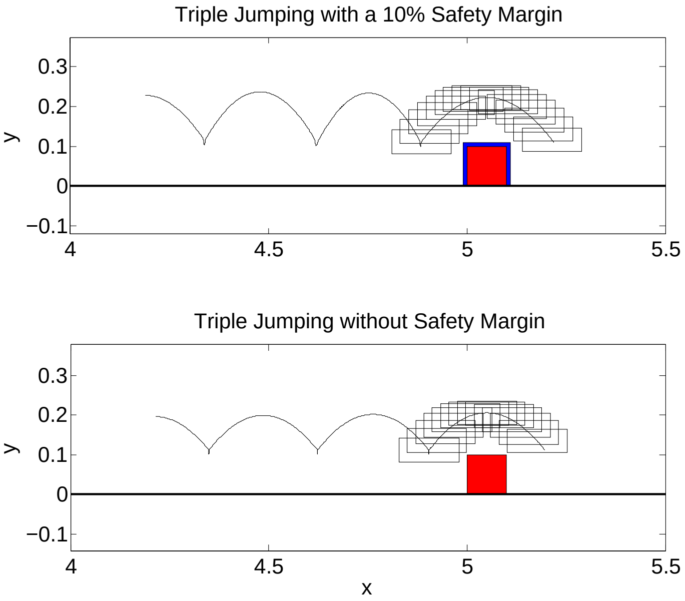


Figure 5.18: The COM trajectory of the robot that jumps over obstacle (h_{obs} = 0.1 (m) and w_{obs} = 0.1) (top) with a 10% safety margin (bottom) without the safety margin.

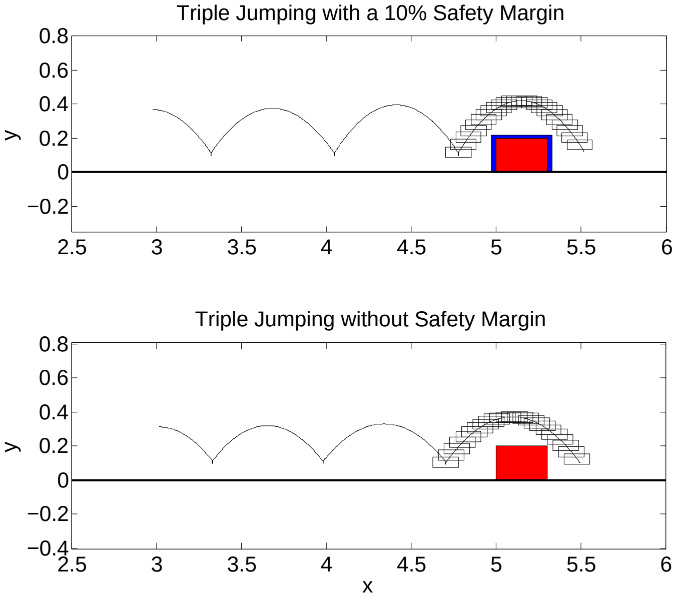


Figure 5.19: The COM trajectory of the robot that jumps over obstacle (h_{obs} = 0.2 (m) and w_{obs} = 0.3) (top) with a 10% safety margin (bottom) without the safety margin.

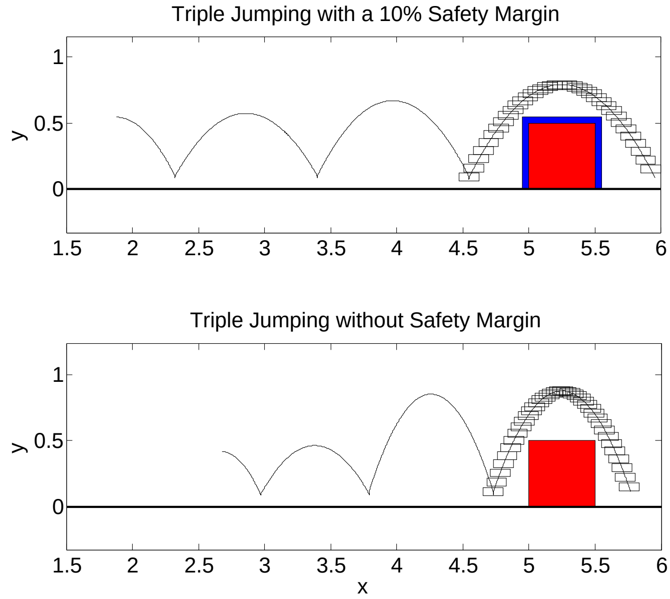


Figure 5.20: The COM trajectory of the robot that jumps over obstacle (h_{obs} = 0.5 (m) and w_{obs} = 0.5) (top) with a 10% safety margin (bottom) without the safety margin.

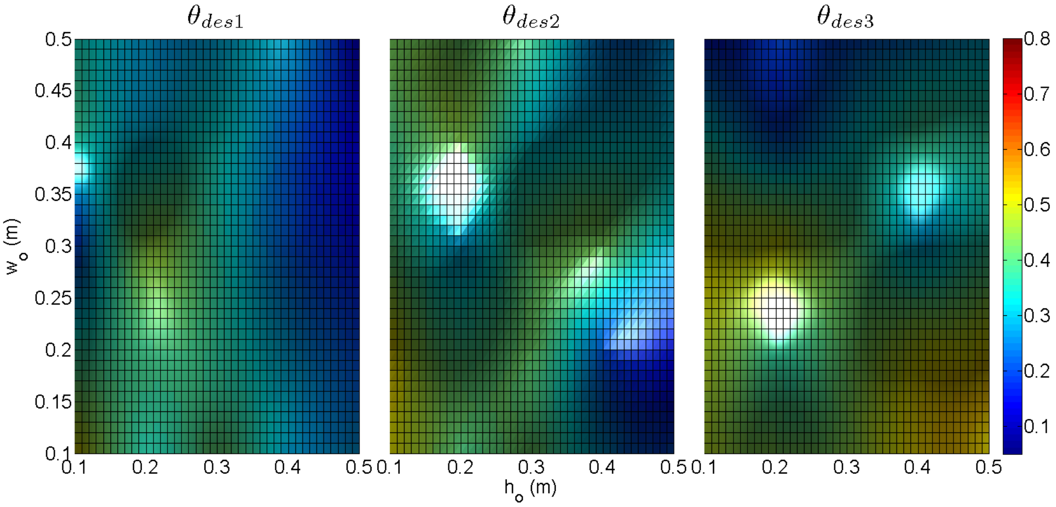


Figure 5.21: The optimum reference angle signals of triple jump controller without safety margin added to obstacle size. The height and width of the obstacle are scaled between 0.1-0.5 (m). The color scale represents the optimum reference angle signal of controller at the (left) first (middle) second (right) third step of triple jump.

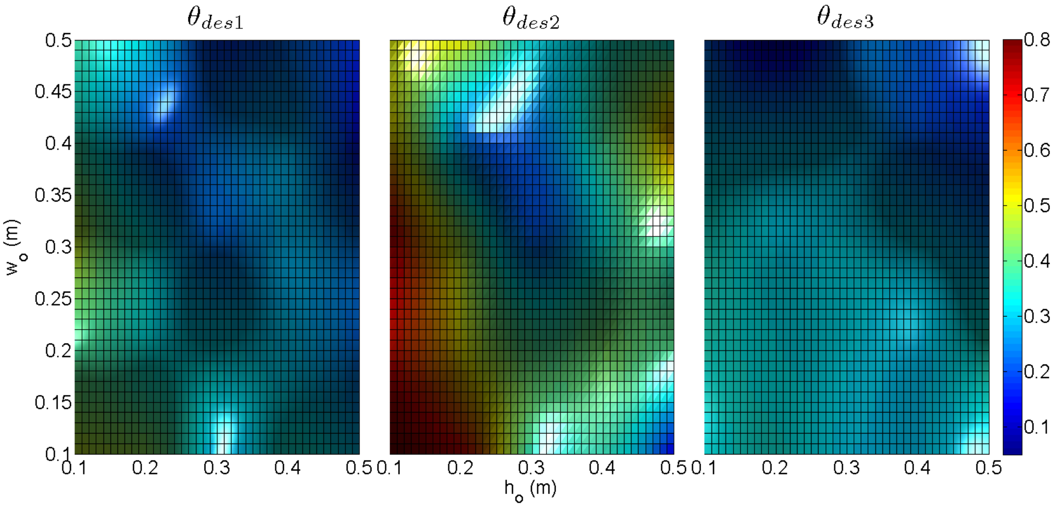


Figure 5.22: The optimum reference angle signals of triple jump controller with a 10% safety margin added to obstacle size. The height and width of the obstacle are scaled between 0.1-0.5 (m). The color scale represents the optimum reference angle signal of controller at the (left) first (middle) second (right) third step of triple jump.

