Trajectory Optimisation Inspired Design for Legged Robotics

Presented by:
Callen Fisher

Prepared for:
A/Prof. A. Patel
Dept. of Electrical Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town in fulfilment of the academic requirements for a Doctor of Philosophy degree in Electrical Engineering

March 9, 2021

University of Cape Town

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

University of Cape Town

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Callen Fisher Signed by candidate

Date: 9 MARCH 2021

Acknowledgements

The success of this project could not have been achieved without the continual support and guidance of my supervisor, A/Prof. Amir Patel. Amir, after 8 years of research we have finally done it! It has been an honour working under you and I am eternally grateful for all you have done for me in shaping my future. We started off in a small room at the back of the Control lab and now we have an extremely successful research lab. Thanks again for always pushing me to give my best.

I would also like to thank the NRF (National Research Foundation) for sponsoring this research under grant number 101245 and 116415. I would also like to thank the following companies for their donations, discounts and advice: Festo, Beckhoff, Ingenia and RS components.

I would personally like to thank Prof. C. Hubicki for always being available to give advice and for taking the time to assist me in my research. Without your support, that RA-L paper would not have been possible! There have also been a number of UCT colleagues who have helped me throughout my research career and have provided invaluable advice! Thank you Ed, Robyn and Justin.

To my MSc minion, Josh, thanks for putting up with me and for designing our robot! I am excited to see what you achieve in your future! Keep up the hard work. To the friends I have made in the Mechatronics lab, thank you for keeping me sane with copious amounts of coffee (special thank you to Do Yeou!)

To my new colleagues at SU, thank you for giving me the opportunity to work alongside you, the change of scenery has done wonders for my productivity. Lastly I would like to thank my parents. Without all the food and coffee you have provided I would not have survived. Sorry about all the late nights and nerd talk.

This PhD has had some rough times: Student protesting, day zero water crisis, no electricity, new full time job and Covid-19... But I made it...

Nomenclature

- PD: Proportional–derivative controller
- CoT: Cost of transport
- CoM: Centre of mass
- GRF: Ground reaction force
- DoF: Degree(s) of freedom
- SLIP: Spring loaded inverted pendulum
- SEA: Series elastic actuator
- CoG: Centre of gravity
- CoP: Centre of pressure
- ZMP: Zero moment point
- (E)KF: (Extended) Kalman filter
- PFL: Partial feedback linearisation

Abstract

Control of legged robots is a non-trivial task, especially when looking at aperiodic (non-steady-state) manoeuvres such as rapid acceleration and deceleration. Observing nature, animals are seen to effortlessly perform these rapid transient manoeuvres, however, robust walking is still considered a complex task in the robotics literature. For robots to successfully explore the unknown world outside of the laboratory environment, these transient manoeuvres need to be thoroughly understood and mastered.

As controlling and designing legged robots is an extremely complex task, this thesis argues that trajectory optimisation methods can be employed to make various aspects of design more tractable. First trajectory optimisation methods were employed to determine how legged robots should accelerate. In nature animals are seen to leap straight into the desired gait, however, in the robotics literature, robots often perform multiple discrete gait transitions, or accelerate extremely slowly. The results of the study revealed that the optimal method to accelerate was to launch straight into the desired gait (for both bipeds and quadrupeds), with a sliding mass template model emerging for all results.

Another discrepancy between the literature and nature is that animals have active spines which have been shown to aid in rapid locomotion tasks. In the robotics literature a number of spines exists, however, which is the optimal morphology for transient manoeuvres? Using Monte Carlo-based trajectory optimisation methods, the rigid, revolute and prismatic spine morphologies were compared, with results showing the prismatic spine was the optimal configuration for long-time-horizon tasks.

Due to transient locomotion requiring accurate and complex whole-body models, resulting in computationally expensive optimisation problems, an "optimisation-inspired" approach (akin to "bioinspiration") was taken to identify heuristics and trends for a monopod robot. Initially, optimal trajectories for a two link leg monopod were analysed. Interestingly, during the stance phase, the axial force of the leg behaved in a "bang-bang" like fashion with fine hip torque control. Furthermore the aerial phase showed correspondence to the popular Raibert style controller. This resulted in the development of a hybrid pneumatic-electric monopod robot as a test-bed. Trajectories were then optimised for the robot to determine if these heuristics held. From these results, a stance phase PD controller for the hip actuator was developed and simulated under disturbances to test robustness. The resulting controller and heuristic was successfully tested on the platform, performing a long-time-horizon motion, which included transient phases of acceleration and deceleration.

Contents

1 Introduction **1**

1.1 Motivation and Background to the Study 1

1.2 Research Questions 3

1.3 Scope and Limitations 3

1.4 Plan of Development 4

1.5 Publications 5

2 Literature Review **6**

2.1 Aspect 1: How do Animals Perform Transient Motions? 7

2.2 Aspect 2: Optimal Morphology for Transient Locomotion 12

2.3 Aspect 3: Can we Control these Transient Motions? 19

2.4 Trajectory Optimisation 22

2.4.1 Direct Methods 25

2.4.2 Through-Contact Methods 26

2.4.3 Cost Functions 28

3 Methodology **30**

3.1 Modelling and Simulating the System . . . 30

3.1.1 Euler-Lagrange Dynamics . . . 31

3.1.2 Trajectory Optimisation . . . 33

3.2 Validating Controllers and Verifying Models . . . 47

3.2.1 Controller Design . . . 48

3.2.2 Support Rig . . . 48

3.2.3 Hybrid Pneumatic-Electric Robot . . . 49

4 How Should Legged Robots Rapidly Accelerate and Decelerate? 51

4.1 Determining How Legged Robots Should Rapidly Accelerate and Decelerate . . . 52

4.2 Analysis of How Legged Robots Should Accelerate and Decelerate . . . 54

4.2.1 Gait Analysis . . . 54

4.2.2 Comparison to Long-Time-Horizon Trajectories . . . 56

4.2.3 Validation of Through-Contact Methods . . . 56

4.3 Template Identification . . . 56

4.4 Summary of Discussion . . . 59

5 What is the Optimal Spine Morphology for Rapid Acceleration and Deceleration Manoeuvres? 60

5.1 Determining the Optimal Spine Morphology . . . 61

5.2 Optimal Spine Morphology Results and Discussion . . . 63

5.2.1 Bootstrapping Results . . . 63

5.2.2 Mass Related Results 65
5.2.3 Stride-Length and Frequency Observations 66
5.2.4 Applied Work Results 66
5.2.5 GRF Observations 68
5.2.6 Energy Efficiency Comparison 69
5.3 Summary of Discussion 70

6 Can Optimisation Methods Inspire Controller Design? 71

6.1 Generating Two Link Monopod Trajectories 72
6.2 Analysis of Two Link Monopod Results 73
6.2.1 Neutral Point Analysis 74
6.2.2 GRF Analysis 75
6.2.3 Proposed Robot 75
6.3 Development of a Pneumatic Actuator Model 76
6.4 Generating Trajectories for the Pneumatic Robot 78
6.5 Analysis of Pneumatic-Electric Monopod Results 81
6.5.1 Neutral Point Analysis 82
6.5.2 Torque Analysis 83
6.5.3 GRF Analysis 83
6.6 Controller Simulation Tests 85
6.6.1 GRF Analysis 86
6.6.2 Neutral Point Analysis 87

6.6.3 Disturbance Analysis 88

6.7 Physical Experiments 90

6.8 Summary of Discussion 93

7 Conclusions and Future Work **94**

A Equations of Motion **108**

B PFL Mapping **111**

C Physical Experiments **113**

C.1 Vertical Experiments 113

C.2 Fixed Body Experiments 114

List of Figures

1.1 An overview of the research covered in this dissertation. 1

1.2 A graphical layout of the dissertation. 4

2.1 Atlas performing a Parkour Manoeuvre. 7

2.2 An image of the Dodinga Hills Guereza, which is a subspecies of monkey, effortlessly leaping through the air. 10

2.3 An example of the Spring Loaded Inverted Pendulum (SLIP) template. 11

2.4 An example of a reduced order model for the MIT cheetah robot. 11

2.5 Animals, such as horses, have adapted to handle these rapid deceleration manoeuvres by throwing their bodies backwards to avoid pitching forward. 12

2.6 A typical Series Elastic Actuator. 14

2.7 A comparison of the load distribution for each motor for the five-bar, four-bar and series articulate leg morphology. 16

2.8 A number of state-of-the-art quadruped robots are shown. 18

2.9 A graphical representation of the Raibert controller. 20

2.10 An example of a robot attached to a boom system. 23

2.11 A comparison of single and multiple shooting methods. 26

3.1 The image shows the three robot types of interest. 31

3.2 A graphical representation of how hybrid dynamics transitions states. 33

3.3 A graphical representation of the *stitching* method. 34

3.4 The different phases of motion for each of the robots is shown. 35

3.5 A graphical representation showing how the trajectory is first broken up into node points (i) and then collocation points (j). 37

3.6 Image showing an example complementarity constraint. 40

3.7 Flow diagram showing how the seed was generated and solved (using regularisation techniques). 41

3.8 The prescribed contact order for the steady-state steps was enforced through constraints for the monopod, biped and quadruped robots. 43

3.9 The different rest positions for the relevant robots used through out this research are shown. 45

3.10 The boom system developed by Joshua Van Zyl. 49

3.11 A SolidWorks render of the monopod robot that was used to test controllers on is shown on the left with the physical robot on the right. 50

4.1 An overview of the research covered in this chapter. 52

4.2 A comparison of the acceleration trajectories for the two cost functions at different time period constraints (2 seconds and 6 seconds). 55

4.3 A comparison of the deceleration trajectories for the two cost functions at different time period constraints (6 seconds and 2 seconds). 56

4.4 Stitching the acceleration, steady-state and deceleration trajectories generates the long-time-horizon trajectory. 57

4.5 The long-time-horizon trajectories resemble the stitched results. 57

4.6 Note how the friction is often clipping at ± 1 (the coefficient of friction). 58

4.7 The magnitude of the horizontal foot velocity while the foot is in stance. 58

4.8 Note how the simple sliding mass 1 DoF model accurately tracks the forward velocity of the acceleration and deceleration manoeuvre of the biped and quadruped robot. 59

5.1 An overview of the main research question for this chapter. 61

5.2 The three spines of interest are shown. 62

5.3 An example velocity profile for a successful trajectory. 64

5.4 The mass versus cost (minimum time) for the optimal results reveal an interesting trend. 65

5.5 A comparison of the stride-frequency and stride-length for the three spine morphologies. 67

5.6 A comparison of the stride-average acceleration and deceleration of the three spines of interest. 67

5.7 The graph shows the average total GRF loading for the optimal trajectories versus mass. 68

5.8 An example of the GRF profile, revealing that the prismatic spine robots have a lower peak force and a longer contact time compared to the other spine morphologies. 69

5.9 Graph showing the average cost of transport for each robot. 69

6.1 An overview of the research covered in this chapter. 72

6.2 Graphs showing the horizontal and vertical position and velocity for the long-time-horizon task. 74

6.3 The foot touch-down position with respect to the neutral point plotted for the three phases of motion. 75

6.4 The rotated GRF profile for the four different velocities and five seed points. . . 76

6.5 The rotated GRF of the optimal seed for each of the four steady-state velocities. 76

6.6 The rotated GRF for the acceleration and deceleration trajectories being stitched to the steady-state step. 77

6.7 The robot being developed at the University of Cape Town by Joshua Van Zyl under supervision of A/Prof. Patel. 77

6.8 A comparison of the reduced order model and real data. 79

6.9 The generalised coordinates of the robot are shown. 79

6.10 The state machine that was enforced for the acceleration, steady-state and deceleration trajectories. 80

6.11 The horizontal and vertical position and velocity are shown for the two steady-state velocities. 82

6.12 The foot touch-down position with respect to the neutral point is shown for the acceleration, steady-state and deceleration trajectories. 83

6.13 Three PD controllers for the three phases of motion were designed using LSQ methods. 84

6.14 Trends emerged throughout for the acceleration, steady-state and deceleration trajectories. 85

6.15 The horizontal and vertical position and velocity are shown for the stitched result and simulation result. 87

6.16 A comparison of the GRF from the stitched result and simulation result. 88

6.17 The foot touch-down position with respect to the neutral point is shown for the simulation results. 88

6.18 A comparison of the stitched result, simulated result and disturbances. 89

6.19 A comparison of the limit cycles for the simulated trajectory and the disturbance trajectory. 90

6.20 A comparison of the stitched result, simulated result and disturbances. 90

6.21 A comparison of the disturbance trajectory to the simulated results. 91

6.22 An image taken of the monopod attached to the mechanical boom providing support in the sagittal plane. 91

6.23 A comparison between the horizontal position and velocity of the simulation results versus the physical experiments are presented. 92

6.24 Static images of a contact event taken while the robot was performing the long-time-horizon task. 92

C.1 Notice how the robot achieved an average vertical height of 0.35 m, with the leg fully compressing during the stance phase. 114

C.2 Notice how the robot achieved an average vertical height of 0.3 m, with the leg not fully compressing during the stance phase. 114

C.3 Experimental results of the fixed body forward tests. 115

List of Tables

2.1 Leg Morphology Results [1] 17

3.1 Robot and Boom Parameters 50

5.1 Design Space 64

5.2 Bootstrap Results (Minimum time results) 65

5.3 t-test results 66

A.1 Leg Morphology Bounds 109

Chapter 1

Introduction

For the past few decades, the robotics community has been developing high speed legged robotic platforms with the main focus on steady-state (periodic) and energy efficient locomotion. Little attention has been given to investigating how these robots should accelerate to these steady-state gaits, how to design these agile robots that are capable of performing these transient manoeuvres and what the controller morphology of these agile robots should be. This is mainly due to the complexities involved with these transient motions (aperiodic), along with modelling these systems (hybrid dynamics). Further, once these problems have been solved, how do you control such a complex and non-linear system to perform these highly agile manoeuvres? These questions will be addressed in this research with a rough overview of the dissertation shown in Figure 1.1.

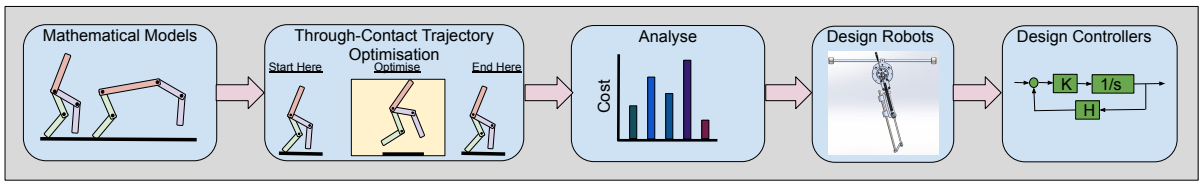

Figure 1.1: An overview of the research covered in this dissertation. Trajectory optimisation methods were employed to first determine how legged robots should accelerate and decelerate. Then investigation into the optimal spine morphology for quadruped robots to perform these acceleration and deceleration manoeuvres was performed. Finally trajectory optimisation methods were used to identify heuristics and trends to develop controllers for a monopod robot that was capable of performing a long-time-horizon task (start and end at rest while travelling a fixed distance), which was experimentally validated.

1.1 Motivation and Background to the Study

The effortless agility exhibited in nature is vastly superior to the current state-of-the-art robotic platforms. Current platforms, like Atlas [2], are pushing the limits of robotics and can perform agile manoeuvres such as running, jumping, rolling and even Parkour [3]. Even with these

advances, walking robustly in robotics is still considered a complex and unsolved problem. Many researchers apply assumptions, such as massless legs [4, 5], infinite friction [4, 6] and simplified reduced order models [7], in order to make designing controllers a more tractable task for their robotic platforms. The problem with many of these assumptions is that they have only been tested on steady-state (periodic) non-agile manoeuvres, however, they may not hold for rapid transient manoeuvres, limiting their feasibility for future developments.

As it is not well understood how robots should accelerate, further complexities arise. Looking at nature, animals jump straight to their desired gait with their feet often slipping [8], however, the robotics literature seems to suggest performing multiple intermediate gait transitions with no slipping [9, 10, 11]. A further comparison to nature reveals that quadruped animals use active spines, but robots seldom do, often making use of passive spines [12, 13, 14, 15, 16], with a few active spines [17, 18, 19, 20, 21, 22, 23]. Therefore, can optimisation methods be used to determine how legged robots should accelerate and what the optimal spine morphology for these manoeuvres is?

Throughout this research, a number of different robot models were used. Each of these models have a number of complexities that need to be addressed. These include:

1. Non-linear dynamics
2. Hybrid dynamics (equations of motion change if the robot is aerial, rear foot on ground or front foot on ground etc.)
3. Multiple input, multiple output systems
4. Under-actuation
5. Possible sliding of contact points
6. Unknown foot contact order

Due to these complications, it is proposed to use trajectory optimisation methods to aid in the design and control of these legged platforms. However, using trajectory optimisation brings in further complexities that need to be addressed: what should the performance metric be for rapid acceleration? For steady-state steps it makes intuitive sense to analyse the performance of the step by looking at the energy used (with a cost of transport metric and an enforced average velocity). However, with transient motions, there is no intuitive cost function to use. Some suggest minimum time, which can result in damage being done to the robot as the motions tend to be rapid and jerky [24]. In this work, the Author argues that energy based cost functions can be used to control rapidity with judicious utilisation of time bounds, along with terminal conditions to enforce the desired motion.

1.2 Research Questions

The current robotics literature is focused on steady-state locomotion and energy efficiency, with limited focus on transient manoeuvres. Currently there are a number of platforms that are pushing the limits of top speed achieved, however, they have no means to rapidly accelerate to these speeds or to decelerate to rest once at these speeds.

Using trajectory optimisation methods, it is possible to bridge the gap between pure simulation based studies (requiring high fidelity models) and physical platforms. The outcome of this research will serve as a guide for future legged robots and will hopefully shift the focus from steady-state energy efficient locomotion towards these more complex transient manoeuvres, which will drastically increase the agility of future legged robotic platforms. This has lead to the following research questions that were asked and answered during the study:

1. How should legged robots accelerate and decelerate? Do multiple intermediate gaits matter or should the robots jump straight to the final gait? This will be explored in Chapter 4.
2. What is the optimal spine morphology for quadruped robots performing rapid transient manoeuvres such as acceleration and deceleration? This will be explored in Chapter 5.
3. Can trajectory optimisation be used to inspire mechanical and controller design? This will be explored in Chapter 6.

1.3 Scope and Limitations

The scope of this research involves firstly determining how legged robots should accelerate and decelerate, along with the identification of a template for these manoeuvres. Secondly, investigation into determining the optimal spine morphology (using Monte Carlo methods) for rapidly accelerating and decelerating quadruped robots. Finally controllers were developed using trajectory optimisation methods. These were first tested in simulation (using trajectory optimisation methods) and then on a monopod legged robot.

Due to the complexity of these manoeuvres, the trajectory optimisation methods had a number of limitations. The number of nodes, robots compared and seed points used were kept as small as possible to keep the study feasible but still useful (Monte Carlo simulations took three months to complete). Through-contact optimisation methods were used with a coefficient of friction of 1 ($\mu = 1$). Controllers were first developed in simulation and were then tested on the platform (which was developed in collaboration with an MSc student, Joshua van Zyl, at the University of Cape Town).

This research only considered the planar case with the physical platform constrained to the sagittal plane by a support rig, in the form of a boom. To simplify the design, all power (batteries and pneumatic air compressors) where kept off-board, reducing the weight and design complexity of the robot. Boom sensors were also used to determine the position states of the robot.