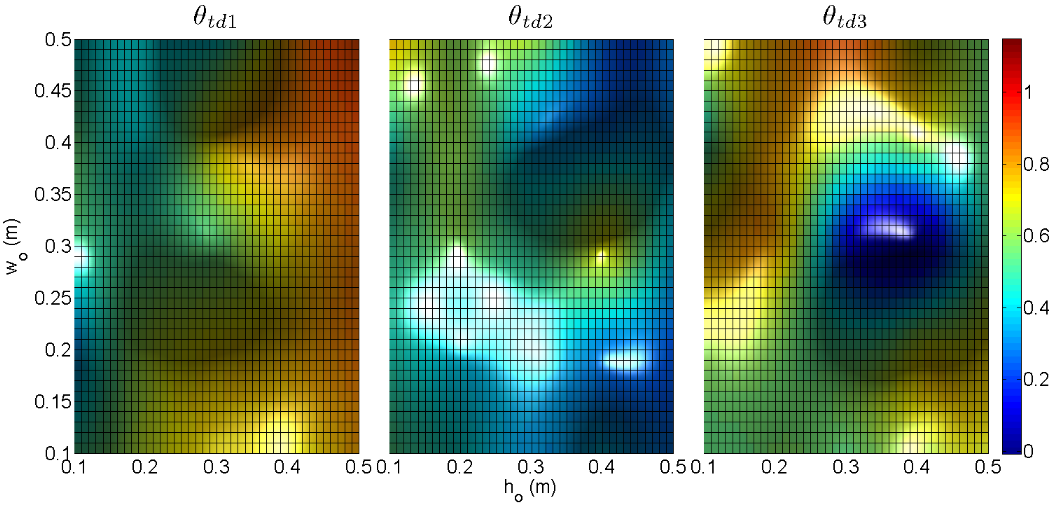


Figure 5.23: The optimum touchdown angle of triple jump controller without safety margin added to obstacle size. The height and width of the obstacle are scaled between 0.1-0.5 (m). The color scale represents the optimum touchdown angle of controller at the (left) first (middle) second (right) third step of triple jump.

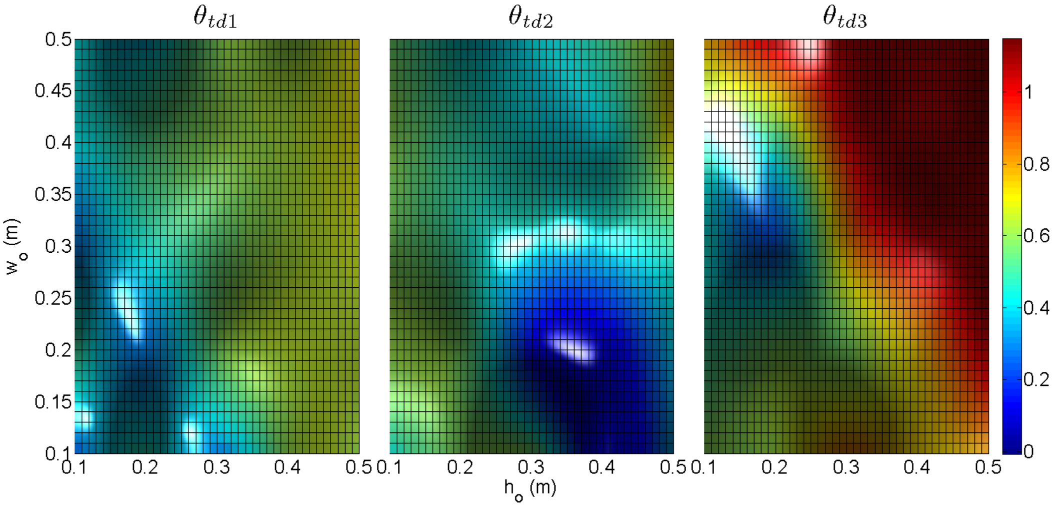


Figure 5.24: The optimum touchdown angle of triple jump controller with a 10% safety margin added to obstacle size. The height and width of the obstacle are scaled between 0.1-0.5 (m). The color scale represents the optimum touchdown angle of controller at the (left) first (middle) second (right) third step of triple jump.

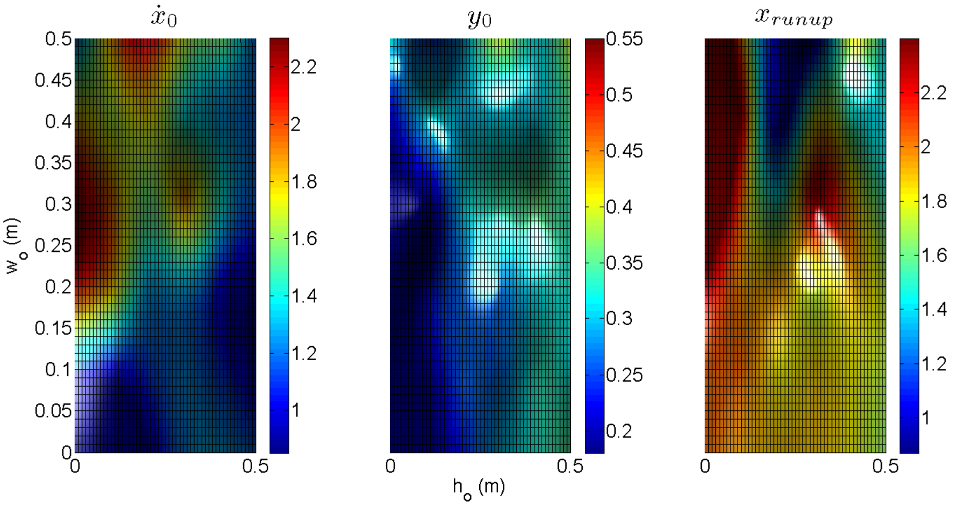


Figure 5.25: The optimum starting conditions of triple jump phase without safety margin added to obstacle size. The height and width of the obstacle are scaled between 0.1-0.5 (m). The color scale represents the optimum (left) initial forward velocity (middle) initial height (right) the initial starting x position of triple jump phase.

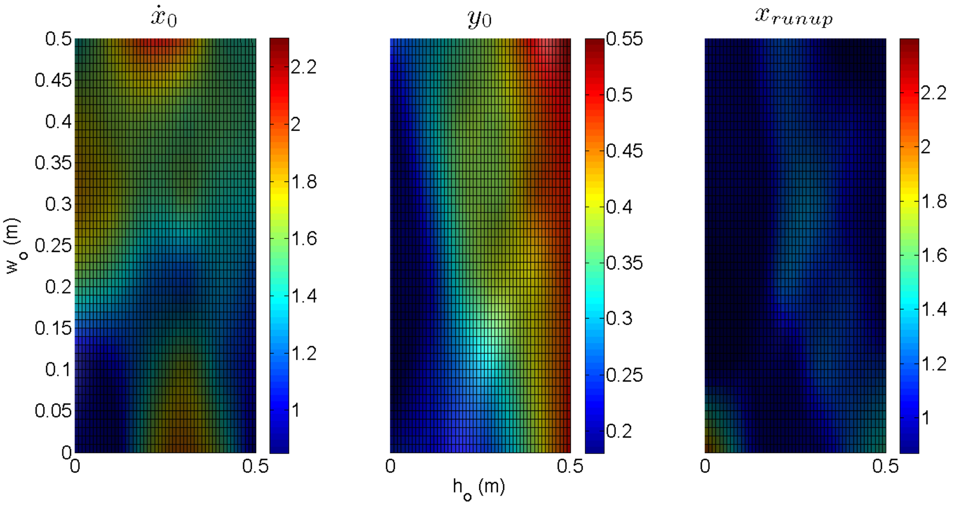


Figure 5.26: The optimum starting conditions of triple jump phase with a 10% safety margin added to obstacle size. The height and width of the obstacle are scaled between 0.1-0.5 (m). The color scale represents the optimum (left) initial forward velocity (middle) initial height (right) the initial starting x position of triple jump phase.

5.7 Conclusion

We have presented the optimal control approach for the rough terrain locomotion of half circular legged robot. The main purpose is that while using suitable algorithms for rough terrain locomotion that found the path of the robot to navigate through the world, we need to consider actions the robot should perform to realize the path safely. We have analyzed the sensitivity of trajectory to variations in initial states and control parameters. We scaled the control parameters based on the response of the system to perturbation in the initial states. We have also evaluated the performance of the controller in the presence of uncertainty in the localization of the obstacle. We have defined a controller that takes into account the safety margin around obstacle. The controllability of this nonlinear system is the other issue that should be considered to find the suitable controller input to bring the system from any initial starting point to desired jumping conditions. Simulation of the system in the environment with an irregularly distributed different size multiple obstacles should be realized.

CHAPTER 6

HALF CIRCULAR LEG TEMPLATE FOR A HEXAPEDAL RUNNING

6.1 Introduction

Template is a low dimensional model that exhibits a targeted behavior of high dimensional biological and robotic systems that have different morphological structures [53]. It reduces modeling complexity that arises from the highly nonlinear characteristics of human and animals biological parts such as muscles, tendons, neurons, joints etc. The high degree of freedom, nonlinear motion of different animals can be captured with a low dimensional templates, and this gives insight to understand the locomotion of complex mechanisms found in nature.

A template based control of the robotic systems is a well-known approach that drives the system to mimic the template dynamics [85]. A spring-mass template model is used to describe accurately the rapid locomotion of human and other animals that have different leg and skeleton morphology [55, 56, 72]. Until know SLIP template has been applied to characterize the locomotion behavior of different types of robot platforms [17, 18, 54, 86, 87]. It was experimentally shown that the dynamics of bipedal SLIP can be anchored a high degree of freedom hexapod robot, called RHex [4, 32]

In this chapter, the dynamic motion of the hexapod robot is analyzed using two different modeling approaches. The goal is to find optimum control inputs and physical parameters of the leg such that the dynamics of templates accurately mimic the COM motion of the hexapod robot. We used the time trajectory of a full body configuration and leg states of hexapod, called Edubot (see Fig. 6.1) a small form-factor modular RHex variant that is used for education and research [1, 3, 88, 89]. We propose a low dimensional circular compliant beam (CCB) template described in Chapter 2 as a candidate template model for the hexapedal running and we compare the predictions of this model with those of widely used Spring Loaded Inverted Pendulum (SLIP) model using the experimental data of Edubot. The data collection and extraction procedures are same as described in [90].

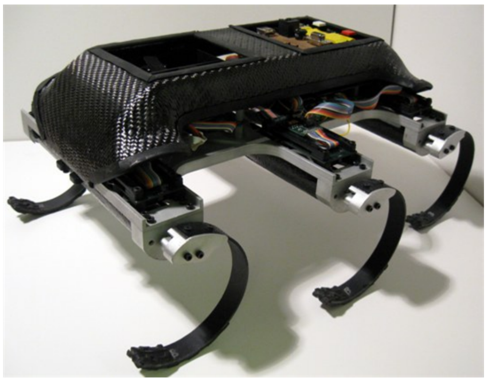


Figure 6.1: Edubot [3]

6.2 Templates for Legged Locomotion

6.2.1 Spring Loaded Inverted Pendulum (SLIP) Template

The SLIP template [56] is a simplified model which is used for representing the COM dynamics of the robot. The six legs of the robot represented by a massless virtual leg with a spring constant, k and rest length, l_0. The virtual leg is attached the point mass body from the COM. The SLIP has a two phase: stance and flight. During the stance phase leg spring applies a force on the body. Figure (6.2) shows the COM trajectory and forces of the SLIP in stance phase. We assume that the virtual toe position does not change during the stance phase. The dynamic equations of the torque controlled SLIP in stance can be written as follows

$$\begin{aligned} \ddot{l} &= -g\ cos\theta - \frac{k}{m}(l - l_0) + l\ \dot{\theta}^2 \\ \ddot{\theta} &= \frac{\tau}{m\ l^2} + \frac{g\ sin\theta}{l} - \frac{2\ \dot{l}\ \dot{\theta}}{l}. \end{aligned} \tag{6.1}$$

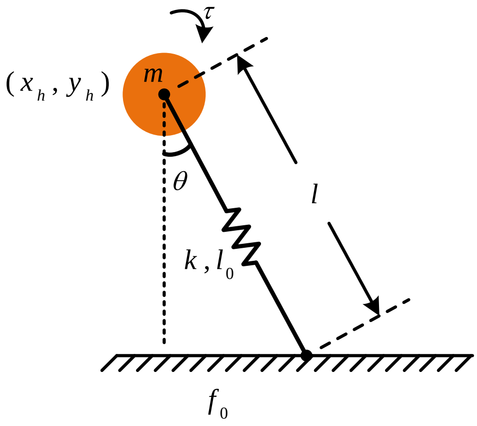


Figure 6.2: The SLIP Model

where θ is the angle of leg defined between gravity vector and leg, l is the instantaneous leg length, k is the spring stiffness constant, m is the mass of body and l_0 is the initial leg length. During stance the leg is free to rotate around its toe. In flight phase, the point body mass is acted upon by only body mass and exhibits a projectile motion.

6.2.2 Half Circular, Compliant Beam (CCB) Template

The details of the this model is given in Chapter 2. We rewrite the dynamic equations as

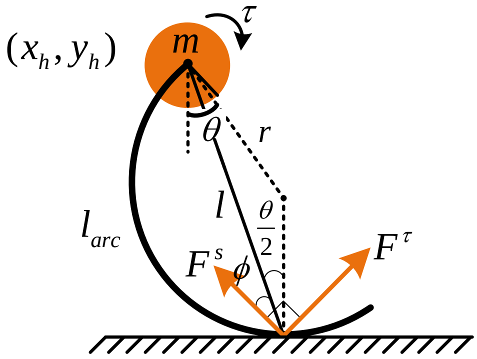


Figure 6.3: The CCB Model

$$\begin{pmatrix} m_b & 0 & 0 & -1 & 0 & 0 \\ 0 & m_b & 0 & 0 & -1 & 0 \\ 0 & 0 & I_{leg} & 0 & 0 & -l_{arc}cos\phi \\ 0 & 0 & 0 & 1 & 0 & -cos\frac{\theta}{2}+\phi \\ 0 & 0 & 0 & 0 & 1 & -sin\frac{\theta}{2}+\phi \\ 1 & 0 & C_1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \\ F^b_x \\ F^b_y \\ F^\tau \end{pmatrix} = \begin{pmatrix} 0 \\ -m_b g \\ -T - F^s l_{arc} sin\phi \\ -F^s_x \\ F^s_y \\ C_2 \end{pmatrix}, \tag{6.2}$$

where

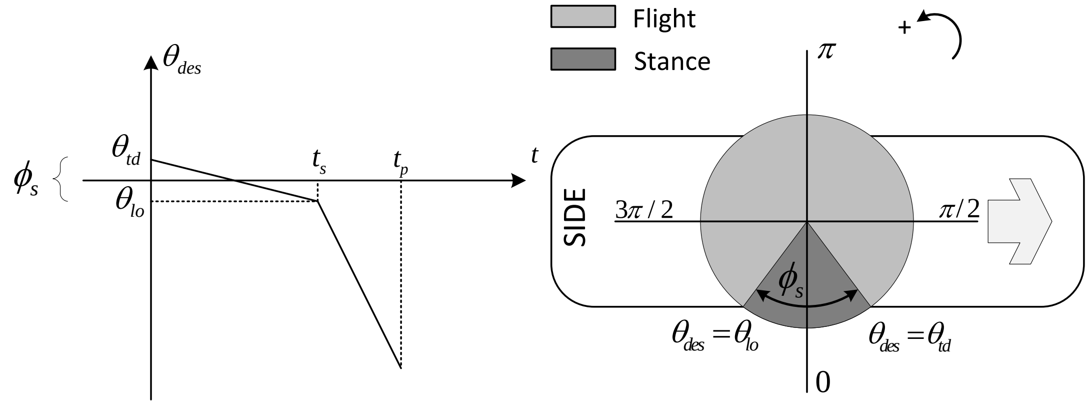
$$\begin{aligned} C_1 &:= \frac{l_{arc}}{(\pi-\theta)^2}(sin\theta + cos\theta(\pi-\theta)) \\ C_2 &:= -2l_{arc}\frac{\dot{\theta}^2}{(\pi-\theta)^3}(sin\theta + cos\theta(\pi-\theta) - sin(\theta)(\pi-\theta)^2). \end{aligned}$$

6.3 Experimental Set-up and Data Collection

All the experiments in this study were performed by the Edubot robot platform [3]. EduBot, (Fig. 6.1), has one rigid body with six actively driven compliant legs. Each leg of the Edubot is enforced by six local PD controllers and the open loop control of the slow and fast swing phases of each tripod performed by a controller, called *Buehler Clock* [4], that is parametrized by four variables: $[t_c, t_s, \phi_s, \phi_o]$. We used equivalent set of control parameters like defined in [91], t_p (duration of whole period), t_s (stance time), θ_{td} (leg touchdown angle) and θ_{lo} (leg liftoff angle). Fig. 6.4 presents the controller structure where ϕ_s is the rotation angle of the leg during slow phase. The period of one leg starts with a ground contact. During the stance phase the leg sweeps $\phi_s = (\theta_{lo} - \theta_{td})$ rad, and its completes the whole phase by rotating $(2\pi - \phi_s)$ rad. during the flight phase.

Table 6.1: Edubot parameters

Variable	Definition	Value	Unit
E	young modulus	9.8	GPa
r	radius of leg	0.059	m



(a) (b)

Figure 6.4: (a)The Buehler two speed profile is defined by 4 parameters (b) Side view of the robot with a desired leg angle of the fligth and stance phase $[\theta_{td}, \theta_{lo}, t_s, t_p]$ [4].

$$\theta_{des} = \begin{cases} \frac{\theta_{lo}-\theta_{td}}{t_s} t + \theta_{td} & 0 \le t < t_s \\ \frac{\theta_{td}-\theta_{lo}-2\pi}{t_p-t_s}(t - t_s) + \theta_{lo} & t_s \le t < t_p \end{cases} \tag{6.3}$$

$$\dot{\theta}_{des} = \begin{cases} \frac{\theta_{lo}-\theta_{td}}{t_s} & 0 \le t < t_s \\ \frac{\theta_{td}-\theta_{lo}-2\pi}{t_p-t_s} & t_s \le t < t_p \end{cases}$$

In this experiment Vicon camera setup is used to capture center of mass (COM) position of the robot during locomotion. This setup includes six cameras with a sampling rate 120 Hz. [90]. In order to start analysis we first obtained accurate COM position data with a deviation of less than 5 mm. Due to the inherent inaccuracies in the Vicon calibration process, the physical ground plane is not aligned with the X-Y plane of the Vicon coordinate frame. We first calibrated the camera system logging the positions of numerous reflective balls which were spread out on the Vicon system's capture area, and then we fitted a surface to these points, which defines the shape and orientation of the ground. The Cartesian body position data ${}^{v}b := [{}^{v}b_x \; {}^{v}b_y \; {}^{v}b_z] \in R^3$ in the Vicon coordinate system, V, are then transformed to World Coordinate System, W, by a homogeneous transformation $h_v^w : V \rightarrow W$.

The control parameters of the robot adjusted carefully to acquire smooth and continuous tripod gaits. We run the robot from one side to other side of the experiment area whose length is approximately three meters. Vicon cameras track and record the position of reflective markers attached to robot and user-defined function fits a rectangular to these markers. Then orientation and geometric center, which is almost near to location of COM, of the robot are calculated with a sampling rate 120 Hz. Another logger on the robot receipts the information about the angular position and velocity of each leg. We synchronized Vicon and leg logger by fitting a spline to both sets of data and evaluating the fitted spline at points in a common time vector with a frequency of 1000 Hz.