1.4 Plan of Development

The dissertation begins with a thorough literature review section that examines the latest research in legged robots and is presented in Chapter 2. Special focus is on agile legged robots as well as trajectory optimisation methods, templates and controller design.

The methodology, presented in Chapter 3, details the trajectory optimisation methods used throughout this research and details how the Monte Carlo simulations were performed. Special focus is given to how the system will be tested (in simulation and on a physical platform) and validated.

Chapter 4 gives insight into how legged robots should accelerate and decelerate and provides a template that can be used to plan these manoeuvres. Once it was determined how robots should accelerate, investigation into determining the optimal spine morphology for rapidly accelerating quadruped robots was conducted and is detailed in Chapter 5.

Chapter 6 details how trajectory optimisation was utilised to design controllers for a monopod robot. These controllers were first tested in simulation and then on the physical platform. The dissertation then ends with the conclusions presented in Chapter 7. The layout of the dissertation is summarised in Figure 1.2.

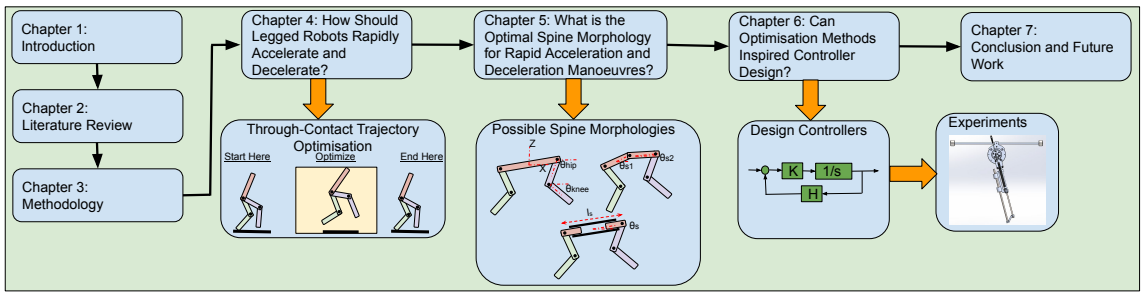

Figure 1.2: The dissertation starts with a brief introduction, followed by the relevant literature and theory development in Chapter 2. The methodology is described in greater detail in Chapter 3. The optimal method to accelerate is detailed in Chapter 4 followed by investigating the optimal spine morphology for quadrupeds in Chapter 5. Control algorithms were developed and implemented in Chapter 6. Finally the conclusion is presented in Chapter 7.

1.5 Publications

The following publications form part of this research:

1. C. Fisher, S. Shield and A. Patel, "The Effect of Spine Morphology on Rapid Acceleration in Quadruped Robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

2. C. Fisher, C. Hubicki and A. Patel, "Do Intermediate Gaits Matter When Rapidly Accelerating?" *IEEE Robotics and Automation Letters*, 4(4), 2019.

3. C. Fisher, J. van Zyl, R. Govender and A. Patel, "Optimisation-Inspired Controller Design for Transient Legged Locomotion," *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

4. C. Fisher and A. Patel, "On the Optimal Spine Morphology for Rapid Manoeuvres in Quadruped Robots," *SAIEE Africa Research Journal*, *Under Review*, 2020.

Chapter 2

Literature Review

In nature, some animals learn to walk and run within hours of being born [25], however, the precise determinants of locomotion are still under debate [26], making it challenging to replicate these motions in legged robotics. Therefore, it is no surprise that legged robotic locomotion is largely considered an unsolved problem, with only a handful of legged robots capable of performing unaided, dynamic locomotion. The majority of legged platforms are restricted to the laboratory environment with the focus remaining on steady-state locomotion and the aims of improving efficiency and robustness (for example robots such as ATRIAS, Mabel and ScarlETH, all detailed below). Very little focus has been given to investigating how to accelerate to, and stop from these steady-state manoeuvres. For a robot to be successfully deployed in the field, it must be capable of performing dynamic and transient motions. These transient motions (rapid acceleration and deceleration) are the focal point of this research, as they will drastically improve the ability of these legged platforms to explore the unpredictable and rough terrain found outside of the laboratory environment. They also aid in performing time-critical tasks (such as search and rescue missions).

Looking at the state-of-the-art platforms, there have been some phenomenal developments over the past decade that have enabled legged robots, such as Atlas [2], to perform dynamic Parkour manoeuvres [3] (see Figure 2.1) that involve complex terrain and motions such as running, jumping and rolling. These are, however, the exception, as most platforms struggle to walk/run without the assistance of a support rig or boom system.

Studying nature and the current robotic platforms, there seems to be three aspects influencing manoeuvrability. These include:

1. How do animals perform these transient motions?
2. What is the optimal mechanical morphology for transient locomotion?

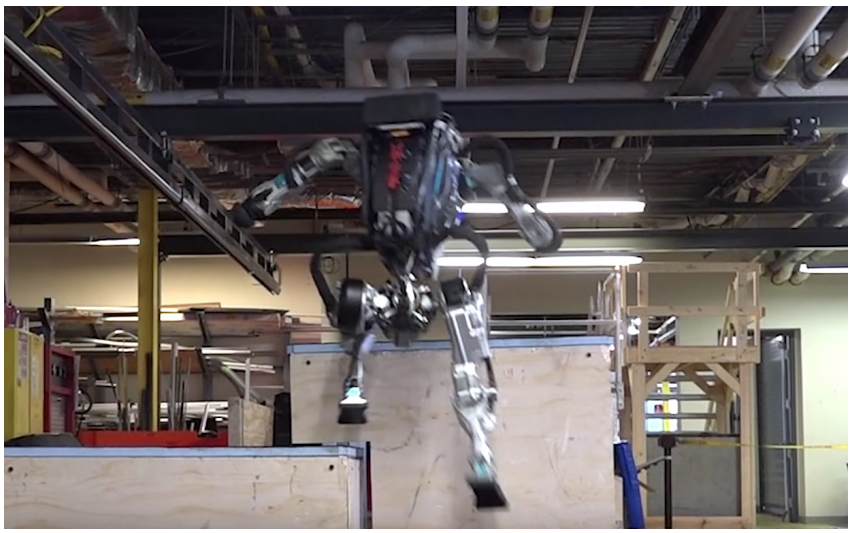

Figure 2.1: An action shot showing the state-of-the-art robot, Atlas, from Boston Dynamics performing a Parkour Manoeuvre. Note how there are no support rigs with all power and control performed on-board in real time. Image from [27].

3. Can we develop controllers to perform these agile motions?

The available literature was explored to provide the current view and developments done on these three key aspects:

2.1 Aspect 1: How do Animals Perform Transient Motions?

Looking at the current state-of-the-art robotic platforms, the majority of the robotics literature is focused on steady-state (periodic) locomotion [4, 6, 7, 28, 29, 30]. In order for legged robots to successfully leave the confines of the laboratory and perform similarly complex manoeuvres exhibited by animals in challenging and unknown environments, we need to fully understand how these animals move. As legged systems experience intermediate contacts with the ground, as well as the environment, it becomes critical to be able to manage contact transitions [31]. These contacts and footholds must be studied and exploited to achieve the desired dynamic task [32] as well as recover from disturbances. Ground contacts often result in highly non-linear and impulsive forces that are numerically challenging to solve in the current methods employed in the robotics literature (more details in Section 2.4). To date, a number of different gaits have been achieved in the robotics literature:

1. Galloping (example literature: [33, 34, 35, 36])
2. Bounding (example literature: [34, 35, 37])
3. Trot (example literature: [33, 36])

4. Pronk (example literature: [33])
5. Walking (most platforms)
6. Running (bipedal) (example literature: [30, 38])

These challenging environments require accurate foot placement, with a number of platforms achieving this through the use of heuristics [33] inspired by intuition or observations made in nature. Another approach is to use hybrid dynamics, along with assumptions such as no slipping (as done on Cassie [7] and ATRIAS [38]). Hybrid dynamics requires *a priori* knowledge of the contact order, which is not feasible for transient manoeuvres that have an unknown contact order. Hybrid dynamics has seen wide use in robots such as ATRIAS [30], MIT Cheetah 2 [39] and Cassie [7]. These models include continuous dynamics for the stance and aerial phases, which are linked by discontinuous events such as touch-down and lift-off, due to foot impacts with the ground. Cassie modelled the contacts as holonomic constraints, and implemented a contact detection algorithm, which analysed the implicit leg force, and if it was above a threshold for 5 ms, then a contact was assumed [7].

On the other hand, robots such as Scout II [34] made use of a reduced-order model where the leg was modelled as two point masses connected by a spring. With the use of hybrid dynamics, multiple simplified models of the Scout II robot were implemented and compared (models consisted of one rear leg, torso and one front leg, modelling the quadruped as a planar robot). It was found that saturation and non rigid torque transition characteristics were required in the model to generate accurate results.

Comparing this to nature, animals are often observed to frequently slip while performing these agile manoeuvres [8] and utilise an unknown foot contact order (unlike periodic gaits that have a pre-defined foot contact order). Studying these rapid manoeuvres, the impulsive contacts seem to result in the muscles applying forces that show signs of "bang-bang" like actuation, or at least attempting to push off the ground as hard as possible [40].

Investigation into arboreal animals has revealed some insight into how they achieve their explosive power to achieve these leaps. Through the use of power modulation, they can increase the available peak power to a value higher than the instantaneous power developed by the muscle alone [41] [1]. This is achieved by transferring the energy from a muscle into a series elastic structure, then releasing the stored potential energy as kinetic energy [41]. This explosive release of power, shows signs of "bang-bang" control. In robotics, it has been determined that explosive leg force is required for dynamic and agile manoeuvres [42]. These manoeuvres characterised by an impulsive change in velocity, requiring a large explosive force. In robotics, actuator power is a limiting factor, resulting in a large number of platforms utilising SEAs.

Hubicki et al. [43] generated a large number of long-time-horizon trajectories of a simple

[1] A similar concept in robotics is series elastic actuators, SEA [42]

monopod robot with a spring leg to determined if near limit cycle behaviour emerged. Further analysis of phase plots revealed indications of "bang-bang" like acceleration [43]. These motions are commonly attributed to the minimum time control of a sliding mass. This means that the task optimal forward velocity of the model during transient motions, such as acceleration and deceleration, matches that of a simple 1 DoF model of a sliding mass [43]. It was hypothesised that by using the sliding mass template, a computationally simple and near optimal locomotion planning path can be generated in real time. This model was only validated on a simple monopod model, however, it was hypothesised that it would hold for more complex models.

A common trend in the literature is to take a biologically-inspired approach to studying locomotion in order to identify heuristics which can be used to design controllers to improve the performance of our robotic systems in terms of agility. However, the available research on manoeuvrability in animals is very limited, with the methods used to achieve these feats of manoeuvrability still being considered as unknown and up for debate. A large portion of the available research is focused on steady-state (periodic) motion.

One study of particular interest, investigated how humans moved from point A to B under time constraints [44]. By enforcing a time constraint, an average velocity was enforced without enforcing the speed at any given moment. For this experiment, each participant was given a stop-watch, and told to arrive at point B in a fixed time (they could not arrive early or late). For large times, the subjects walked the entire distance, while for short times they ran the whole way (as expected).

The results become more applicable for intermediate time periods. Subjects were seen to use a mixture of walking and running. It was observed that they walked at slow speeds (where it is more energetically optimal) and ran at high speeds (again more energetically optimal), and avoided intermediate speeds (such as a fast walk or slow run/jog). Using a metabolic cost model, similar results were achieved using optimisation methods. When adding a gait transition cost (for example a cost to change from walking to running), the resulting optimal trajectory had only one gait transition, with no intermediate gaits/velocities. However, when looking at the robotics literature, conflicting results emerge, with some researchers suggest accelerating by implementing multiple gait transitions [10] while others suggest hopping in the correct foot fall pattern and then slowly accelerate [9, 11], which does not match observations found in nature.

In [45] gait switching was performed on a hexapod robot, making use of a feedforward open loop approach. A number of strategies were implemented, the first being if the two gaits overlap, then a transition can be performed at the intersection. The second approach was a more complex approach that involved switching one leg at a time, with special attention to ensure the robot remained in a valid configuration and did not fall over. These methods were tested on two different hexapod robots. This research did not consider rapid transitions from rest, and only focused on intermediate gait transitions for switching tasks (for example RHex switched from the tripod gait to the stair climbing gait).

Other studies have shown that steady-state locomotion may not be optimal, if not implemented at the correct velocity. Specifically, the literature suggests performing a mixture of gaits (walking and running) to achieve the net desired velocity, instead of running at the desired velocity which may be energetically expensive (not optimal) [46]. Therefore, these trajectories only require two optimal gaits and transitions between them. One must note that these experiments were done without considering fatigue and standing stationary [44]. It is evident that the energy-optimal method (in the mathematical sense) to rapidly move from point A to B is still unknown.

A recent trend worth studying is the sport Parkour, where humans navigate from point A to B in the most elaborate ways possible. These often involve leaps for foot/hand holds between walls, banisters and railings. Similar dynamic motions are exhibited by arboreal animals (see Figure 2.2) that leap between branches and mountain goats which are observed to bound up cliff faces [41]. From observing these animals, it is clear that by maximising their leap distances, the number of potential available footholds is maximised, making the navigation of these complex terrains simpler [41]. For complex motions like these, defining the contact order *a priori* is not feasible as slipping will occur, resulting in the animal having to adapt in real time. This sport has gained traction in the robotics literature and has been performed by Atlas [27] and RHex [47].

Figure 2.2: An image of the Dodinga Hills Guereza, which is a subspecies of monkey, effortlessly leaping through the air. Image from [48].

Robots like ATRIAS and ScarlETH were focused on energy efficient locomotion, with the actuators on ScarlETH only providing positive power, and never removing energy from the system [49]. ATRIAS on the other hand focused on energy efficiency and could smoothly transition from rest to a running gait of 2.5 m/s [30]. Due to the dynamics being designed around the Spring Loaded Inverted Pendulum (SLIP) model (see Figure 2.3), the robot was capable of attenuating disturbances as well as recycling energy from step to step [38].

Other platforms make use of reduced order models to simplify their dynamics. Robots, such as Cheetah Mini [51] and MIT Cheetah 3 [33], used reduced order models to implement MPC

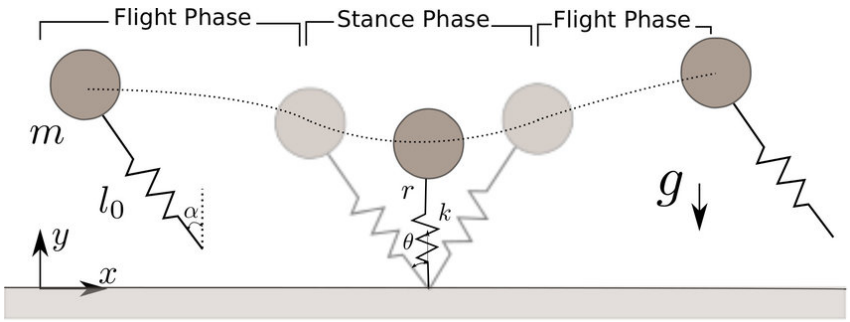

Figure 2.3: An example of the Spring Loaded Inverted Pendulum (SLIP) template. This reduced order model accurately predicts the CoM motion of a number of animals. Image from [50].

controllers. The MIT Cheetah 3 modelled the robot as a single rigid body subject to forces at the contact patches, and ignored the dynamics of the legs. An example of these simplified models is shown in Figure 2.4.

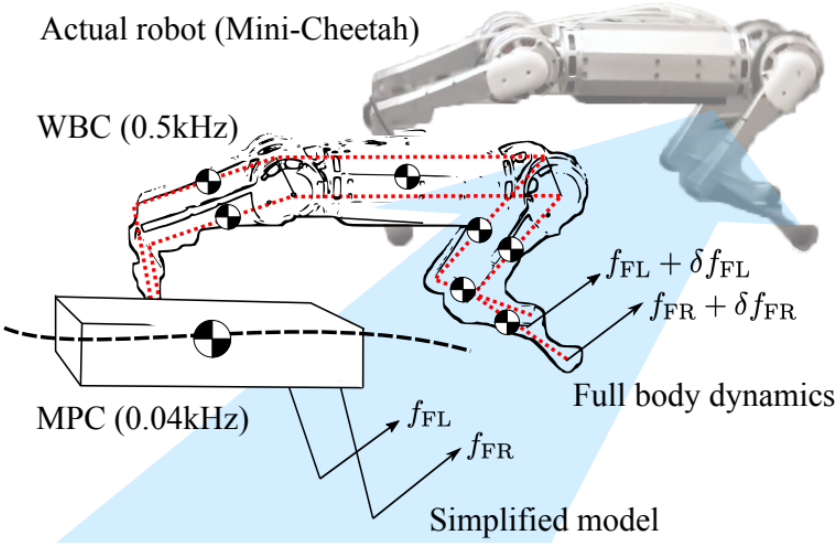

Figure 2.4: An example of a reduced order model for the MIT cheetah robot. Note how the bottom model ignores the dynamics of the legs and is used for the MPC controller. The middle model is more accurate and includes the whole body dynamics. Image from [51].

Through insight from these reduced order models, a number of stable locomotion behaviours were generated for the ATRIAS robot [38]. This revealed that robust $3D$ bipedal walking and running on rough unstructured terrain was possible using a passive dynamics approach. Other platforms such as the MIT Cheetah 2 [39] made use of reduced models to generate limit cycles in simulation which were later validated on the platform.

Rapid acceleration and deceleration is vital for survival in nature, however, there are a number of constraints to the maximum achievable acceleration/deceleration [52]. Studies on Polo ponies and racing greyhounds have determined that the factors that limit the maximum possible acceleration/deceleration include a mechanical and a physiological constraint [52].

The limiting factor is dependant on the speed of the animal. At low speeds, it was found that

the maximum acceleration possible was limited by geometric constraints, such as over pitching [52]. During acceleration, the nose of the animal pitches up (nose up) and during deceleration, the tail pitches up (tail up). Animals have adapted to achieve these motions by throwing their bodies backwards during rapid deceleration manoeuvres, often skidding to rest as seen in Figure 2.5. If acceleration/deceleration increases, the nose up/tail up condition will cause the animal to become unstable and topple over. At higher speeds, the available muscle power becomes the limiting factor. This study focuses on a model which only considered a rigid spine. Due to the limited available muscle power, animals use power modulation to increase the available peak power to achieve these advanced locomotion tasks [41].

Figure 2.5: Animals, such as horses, have adapted to handle these rapid deceleration manoeuvres by throwing their bodies backwards to avoid pitching forward. Image from [53].

It was noted in [52] that other factors, such as traction, could also limit the available acceleration limits, however, slipping was not observed in their study. Watching slow motion videos of cheetahs and greyhounds accelerating, it is obvious that their feet often slip during these manoeuvres [8], making it important to allow for slipping in simulation based studies.