Since both model live in the sagittal plane we will use the sagittal plane projection of the COM tracjectory obtained from Vicon data. When the robot runs between two points, the horizontal trajectory of COM may not be parallel to one of the axis of Vicon system. In this case, we need to find a straight line which defines the direction of motion. A straight line can be obtained from least squares polynomial fitting of horizontal trajectory. Figure (6.5) shows the projection of COM trajectory to the sagittal plane.

The x, y, z points in the Vicon coordinate frame,V can be represented in the sagittal Plane, S with the below matrix,

$$\begin{bmatrix} b_x & b_y & 0 & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \mathbf{RP} \tag{6.4}$$

where **b** is the position vector of COM in sagittal plane, **R** is the rotation and translation matrix and **P** is the orthogonal projection matrix. **R** and **P** can be written as

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{pmatrix} \begin{pmatrix} cos(\beta) & sin(\beta) & 0 & 0 \\ -sin(\beta) & cos(\beta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where the first matrix in **R** is translation matrix and the others are rotation matrices about the z and x axis of the Vicon coordinate frame, respectively. The angle β is the slope of the approximate trajectory line in horizontal plane.

Using above matrices, the COM trajectories of each run are projected to the sagittal plane. To fit the models to experimental data, we need to define stances of each run. In this study we used the tripod gait controller [4, 92] that is characterized by the simultaneous motion of the front and rear leg on the one side and middle leg on the other side. While the robot operating in the tripod gait, it has three different operation modes: single support, double support and flight phase. We only consider the single support phase where the robot is propelled by only one tripod (left or right).

In our analysis, the exact measurement of the COM height is very important to obtain accurate fitting results. The Vicon system has a limited area where the measurement of the rigid body pose is done accurately. To acquire the this area, we swept the robot on the Vicon area, and then we found the region which has acceptable measurement error. We filtered the data where the robot was on the outside of this permissible region.

The goodness of the running is the other important issue of our data filtering. Since templates used in this study live in the sagittal plane, we will use the sagittal plane projection of the COM trajectory obtained by Vicon system. When the robot runs between two points, the horizontal trajectory of COM may not be parallel to one of the axis of Vicon system. In this case, we need to find a straight line which defines the direction of motion. A straight line can be obtained from least squares polynomial fitting of horizontal trajectory. Since the robot is not always able to walk or run along a straight line, sometimes this fitting result gives us unacceptable errors. We defined another criterion for the goodness of the run. Firstly, we fitted a straight line, i.e. $y = ax + b$, to the horizontal trajectory of COM in W, and then we calculated the orthogonal distance between each point and fitted line from the below formula

$$d(i) = \frac{\left|{}^{w}b_y(i) - (a.{}^{w}b_x(i) + b)\right|}{\sqrt{1 + a^2}}. \tag{6.5}$$

We filtered the data which lie the outside of the interval $\mu \pm \alpha$, where μ is the mean and α is the standard deviation of the *distance* vector. The Cartesian body position data ${}^{w}b := [{}^{w}b_x \; {}^{w}b_y \; {}^{w}b_z] \in R^3$, in the World coordinate frame, W, are then projected to the sagittal Plane, S, by a homogeneous transformation $h_w^s : W \rightarrow S$ defined as

$$h_w^s = \begin{pmatrix} \frac{1}{\sqrt{1+a^2}} & 0 & 0 \\ \frac{a}{\sqrt{1+a^2}} & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} {}^{s}b_x \\ 0 \\ {}^{s}b_z \end{pmatrix} = h_w^s . \begin{pmatrix} {}^{w}b_x \\ {}^{w}b_y \\ {}^{w}b_z \end{pmatrix}. \tag{6.6}$$

Fig. 6.5 shows the projection of COM trajectory to the sagittal plane. The red line is the real horizontal trajectory of the robot. The sagittal plane, S, is perpendicular to the fitted straight line. To calculate the net force acting on body during one stride, we first fitted a cubic spline to x and y trajectory of COM and then we double differentiated them. This provided us estimated value of the instantaneous body COM acceleration in S.

$$\mathbf{F}_b = m\frac{d^2}{dt^2} \quad \mathbf{b} = [m\frac{d^2}{dt^2}b_x, m\frac{d^2}{dt^2}b_y] \tag{6.7}$$

where m is the body mass.

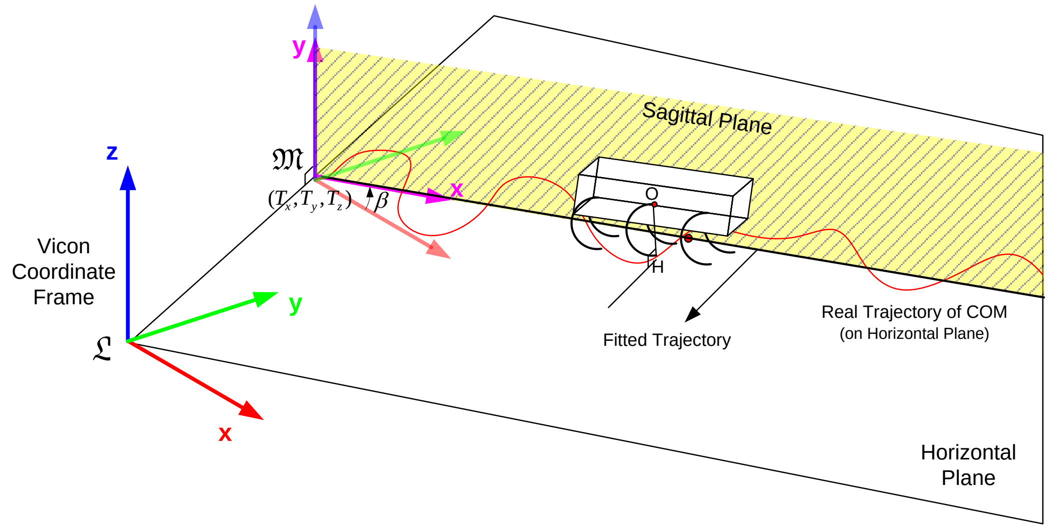


Figure 6.5: Sagittal plane movement of the robot in the world coordinate frame

6.4 Fitting Models to Hexapod Running Data

6.4.1 Fitting Procedure

6.4.1.1 Stance Extraction

The system identification study in this section only considers the single tripod support phase of the robot; therefore, we need to extract the stance phase of each tripod for the gaits that have double stance phase (or in other words that do not exhibit any flight phase). Since our experimental platform was not equipped with any sensor to detect the touchdown and liftoff conditions of leg while running, we performed offline stance detection procedure i.e. time-shifted target speed based stance extracting method that uses mean target velocities of the both left and right tripod as a template signal in order to find best delay with a minimum absolute error. The template signal provided by Buehler clock was shifted along the time axis and at each step we minimized the total absolute error between the actual and target velocity that was calculated by

$$e = \sum_{i=1}^{n} \left| \dot{\theta}(i) - \dot{\hat{\theta}}(i+k) \right|, \tag{6.8}$$

where $\dot{\theta}$ is the actual hip velocity, $\dot{\hat{\theta}}$ is the target hip velocity, n is the number of samples, and k is the delay.

The stride of the robot, where the template model is valid, is defined by the slow phase of shifted target velocity with a minimum error. Figure (6.6-top) gives the extracted strides of the one run. The left and right tripods are represented by a different line type. Figure (6.6-middle) and Fig. (6.6-bottom) show the actual, target and shifted mean velocity of the one tripod. As seen from the figure the stance phase is approximated by a slow phase of the shifted target velocity.

The stride detection procedure explained above is the first step of our system identification study. We will try to optimize the parameters of the SLIP and CCB templates that fit the stance COM trajectory of the hexapod platform. After obtaining the approximate leg parameters of the CCB template, we will change the stance detection procedure with a geometrical one that uses the half circular leg structures as a base for extraction.

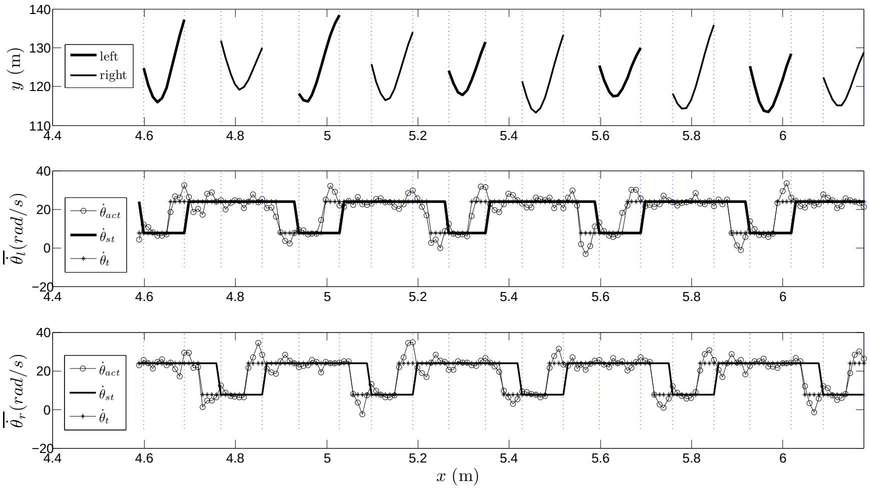


Figure 6.6: The stance phases of one run (top), mean of the actual $\dot{\theta}_{act}$, target angular velocity $\dot{\theta}_t$ and shifted target angular velocity $\dot{\theta}_{st}$ of the left (middle) and right tripod (bottom).