2.2 Aspect 2: Optimal Morphology for Transient Locomotion

The state-of-the-art legged platforms apply a number of different actuators and mechanical designs in order to achieve their desired task. If dynamic locomotion is going to be investigated for legged robots, careful attention must be given to the actuator selection, as these motions pose a number of demanding requirements [54, 55]. Some of these requirements include:

1. High impact robustness
2. Fast motion tracking
3. Low impedance

4. Force controllability
5. High torque
6. Light weight

A number of actuators have been used such as:

1. Pneumatic (example literature: [32, 56])
2. Hydraulic (example literature: [55, 57, 58])
3. Direct drive electric (example literature: [36, 59])
4. Geared electric (example literature: geared [60] or belt/cable driven [49, 55, 61])
5. SEA (example literature: [7, 30, 49, 61, 62])

Each actuator has its advantages and disadvantages. In the $1980's$ pneumatics was very popular and some of the first agile legged robots were developed by Raibert [32, 56]. In his seminal research, Raibert demonstrated that dynamic motions can be achieved using simple model independent controllers. The Raibert controller was shown to work on monopods, bipeds and quadruped robots, each with a pneumatic leg (actuator controls the leg length) connected to the body through a hydraulic actuator (controls the leg angle) to generate hip torques [32].

In line with pneumatic actuators, hydraulic actuators have also been used in robotic platforms such as BigDog [57] and HyQ [55, 58]. BigDog was developed as a military robot to carry large loads over rough terrain. The robot weighed 90 kg and could carry a payload of 50 kg. On board was a two cylinder combustion engine providing 3000 PSI for the hydraulic actuators. Even with its weight, the robot could run at 5 m/s (and climb 45° slopes) and jump over a 1 m ditch thanks to the power of the hydraulic actuators. On the other hand, HyQ also made use of hydraulics (as well as electric actuators) and also weighed 90 kg, however, it could not carry the same load as BigDog. Each leg of HyQ weighed 4.5 kg with electric actuators to control the abduction/adduction of the leg.

Due to the high power-to-weight ratio of hydraulic and pneumatic actuators, as well as their ability to sustain large impact loads, makes them ideal for legged robotic platforms [55]. There are, however, large differences between these actuators. Due to the lack of compressibility of the oil in hydraulic systems, the cylinder response is stiff compared to pneumatic cylinders which operate using a compressible gas such as air or nitrogen. For hydraulic systems, the dynamic response strongly depends on the volumes inside the hydraulic tubing and the bulk modulus of the oil [55] taking into account the amount of trapped air in the tubes as well as the tubes elasticity. For the pneumatic cylinder, force is only applied once the air has suitably compressed, causing a delay in the response.

Compliance in actuators seems to play a crucial role in legged robotics, as well as to shield the motor and gearbox from impulsive loads experienced during touch-down events [34]. Some compliance has been added to platforms by using a belt or cable system to translate the motor torque to the joints [55]. Compliance has the negative effect of increasing the degree of underactuation, making it harder to control the system [63].

Other platforms such as Baleka [60] and Minitaur [59, 36] have opted for virtual compliance. Baleka made use of a low gear ratio (planetary gear box with a gear ratio of 5) making it quasi-direct drive, whereas Minitaur was a direct drive quadruped robot. Mabel [64] on the other hand made use of virtual constraints within the HZD formulation, along with force control to create a virtual compliant element, effectively allowing the leg stiffness to be varied. Mabel contained springs in its drive train to enhance both energy efficiency and agility [65].

Robots such as the MIT Cheetah 2 [37] and 3 [33] made use of proprioceptive actuators to mitigate high impacts [34] and control its GRF and leg stiffness, without the need of additional sensors or compliance in joints or feet [31]. Other platforms include compliance via additional springs. This has resulted in the development of series elastic actuators (SEAs), Figure 2.6.

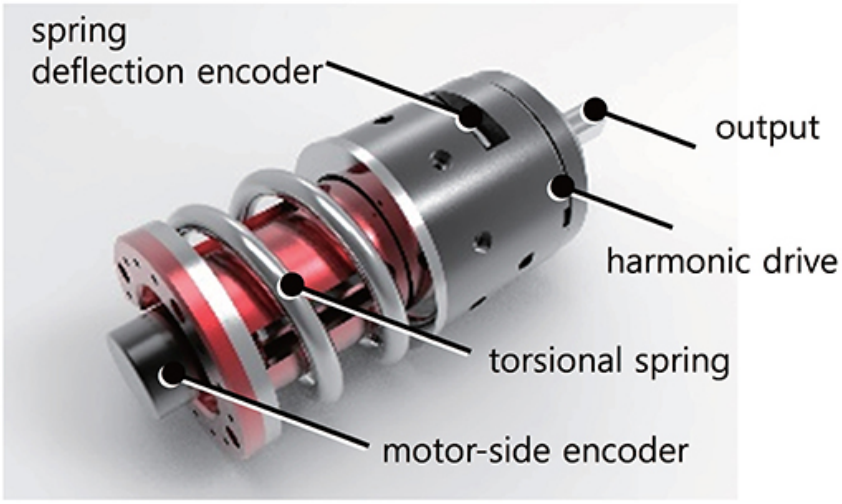

Figure 2.6: A typical Series Elastic Actuator. The actuator consists of a torsional spring, encoders to measure motor position and spring deflection and harmonic drive. Image from [66].

SEAs have become very popular in legged robotics as they are capable of achieving a higher peak power then standard actuators, along with an increase in efficiency [41, 49], alleviating the limiting factor of power-to-weight ratio. Another benefit of SEAs is that they protect the gearbox from the impulsive loads experienced during ground contacts [55]. However, one of the disadvantages of SEAs is that they reduce the control bandwidth of the actuator (control bandwidth is inversely proportional to the spring stiffness [60]). Furthermore, little attention has been given to selecting/tuning these parameters or for selecting a desired leg stiffness, with [67] selecting the stiffness to that of a similar mass quadruped animal.

Quadrupedal robots, such as ScarlETH [49] and AnyMAL [62], made use of SEAs, as well as

bipedal robots, such as ATRIAS [30, 61] and Cassie [7]. Cassie made use of leaf springs, whereas ATRIAS made use of custom built motors with a harmonic gear ratio of 50:1 [30]. ScarlETH had 30 Nm per joint which allows natural dynamics, as well as temporary storage of energy which improved passive adaptability [49]. AnyMAL had a highly integrated SEA with a novel joint design which enabled it to change its leg morphology from **X**-type to **O**-type [62], due to its range of motion.

With the large range of actuators to choose from, many research groups rely on trial and error to select the appropriate actuators. However, trajectory optimisation methods can be utilised to aid in actuator selection. In [68], a nonlinear optimisation process was developed that integrated controller design for GRF selection, as well as mechanical design to selected the motor/gearbox pair. Simulation, along with analytical methods, were used to select the actuator in [69] by including the task as well as motor thermal dynamics.

In [70] a novel computational approach was taken to optimising robot parameters along with fine tuning motion parameters. This ensured that the resulting morphology chosen for the robot enabled it to perform the desired motion. Results were applicable to a number of actuator types and was tested and validated on a number of robotic platforms. This method produced accurate results with less iterative design cycles.

The second crucial mechanical design choice in developing robotic platforms is selecting the appropriate leg configuration. There are two common leg morphologies that are implemented in legged robotic platforms. Robots such as Minitaur [36, 59], ATRIAS [2, 71] and Baleka [60] make use of the scissor type leg (four or five bar parallel link morphology), whereas others make use of a two link leg (series articulate morphology), generally with both motors at the hip, with a variety of methods to translate the torque to the knee joint. Some robots have opted to use some form of cable/chain pulley system to translate the torque, others have made use of a four/five bar linkage system [33].

Some of the key aspects to focus on when choosing a leg morphology include [60]:

1. Functional workspace
2. Force transparency (Impact Mitigation Factor [72])
3. Force amplification
4. Robustness
5. Load sharing between the actuators

According to [72], the force amplification and load sharing between actuators are related with the concept of a force ellipsoid. Often designers choose the five bar mechanism over the

four bar to reduce the width of the leg. The four bar mechanism requires the motors and gearboxes to be located co-axially, whereas the five bar allows motors to be located next to each other. In [60] a comparison between the load distribution between the parallel linkages (four and five bar morphologies) and the series articulate morphology was conducted. The parallel mechanisms had a significantly better load distribution. In numerous locations, the series articulate morphology barely used one of the actuators as seen in Figure 2.7, with the load distribution for the parallel linkages to be fairly evenly distributed.

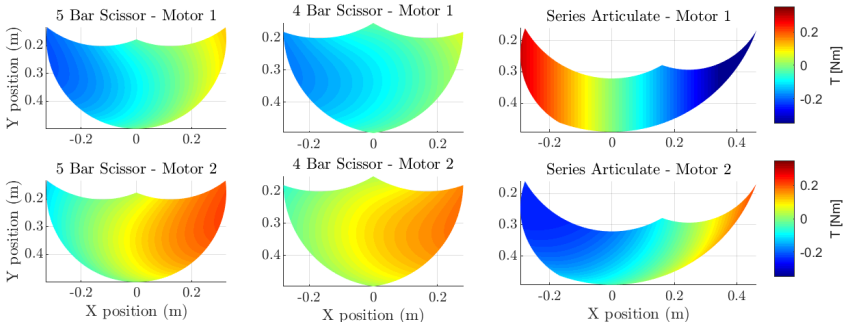

Figure 2.7: A comparison of the load distribution for each motor for the five-bar, four-bar and series articulate leg morphology. Image from [60].

Due to the complexity of a number of the robots, as well as the built in compliance, a number of approaches have been taken to model these platforms. Robots such as ScarlETH [49], ATRIAS [30, 61] and Cassie [7] make use of the SLIP template to model their dynamics. This simplification works as the CoG of the legs was near the hip, as the hip and knee actuators are located in the hip, making the legs lightweight (modelled as massless springy legs).

A large-scale Monte Carlo simulation study [73] was performed to determine if SLIP dynamics were energetically optimal. This was achieved by embedding the SLIP dynamics (and Raibert controllers) into a three link monopod and optimising for energy efficiency, comparing the results to optimisations without the enforced SLIP dynamics. 100 robots were randomly generated from a design space, with 50 seed points optimised for each case. It was found that the SLIP-Raibert approach required almost twice as much energy to transverse a given distance.

Another study investigated energy efficiency during a periodic step with regards to the leg bend direction for bipedal robots [74]. Comparing ostriches to humans, one can notice the difference in leg bend direction, however, biological studies cannot determine which morphology is optimal for robotics due to the vast differences between the biological and engineering domains. Using large-scale Monte Carlo simulations, the effects of leg bend direction on gait efficiency was determined. it was found that, for a large variety of bipedal robots, that the optimal leg bend direction was backwards. In total 10 000 optimisation attempts were executed for each bend direction, with a convergence rate of 9.3% for the forward bending leg and 11.6% for the backwards bending leg.

A previous study that the Author was involved in, investigated a long-time-horizon task (start and end at rest, while travelling 30 body lengths) on legged planar quadruped robots to

determine the optimal leg bend direction [1]. For this research, four leg morphologies were considered:

1. Knees facing forward
2. Knees facing backwards
3. Knees facing inwards (forming an **X**)
4. Knees facing outwards (forming an **O**)

In total, 100 sets of robot parameters were generated at random (for each set of robot parameters, four robots were generated, each with a different leg configuration). Large-scale Monte Carlo simulations were performed on the long-time-horizon task. These simulations made use of through-contact methods, allowing the optimiser to determine the contact order as well as 3-point collocation with 300 node points [1]. In total, a convergence rate of 17.8 % was achieved with the results shown in Table 2.1.

Table 2.1: Leg Morphology Results [1]

Config	Cost	Acceleration	Deceleration
Forward	$(16.67 \pm 5.8)\%$	$(28.00 \pm 6.3)\%$	$(7.14 \pm 5.2)\%$
Rear	$(2.38 \pm 2.4)\%$	$(13.99 \pm 4.9)\%$	$(45.24 \pm 7.0)\%$
O-type	$(0.0 \pm 0.0)\%$	$(22.01 \pm 5.9)\%$	$(9.52 \pm 4.9)\%$
X-type	$(80.95 \pm 6.1)\%$	$(35.99 \pm 6.8)\%$	$(38.10 \pm 6.5)\%$

The third design consideration is only applicable to quadruped robots. Animals exploit spine actuation during high speed locomotion (such as turning, acceleration and deceleration), which is paramount to survival during hunting and evading predators [75]. The flexible spine has been shown to be fundamental for these manoeuvres [76] and it was noted that not all speed is manifested in the legs of animals. Hildebrand observed that the spine of the cheetah increased the stride-length by 5 % which resulted in an overall increase in the top speed by 10 % [77, 78]. This shows that multiple aspects work together to generate the superior agility observed in nature.

The spine has only recently been incorporated into robotic platforms, with a number of novel spine designs being utilised. These have generally been focused on steady-state locomotion and energy efficient manoeuvres [22, 79]. Studies have not investigated the effect of spine motion on transient manoeuvres such as acceleration and deceleration. Nor has the optimal spine morphology for these manoeuvres been investigated. This is partly due to the complexities in studying and modelling these motions.

As the Author was only focused on planar motion, only planar 1 DoF spine morphologies were considered. Flexible spines are frequently found in nature in many different shapes and sizes with different degrees of flexibility, however, these cannot be directly applied to robotics due

to a number of complexities. A bio-inspired approach was taken where design principles were extracted from nature, transferred to the engineering domain and applied to robotics (some state-of-the-art quadruped robots are shown in Figure 2.8). Looking at nature, flexible spines (consisting of a large number of actuated revolute joints connecting rigid vertebrae) are often used. However, these are infeasible in high speed agile robotics due to their complexity in mathematical modelling of the system, designing a suitable controller and due to the weight from the large number of actuators required.

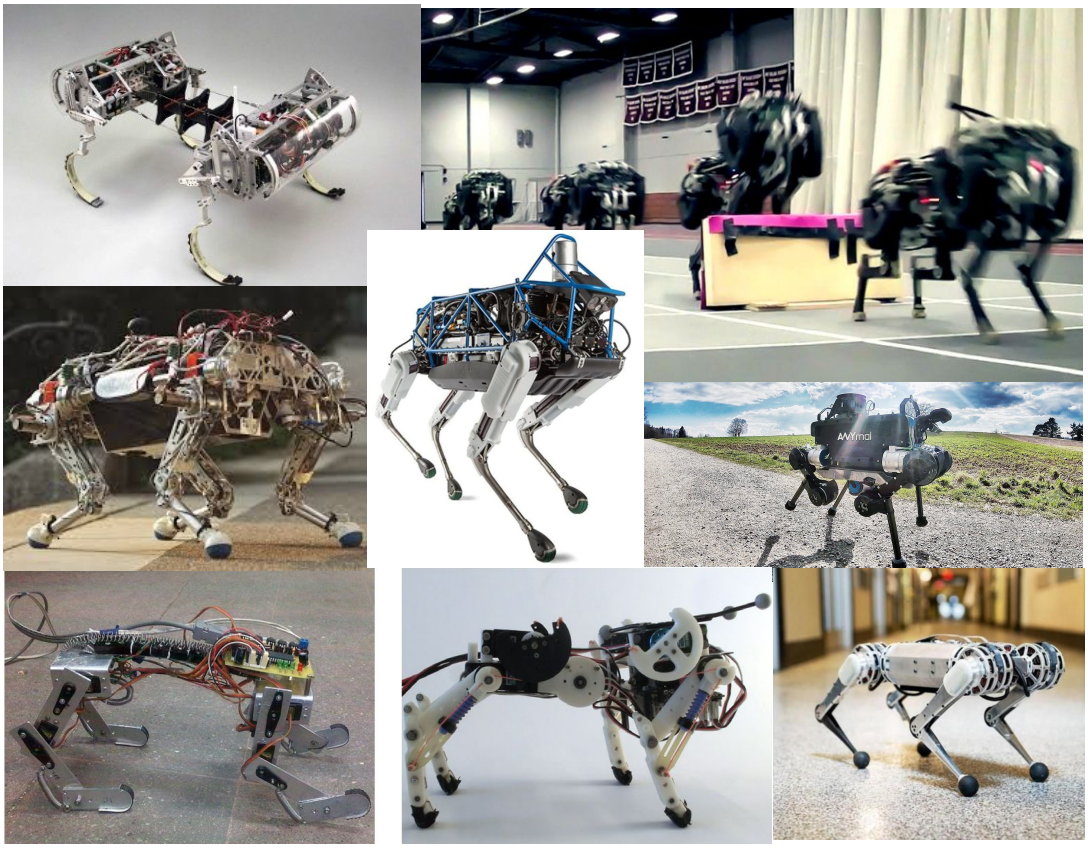

Figure 2.8: A number of state-of-the-art quadruped robots are shown. Images from (left to right, top to bottom): [80] [81] [82] [83] [84] [85] [85] [86].

There are thus two ways to realize a flexible spine in robots. The first is to reduce the degrees of freedom of the spine to one, resulting in the revolute spine. The spine will still encompass the bending nature of the flexible spine seen in nature, however, it is feasible for robotics as it requires only one actuator and two rigid links. Currently there are a number of robots that utilise a revolute spine: [12, 14, 15, 20].

The second method is to study the resulting motion of the flexible spine in nature to determine what the overall effect of the spine is. The flexible spine has been shown to increase the stride-length [77, 78] by pulling its legs in when the front foot is on the floor and pushing out the legs when the rear leg is on the floor. When this is converted into the engineering domain, a prismatic spine emerges. This is feasible in robotics as there are no internal organs restricting the motion. Examples of this spine morphology in the literature can be found here: [13, 87].

2.3 Aspect 3: Can we Control these Transient Motions?

Legged robots are inherently under-actuated, making controlling them a challenging task in the presence of intermittent contacts with the ground [33]. This is commonly solved using reduced order models and templates, making controller design more tractable [61]. One such template, which was inspired from heuristic knowledge, is the SLIP (Spring Loaded Inverted Pendulum) template, that accurately describes the CoM position during steady-state locomotion of many legged animals [61]. These templates have been extended to 3D and have been implemented for turning, however, the majority of this work was done in simulation and the turns were not aggressive [88, 89].

Using these reduced order models to design controllers for robotic systems generally relies on heuristics and often suffers from robustness issues [90], due to no formal link between the template and whole-body system [91]. In [92], they suggest designing initial controllers on the reduced order models and to validate/tune them on the more complex robot model before experimentally testing them. Controllers designed for these complex whole-body systems rely on accurate and complex models that are inherently computationally expensive, resulting in robust agility out of the reach of most robotic platforms.

One problem with heuristics is being able to identify and extract them from manoeuvres. In [93] they developed a data driven approach to designing and extracting these heuristics. This enabled them to avoid having to design meaningful cost functions for high fidelity models to generate the desired motions. Instead, they embedded the heuristic into the cost function, biasing the optimiser to the desired gait, while still allowing it to explore the search space for a more optimal result. This method allowed for the use of less accurate models and dynamics, with result being tested and validated on the Mini Cheetah robot.

A common observation made by a number of researchers, notably Raibert, suggests that simple controllers (for example the Raibert controller, Figure 2.9), can generate stable dynamic locomotion in legged robotics using simple model independent controllers [32]. A wide range of dynamic locomotion was achieved on a number of different platforms such as monopods, bipeds and quadrupeds [32]. These controllers stabilise the body when the foot is in contact with the ground, while the foot is aerial, the leg is rotated into the correct position for the next touch-down phase. It has been hypothesised that these simple controllers can be modified to achieve rapid acceleration and deceleration manoeuvres.