6.4.1.2 Data Processing

Our data consist of three different post-processed data sets whose gait parameters are given in Tab. 6.2. W_c is the angular speed of the leg [rad/sec], ϕ_s is the angle of the stance phase centered on ϕ_0. Duty cycle (dc) is the percentage of the time spent in the stance phase. While the robot is walking, the dc value is more than 0.5. The dc

value which is smaller than 0.5 is used for running and jogging. K_p and K_d is the normalized proportional and derivative gain of the controller, respectively.

Table 6.2: Parameters of each gait

Parameter	Running	Jogging	Pace
ω_c	19.0	17.4	10
ϕ_s	0.75	1.12	1.0
ϕ_0	0.35	0.20	0.2
dc	0.30	0.45	0.6
k_p	0.56	0.56	0.7
k_d	0.03	0.03	0.0

In section 6.2, we have presented the dynamic equations of the SLIP and CCB model as a function of unknown leg parameters. In this section, we use these models to estimate unknown leg and controller parameters of each gait. To obtain a continuous trajectory $(b_x(t), b_z(t)) \in S$, we fitted a cubic spline to data points

$$ {}^{s}b = \begin{pmatrix} {}^{s}b_x(k) \\ {}^{s}b_z(k) \end{pmatrix}, t(k), k = 1...N, \tag{6.9}$$

where N is the number of samples in a stance phase. An estimated COM trajectory is obtained by double differentiating the body accelerations. The initial states of the SLIP and CCB template models are identical to the initial states of the fitted trajectories, i.e.

$$\begin{aligned}
x_{slip}(0) &= x_{ccb}(0) = b_x(t)_{t=t(1)}, \\
\dot{x}_{slip}(0) &= \dot{x}_{ccb}(0) = \left.\frac{db_x(t)}{dt}\right|_{t=t(1)} \\
y_{slip}(0) &= y_{ccb}(0) = b_z(t)_{t=t(1)}, \\
\dot{y}_{slip}(0) &= \dot{y}_{ccb}(0) = \left.\frac{db_z(t)}{dt}\right|_{t=t(1)} \\
\theta_{slip}(0) &= \theta_{ccb}(0) = \theta(t)_{t=t(1)}, \\
\dot{\theta}_{slip}(0) &= \dot{\theta}_{ccb}(0) = \left.\frac{d\theta(t)}{dt}\right|_{t=t(1)}
\end{aligned} \tag{6.10}$$

6.5 Fitting Prodecure

6.5.1 Fitting to Individual Stance Phase of One Gait

The system identification study is achieved in two steps. In the first step, simulations of stance phases of template models for different gaits are performed over the same period of time as the data trajectory and the leg parameters, i.e. $[k, l_0, f_0]$ for SLIP and $[EI, r_0]$ for CCB template, as well as controller parameters $[K_p, K_d, \phi_{lo}, \phi_{td}, t_s]$ are obtained by minimizing error function defined as

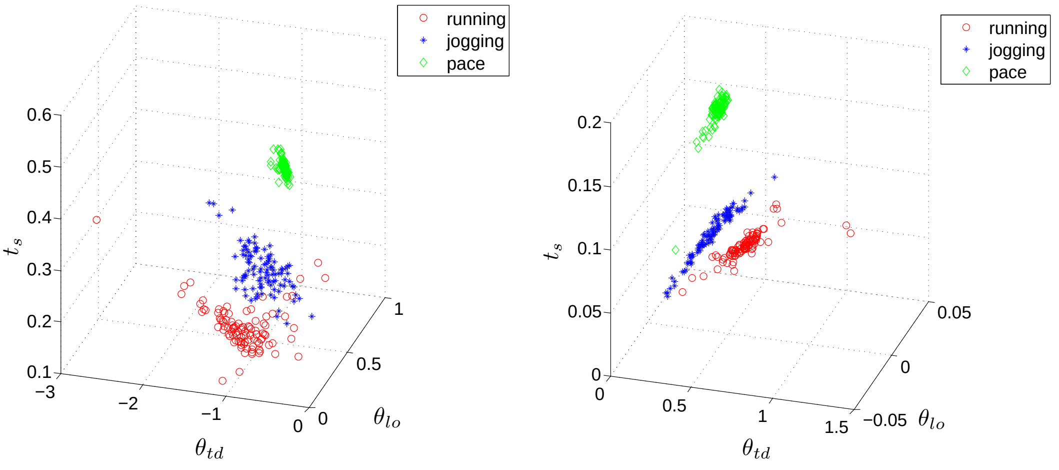
$$e = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|X_{real,i} - X_{model,i}\right\|_{L_2}^{2}} \tag{6.11}$$

where $X \in (x, y)$. This nonlinear unconstrained optimization problem is solved by Nelder-Mead search techniques [63]. Table 6.3 summarizes the results of the first step of system identification study.

In the second step, we fixed the leg parameters and gains of PD controller and we optimized the parameters of the reference signal function of PD controller $[\phi_{lo}, \phi_{td}, t_s]$ to obtain more accurate results. Fig. 6.7 shows the fine tuned controller parameters of each stance. The CCB template errors calculated by (6.11) and are less than that of SLIP template errors for all gait types (see Fig. 6.8).

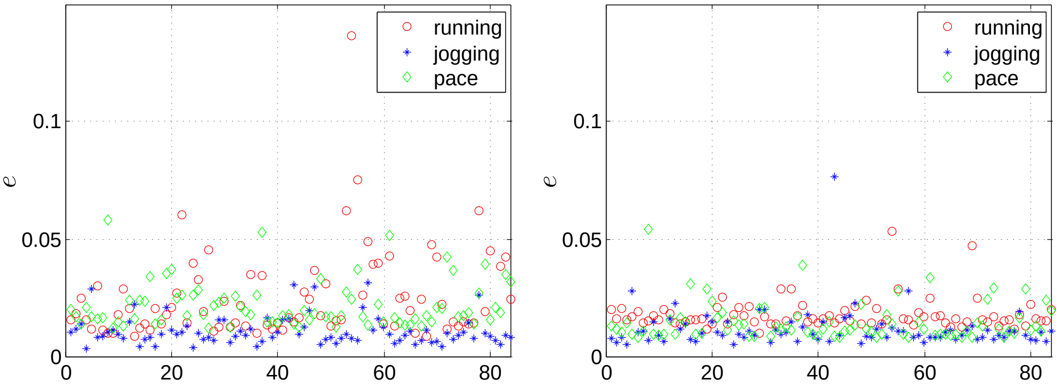
Table 6.3: The results of the first step of the system identification.

SLIP	gait	k	l_0	K_p	K_d
	running	4543.2	0.1300	3.66	35.50
	jogging	1979.2	0.1367	50.99	13.08
	pace	3222.6	0.1300	133.15	1.79
CCB	gait	EI	r_0	K_p	K_d
	running	0.8729	0.0666	724.39	12.63
	jogging	0.5911	0.0673	492.16	17.37
	pace	0.8196	0.0681	54.21	12.44



(a) (b)

Figure 6.7: The controller parameters [$\phi_{lo}, \phi_{td}, t_s$] of (a) SLIP model (b) CCB template obtained in the second step



(a) (b)

Figure 6.8: Fitting errors of (a) SLIP template (b) CCB template

6.5.2 Fitting to All Stance Phase of One Gait

In the previous section, the two models were fitted to individual stance phases of RHex COM trajectory for a single speed with a performance measure defined as an error between the fitted and real trajectory. This method gives different touchdown angle, length and stiffness values over the one gait type that actually should be same. In this section, we have tried to optimize all parameters by fitting the both model to all stances of single speed. For these purpose, we select 100 stances for running, 200 stances for jogging and walking. The cost function is defined as

$$nerr = \frac{1}{M}\sum_{k=1}^{M}\left(\frac{\sqrt{\frac{1}{N}\sum\limits_{i=1}^{N}\left\|X_{real,i} - X_{model,i}\right\|_{L_2}^{2}}}{\sqrt{\frac{1}{N}\sum\limits_{i=1}^{N}\left\|X_{real,i}\right\|_{L_2}^{2}}}\right) \tag{6.12}$$

where M is the total stance number of one gait, N is the sample number of a one stance, $X_{real} \in (x, y, \dot{x}, \dot{y})$ is the experimentally obtained states of the robot COM, and $X_{model} \in (x, y, \dot{x}, \dot{y})$ is the model generated states for the same time instant, i. The error is normalized with respect to the experimental flow.

Table 6.4: The results of the first step of optimization.

Model	Gait	Cost	Optimized Parameters					
			EI	K_p	K_d	θ_{td}	θ_{lo}	t_s
CCB	Running	0.0795	0.1294	46.8497	0.4427	0.6641	-0.0054	0.4573
	Jogging	0.1196	0.1572	24.9387	0.5358	0.5908	-0.0011	1.0257
	Pace	0.4045	0.2630	10.7346	0.0757	0.4271	0.0270	0.1555
			Optimized Parameters					
			k	fx_0				
SLIP	Running	0.2829	2296.4	0.0532				
	Jogging	0.2003	2479.6	0.0449				
	Pace	0.3954	12446	-0.0095				

The optimization is performed in three steps. In the first step, we find the optimum $[k, fx_0]$ parameters of SLIP model and $[EI, K_p, K_d, \theta_{td}, \theta_{lo}, ts]$ parameters of CCB model by minimizing the cost function (6.12). The resulting parameters of two model for three gaits are given in Table 6.4. The errors of CCB model for all gait type are less than that of SLIP model.