Since the development of the Raibert controller, it has been used in a wide range of platforms in one form or another, and has been said to be the most famous example of insight based bipedal control [38]. Robots such as ScarETH [49] and Scout II [34], made use of the Raibert controller for the flight phase (selecting the correct touch-down angle), with ATRIAS's controller showing clear signs of Raibert control. Studies on humans performing a number of tasks, such

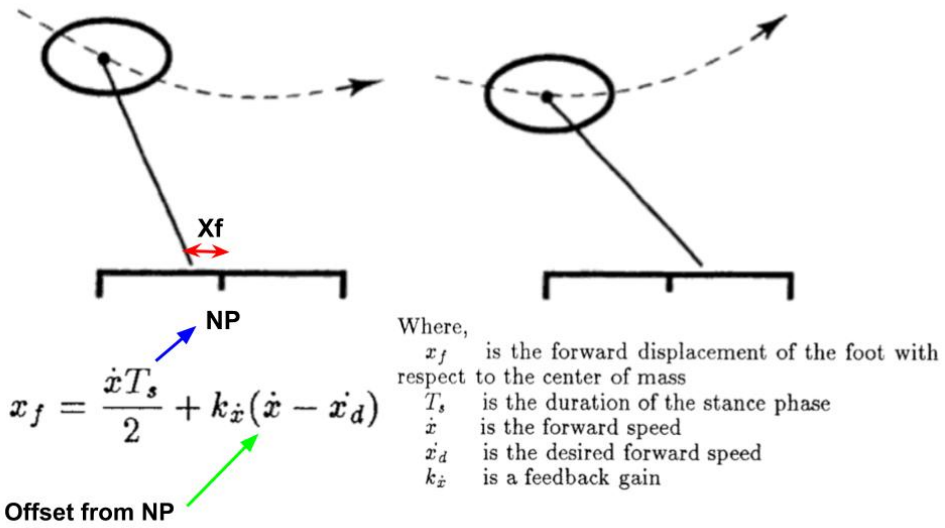

Figure 2.9: A modified image from [94]. Image shows how the trajectory changes depending on the position of the foot. If the foot was placed behind the neutral point (NP, as seen on the left), the robot will tend to accelerate forward, whereas if the foot was placed in front of the neutral point, the robot will tend to decelerate. The NP is calculated using the forward velocity, and estimates of the stance time. The offset of the foot is calculated using a gain multiplied by the difference between the current velocity and the desired velocity.

as acceleration, deceleration and constant speed running, revealed signs of a Raibert controller, in terms of foot touch-down position relative to the neutral point [95].

To date, limit cycles have been fundamental in the control and analysis of these complex legged systems. Hubicki et al. [43], used trajectory optimisation to determine if limit cycles naturally emerged from optimal trajectory optimisation solutions from complex tasks that involved long-time-horizons as well as transients, such as acceleration and deceleration with added disturbances. For multiple tasks, optimal results often approached, or exactly matched limit cycles, proving they can be used as a locally optimal planning path.

A number of simulation studies have been conducted on a variety of non-periodic manoeuvres such as kicks and jumps [96]. In [96] a 26 DoF simulated biped (using DynaMechs) was shown to perform these manoeuvres using real time conic optimisation to select the appropriate joint torques and contact forces. Each motion (kick or jump) was characterized by an appropriate task space, which was solved by the optimiser, with user defined priorities between the different tasks.

In [89] a 3D-SLIP model was utilised to perform turning manoeuvres of a simulation of a 26 DoF model of a human. Controllers were developed to transition from periodic running into a turn. Robots such as the MIT Cheetah 3 also investigated turning, with yaw rates of up to $180°/s$, as well as being able to climb stairs [33]. The Oncilla robot [97] was used to investigated six different turning strategies:

1. Varying step size with stance time remaining constant
2. Varying the step speed with a constant step size (failed on the robot)
3. Varying the step size with a constant step speed (failed on the robot)
4. Varying motor torque
5. Making use of abduction and adduction dynamically

MPC has also seen significant use on legged platforms. The MIT Cheetah 3 had general balance and locomotion controllers embedded into the modular control architecture which enabled the robot to handle terrain disturbances through reactive gait modifications [37]. MPC was implemented to determine the GRF by formulating the simplified dynamics as a convex optimisation problem with a 0.5 second horizon that solved in under 1 ms [33]. Experimental results showed the controller was capable of stabilising a variety of different gaits using the same gains and weights. Outputs from the MPC controller were used to calculate the joint torques and to stabilise the robot without the need of additional feedback controllers.

The Mini Cheetah implemented MPC as well as impulse control [51]. The MPC found the optimal reaction force profile over a long-time-horizon using the simplified model of the robot. Using the outputs of the MPC controller, the impulse controller calculated the required joint torques. The MIT Cheetah 2 also implemented impulse control for feedforward force profiles to perform the bounding gait [39]. A discrete gait pattern stabiliser was also implemented to maintain the footfall sequence and timing. Feedback controllers were required to balance the robot during the stance phase.

Through the use of feedback controllers, ATRIAS [30] was capable of walking in $3D$, with the use of reduced models. A number of behaviours were generated, which did not require any prior planning. By blending and combining multiple behaviours, dynamic locomotion was achieved [38]. Each behaviour had a controller, which were blended together depending on the leg force, rather than discretely switching between controllers.

Robots such as Cassie [7] took a different approach by posing the control as a nonlinear program where the robot dynamics were modelled as a constrained system with compliance. Little Dog [98] took this a step further and deployed state-of-the-art learning, planning and control to achieve fast locomotion on rough terrain. The controller learned the optimal foot position using a terrain template as well as ZMP (Zero Moment Point) algorithms.

On the other hand, ZMP has been utilised for slow but stable bipedal locomotion, making use of flat footed contacts. The downfall of this approach was that the contacts required planning with respect to the environment to enforce the ZMP criteria, and therefore, it was not suited to unstructured terrain [38]. For the ZMP criterion to be satisfied, the zero moment point must lie inside the support polygon of the feet of the robot [99].

In order to investigate transient locomotion, such as rapid acceleration and deceleration, we need to develop a method to test these motions. Generally, initial investigations use planar robots that need to be constrained in the sagittal plane using a boom or support rig, with some steady-state locomotion studies performed on treadmills in the laboratory [100].

Human studies on treadmills, especially those involving gait transitions [101], have found discrep-ancies between the results from treadmills compared to overground running. A few of these discrepancies will not effect robots such as:

1. Visual stimulation (optical flow) is lacking, therefore, the subject does not feel like they are accelerating [101].
2. Short treadmills may limit acceleration as fear of reaching the end of the treadmill.
3. Subjects may alter their gait as they could be intimidated by the treadmill if they are not use to running on one.
4. Transition speeds between gaits are higher when on ground versus on a treadmill [101].
5. Lower magnitudes of acceleration are experienced on a treadmill [101].

For robotics, there are a number of critical problems with testing on treadmills. It has been noted that when accelerating on a treadmill, the linear inertia is zero compared to overground acceleration (you effectively hop into the air and swing your legs faster to accelerate on a treadmill, compared to accelerating your mass when running overground) [101]. However, it has been shown that for constant velocity motions, running on a treadmill is mechanically equivalent to overground running. These treadmills are, however, costly to purchase or develop.

Therefore, for the majority of research groups, these treadmills are not an option to test transient manoeuvres. Instead boom like systems are used, Figure 2.10, where instead of a robot accelerating in a straight line, they are constrained to run in a circle and the boom supports the robot in the sagittal plane. The longer the boom arm, the straighter the robot runs. The downfall with these systems is the added mass of the boom that the robot must support and accelerate. An alternate method is to have an active boom which compensates for its mass and inertia.

2.4 Trajectory Optimisation

It has been suggested that locomotion in animals can be cast as an optimisation problem where some cost function is minimised [103]. Therefore, trajectory optimisation can be used to design optimal gaits for legged robots. Trajectory optimisation is a mathematical approach to

Figure 2.10: An example of a robot attached to a boom system. The boom constrains ATRIAS in the sagittal plane, allowing it forward, vertical and pitching motion. The vertical and horizontal motion is constrained to the arc of the boom, however, the longer the boom arm, the less of an effect this has on the motion. Image from [102].

generating a path (trajectory) subject to a number of constraints and variable bounds. These methods select the inputs to the system (controls) as a function of time. A solution is deemed feasible when it satisfies all the problem requirements, such as constraints and bounds. However, a solution is deemed optimal once the solution is feasible and has minimised the user defined cost (objective) function [104]. The solver achieves this by varying the decision variables between their bounds to satisfy the constraints in order to find a feasible solution. The decision variables generally contain the time, state trajectories, $x(t)$, and the control trajectories, $u(t)$.

In order to guide the solver to a realistic solution, a number of constraints are applied to the problem. These include boundary conditions (for example the initial and terminal conditions), system dynamics (for example the robot EoM) and path constraints (for example the robot cannot penetrate the ground). Bounds are applied to the decision variables to narrow the search space and enforced conditions.

For this research, only single phase continuous time trajectory optimisation problems were considered (continuous dynamics throughout the entire trajectory). Many systems implement hybrid dynamics which consists of multiple phases of continuous dynamics, connected with a discrete mapping (contact events). These contact events need to be *a priori* defined.

Optimal locomotion planning represents a significant computational challenge [43], and has been tackled in a number of ways:

1. Constrain the problem to determine feasibility instead of optimality
2. Simplify the robot dynamics using templates
3. Simplify the task (periodic motion instead of long-time-horizon tasks)
4. Optimisation assumptions (no slipping, fixed contact order, no motor model, infinite torque actuators etc.)

Due to the complexities of the systems being optimised, as well as some of the required tasks (long-time-horizon task), numerous assumptions and simplifications are often implemented to make the trajectory optimisation problem more tractable. One such simplification is to utilise simple models, such as templates, to represent the dynamics of the system. Often these models have further simplifications, such as infinite available torque [29], massless legs [4, 5] and no slipping allowed [4, 6] (infinite friction).

The reason why templates are popular in the literature is due to the CoM trajectory of a number of walking and running gaits in animals being described using these simple models (such as the SLIP model) that contain springy simple legs [38]. These templates exhibit the underlying motion, however, they use a reduced set of variables [39], reducing the computational complexity of the system. These simple models generally have massless virtual-legs with point feet that are not allowed to slip, as well as a point mass body. Due to the complexity of the robotic systems being developed, these simplified models are often used to design controllers as the complexity of the system is reduced making the study more tractable [7]. The use of these simple models bridges the gap between heavily model based methods and model free heuristic methods (such as Raibert controllers) [39].

The shortfall of this method is that it does not scale well to more complex and dynamic systems, as these systems have non-negligible leg mass that has an effect on the body roll during the trot gait and body pitch during the bounding gait [105]. These effects are less significant during slow speed periodic gaits, however, they cannot be ignored during higher speeds and non-periodic gaits. This has resulted in a focus area of the literature to perform system identification to accurately estimate the inertial parameters to account for these effects [106].

In order for robots to explore the environment, more dynamic tasks (such as rapid acceleration, deceleration and turning) need to be investigated. Currently these tasks are tackled using full body detailed models to generate controllers, which require a large amount of computation power and are extremely challenging to solve in real time.

Another approach is to simplify the task at hand, for example a single steady-state step instead of a long-time-horizon task. These simpler tasks, such as limit cycle (periodic, steady-state) locomotion, have been a large focal point in the literature, as robots that can successfully stabilise a limit cycle, can in theory move indefinitely [43]. However, little attention has been devoted to accelerating to and from these limit cycles.

Problems can be further simplified by enforcing a contact order. Often inaccurate integrators are also used, such as Backwards-Euler compared to higher-order collocation methods, as well as inaccurate contact models, such as a soft contact model [107, 108], where the gains for the contact have to be manually chosen. A recent study has explored using trajectory optimisation methods on problems without perfect terrain information [109] (such as uncertainty in the friction coefficient and terrain).

Optimisation methods are generally classified into two broad methods: Direct and Indirect. This research focuses on direct methods, which discretise the optimisation problem, converting it into a nonlinear program (known as transcription methods) [104]. The discretisation of the continuous time trajectories is done by approximating the trajectory using polynomial splines.

2.4.1 Direct Methods

Majority of the solvers (this research uses IPOPT [110]) require an initial guess or seed point to start the optimiser [104]. For simple problems, a wide range of seed points will result in an optimal solution, however, as the problem becomes more complex, a bad seed point can result in the solver getting stuck or finding a local minima. Seeds become very important depending on the type of methods being applied [104]:

- *Single Shooting*: Transcription methods are used to convert the continuous time problem into a non-linear problem, which can be solved using commercial solvers. Single shooting methods are the easiest method to transcribe an optimal control problem, however, it only works on simple systems with simple dynamics. The trajectory is approximated using a simulation. By adjusting the initial conditions and the function that approximates the inputs (for example a zero order hold function), a trajectory will be generated. As the trajectory is simulated, no path constraints can be applied to the problem, as only the initial and final variables are defined [104].
- *Multiple Shooting*: This is a more robust extension to Single Shooting methods as the trajectory is divided into segments, with each segments trajectory approximated using a simulation. This effectively creates a large number of smaller nonlinear optimisation problems, which are coupled. Each individual problem has fewer variables and is, therefore, easier to solve [104]. It is also easier to implement path constraints, as some intermediate variables (between each segment) are available. As more segments are used and they get shorter, the relationship between the decision variables, objective function and the constraints becomes more linear. Initially the end of one segment will not line up with the start of the next segment. This is known as a defect, which needs to be minimised to create a smooth trajectory.
- *Orthogonal Collocation*: The system dynamics are enforced through nonlinear constraints, and therefore, the solver needs to find a feasible solution before attempting to find a

locally optimal solution. The above shooting methods always return a dynamically feasible solution, as the dynamics are simulated [111]. As the trajectories are parametrized, the initial guess can be generated for the state trajectory, $x(t)$, which is easier to determine than just the initial values required for the above methods. Path constraints are easily implemented as the optimiser has access to all the variables. These methods generate large optimisation problems, however, they are sparse (with dedicated solvers optimally designed for sparse problems).

An image showing the comparison between these methods is shown in Figure 2.11. Throughout this research, 3-point orthogonal collocation methods are used (detailed in Chapter 3).

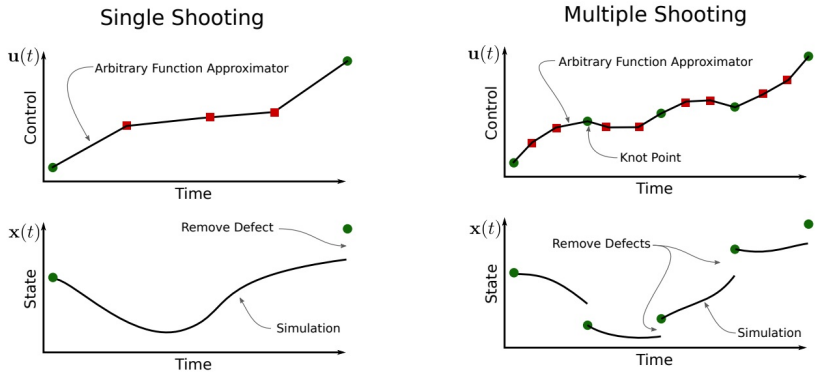

Figure 2.11: The graph on the left shows an example of single shooting, where the control signals are approximated by a function, and the simulation of the state-trajectory. The graph on the right shows an example of multiple shooting where the state trajectory has been divided into a number of simulations. Image from [104].

These methods rely on continuous time trajectories (including the system dynamics) and therefore, contacts, which alter the dynamics of the system, are not intuitively handled. Therefore, through-contact methods were implemented.

2.4.2 Through-Contact Methods

Direct methods are often used to plan locally optimal trajectories for robotic systems. These systems often involve impacting the ground (for example legged robots running). Most state-of-the-art techniques require *a priori* defining the contact order, requiring prior knowledge of the contact order. This drastically limits the search space and eliminates all potential non-intuitive solutions that may be optimal [112]. These methods connect the different modes (aerial, one foot on floor etc.) through a discontinuous mapping (at the velocity level) using hybrid dynamic methods.

Up until a guard condition (foot contacting the ground), the trajectories are described using smooth dynamics. The impact is modelled as a discontinuous jump in state space (a change

in velocity as energy is dissipated through the impulsive collision). The new set of dynamics (with the foot fixed to the floor, after the guard condition) are smooth for the remainder of the contact, until the lift-off guard condition. Constraints are added to ensure the foot is in the correct position for the guard condition to be met, ensuring each segment is connected through the hybrid events. This method does not allow the foot to slip, assuming infinite friction.

Alternative methods use continuous dynamics to solve the contacts through modelling the floor as a spring and damper system. These are called soft contact models, and result in stiff dynamics and generally require small sample times, increasing the complexity of the optimisation problem [111]. This requires hand-tuning the dynamics of the floor, which is useful when running on a soft ground that can easily be modelled.

In general, these impulsive collisions complicate trajectory optimisation methods as they are highly nonlinear and result in large impulsive forces with large changes in velocities. Recently, success has been achieved using through-contact methods, where the desired contact order is not required *a priori*. The order is solved by the optimiser, allowing all gait patterns to be explored.

The contacts are formulated as a set of complimentarity constraints, with the contacts modelled as inelastic collisions, with a coulomb friction model allowing the foot to slip if the friction cone is violated [111]. As orthogonal collocation methods are implemented, no simulation is required as the dynamics are implemented using constraints, with the trajectories being approximated using high order splines, which are evaluated at collocation points [111]. Through the use of slack variables, through-contact methods can make and break contacts, allowing the optimiser to determine the optimal contact order [112]. How this method has been implemented in the literature has generally resulted in poor accuracy results, making these studies infeasible to be implemented on a physical platform. However, recent advancements in the literature [112] has shown how to improve this accuracy.

The common method of implementing these through-contact methods typically relied on first-order Backwards-Euler integration methods which suffer from bad accuracy [112]. These methods, therefore, required a large number of node points (time-steps, N) for sufficient accuracy (accuracy of $O(1/N)$). By implementing higher order orthogonal collocation methods, such as 3-point collocation, the dynamics are solved with third order polynomials, resulting in a smoother representation of the state trajectories at each node point and collocation point, resulting in an accuracy of $O((1/N)^{2K-1})$ [112].

This method constrains the contact mode switching event to occur at mesh points (node points) and enforces the contact to remain for the duration of the relevant collocation points. This is done through restricting when the algebraic complimentarity constraints can activate or deactivate. These constraints can apply the impulsive transition between modes, provided sufficient freedom is given to adjust the time-step, allowing a contact to occur at a mesh point.

The advantage of this is that no contact detection algorithm is required [112], however, a variable time-step is required.

By implementing these complimentarity constraints, the trajectory optimisation problem becomes notoriously difficult to solve. Two methods are commonly used throughout the literature to solve these constraints. The first is the penalty method [113, 114] and the second is the *epsilon* relaxation method [112, 115].

There are a number of reasons why the penalty method was not implemented in this research. The first being that there are multiple terms in the objective function (the penalty and the cost function), so it is unclear which is getting optimised. The penalty needs to be scaled, varying the scale effects the solution. As the contacts are being solved in the objective function, these "constraints" do not need to be satisfied, whereas using the epsilon relaxation method, it is guaranteed that the contacts are solved (a requirement for a feasible solution).

For the *epsilon* relaxation method, the complementarity constraints (in the form of equality equations, $\alpha * \beta = 0$) are transformed into inequality constraints, which are relaxed by a parameter ϵ, $\alpha * \beta < \epsilon$. To help the solver find a solution, the problem is solved multiple times, first, ϵ starts large ($\epsilon = 1000$) and after each successful solve, ϵ is reduced until a sufficient accuracy has been achieved.