Table 6.5: The mean, maximum and minimum leg radius of CCB model and leg length of SLIP model that are calculated by using optimum θ_{td} and fx_0 given in Table 6.4

Model	Gait	Radius of Leg (m)		
		r_{mean}	r_{max}	r_{min}
CCB	Running	0.0615	0.0650	0.0549
	Jogging	0.0613	0.0659	0.0570
	Pace	0.0539	0.0556	0.0524
		Initial Length of Leg (m)		
		l_{0mean}	l_{0max}	l_{0min}
SLIP	Running	0.1222	0.1278	0.1117
	Jogging	0.1209	0.1288	0.1135
	Pace	0.1035	0.1067	0.1005

In the second step, we calculated the initial leg length of SLIP model and radius of CCB model for all individual stance phase of one gait assuming that the leg hits the ground with an undeflected configuration. The mean, maximum and minimum values of the leg length and radius are given in Table 6.5. The leg length and radius calculated in this step are utilized to extract stance data only using the model structures and eliminate the dependency on experimental stance calculation method described previously in Section 6.4.1.1.

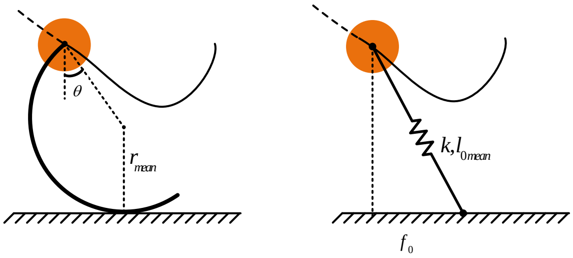


Figure 6.9: The geometrical stance extraction procedure of SLIP and CCB models. The dashed part of the COM trajectory is eliminated since the height of the COM in this part does not satisfy the geometrical constraint given in (6.14)

In the last step of optimization, we fixed the leg parameters $[k, l_{0mean}]$ of SLIP model and $[EI, r_{mean}]$ of CCB model and we optimized fx_0 and $[K_p, K_d, \theta_{td}, \theta_{lo}]$ by eliminating experimental stance data (the dashed part of the trajectory, see Fig. 6.9) where the height of the robot is higher than

$$\begin{aligned} y > \quad & l_{0mean}\left(\sqrt{1-\left(\frac{fx_0}{l_{0mean}}\right)^2}\right) && \text{for SLIP} && (6.13) \\ y > \quad & r_{mean}(1+\cos(\theta_{td})) && \text{for CCB.} && (6.14) \end{aligned}$$

Table 6.6: The optimization results of the last part. We fixed the leg parameters ($[EI, k]$ are given in Table 6.4 and $[l_{0mean}, r_{mean}]$ are given in Table 6.5) and we re-optimized the control parameters of both models.

Model	Gait	Cost	Optimized Parameters			
			K_p	K_d	θ_{td}	θ_{lo}
CCB	Running	0.1043	18.8760	0.7243	0.8319	-0.0051
	Jogging	0.1438	3.7335	0.4220	0.6844	-0.0014
	Pace	0.3605	19.9981	0.2342	0.6503	0.1442
			fx_0			
SLIP	Running	0.2447	0.0567			
	Jogging	0.2560	0.0447			
	Pace	0.3197	-0.0091			

6.6 Conclusion

In this section, we validated that the CCB template is capable of estimating the trajectory of the COM of hexapedal running for a large range of speeds. We compare the efficiency of this model with a widely used SLIP template and we show that the normalized error between real and model generated trajectory is about 2% for all range of speeds, which are smaller than the error of SLIP model. It is emphasized that this more complete model can be implemented on RHex type of robots to derive controllers that provides wide range of stable and robust dynamic locomotion.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this thesis, we have focused on the characterization and optimization of running with curved legs. Although the spring-loaded-inverted-pendulum (SLIP) model characterizes the essential aspects of running dynamics of humans and animals, because of its excessive simplicity this model cannot capture the fundamental aspects of running with curved legs such as rolling contact during stance phase and nonlinear stiffness of the leg, which are inherent properties of the curved leg. In this thesis we have offered a computationally efficient, physically accurate and more realistic low dimensional model which inherently satisfies rolling contact and variable stiffness properties and is intended for use as a template to understand dynamics of legged locomotion. We also experimentally validated that our model captures nonlinear stiffness property of the leg very accurately. The proposed leg model is not limited to RHex-like robots, it can be used to understand the dynamics of locomotion with compliant curved structures other than half circular one. This energy based compliance calculation procedure can be also generalized to understand the dynamics of a wide range of motion and various behaviors with an arbitrary shape compliant leg.

We designed controllers based on optimal control theory and analyzed the different locomotive behavior of the circular compliant legged monopod such as running and jumping over obstacles. We have used simple open-loop control strategies since they do not require rich sensory information (especially without notice of body state). Despite the simplicity of controller structures used throughout the thesis, we think that the results obtained using standard optimization techniques will give an idea how the feedback control policies should be structured to obtain reactive, stable and more robust controller for the autonomous dynamic locomotion. The stability of the designed controllers have been analyzed using Apex return map, and we have shown that the motion of the system under forward speed and height controllers are neutrally stable.

We have presented the optimal control approach for the rough terrain locomotion of half circular legged robot. Motivating by the performance of the long and high triple jumper athletes, we have defined optimal control problem that finds the optimum take-off conditions for the triple jump motion considering different obstacle sizes. We have examined the sensitivity of triple jump trajectory to variations in initial states, control parameters and localization of obstacle. We have also defined a controller that takes into account the safety margin around an obstacle. The current study is limited to one obstacle whose localization is known in the environment. We need to consider irregularly distributed different size multiple obstacles to obtain more realistic controller structure.

Finally, the new compliant circular beam template is anchored to produce a representative model to capture the dynamics of hexapedal locomotion. We compare the efficiency of this model with a widely used SLIP template. The results of this study confirms that this candidate template can be used to understand the locomotion of the complex biological systems.

The next step of our study is the experimental verification of our controllers which requires the parameter identification of real robot platform and reliable high-bandwidth state feedback. The physical limitations of the robot platform such as torque limits of the hip motor or maximum deflection of the leg can be added as a constraint to the optimal control problems to enhance practical applicability of the controller. The performance of the controller should be investigated for other leg design and system parameters.

Throughout the thesis, we have analyzed the effects of circularity and compliance properties of the leg on the dynamics of one legged robot. Although studies about monopod locomotion may improve knowledge about dynamics of multi-legged locomotion, their pay load capabilities, stability and limited locomotion properties such as performing only hopping motion makes them hard to use in real time applications.

The circular, compliant leg model will be extended to two, four or six legged models with increasingly capable dynamic abilities. We hope that our research will serve as a base for future studies on multi-legged locomotion with a circular and compliant leg.

REFERENCES

[1] K. C. Galloway, J. E. Clark, and D. E. Koditschek, “Design of a tunable stiffness composite leg for dynamic locomotion,” in *ASME IDETC/CIE*, 2009.

[2] Y. O. Aydin, K. C. Galloway, Y. Yazicioglu, and D. E. Koditschek, “Modeling the compliance of a variable stiffness c-shaped leg using castigliano’s theorem,” *ASME Conference Proceedings*, vol. 2010, no. 44106, pp. 705–713, 2010.

[3] J. Weingarten, D. E. Koditschek, H. Komsuoglu, and C. Massey, “Robotics as the delivery vehicle: A contextualized, social, self paced, engineering education for life-long learners,” *Robotics Science and Systems Workshop on Research in Robots for Education*, 2007.

[4] U. Saranli, M. Buehler, and D. E. Koditschek, “RHex: a simple and highly mobile robot,” *International Journal of Robotics Research*, vol. 20, no. 7, pp. 616–631, July 2001.

[5] S. J. Gould, “Kingdoms without wheels,” *Natural History*, vol. 90, no. 3, p. 42, Mar. 1981.

[6] J. L. Kavanau and D. H. Brant, “Wheel-running preferences of peromyscus,” *Nature*, vol. 208, no. 5010, pp. 597–598, 1965.

[7] I. Walker, “Ideas in theoretical biology why legs and not wheels?” *Acta Biotheoretica*, vol. 39, no. 5010, pp. 151–155, 1991.

[8] M. LaBarbera, “Why the wheels won’t go,” *The American Naturalist*, vol. 121, no. 3, pp. 395–408, 1983.

[9] J. Wong, *Theory of ground vehicles*. John Wiley & Sons, 2008.

[10] M. Bekker, *Theory of land locomotion: mechanics of vehicle Mobility*. Ann Arbor, MI: University of Michigan Press, 1956.

[11] R. Full, K. Earls, M. Wong, and R. Caldwell, “Locomotion like a wheel?” *Nature*, vol. 365, no. 6446, p. 495, Oct. 1993.

[12] R. L. Caldwell, “A unique form of locomotion in a stomatopod-backward somersaulting,” *Nature*, vol. 282, pp. 71–73, 1979.

[13] J. Brackenbury, “Caterpillar kinematics,” *Nature*, vol. 390, no. 6659, pp. 453–453, 1997.

[14] M. Garcia-Paris and S. M. Deban, “A novel antipredator mechanism in salamanders: rolling escape in hydromantes platycephalus.” *Journal of Herpetology*, vol. 29, pp. 149–151, 1995.