This still does not solve the problem of having a large number of complementarity constraints (each evaluated at the node and collocation points). In order to keep the accuracy, but reduce the number of complementarity constraints (making it easier to solve), additional slack variables were added, and the complementarity constraints were only evaluated at the node points, with the contact held for the duration of the collocation points [112]. Further details of how this was implemented can be found in Chapter 3.

2.4.3 Cost Functions

There are several approaches taken when optimising for a gait. One method is to optimise for energy efficiency, another is to optimise for mechanical strain or controller heuristics. A large focus has been devoted to energy efficient locomotion focusing on steady-state periodic motions. In nature, most mid-sized quadruped mammals make use of the galloping gait at high speeds due to its energy efficiency [105]. This is not observed in robotics due to most robots not transitioning past the transitional speed to make the gait optimal, as well as robots lacking lateral or roll dynamics [33] (either due to the support system or a $2D$ simplification). When investigating these motions using trajectory optimisation methods, a number of intuitive cost (objective) functions are available and widely used. However, as little attention has been given to transient motions, an intuitive cost function does not exits.

Some authors have suggested that energy should not be optimised for small tasks (such as acceleration), as it is a small fraction in comparison to the daily energy budget. However, it has been argued that some form of energy cost is used in every day activities [44]. The evidence for this is that evolution does not know the difference between energy saved on one large task or the equivalent energy saved across multiple smaller tasks [44].

An alternate cost function that has been extensively used in the robotics literature, is that of vertical agility [41]. Vertical agility is a measure of the maximum jump height divided by the frequency of jumps, and is measured in vertical meters per second [41]. As it is a measure of the vertical climb rate, the vertical agility of any system is limited to its power-to-weight ratio, as well as a theoretical maximum achievable value [41]. This metric has been used to compare a number of different legged robots [41]. Researchers have hinted that if a robot is good at vertical hopping then it will be good at acceleration (by leaning forward and hopping) [41, 116].

Chapter 3

Methodology

This research investigates three specific problems, namely: determining how legged robots should accelerate (positive and negative acceleration), the optimal spine morphology (rigid, revolute or prismatic) for quadruped robots to perform these long-time-horizon manoeuvres (start and end at rest, travelling a fixed distance), and finally how to control a legged platform (monopod robot) using inspiration from trajectory optimisation methods. The robots of interest can be seen in Figure 3.1. As can be seen, two leg morphologies were used, namely the two link leg and the prismatic leg. In this chapter, the methods used are explained for both leg configurations, however, in later chapters, the selected leg configuration is explicitly stated.

The three problems of interest tie into the research questions (addressed in Section 1.2) with some overlap between questions. In order to address these questions, a large number of mathematical models and simulations of the relevant robots were required. This required designing the robot in SolidWorks (done by Joshua Van Zyl) and developing mathematical models which were used in trajectory optimisation problems to inspire the controller design. These mathematical models were also used with Monte Carlo simulations to determine the optimal spine morphology for quadrupedal robots. The designed controllers were first simulated using trajectory optimisation methods and then validated on the physical platform. This chapter aims to describe the above mentioned methods used throughout this research as well as to give the required details on the modelling, controller design and verification process followed.

3.1 Modelling and Simulating the System

Modelling and simulating a system is a very broad category. In this research it covers three main aspects. These include: mathematically modelling the system to determine the equations of motion (EoM) using Euler-Lagrange Dynamics, trajectory optimisation to generate trajectories

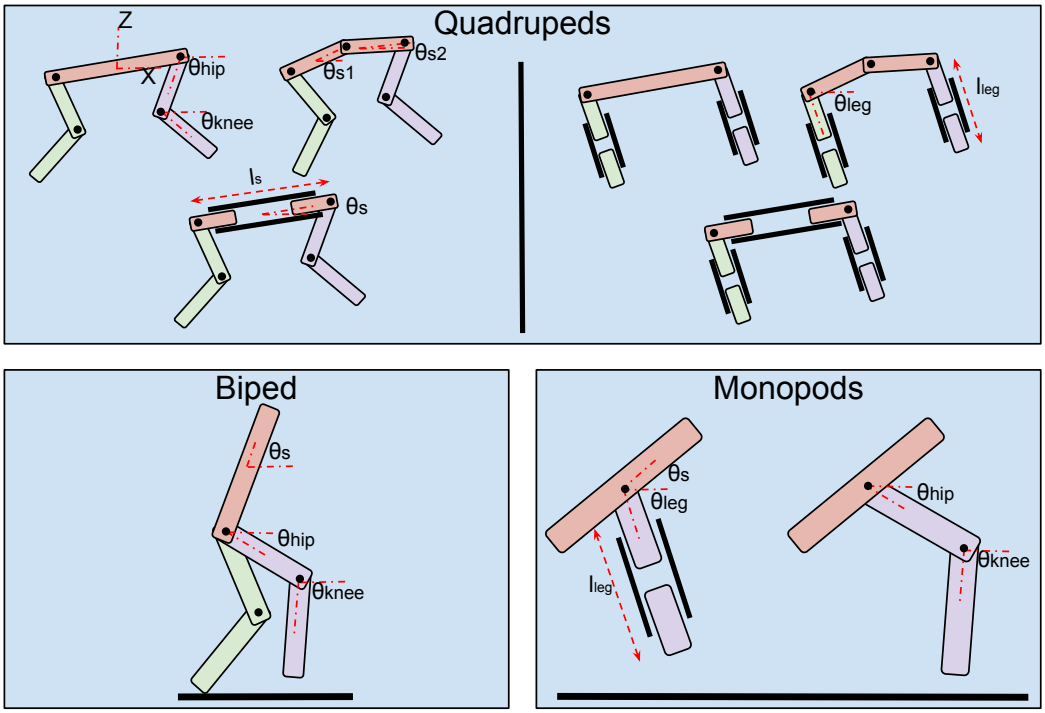

Figure 3.1: The image shows the three robot types of interest. On the top are the quadrupeds, bottom left the biped and bottom right the monopods. The three quadruped spines of interest in this research consist of the rigid, revolute and prismatic spine. Both the two link leg morphology and the prismatic leg morphology are shown. For the quadruped, the leg morphology takes on an "X" formation, with the rear and front knees pointing inwards.

to analyse and finally, inspiring the designing of controllers. These methods are described in greater detail below:

3.1.1 Euler-Lagrange Dynamics

The EoM were generated using Euler-Lagrange dynamics [117] in the form of the manipulator equation:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{B}\boldsymbol{\tau} + \mathbf{A}\boldsymbol{\lambda} \tag{3.1}$$

where $\mathbf{q}$ are the generalised coordinates of the robot (using absolute angles relative to the inertial frame) along with the generalised forces and torques represented by $\boldsymbol{\tau}$. $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is the Coriolis matrix, $\mathbf{G}$ is the gravitational potential matrix, $\mathbf{B}$ maps the applied torques and forces to the generalised coordinates and $\mathbf{A}$ maps the external forces (such as ground reaction forces) to the generalised coordinates. The ground reaction forces (GRF) are represented by $\boldsymbol{\lambda}$ and consist of the following for each leg:

$$\boldsymbol{\lambda} = [\lambda_x, \lambda_z]' \tag{3.2}$$

Sub-script x represents the horizontal component and z the vertical component of the GRF, as its a planar model, there was no y component. The generalised coordinates are as follows:

$$\begin{aligned}
\mathbf{q}_{Monopod} &= [X, Z, \theta_s, \mathbf{q}_{leg}]' \\
\mathbf{q}_{Biped} &= [X, Z, \theta_s, \mathbf{q}_{leg_{left}}, \mathbf{q}_{leg_{right}}]' \\
\mathbf{q}_{Rigid} &= [X, Z, \theta_s, \mathbf{q}_{leg_{rear}}, \mathbf{q}_{leg_{front}}]' \\
\mathbf{q}_{Revolute} &= [X, Z, \theta_{s1}, \theta_{s2}, \mathbf{q}_{leg_{rear}}, \mathbf{q}_{leg_{front}}]' \\
\mathbf{q}_{Prismatic} &= [X, Z, \theta_s, l_s, \mathbf{q}_{leg_{rear}}, \mathbf{q}_{leg_{front}}]'
\end{aligned} \tag{3.3}$$

where $\mathbf{q}_{leg_{...}}$ can be one of the following (depending if the two link leg configuration or the prismatic configuration is being used):

$$\begin{aligned}
\mathbf{q}_{two\ link} &= [\theta_{hip}, \theta_{knee}] \\
\mathbf{q}_{prismatic} &= [\theta_{leg}, l_{leg}]
\end{aligned} \tag{3.4}$$

with the applied forces and torques for each robot as follows:

$$\begin{aligned}
\mathbf{u}_{Monopod} &= [\mathbf{u}_{leg}]' \\
\mathbf{u}_{Biped} &= [\mathbf{u}_{leg_{left}}, \mathbf{u}_{leg_{right}}]' \\
\mathbf{u}_{Rigid} &= [\mathbf{u}_{leg_{rear}}, \mathbf{u}_{leg_{front}}]' \\
\mathbf{u}_{Revolute} &= [\tau_s, \mathbf{u}_{leg_{rear}}, \mathbf{u}_{leg_{front}}]' \\
\mathbf{u}_{Prismatic} &= [F_s, \mathbf{u}_{leg_{rear}}, \mathbf{u}_{leg_{front}}]'
\end{aligned} \tag{3.5}$$

where $\mathbf{u}_{leg_{...}}$ can be one of the following (depending if the two link leg configuration or the prismatic configuration is being used):

$$\begin{aligned}
\mathbf{u}_{two\ link} &= [\tau_{hip}, \tau_{knee}] \\
\mathbf{u}_{prismatic} &= [\tau_{leg}, F_{leg}]
\end{aligned} \tag{3.6}$$

Further details of how the EOM were generated can be found in Appendix A.

3.1.2 Trajectory Optimisation

Trajectory optimisation is a mathematical tool used to generate a trajectory that minimises (or maximizes) a measure of performance (a cost function) while satisfying a number of constraints and variable bounds [104]. The trajectory of each variable was divided into N node points. Constraints, such as the EoM, were applied to ensure a feasible transition from node to node. Constraints were satisfied by varying the decision variables between their respective bounds until a feasible or (locally) optimal solution was found.

As seen from the literature review, legged systems are hybrid dynamic, with discrete contact and lift-off events as seen in Figure 3.2. These methods are acceptable for steady-state locomotion where the contact order is known, however, for rapid manoeuvres, the contact order is unknown. Therefore, a method that allowed the optimiser to pick the contact order was utilised (using through-contact methods [111], detailed below).

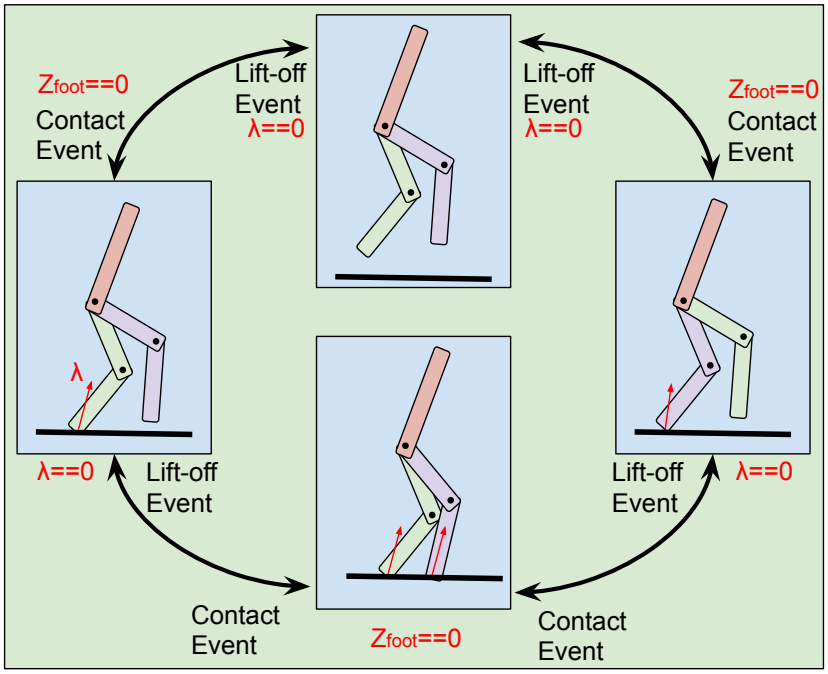

Figure 3.2: Hybrid dynamic methods use discrete mappings (activated at foot touch-down or lift-off events) to transition between aerial and stance phases. During this mapping, the EoM of the system are changed (to either add GRF or remove them). These methods require a known contact order and do not allow for the foot to slip.

For this research, the selected optimisation task was the long-time-horizon task. This task involved the robots starting and ending at rest while travelling a fixed distance. This resulted in an acceleration phase, steady-state phase as well as a deceleration phase. When the acceleration and deceleration phase where of particular interest, the task was split into the three components and optimised separately. First steady-state steps were generated (starting at the apex of the steady-state step), followed by acceleration trajectories (from rest to the apex, $\dot{Z} = 0$, of the steady-state trajectory) were generated. Finally deceleration trajectories from the apex

of the steady-state step to the rest configuration were generated. For the acceleration and deceleration tasks, the contact order, and number of contacts, were not enforced and selected by the optimiser. These three phases were then *stitched*[1] together to form the long-time-horizon task, which is graphically represented in Figure 3.3. This enabled three smaller trajectories to be optimised, allowing for more nodes (resulting in an increase in the accuracy of the trajectory) to be used for each of the trajectories.

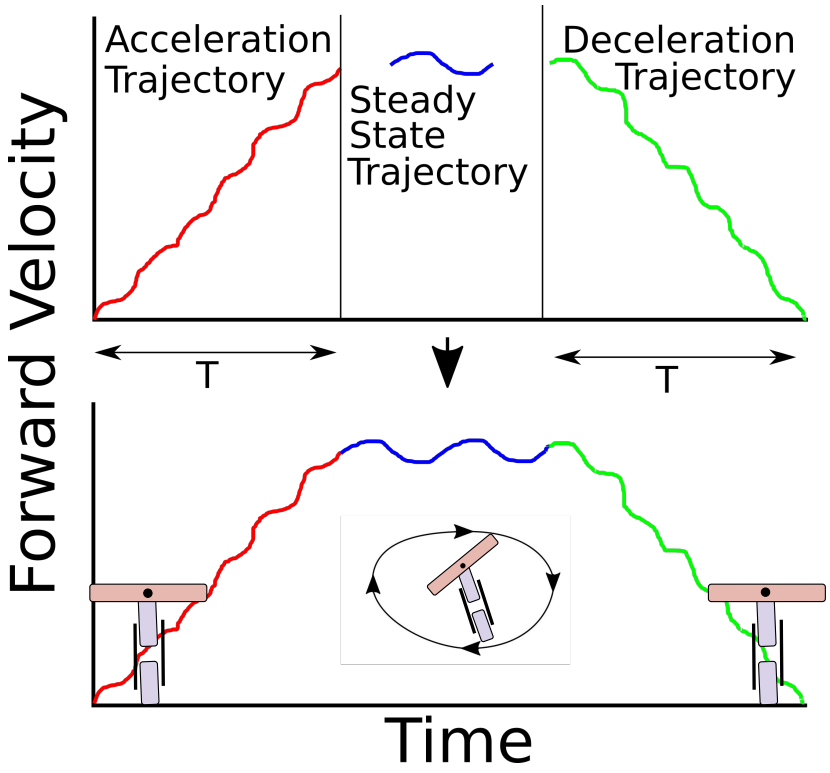

Figure 3.3: A graphical representation of the *stitching* method. First steady-state trajectories were generated. Then acceleration and deceleration trajectories were generated and stitched to the steady-state trajectory, resulting in the long-time-horizon task [43]. To cover more distance, multiple steady-state steps can be stitched together, as they are periodic.

The robot models optimised in this research consisted of highly non-linear hybrid dynamic systems which were extremely complex to solve. The different states the system could take included: Aerial phases, one foot on the floor, or for the biped/quadruped robots, both feet on the floor, as seen in Figure 3.4). Due to this complexity there was no guarantee that the solver would find an optimal solution on the first attempt, often getting stuck in local minima. Therefore, to increase the likelihood that the solver will find an optimal solution, and to explore the search space, multiple optimisations were performed from a randomly generated seed points (the starting point of the optimiser) using uniformly distributed random numbers between the variable bounds [74]. The optimal solution was then taken as the seed that resulted in the lowest cost value.

In order to improve the convergence rate and the time taken to converge, a number of heuristics

[1]The end of the acceleration trajectory was constrained to the start of the steady-state step. The start of the deceleration trajectory was constrained to the end of the steady-state step.

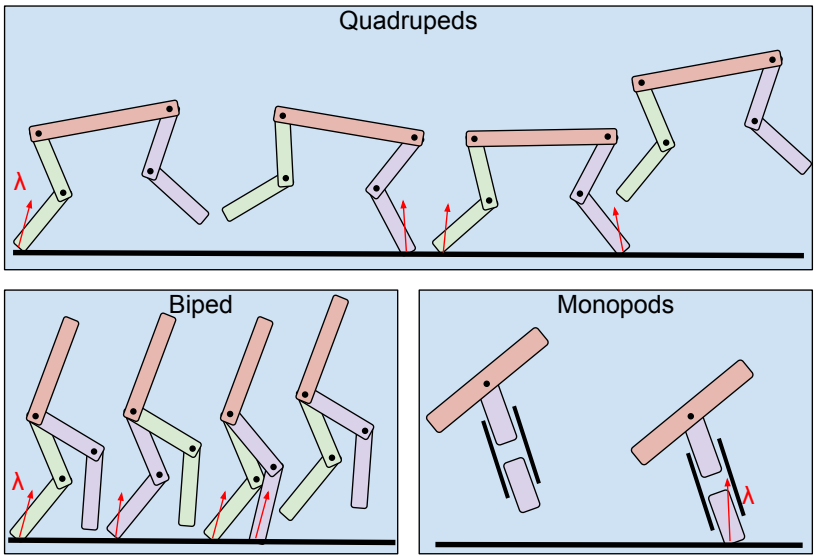

Figure 3.4: The different phases of motion for each of the robots is shown. These include the two phases for the monopod (aerial, foot on the floor), four phases for the biped (aerial, one foot on the floor, both feet on the floor) and the quadruped (aerial, one foot on the floor, both feet on the floor).

were employed [112, 118]. One such method employed involved only randomizing the initial values of the generalised coordinates and fixing the remaining variables of the seed to 0.01 [43]. Another method used was to solve the seed iteratively, each iteration adding more constraints until all the constraints had been added (with the solver tolerances being set low, $tol = 1E-4$ [112], and the cost function fixed to a constant value). Once all the constraints had been added, the solver was warm started with tighter tolerances, $tol = 1E-6$, and the desired cost function.

An additional problem was that rapid acceleration and deceleration has not been a focal point in the literature, and therefore, there are no widely accepted cost functions to choose from. When looking at steady-state motion, commonly the energy based cost function is used, as energy efficiency is an important concept for robotics [119], however, this does not promote explosive motions. The Author decided to use an energy based cost for these manoeuvres, with additional constraints to enforce a rapid manoeuvre. These constraints included:

1. Enforcing a minimum distance the robot must travel in a fixed time
2. Enforce an average/final velocity
3. Start at rest and end at the apex of a high speed gait (or vice versa) with limited available time or distance.

The choice of cost function depended on the optimisation task at hand, with some of the following cost functions being used:

$$
\begin{aligned}
J_1 &= \sum_{i=1}^{N} h(i) \\
J_2 &= \sum_{i=1}^{N} \frac{\tau(i)^2}{x_0(N)} h(i) \\
J_3 &= \sum_{i=1}^{N} \tau(i)^2 h(i)
\end{aligned}
\tag{3.7}
$$

where J_1 is minimum time cost function (where $h(i)$ is the time duration for the i^{th} node), J_2 is a cost of transport, CoT, cost function (only applicable to electric actuated robots, the focus of this research). With $x_0(N)$ being the final distance travelled. J_3 is a torque based heat cost function. It must be noted that J_2 is not a true CoT and is rather the heat based cost function, J_3, normalized by the total distance travelled. It is also important to note that all optimisations are enforced to start at zero forward displacement and zero time.

The trajectory optimisation problem was run in GAMS [120] (General Algebraic Modelling System) software using the IPOPT solver [110]. In order to further guide the optimiser to a solution, a number of constraints and bounds were applied. Some of the common constraints and bounds are listed below. As a number of different trajectory optimisations were run throughout this research, the specifics on which constraints were used are given in the relevant chapters.

3.1.2.1 Collocation

The trajectories of each generalised coordinate was discretised into N finite elements (time periods or node points, designated i) using polynomials, where N was between 50 and 300 depending on the optimisation task (further details can be found in the relevant chapters). The trajectory between each node point was represented using a Runge-Kutta basis with 3 collocation points [112] as seen in Figure 3.5, designated j. Therefore, the trajectory was divided into N points, ($i \in [1, N]$), each with 3 collocation points, ($j \in [1, 3]$). Three-point Radau was used to solve the differential equations, (3.1), at the selected time points [112, 121], with an accuracy of h^{2K-1} (where h is the time-step and K is 3). This was done by implementing the following equations for each state trajectory ($\mathbf{q}$ and $\dot{\mathbf{q}}$):

$$
\begin{aligned}
\mathbf{q}(i,j) &= \mathbf{q}_0(i) + h(i) * \sum_{k=1}^{3} \mathbf{a}(k,j)\dot{\mathbf{q}}(i,k) \\
\dot{\mathbf{q}}(i,j) &= \dot{\mathbf{q}}_0(i) + h(i) * \sum_{k=1}^{3} \mathbf{a}(k,j)\ddot{\mathbf{q}}(i,k) \\
\mathbf{q}_0(i) &= \mathbf{q}(i-1,3) \\
\dot{\mathbf{q}}_0(i) &= \dot{\mathbf{q}}(i-1,3)
\end{aligned}
\tag{3.8}
$$

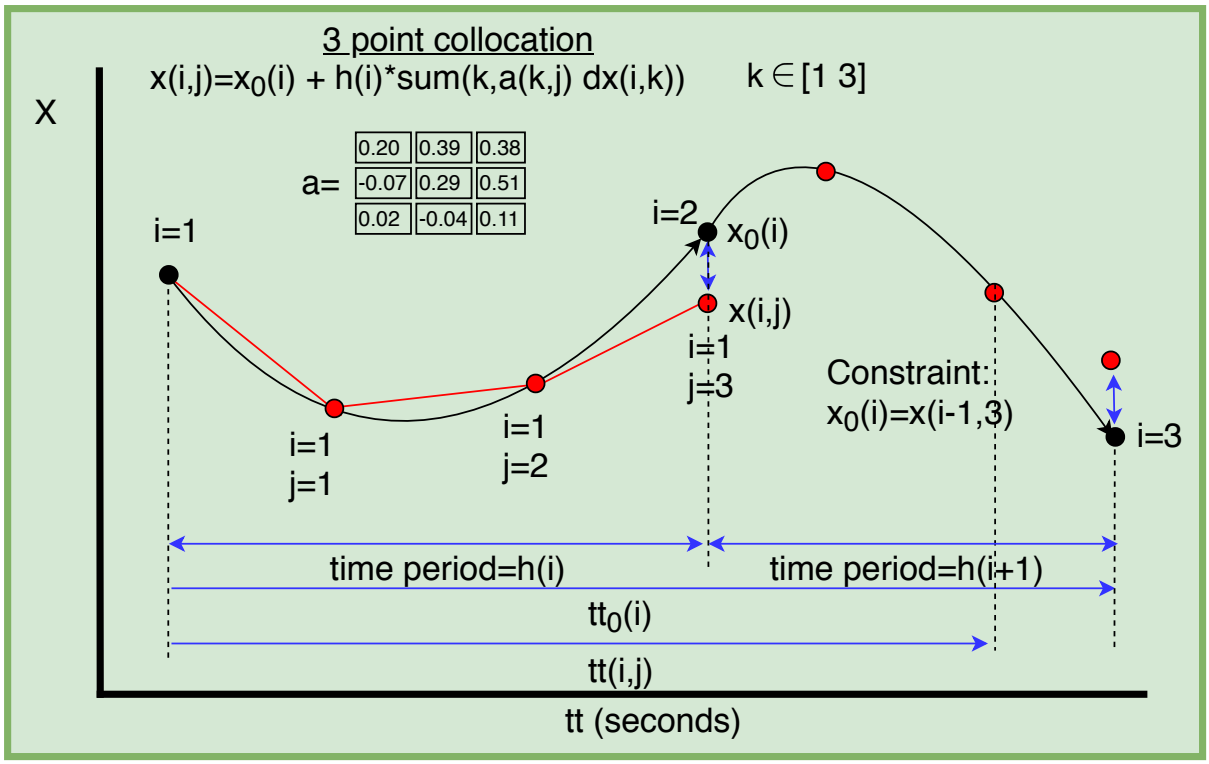

Figure 3.5: A graphical representation showing how the trajectory is first broken up into node points (i) and then collocation points (j). Additional constraints to ensure the trajectory is smooth are also shown.

The first two equations implement the 3 point Radau to integrate the equations of motion from node i to $i+1$ whereas the third and fourth equation ensure that the third collocation point of node i was located at the same point as node $i+1$ to ensure a continuous state trajectory. The difference between $\mathbf{q}(i,j)$ and $\mathbf{q}_0(i)$ was that $\mathbf{q}(i,j)$ acts on i (node point) and j (collocation point), whereas $\mathbf{q}_0(i)$ only acts on i. The 3 point collocation matrix was as follows [112]:

$$
\mathbf{a} = \begin{bmatrix}
0.19681547722366 & 0.39442431473909 & 0.37640306270047 \\
-0.06553542585020 & 0.29207341166523 & 0.51248582618842 \\
0.02377097434822 & -0.04154875212600 & 0.11111111111111
\end{bmatrix}
\tag{3.9}
$$

As contacts are required to occur at node points (and not between node points), a variable

time-step was required. The time-step between each finite element was constrained as follows:

$$\begin{aligned} 0.5h_M \leq h(i) \leq 2h_M \\ tt(i,j) = tt_0(i) + h(i)\sum_{k=1}^{3}\mathbf{a}(k,j) \\ tt_0(i) = tt(i-1,3) \end{aligned} \tag{3.10}$$

where $h(i)$ was the time-step for the i^{th} node and h_M was set to T/N, where T was set depending on the optimisation task (see relevant chapter for more details). T was used as a scaling factor for the node time and was set according to the expected time the task would take (for the Monte Carlo simulations, the expected time was not known so multiple T values were used). tt was the total time from the start of the trajectory to the relevant node point and collocation point. The first equation bounds the minimum and maximum time period for the node trajectory, whereas the second equation splits up the node time among the collocation points and the third equation ensures the start time of node i ($tt_0(i)$) matches the end time of node $i-1$ at the third collocation point ($tt(i-1,3)$).

3.1.2.2 Through-Contact Optimisation Method

As these transient manoeuvres are poorly understood, the contact order (gait) was not enforced and it was left to the optimiser to select the optimal gait for these manoeuvres, using through-contact optimisation methods [111]. Previous studies using this method have shown that it is a vital tool in studying complex locomotion [122, 123]. There are some downfalls to this method, such as requiring a large number of complimentary constraints to ensure that the GRF only acts when the foot is on the ground [111]. In order to implement this method, the horizontal component of the ground reaction force at each foot was split into the positive, λ_x^+, and negative, λ_x^-, component as follows:

$$\boldsymbol{\lambda} = [\lambda_x^+ - \lambda_x^-, \lambda_z]' \tag{3.11}$$

with the following positive variables (duplicated for each foot):

$$\lambda_z, \lambda_x^+, \lambda_x^- \geq 0 \tag{3.12}$$

First the friction cone was enforced using the following constraints:

$$\mu\lambda_z - \lambda_x^+ - \lambda_x^- \geq 0 \tag{3.13}$$

with the coefficient of friction, μ, being 1 for all simulations. The magnitude of the relative tangential velocity at the contact was calculated with the following constraints:

$$\begin{aligned} \gamma + \Psi(\mathbf{q}, \dot{\mathbf{q}}) &\geq 0 \\ \gamma - \Psi(\mathbf{q}, \dot{\mathbf{q}}) &\geq 0 \\ \gamma &\geq 0 \end{aligned} \tag{3.14}$$

where γ was the magnitude of the relative velocity and $\Psi(\mathbf{q}, \dot{\mathbf{q}})$ was the relative tangential velocity. This requires that γ was a positive variable (third equation in the above set of equations). In order to ensure that the ground reaction force was only applied to the foot when it was on the ground, the following complementarity constraint was applied (shown in Figure 3.6):

$$\phi(\mathbf{q})^T \lambda_z = 0 \tag{3.15}$$

where $\phi(\mathbf{q})$ is the vertical height of the foot (positive by nature). For this equation to be satisfied, a ground reaction force can only be applied when the foot is on the ground and must be zero if the foot is not on the ground. In order to ensure that the friction force lies on the friction cone when the foot slides, the following complementarity constraint was applied:

$$(\mu\lambda_z - \lambda_x^+ - \lambda_x^-)^T \gamma = 0 \tag{3.16}$$

Finally, the following two complementarity constraints were applied to ensure that the horizontal ground reaction force opposes the sliding motion of the foot:

$$\begin{aligned} (\gamma + \Psi(\mathbf{q}, \dot{\mathbf{q}}))^T \lambda_x^+ &= 0 \\ (\gamma - \Psi(\mathbf{q}, \dot{\mathbf{q}}))^T \lambda_x^- &= 0 \end{aligned} \tag{3.17}$$

The above constraints were duplicated for each foot. All contacts were modelled as inelastic collisions [73, 111] and can have one of two modes: slipping or sticking. Slipping was modelled using a simple coulomb friction model [111].

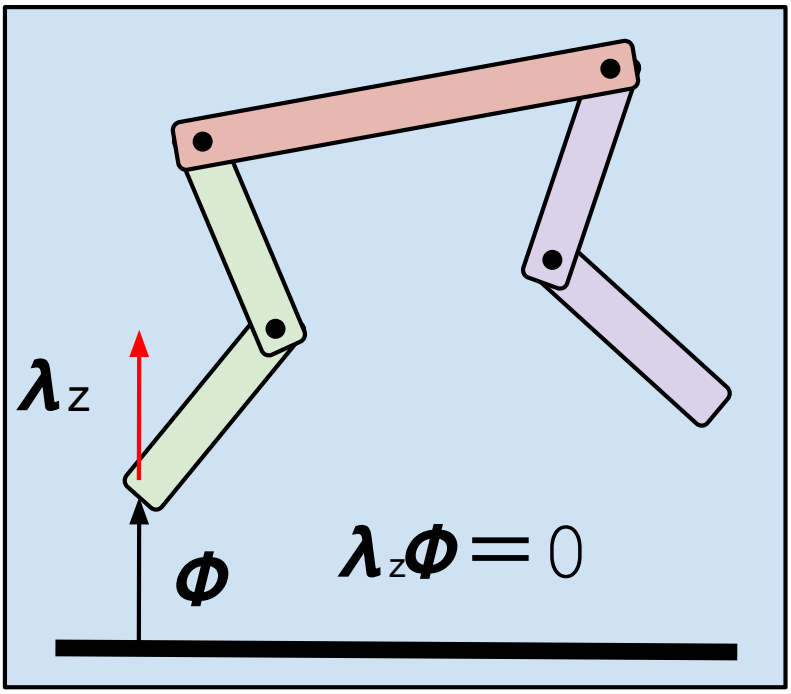

Figure 3.6: Image showing an example complementarity constraint. Note how the foot height, multiplied by the vertical GRF must equal zero. Therefore, if the foot is in the air, the GRF must be zero and vice versa.

To further improve convergence times, regularisation methods in the form of ϵ relaxation techniques [113] were employed to solve complementarity constraints ($\alpha\beta = 0$) from the through-contact optimisation methods. This was done by re-phrasing the above constraints as follows:

$$\alpha\beta \leq \epsilon \tag{3.18}$$

where α and β form the two parts of the complimentary constraints, and are both positive variables. Initially ϵ was set to 1000 and the problem was solved iteratively 8 times. After each successful solve, ϵ was divided by 10. As soon as a solution was not found, the next seed point was run. If it succeeded 8 times, then the complementarity constraints were considered solved (within an acceptable tolerance, $\epsilon = 1E-4$) and the solution was saved.

In order to further simplify the problem and increase the convergence rate, these complementarity constraints were summed across the collocation points (j) and only evaluated at node points

(i) as follows:

$$\begin{aligned} \alpha'(i)\beta'(i) &\leq \epsilon \\ \alpha'(i) &= \sum_{j=0}^{K} \alpha(i,j) \\ \beta'(i) &= \sum_{j=0}^{K} \beta(i,j) \end{aligned} \tag{3.19}$$

Once solved, the complementarity constraint will equal zero ($\alpha(i,j)\beta(i,j) = 0$, within the tolerance of the optimiser and the value of ϵ). The seed generation and solve method is shown in the flow diagram in Figure 3.7.

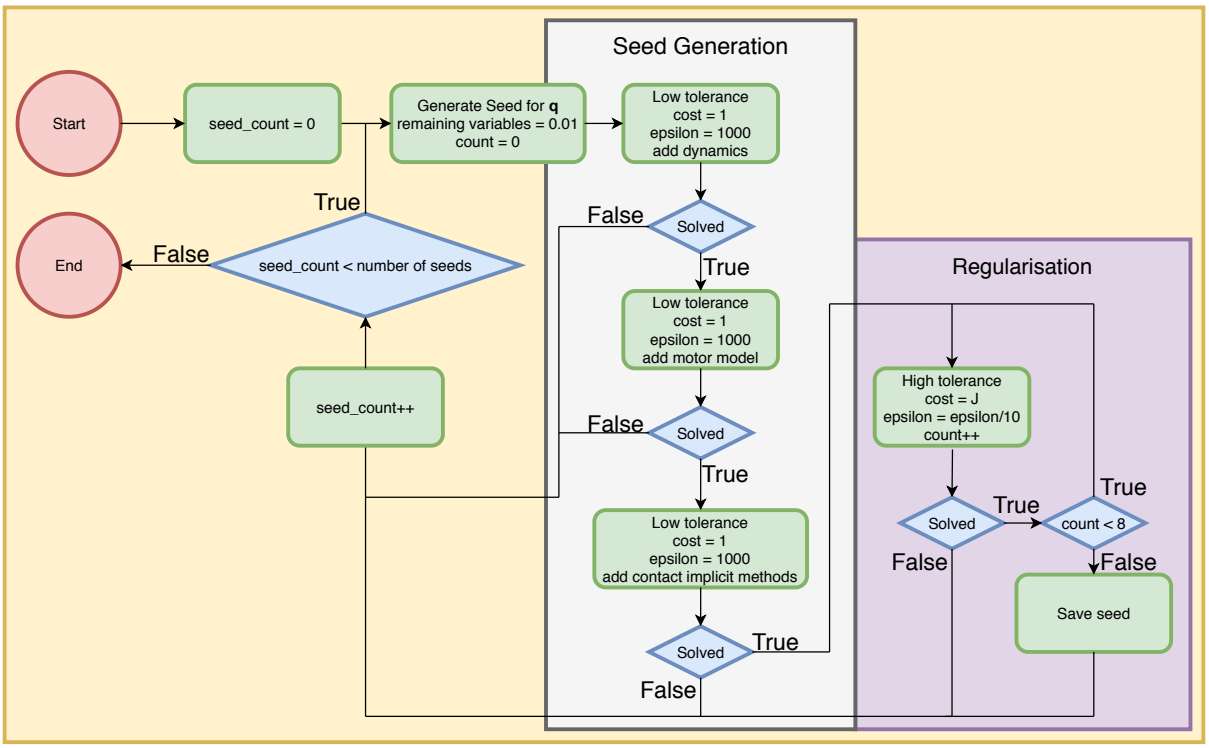

Figure 3.7: Flow diagram showing how the seed was generated and solved (using regularisation techniques). The regularisation techniques consisted of solving the problem iteratively 8 times, each time reducing the epsilon value. if it failed to solve, it would break out of the loop and start a new seed if it hadn't reached its maximum number of seeds.

Initially the seed counter was set to zero and loops through the required number of seeds. For each loop, a random seed was generated. An initial set of constraints were added to the problem and it was solved with a low solver tolerance and a constant cost function. If a solution was not found, it goes to the next seed loop. If a solution was found, further constraints were added. This was repeated until all constraints had been added. The tolerance of the solver was then increased, ϵ was set to 1000 and the correct cost function was added. This was then optimised up to 8 times. If a solution was not found, it moved to the next seed loop. If a solution was found, ϵ was divided by 10 and the problem was optimised again. Once all 8 optimisations had

occurred, the solution was saved and the next seed loop was run.

3.1.2.3 Hard Stops for Prismatic Actuators

For the prismatic actuators, hard stops were implemented to stop the actuator from over extending or over contracting. These hard stops were implemented using complimentary constraints (similar to the GRF) as follows (for the minimum length):

$$\begin{aligned} \alpha_{HS}(i,j) &= l(i,j) - l_{min} \\ \beta_{HS}(i,j) &= F_{HS}(i,j) \\ \alpha_{HS}(i,j)\beta_{HS}(i,j) &= 0 \end{aligned} \tag{3.20}$$

The same method shown above of summing α and β across the collocation points and evaluating at the mesh points was implemented for the hard-stops (eq 3.19). For the maximum length hard stop constraint, the following formulas were used:

$$\begin{aligned} \alpha_{HS}(i,j) &= l_{max} - l(i,j) \\ \beta_{HS}(i,j) &= F_{HS}(i,j) \\ \alpha_{HS}(i,j)\beta_{HS}(i,j) &= 0 \end{aligned} \tag{3.21}$$

3.1.2.4 Steady-State Fixed Contact Constraints

As the long-time-horizon trajectory was occasional broken up into its three phases (acceleration, steady-state and deceleration) for some of the optimisations performed, the steady-state trajectories were generated using a fixed contact order instead of using through-contact methods, as the contact order was known *a priori*. This contact order can be seen in Figure 3.8.