[15] J. Diamond, “Transport mechanisms: the biology of the wheel,” *Nature*, vol. 302, pp. 572–573, 1983.

[16] R. H. Armour and J. F. Vincent, “Rolling in nature and robotics: a review,” *Journal of Bionic Engineering*, vol. 3, no. 4, pp. 195 – 208, 2006.

[17] M. H. Raibert, *Legged robots that balance*. Cambridge, MA: The MIT Press, 1986.

[18] M. H. Raibert, H. B. Brown, E. Hastings, J. Koechling, K. N. Murphy, S. S. Murthy, and A. J. Stentz, “Dynamically stable legged locomotion,” MIT Artificial Intelligence Laboratory, Tech. Rep., 1989.

[19] P. Gregorio, M. Ahmadi, and M. Buehler, “Design, control, and energetics of an electrically actuated legged robot,” *IEEE Trans. Systems, Man, and Cybernetics*, vol. 27, pp. 626–634, 1997.

[20] J. Prosser and M. Kam, “Vertical control for a mechanical model the one-legged hopping machine,” in *Proceedings of First IEEE Conference on Control Applications*, Dayton, USA, Sep 1992, pp. 136–141.

[21] J. Prosser and M. Kam, “Control of hopping height for a one-legged hopping machine,” *Mobile Robots VII*, vol. 1831, pp. 604–612, 1992.

[22] H. Rad, P. Gregorio, and M. Buehler, “Design, modeling and control of a hopping robot,” in *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS ’93*, vol. 3, Jul 1993, pp. 1778 –1785 vol.3.

[23] G. Zeglin, “Uniroo: a one legged dynamic hopping robot,” Master’s thesis, Massachusetts Institute of Technology, Cambridge, 1991.

[24] T. McGeer, “Passive dynamic walking,” *The International Journal of Robotics Research*, vol. 9, no. 2, pp. 62–82, Apr. 1990.

[25] J. Donelan, R. Kram, and A. Kuo, “Mechanical work for step-to-step transitions is a major determinant of the metabolic cost of human walking,” *Journal of Experimental Biology*, vol. 205, no. 23, pp. 3717–3727, 2002.

[26] A. Kuo, J. Donelan, and A. Ruina, “Energetic consequences of walking like an inverted pendulum: Step-to-step transitions,” *Exercise and Sport Sciences Reviews*, vol. 33, no. 2, pp. 88–97, 2005.

[27] P. G. Adamczyk, S. H. Collins, and A. D. Kuo, "The advantages of a rolling foot in human walking." *The Journal of experimental biology*, vol. 209, no. 20, pp. 3953–63, Oct. 2006.

[28] R. Tedrake, T. Zhang, M. fai Fong, and H. Seung, "Actuating a simple 3d passive dynamic walker," in *Proceedings of 2004 IEEE International Conference on Robotics and Automation, ICRA '04.*, vol. 5, May 2004, pp. 4656 – 4661 Vol.5.

[29] F. Asano and Z.-W. Luo, "On energy-efficient and high-speed dynamic biped locomotion with semicircular feet," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2006, pp. 5901 –5906.

[30] P. C. Lin, "Proprioceptive sensing for a legged robot," Ph.D. dissertation, The University of Michigan, 2005.

[31] E. Z. Moore, "Leg design and stair climbing control for the RHex robotic hexapod," Master's thesis, McGill University, 2002.

[32] R. Altendorfer, N. Moore, H. Komsuoglu, M. Buehler, H. B. Brown Jr., D. McMordie, U. Saranli, R. J. Full, and D. E. Koditschek, "RHex: a biologically inspired hexapod runner," *Autonomous Robots*, vol. 11, no. 3, pp. 207–213, 2001.

[33] U. Saranli and D. Koditschek, "Template based control of hexapedal running," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, 2003, pp. 1374–1379.

[34] L. L. Howell, *Compliant mechanisms.* New York: John Wiley and Sons, 2001.

[35] J. Y. Jun and J. E. Clark, "Effect of rolling on running performance," in *2011 IEEE International Conference on Robotics and Automation.* IEEE, May 2011, pp. 2009–2014.

[36] J. Y. Jun, D. Haldane, and J. E. Clark, "Compliant leg shape, reduced-order models and dynamic running," in *2010 International Symposium on Experimental Robotics*, 2010, pp. 1–15.

[37] H. Langhaar, *Energy methods in applied mechanics.* New York: John Wiley, 1962.

[38] R. Budynas, *Advanced strength and applied stress analysis.* New York, NY: McGraw-Hill, 1998.

[39] R. Budynas and K. Nisbett, *Shigley's mechanical engineering design.* New York, NY: McGraw-Hill, 2008.

[40] Sayginer, E, "Modeling the effects of half circular compliant legs on the kinematics and dynamics of a legged robot," Master's thesis, Middle East Technical University, 2010.

[41] F. Iida and R. Tedrake, "Motor control optimization of compliant one legged locomotion in rough terrain," in *Intelligent Robots and Systems, IROS 2007. IEEE/RSJ International Conference on*, Nov 2007, pp. 2230 –2235.

[42] J. Vermeulen, D. Lefeber, and B. Verrelst, "Control of foot placement, forward velocity and body orientation of a one-legged hopping robot," *Robotica*, vol. 21, no. 01, pp. 45–57, 2003.

[43] T. Y. Wu, T. J. Yeh, and B. H. Hsu, "Trajectory planning of a one-legged robot performing a stable hop," *International Journal of Robotics Research*, vol. 30, no. 8, pp. 1072–1091, 2011.

[44] T. Y. Wu and T. J. Yeh, "Optimal design and implementation of an energy-efficient biped walking in semi-active manner," *Robotica*, vol. 27, no. 06, pp. 841–852, 2009.

[45] K. Mombaur, M. Scheint, and M. Sobotka, "Optimal control and design of bipedal robots with compliance," *At-Automatisierungstechnik*, vol. 57, no. 7, pp. 349–358, 2009.

[46] C. Chevallereau and Y. Aoustin, "Optimal reference trajectories for walking and running of a biped robot," *Robotica*, vol. 19, no. 05, pp. 557–569, 2001.

[47] K. Mombaur, H. Bock, J. Schloder, and R. Longman, "Open-loop stability - a new paradigm for periodic optimal control and analysis of walking mechanisms," in *2004 IEEE Conference on Robotics, Automation and Mechatronics,*, vol. 2, Dec 2004, pp. 704–709.

[48] G. Bessonnet, P. Seguin, and P. Sardain, "A parametric optimization approach to walking pattern synthesis," *International Journal of Robotics Research*, vol. 24, no. 7, pp. 523–536, 2005.

[49] R. L. Tedrake, "Applied optimal control for dynamically stable legged locomotion," Ph.D. dissertation, Massachusetts Institute of Technology, 2004.

[50] J. Weingarten, G. Lopes, M. Buehler, R. Groff, and D. Koditschek, "Automated gait adaptation for legged robots," in *Proceedings of 2004 IEEE International Conference on Robotics and Automation, ICRA '04.*, no. 3, 2004, pp. 2153–2158.

[51] J. Y. Jun and J. Clark, "Dynamic stability of variable stiffness running," in *Proceedings of 2009 IEEE International Conference on Robotics and Automation, ICRA '09.*, May 2009, pp. 1756 –1761.

[52] U. Saranli, A. A. Rizzi, and D. E. Koditschek, "Model-based dynamic self-righting maneuvers for a hexapedal robot," *International Journal of Robotics Research*, vol. 23, no. 9, pp. 903–918, 2004.

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

[54] R. Altendorfer, U. Saranli, H. Komsuoglu, D. Koditschek, J. Brown, H.Benjamin, M. Buehler, N. Moore, D. McMordie, and R. Full, "Evidence for spring loaded inverted pendulum running in a hexapod robot," in *Experimental Robotics VII*, ser. Lecture Notes in Control and Information Sciences. Springer Berlin Heidelberg, 2001, vol. 271, pp. 291–302.

[55] R. Blickhan and R. Full, "Similarity in multilegged locomotion: Bouncing like a monopode," *Journal of Comparative Physiology A*, vol. 173, pp. 509–517, 1993.

[56] R. Blickhan, "The spring-mass model for running and hopping," *Journal of Biomechanics*, vol. 22, no. 11-12, pp. 1217 – 1227, 1989.

[57] E. Sayginer, "Modeling the effects of half circular compliant legs on the kinematics and dynamics of a legged robot," Master's thesis, Middle East Technical University, 2010.

[58] M. M. Ankarali, E. Sayginer, Y. Yazicioglu, A. Saranli, and U. Saranli, "A dynamic model of running with a half-circular compliant leg," in *15th International Conference on Climbing and Walking Robots (CLAWAR)*, Jul 2012.

[59] H. Goldstein, *Classical Mechanics*, 2nd ed. Reading, MA: Addison-Wesley Publishing Company, 1980.

[60] K. Galloway, J. Clark, M. Yim, and D. Koditschek, "Experimental investigations into the role of passive variable compliant legs for dynamic robotic locomotion," in *Proceedings of 2011 IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 1243 –1249.

[61] K. C. Galloway and J. E. C. et al, "Variable stiffness legs for robust, efficient, and stable dynamic running," *Journal of Mechanisms and Robotics*, Jan 2013.

[62] C. S. Cai and B. Roth, "On the planar motion of rigid bodies with point contact," *Mechanism and Machine Theory*, vol. 21, no. 6, pp. 453–466, 1986.

[63] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[64] M. D. Lasa and M. Buehler, "Dynamic compliant walking of a quadruped robot: preliminary experiments," in *Proc. International Conference on Climbing and Walking Robots*, 2000.