This was implemented by taking the total number of nodes, N, and dividing them among the phases (aerial, foot on the ground, etc.) and for each phase adding the following constraints (acting on the relevant node points for that phase):

$$\begin{aligned} aerial &: \lambda_z = 0 \\ foot\ on\ floor &: \phi(\mathbf{q}) = 0 \end{aligned} \tag{3.22}$$

where $\phi(\mathbf{q})$ was the vertical height of the relevant foot. For example, taking a phase of the biped steady-state trajectory when the left foot is on the floor and the right foot is in the air,

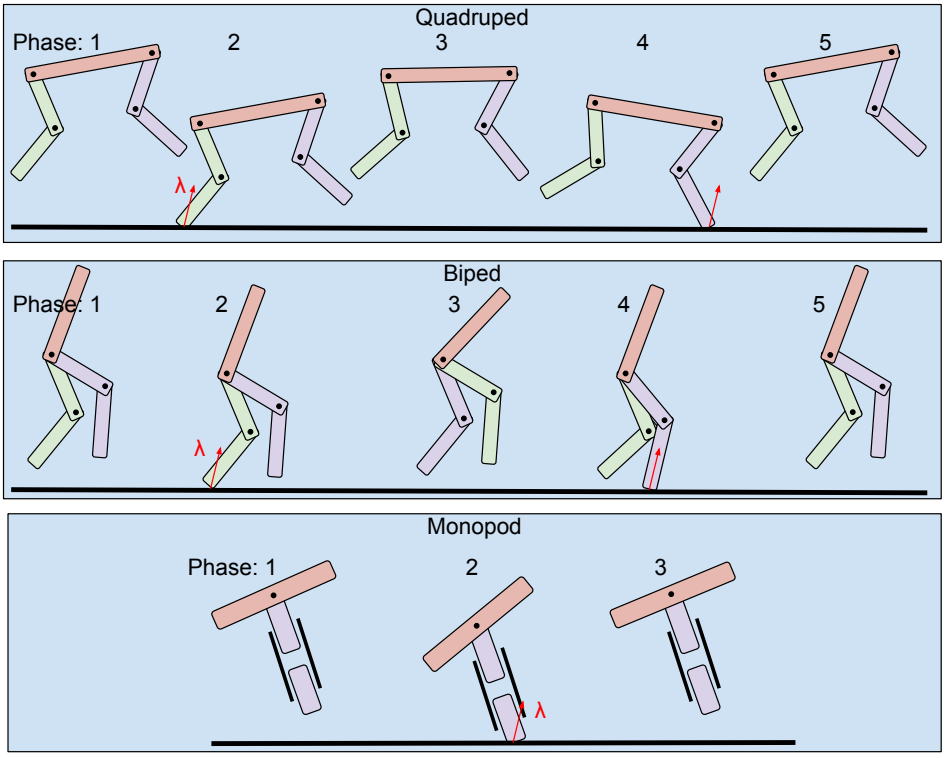

Figure 3.8: The prescribed contact order for the steady-state steps was enforced through constraints for the monopod, biped and quadruped robots. The different phases are shown.

the above constraints will force $\phi_{left}(\mathbf{q}) = 0$ (foot on the floor), and $\lambda_{z_{right}} = 0$, forcing the right foot to be in the air.

3.1.2.5 Joint Angles

Absolute angles were used to generate the EoM, (3.1), as they simplify the equations compared to relative angles [124] (see Appendix A). Therefore, constraints were enforced on the leg angles to restrict the motion and velocity to a feasible range (depending on the robot and leg morphology being optimised). For the two link leg, the constraints were as follows:

$$\begin{aligned} lower\ bound < \theta_{hip} - \theta_{spine} < upper\ bound \\ lower\ bound < \theta_{knee} - \theta_{hip} < upper\ bound \\ lower\ bound < \omega_{hip} - \omega_{spine} < upper\ bound \\ lower\ bound < \omega_{knee} - \omega_{hip} < upper\ bound \end{aligned} \tag{3.23}$$

For the quadruped robot, these constraints enforced the "X" configuration. For the prismatic version of the leg morphology the following constraints were applied:

$$\begin{aligned} lower\ bound < \theta_{hip} - \theta_{spine} &< upper\ bound \\ lower\ bound < \quad l \quad &< upper\ bound \\ lower\ bound < \omega_{hip} - \omega_{spine} &< upper\ bound \\ lower\ bound < \quad \dot{l} \quad &< upper\ bound \end{aligned} \tag{3.24}$$

These bounds can be found in Appendix A.

3.1.2.6 Motor Model

In order to make the results applicable to robotic platforms, a linear motor power model was implemented, taking into account the stall torque and no load speed of the specified motors [74] (and modified for the force actuator). This was implemented through constraints as follows (bearing in mind that relative velocities were used):

$$\begin{aligned} -\tau_{max} - \frac{\tau_{max}}{\omega_{max}}\omega(i) \leq \tau(i) \leq \tau_{max} - \frac{\tau_{max}}{\omega_{max}}\omega(i) \\ -F_{max} - \frac{F_{max}}{\dot{l}_{max}}\dot{l}(i) \leq F(i) \leq F_{max} - \frac{F_{max}}{\dot{l}_{max}}\dot{l}(i) \end{aligned} \tag{3.25}$$

where τ_{max} was the stall torque of the motor, ω_{max} was the no load speed of the motor, F_{max} was the maximum force of the linear actuator and $\dot{l}_{max}$ was the no load speed of the linear actuator. It must be noted that these equations do not hold for the pneumatic actuator which is detailed in Chapter 6.

3.1.2.7 Terminal Conditions

These constraints varied depending on the optimisation task and are detailed in the relevant chapters. Example terminal conditions are given below and can be seen in Figure 3.9:

- Acceleration Trajectories: terminate at the apex of a steady-state trajectory:

$$\begin{aligned} \mathbf{q}_0(N) &= \mathbf{q}_0(1)_{steady-state\ trajectory} \\ \dot{\mathbf{q}}_0(N) &= \dot{\mathbf{q}}_0(1)_{steady-state\ trajectory} \end{aligned} \tag{3.26}$$

- Steady-State Trajectories: terminate to enforce periodicity:

$$\begin{aligned} \mathbf{q}_0(N) &= \mathbf{q}_0(1) \\ \dot{\mathbf{q}}_0(N) &= \dot{\mathbf{q}}_0(1) \end{aligned} \tag{3.27}$$

- Deceleration Trajectories: terminate at the rest position:

$$\begin{aligned} \mathbf{q}_0(N) &= \mathbf{q}_{rest} \\ \dot{\mathbf{q}}_0(N) &= \dot{\mathbf{q}}_{rest} = 0 \end{aligned} \tag{3.28}$$

- Long-Time-Horizon Trajectories: terminate at the rest position:

$$\begin{aligned} \mathbf{q}_0(N) &= \mathbf{q}_{rest} = \mathbf{q}_0(1) \\ \dot{\mathbf{q}}_0(N) &= \dot{\mathbf{q}}_{rest} = 0 \end{aligned} \tag{3.29}$$

The above constraints do not apply to the "x" generalised coordinate in $\mathbf{q}$, as the robots were required to travel a certain distance.

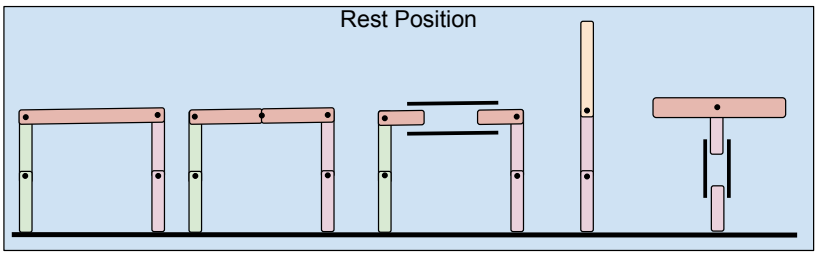

Figure 3.9: The different rest positions for the relevant robots used through out this research are shown.

3.1.2.8 Average Velocity

For the steady-state steps, an average velocity constraint was enforced (note that the initial and terminal velocities were not specified, only the average was constrained). This was enforced using the following constraint:

$$\dot{x}_{average} = \frac{x_0(N) - x_0(1)}{tt_0(N) - tt_0(1)} \tag{3.30}$$

where $x_0(N)$ was the final distance reached and $x_0(1)$ was the initial starting point ($x_0(1) = 0$). $tt_0(N)$ was the final time and $tt_0(1)$ was the starting time ($tt_0(1) = 0$). The constant $\dot{x}_{average}$ was the enforced average velocity.

3.1.2.9 Time Upper Bound

In order to enforce a rapid manoeuvre, an upper bound was placed on the total time allowed for the trajectory. This was implemented using the following constraint:

$$\sum_{i=1}^{N} h(i) < T_{max} \tag{3.31}$$

The upper bound on time, T_{max}, was hand chosen, by manually reducing it until the optimiser could no longer find a feasible solution.

3.1.2.10 Bounds

To restrict the problem size, bounds were enforced on all the decision variables. The optimiser varies these variables between their bounds to find a trajectory that satisfies the above constraints and minimises the specified cost function.

The ground reaction forces, $\boldsymbol{\lambda}$, were bounded to five times the mass of the robot, $5m_{robot}g$, where g was the gravity constant, 9.81 m/s^2. This ensured that excessively large ground reaction forces were not experienced by the platforms, which could result in failure of the mechanical structure.

All initial conditions are enforced using bounds (whereas the terminal conditions were enforced through constraints) by setting the upper and lower bound of the first node to the desired value. Some example initial conditions for the different tasks were as follows:

- Acceleration Trajectories (Start at the rest position):

$$\begin{aligned} \mathbf{q}_0(1) &= \mathbf{q}_{rest} \\ \dot{\mathbf{q}}_0(1) &= \dot{\mathbf{q}}_{rest} = 0 \end{aligned} \tag{3.32}$$

- Steady-State Trajectories (Enforce the apex condition):

$$\begin{aligned} \dot{z}_0(1) &= 0 \\ x_0(1) &= 0 \end{aligned} \tag{3.33}$$

- Deceleration Trajectories (Start at the apex of the steady-state trajectory):

$$\begin{aligned} \mathbf{q}_0(1) &= \mathbf{q}_0(1)_{steady-state\ trajectory} \\ \dot{\mathbf{q}}_0(1) &= \dot{\mathbf{q}}_0(1)_{steady-state\ trajectory} \end{aligned} \tag{3.34}$$

- Long-Time-Horizon Trajectories (Start at the rest position):

$$\begin{aligned} \mathbf{q}_0(1) &= \mathbf{q}_{rest} \\ \dot{\mathbf{q}}_0(1) &= \dot{\mathbf{q}}_{rest} = 0 \end{aligned} \tag{3.35}$$

For the above constraints, the "x" generalised coordinate in $\mathbf{q}$ was always set to zero. The remaining variables that are not bounded through constraints were bounded sufficiently high as to not bias the solution but to restrict the size of the problem.

3.1.2.11 Decision Variables

The above methods result in a large number of complementarity constraints that were solved using *slack* variables. The decision variables for the optimisation problem were as follows:

$$\mathbf{x} = [\mathbf{q}(i,j); \dot{\mathbf{q}}(i,j); \ddot{\mathbf{q}}(i,j); \mathbf{q}_0(i); \dot{\mathbf{q}}_0(i); \boldsymbol{\tau}(i); \mathbf{h}(i); \mathbf{tt}(i,j); \mathbf{tt}_0(i); \boldsymbol{\lambda}(i,j); \mathbf{slack}(i,j)] \tag{3.36}$$

where $\mathbf{q}(i,j)$ and its derivatives, along with $\boldsymbol{\tau}(i)$ and $\boldsymbol{\lambda}(i,j)$ were from the EoM, (3.1), $\mathbf{h}(i)$ was the time period between the node points and the $\mathbf{slack}(i,j)$ variables were due to the complimentary constraints. As higher order collocation methods (detailed above) were implemented to improve the accuracy of the results and integrate from node point to node point, the optimiser required access to all the state variables ($\mathbf{q}(i,j)$ and $\mathbf{q}_0(i)$ and their derivatives). The difference between $\mathbf{q}(i,j)$ and $\mathbf{q}_0(i)$ was that $\mathbf{q}(i,j)$ acts on i (node point) and j (collocation point), whereas $\mathbf{q}_0(i)$ only acts on i. $\mathbf{tt}(i,j)$ contained the total time from node 1 to the i^{th} node and j^{th} collocation point, whereas $\mathbf{tt}_0(i)$ contained the total time from node 1 to the i^{th} node point.

3.2 Validating Controllers and Verifying Models

In order to validate the designed controllers, a hybrid pneumatic-electric monopod was developed. As a model of the pneumatic actuator was required, physical tests were performed on the cylinder, with all the required data being logged at 1 kHz. A simple model was then curve fitted to the data and used in the trajectory optimisation model, see Chapter 6 for more detail.

3.2.1 Controller Design

In order to develop controllers for the robot, trajectories were first generated for the long-time-horizon task, using trajectory optimisation methods. These were thoroughly analysed to determine if heuristics or controllers, such as Raibert or PD controllers, naturally emerged from the trajectories. Least squares methods were implemented to tune PD controllers for the stance phase. Once these controllers were determined and tuned, they were simulated on the robot using trajectory optimisation methods to determine if they were adequate to perform the long-time-horizon task. See Chapter 6 for more details.

For each phase (acceleration, steady-state and deceleration) a unique stance controller was generated. When the robot was aerial, the Raibert controller was implemented. A state machine to control the pneumatic cylinder was implemented, using the leg angle, length and height of the robot (to determine if the foot was on the floor), the states of the state machine were transitioned. To test the robustness of the controllers, disturbances were added during the acceleration and deceleration phase.

3.2.2 Support Rig

In order to investigate transient locomotion, such as rapid acceleration and deceleration, a method to test these motions needed to be developed. Generally, initial investigations use planar robots that need to be constrained in the sagittal plane using a boom or support rig, with robots such as the MIT cheetah performing steady-state locomotion studies on treadmills in the laboratory.

For robotics, there are a number of critical problems with testing on treadmills (as discussed in Chapter 2). It has been noted that when accelerating on a treadmill, the linear inertia is zero compared to overground acceleration (you effectively hop into the air and swing your legs faster to accelerate on a treadmill, compared to accelerating your mass when running overground) [101]. However, it has been shown that for constant velocity motions, running on a treadmill is mechanically equal to overground running [101].

Some techniques have been developed to narrow the gap between accelerating on ground and on a treadmill. These include inertia compensated treadmills (they include a band that adds an external force on the torso) [125]. Some of the downfalls to this method is that the inertia effects are experienced by the whole body when accelerating over ground, however, with inertia treadmills, the effects are only experienced where the belt is attached to the torso. These systems are costly to purchase or develop. Other systems make use of motorless treadmills that create resistance using magnetic force [126].

Therefore, for the majority of research groups, these treadmills are not an option to test transient manoeuvres. Instead boom like systems are used, where instead of a robot accelerating in a straight line, they are constrained to run in a circle and the boom supports the robot in the sagittal plane. The longer the boom arm, the straighter the robot runs. The downfall with these systems is the added mass of the boom that the robot must support and accelerate.

As this study focuses on planar motion, a support rig was developed by Joshua Van Zyl (an MSc student at the University of Cape Town, supervised by A/Prof. Patel) to keep the robot in the sagittal plane. The support system consisted of a four bar mechanism (allowing vertical motion, constrained to an arc) capable of pivoting (allowing forward motion, constrained to an arc). Due to the length of the boom, over 2 m, the horizontal and vertical motion was considered linear. The support system had a number of sensors to determine the horizontal and vertical velocity and position through the use of a Kalman Filter [127]. These sensors included encoders (4096 counts per revolution), an accelerometer (MPU 9250) and a LIDAR (Garmin LiDAR Lite v3). The system can be seen in Figure 3.10

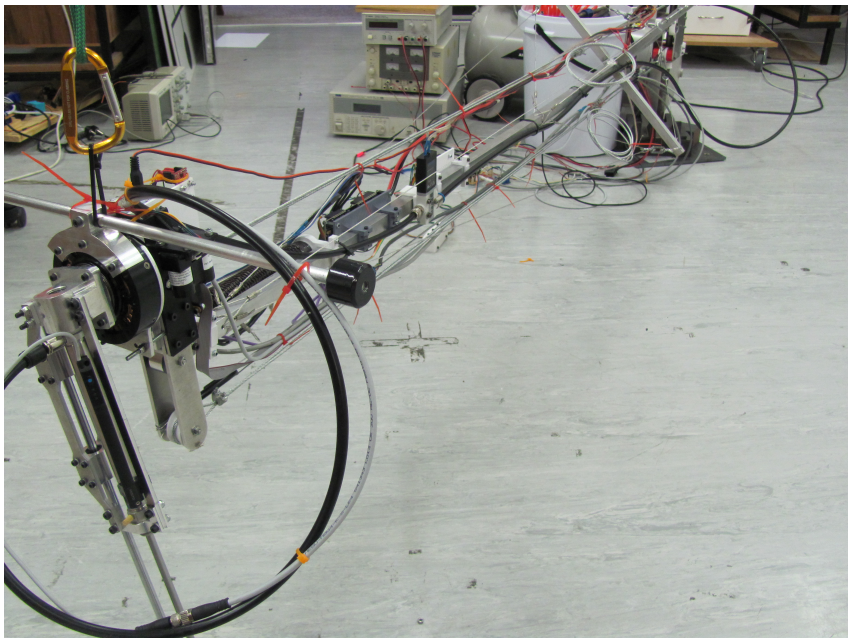

Figure 3.10: The boom system developed by Joshua Van Zyl. It made use of cables to keep the centre beam under compression. The boom was free to pivot allowing forward and vertical motion of the robot.

3.2.3 Hybrid Pneumatic-Electric Robot

The robot was designed in SolidWorks and built by Joshua Van Zyl. The robot can be seen in Figure 3.11. The robot consisted of a pneumatic cylinder, Festo $DSNU-16-125-V-A-Q$, attached to a support rail for the leg length. The cylinder and support rail was attached to a geared DC motor, T-Motor $A80-6$ (with a gear ration of 6 and an O-Drive motor controller), allowing the leg to rotate using position or torque control. An encoder attached to the motor gave leg angle feedback, with a position sensor attached to the leg giving leg length feedback

(using hall sensors, Festo $SDAT - MHS$). The entire robot was attached to the boom system via a pivot with an encoder attached, allowing the body of the robot to rotate. The robot parameters, as well as boom parameters can be seen in Table 3.1. The robot was controlled using a Teensy 4.0 micro controller with a control loop of 2 kHz, however, motor commands were only sent every 500 Hz. The designed controllers were implemented on the robotic platform and compared to the simulation results.

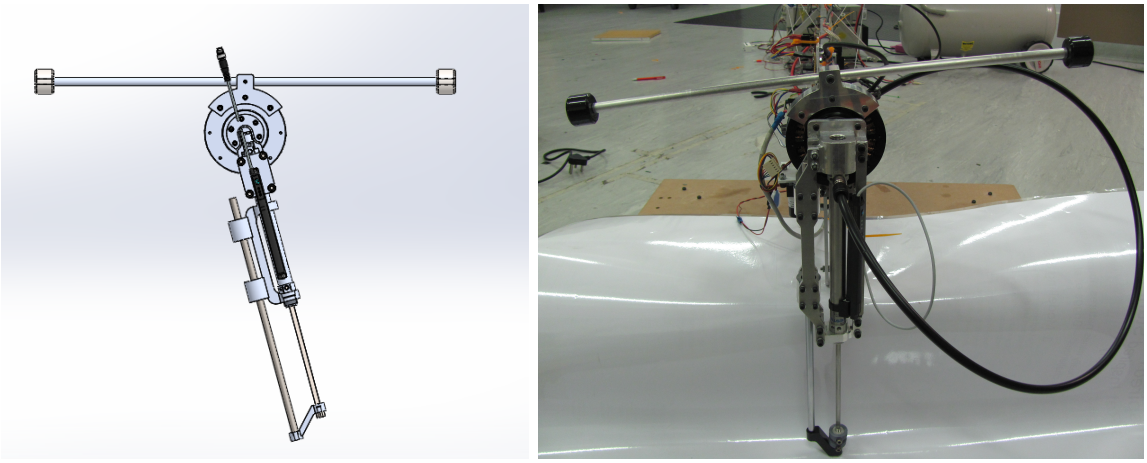

Figure 3.11: A SolidWorks render of the monopod robot that was used to test controllers on is shown on the left with the physical robot on the right. The robot consisted of a pneumatic leg (with a support rail system) with a torque actuator at the hip. The body consisted of a beam with mass attached to the end to increase the mass and inertia of the body. The robot was designed and built by Joshua Van Zyl.