[65] 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.

[66] S. H. Strogatz, *Nonlinear dynamics and chaos: With applications to physics, Biology, Chemistry And Engineering*. Westview Press, 1994.

[67] H. K. Khalil, *Nonlinear systems*, ser. Interdisciplinary Applied Mathematics. Prentice Hall, 1996, vol. 122, no. 0.

[68] M. Vidyasagar, *Nonlinear systems analysis*, P. Janzow, J. Wenzel, C. Goffie, and A. Dworkin, Eds. Prentice Hall, 1993.

[69] S. Sastry, *Nonlinear systems: Analysis, stability and control.* Springer, 1999, vol. 10.

[70] P. Holmes, R. 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.

[71] C. T. Farley, "Leg stiffness and stride frequency in human running," *Journal of Biomechanics*, vol. 29, no. 2, pp. 181–186, Feb 1996.

[72] C. T. Farley, J. Glasheen, and T. A. McMahon, "Running springs: speed and animal size." *The Journal of experimental biology*, vol. 185, pp. 71–86, Dec 1993.

[73] K. Takeuchi, S. Kuswadi, H. Nakaura, and N. Sampei, "Continuous hopping motion control experiment of one linear actuator robot," in *SICE Annual Conference*, 2002.

[74] S. Kuswadi, A. Ohnishi, A. Takahashi, M. Sampei, and S. Nakaura, "A one linear actuator hopping robot: modeling and control," *Advanced Robotics*, vol. 17, pp. 709–737, 2003.

[75] M. Kovac, M. Schlegel, J.-C. Zufferey, and D. Floreano, "Steerable Miniature Jumping Robot," *Autonomous Robots*, vol. 28, no. 3, pp. 295–306, 2010.

[76] J. Zhao, N. Xi, F. Cintron, M. W. Mutka, and L. Xiao, "A single motor actuated miniature steerable jumping robot," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 4274–4275.

[77] J. Zhao, J. Xu, B. Gao, N. Xi, F. Cintron, M. Mutka, and L. Xiao, "Msu jumper: A single-motor-actuated miniature steerable jumping robot," *Robotics, IEEE Transactions on*, vol. 29, no. 3, pp. 602–614, 2013.

[78] C. Papadopoulos, A. Glavroglou, G. Groulos, and L. Tsarouchas, "A biomechanical analysis of the support phase during the preparation and take-off in long and high jumping," in *Proceedings - International Symposium on Biomechanics in Sports*, 1995.

[79] R. M. Alexander, “Optimum take-off techniques for high and long jumps.” *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, vol. 329, no. 1252, pp. 3–10, Jul. 1990.

[80] R. Armour, K. Paskins, A. Bowyer, J. Vincent, and W. Megill, “Jumping robots: a biomimetic solution to locomotion across rough terrain,” *Bioinspiration & Biomimetics*, vol. 2, no. 3, p. S65, 2007.

[81] J. H. Sanders, “The takeoff in the long jump & other running jumps,” in *17 International Symposium on Biomechanics in Sports*, 1999.

[82] A. Seyfarth, A. Friedrichs, V. Wank, and R. Blickhan, “Dynamics of the long jump.” *Journal of biomechanics*, vol. 32, no. 12, pp. 1259–67, Dec. 1999.

[83] A. Seyfarth, R. Blickhan, and J. L. V. Leeuwen, “Optimum take-off techniques and muscle design for long jump,” *J Exp Biol*, vol. 203, no. Pt 4, pp. 741–750, Feb 2000.

[84] J. Perttunen, H. Kyrolainen, P. V. Komi, and A. Heinonen, “Biomechanical loading in the triple jump,” *Journal of Sports Sciences*, vol. 18, no. 5, pp. 363–370, 2000.

[85] U. Saranli and D. E. Koditschek, “Template based control of hexapedal running,” in *Proceedings of the IEEE Conference of Robotics and Automation*, Taipei, Taiwan, 2003.

[86] I. Poulakakis and J. Grizzle, “Formal embedding of the spring loaded inverted pendulum in an asymmetric hopper,” in *In Proceedings of the European Control Conference*, 2007.

[87] D. E. Koditschek, R. J. Full, and M. Buehler, “Mechanical aspects of legged locomotion control,” *Arthropod Structure & Development*, vol. 33, no. 3, pp. 251 – 272, 2004.

[88] C. Li, P. B. Umbanhowar, H. Komsuoglu, D. E. Koditschek, and D. I. Goldman, “Sensitive dependence of the motion of a legged robot on granular media,” *Proceedings of National Academy of Science (PNAS)*, vol. 106, no. 9, pp. 3029–3034, Feb. 2009.

[89] H. Komsuoglu, K. Sohn, R. J. Full, and D. E. Koditschek, “A physical model for dynamical arthropod running on level ground,” in *Proceedings of International Symposium on Experimental Robotics*, 2008.

[90] H. Komsuoglu, A. Majumdar, Y. O. Aydin, and D. E. Koditschek, “Characterization of dynamic behaviors in a hexapod robot,” in *Proceedings of International Symposium on Experimental Robotics*, 2010.

[91] J. Y. Jun, “Characterization and optimization of running with curved legs,” Ph.D. dissertation, The Florida State University, 2011.

[92] J. Weingarten, G. Lopes, M. Buehler, R. Groff, and D. Koditschek, “Automated gait adaptation for legged robots,” in *Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004 IEEE International Conference on*, vol. 3, May 2004, pp. 2153 – 2158 Vol.3.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: ÖZKAN AYDIN, Yasemin
Nationality: Turkish (TC)
Date and Place of Birth: 22.11.1981, Bursa
Marital Status: Single
Phone: 0 312 2104558
Fax: 0 312 2102304

EDUCATION

Degree	Institution	Year of Graduation
Visiting Scholar	University of Pennsylvania	2009-2010
B.S.	Yıldız Technical University	2005
High School	Bursa Anatolian Health Vocational School	1999

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
09/2004-01/2005	Yildiz Technical University	Student Assistant
09/2004-01/2005	Yilmaz Electric Automation	Trainee
07/08-2004	TOFAS Automotive Industry	Trainee
06/07-2004	Oyak-Renault Automotive Industry	Trainee
07/08-2003	BOSEN Natural Gas Station	Trainee

PUBLICATIONS

Journal Publications

Y. Ozkan Aydin, A. Saranli, Y. Yazicioglu, U. Saranli and K. Leblebicioglu, “Optimal Control of a Half Circular Compliant Legged Monopod,” In: *Control Engineering Practice - Under Review*, 2014

Y. Ozkan Aydin, and U. Saranli "A Circular - Compliant Beam Template for a Hexapod Robot" In: *Under Preperation*, 2014

International Conference Publications

Y. Ozkan Aydin, K. Galloway, Y. Yazicioglu, D. Koditschek, “Modeling the Compliance of a Variable Stiffness C-Shaped Leg Using Castigliano’s Theorem,” In: *ASME International Design Engineering Technical Conferences (IDETC), Montreal, Quebec, Canada*, 2010

Haldun Komsuoglu, Anirudha Majumdar, Yasemin Ozkan Aydin Daniel E. Koditschek, "Characterization of Dynamic Behaviors in a Hexapod Robot" In: *International Symposium on Experimental Robotics*, December 2010

Y. Özkan Aydın and K. Leblebicioğlu , “Single Step Optimal Control Of The Half Circular Legged Monopod ,” In: *12th International Workshop on Research and Education in Mechatronics, Kocaeli, Türkiye*, 2011

National Conference Publications

Y. Özkan Aydın, A. Saranlı, and K. Leblebicioğlu, “Dairesel Dört Esnek Bacaklı Robotun Dinamik Modeli,” In: *TOK 2013, Otomatik Kontrol Türk Milli Komitesi, Malatya, Türkiye*, 2013

Y. Özkan Aydın, and K. Leblebicioğlu, “Harita Etiketlerinin Eniyi Şekilde Yerleştirilmesi Problemi” In: *TOK 2013, Otomatik Kontrol Türk Milli Komitesi, Malatya, Türkiye*, 2013

Y. Özkan Aydın, and K. Leblebicioğlu, “Tek Bacaklı Robotla Engel Atlama” In: *TOK 2012, Otomatik Kontrol Türk Milli Komitesi, Niğde, Türkiye*, 2012

Y. Özkan Aydın, and K. Leblebicioğlu, “Esnek Tek Bacaklı Robot için En Iyi Denetimci Tasarımı” In: *TOK 2011, Otomatik Kontrol Türk Milli Komitesi, İzmir, Türkiye*, 2011

GRADUATE COURSES

Advanced dynamics
Applied optimal control
Stability Theory of Dynamical Systems
Introduction to Robotics, Robot Vision
Optimization
Physiological Control Systems Analysis
Linear Systems Theory I,II, Discrete Time Systems
Artificial Intelligence, NeuroComputers
Switching and Automata Theory I,II

SUMMER SCHOOL CLASSES

ETH Summer School on Dynamic Walking and Running with Robots (40 hours), ETH, Zurich, July 2011
ETH Summer School on Soft Robotics (40 hours) , ETH, Zurich, July 2012

AWARDS

The Scientific and Technological Research Council of Turkey
International Research Fellowship
The Scientific and Technological Research Council of Turkey PhD fellowship
B.S. with High Honors , I completed B.S degree in 3.5/4 years
Turkey undergraduate fellowship
Top thirty ranks in 2000 year’s Turkish University Entrance Exam
Graduated with the highest honors from high school
119