Table 3.1: Robot and Boom Parameters

I_B	0.01734 kgm^2
m_B	850 g
$I_{L\ min}$	0.00640 kgm^2
$I_{L\ max}$	0.01671 kgm^2
m_L	385 g
L_{min}	235 mm
L_{max}	360 mm
τ_{stall}	12.25 Nm
$\omega_{no\ load}$	44.5 rad/s
F_{max}	160 N
Boom parameters	
Length	2 m
Weight	1.442 kg
Height	375 mm
I_{xx}	2.8614 kgm^2
I_{zz}	2.8572 kgm^2

Chapter 4

How Should Legged Robots Rapidly Accelerate and Decelerate?

Rapid transient locomotion (acceleration and deceleration) is a vastly complex topic involving highly non-linear hybrid dynamic systems performing fast and complex manoeuvres. Transient locomotion is poorly understood in the robotics community as they involve aperiodic motion and unknown foot contact order.

The robotic literature seems to contradict observations made in nature when it comes to how legged systems should accelerate and decelerate. Observations in nature (the greyhound [40, 128] and humans [44]) suggest a direct transition from the current gait (or rest) to the desired gait (or top speed), whereas the robotic literature [9, 10, 11] takes the approach of multiple gait transitions (walking to bound, bound to gallop etc.).

This Chapter aims at answering the question: ***Should multiple gait transitions be used when rapidly accelerating or decelerating?*** In this chapter, a number of experiments were performed to answer this question. These experiments included:

- Steady-state steps were generated for the biped and quadruped robot.
- Acceleration trajectories were generated from rest to the apex of the steady-state step.
  - Multiple cost functions were used and compared.
  - Multiple time periods for each cost function were enforced to compare different rapid trajectories.
- The same set of experiments were performed for the deceleration trajectories.
- These results were stitched together and compared to long-time-horizon trajectories.

The emerging gaits were analysed and compared to determine how these legged robots should rapidly accelerate and decelerate (multiple gait transitions or launch straight to the desired gait). Figure 4.1 provides an overview of this chapter and the robots used.

The below results were published in [129].

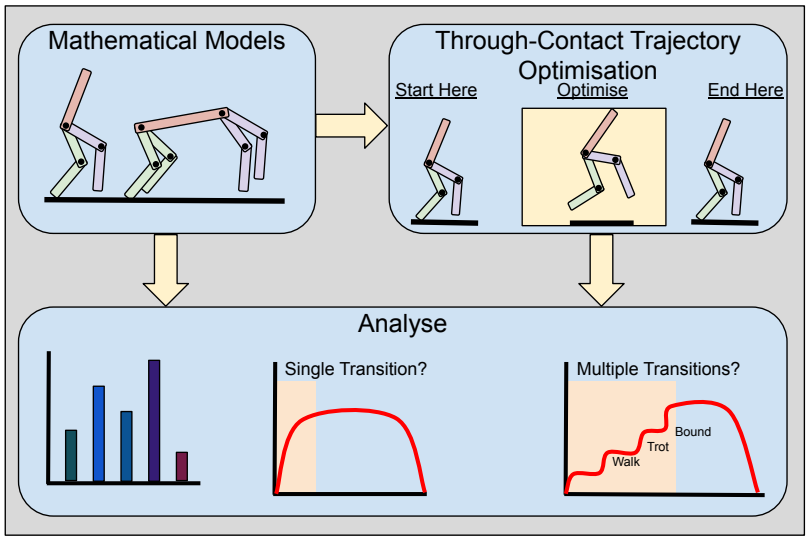

Figure 4.1: Multiple acceleration and deceleration trajectories were generated in this chapter with different cost functions and time constraints to determine the optimal method for rapid acceleration and deceleration. These were then analysed to draw conclusions and determine if multiple gait transitions are optimal or if robots should launch straight into their desired gait/speed.

4.1 Determining How Legged Robots Should Rapidly Accelerate and Decelerate

Due to the complexity of the models (and as only the acceleration and deceleration phases were of interest) the long-time-horizon trajectory was split into three distinct phases: acceleration, steady-state steps and deceleration. First steady-state steps (with the CoT, J_2, cost function) were generated for the bounding gait for the quadruped and running gait for the biped, using a fixed contact order, with an average velocity enforced to 5 m/s. The reason why the bound gait was used instead of the galloping gait for the quadruped was due to the planar dynamics which are suitable for gaits without lateral or roll dynamics [33]. The out of phase nature of the galloping gait results in lateral and roll dynamics, requiring a three dimensional model [35]. The generalised coordinates for the two legged quadruped was modified for this planar four legged model, q_{leg} (3.3), was duplicated to represent four legs.

As multiple cost functions were being compared for the acceleration and deceleration trajectories, it was decided to make these trajectories start and end at the same point (either rest configuration or the apex of a steady-state trajectory). Therefore, only one cost function was used to generate

a steady-state trajectory for the biped and quadruped robot.

The following modifications (same for the biped and quadruped), for the steady-state step, were made to the trajectory optimisation method described in Chapter 3:

1. Number of nodes: $N = 50$
2. Expected total time: $T = 0.3\ s$
3. Cost function: J_2 was used
4. Fixed contact order was enforced
5. Average velocity was enforced at 5 m/s.

Following the generation of the steady-state steps, the acceleration and deceleration trajectories were generated with different maximum total times (enforced using constraints) for the two cost functions (CoT, J_2, and heat based cost, J_3). For each time period, 10 trajectories were generated, with the lowest cost trajectory being selected as the optimal trajectory for that time period and cost function. The gaits that naturally emerged under the different time constraints and cost functions were analysed for trends. As the maximum total time decreased, it enforced a rapid trajectory to emerge. A range of maximum time periods were used, ranging from 2 seconds to 6 seconds (identified by trial and error). For these trajectories, a contact order was not enforced and was selected by the optimiser.

The following modifications, for the acceleration and deceleration trajectories, were made to the trajectory optimisation method described in Chapter 3:

1. Number of nodes: $N = 150$
2. Enforced maximum total times: $T = [6, 5, 4, 3, 2]\ s$
3. Cost function: J_2 (CoT) and J_3 (heat based) were used.
4. Upper bound on time to enforce rapid acceleration/deceleration.

The acceleration, multiple steady-state steps and the deceleration trajectories were stitched together to form the long-time-horizon trajectory (Section 3.1.2). The acceleration trajectories started from rest and terminated at the apex of the steady-state step and the deceleration trajectories started from the apex of the steady-state step and terminated at rest.

In order to compare the stitched results to a full trajectory generated by the optimiser, long-time-horizon trajectories were optimised and the results analysed. The following modifications were made to the trajectory optimisation method described in Chapter 3:

1. Number of nodes: $N = 300$
2. Expected total time: $T = 7\ s$
3. Upper bound on total time set to 7 s
4. Cost function: J_2 was used.

As can be seen, stitching allowed for higher accuracy (more nodes) in the optimisations while still remaining tractable. In total, 150 nodes were used for each transient trajectory and 50 nodes for the steady-state step, whereas the long-time-horizon trajectories that were generated used 300 nodes (and contained multiple steady-state steps).

4.2 Analysis of How Legged Robots Should Accelerate and Decelerate

An animation of the results can be found here:
`https://www.youtube.com/watch?v=QhJD2Es7UlU`.
The CoM (of the torso) velocity profiles of the acceleration trajectories for both cost functions and different time duration's (shortest time period of 2 seconds and longest of 6 seconds) are shown in Figure 4.2. For the CoT trajectories, a walking gait naturally emerged for the long time periods (6 seconds). Therefore, the energy expended to cover the distance was optimal at that velocity. However, for the shorter time period, the CoT trajectories launch straight into the top speed, as seen for the heat based cost function. When time was not a limiting factor, it was observed for the heat based trajectories that the robots remain standing stationary until the last moment and then launches straight into the highest speed gait.

4.2.1 Gait Analysis

For the CoT trajectories for the longest time periods, 6 seconds, the robots remained grounded (walking) until the quadruped robot reached 1.7 m/s, and then proceeded to accelerate to the desired speed. This resulted in the quadruped walking for 81 % of the trajectory. The biped walked until it reached a speed of 2.3 m/s, resulting in walking for 80 % of the trajectory.

The same CoM (of the torso) velocity profiles for the deceleration trajectories are shown in Figure 4.3. Once again a walking gait emerged for the CoT trajectories with the quadruped remaining grounded from 1.37 m/s (slower than the acceleration trajectories) and walked for 93 % of the trajectory. The biped remained grounded from 1.7 m/s resulting in walking for 92 % of the trajectory. Once again the heat based trajectories decelerated as quickly as possible to the rest position (along with the time constrained CoT trajectory).

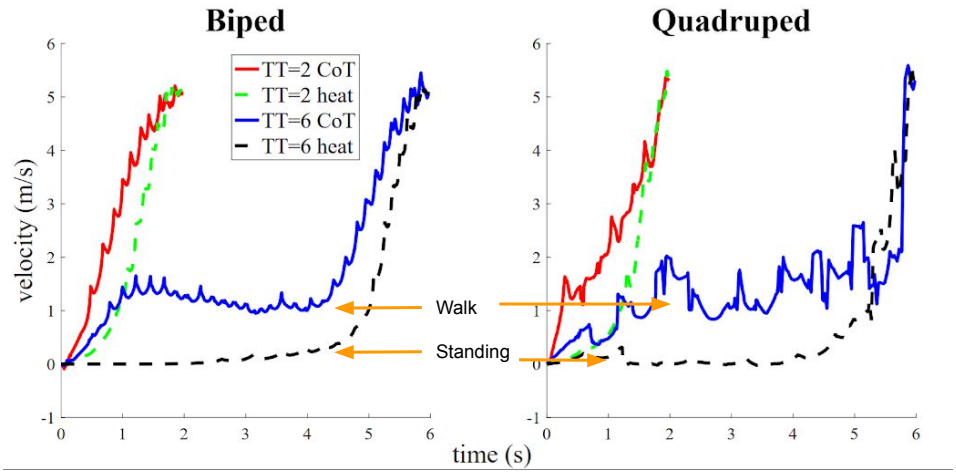

Figure 4.2: A comparison of the acceleration trajectories for the two cost functions at different time period constraints (2 seconds and 6 seconds). Note how the heat based acceleration trajectory for the long time period remained stationary until it accelerates straight to the highest speed gait, whereas the cost of transport trajectory walks first and then accelerates to the top speed. Results are consistent across the biped and quadruped models and matched observations in nature.

These trajectories require a number of steps to accelerate from rest to their top speed, however, no periodic limit cycles emerged and were, therefore, considered a single gait transition. The quadruped required approximately 8 steps, 2 seconds, to accelerate and 3 steps to stop. The biped required 10 steps to accelerate and 3 to decelerate.

For the short time constrained trajectories, both the cost functions resulted in the robots launching (accelerating) straight into their desired gait without any intermediate gaits (which is often done in the literature [9, 10, 11, 130]). This matches observations made in nature [40, 128, 131]. For the longer time periods, the CoT trajectories, the robots first walked and then at the last moment launched into the desired gait, with the same happening for the deceleration trajectories. The walk remained a roughly constant velocity, with a noticeable transition from walk to the desired gait, without performing a slow ramp in speed as is often done in the literature [9, 11]. On the other hand, for the heat based cost function with the longer time periods, the robots remained stationary until the last moment where they launched straight into their desired gait. The same was observed for the deceleration trajectories. Stitching these trajectories together resulted in a long-time-horizon trajectory that can be seen in Figure 4.4.

For all of the above results, the robots launched straight into their desired gaits, without multiple transitions. The quadruped robot avoided gaits such as the trot. It is hypothesised that it is due to the lack of roll/yaw dynamics in the planar model. These results raise concerns about the required power to achieve these motions. This power must be generated by the actuators and the mechanical structure must be strong enough to handle the loads. As a motor model was implemented, these trajectories are realizable on a physical platform, however, for future platform development it is advised to analyse these loads and to develop a platform around these requirements. Previous studies have shown that transient locomotion requires special design considerations [60, 122, 132].

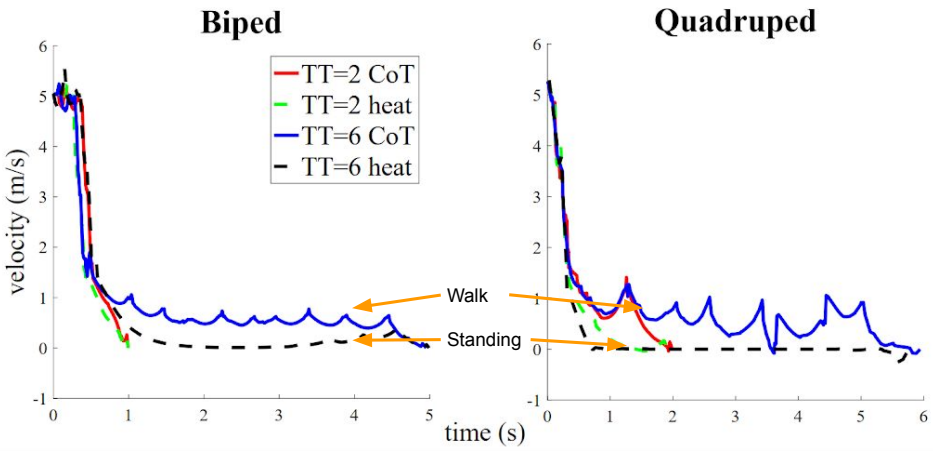

Figure 4.3: A comparison of the deceleration trajectories for the two cost functions at different time period constraints (6 seconds and 2 seconds). Note how the heat based acceleration trajectory for the long time period decelerates straight to rest and remained stationary until the end, whereas the cost of transport trajectory decelerates to a walking gait and then to rest. The results are consistent across the biped and quadruped models and observed in nature.

4.2.2 Comparison to Long-Time-Horizon Trajectories

The above results were generated by stitching multiple trajectories together and did not naturally emerge from the optimiser (acceleration, steady-state and deceleration phases were enforced). Therefore, full long-time-horizon trajectories were also generated to determine what naturally emerges from the optimisations. These trajectories started and ended at rest with the robot travelling a fixed distance. These results can be seen in Figure 4.5 and validate what was observed in the acceleration and deceleration results above, with the robots still launching straight into the desired gait.

4.2.3 Validation of Through-Contact Methods

Many previous works prescribe foot contact order and do not allow for slipping. To validate the need for these through-contact optimisation methods that allow the feet of the robot to slip, an example of the effective friction can be seen in Figure 4.6. This highlights the fact that the foot often lies on the friction cone (when $effective\ lambda = 1$), indicating that the foot can begin to slip, as seen in Figure 4.7.

4.3 Template Identification

From analysing the trajectories in Figure 4.2, 4.3 and 4.5, a pattern emerged for the velocity profiles. Looking at the velocity trajectory of a "bang-coast-bang" [133] like control (minimum

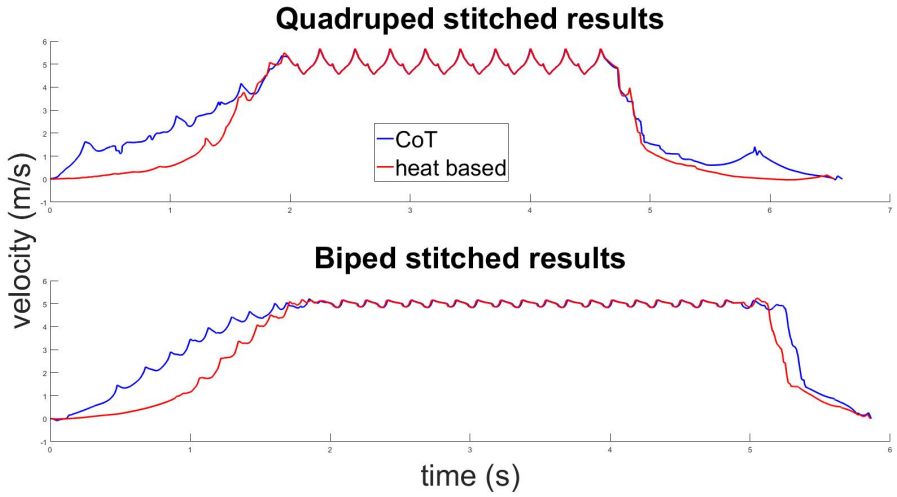

Figure 4.4: Stitching the acceleration, steady-state and deceleration trajectories generates the long-time-horizon trajectory. As the steady-state trajectories are periodic, multiple can be stitched together to cover a larger distance.

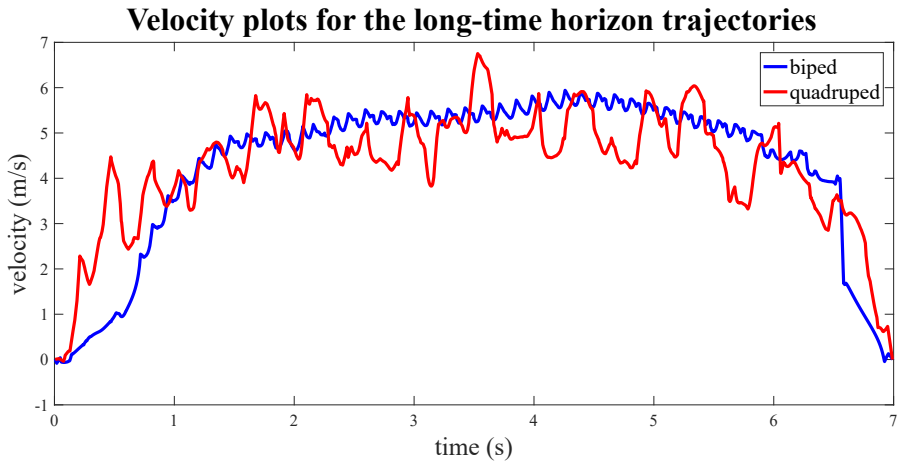

Figure 4.5: The long-time-horizon trajectories resemble the stitched results. It is interesting to note that these trajectories also accelerated straight to the top speed gait with no intermediate gait transitions. These results validate the *stitching* approach to generating long-time-horizon trajectories.

time) problem of a sliding mass [43], under the influence of a time varying force in the presence of viscous friction, a similar profile emerged.

Previous research by Hubicki et al. revealed that the velocity profile of a single monopod hopper matched the profile of the sliding mass "bang-coast-bang" problem. They further hinted that this template may hold for more complex models [43]. This was proved in the above results and was published in [129]. The dynamics of the sliding mass problem were as follows:

$$\ddot{x}(t) = \frac{F(t) - c\dot{x}(t)}{m} \tag{4.1}$$

where the time varying force applied to the mass, m, was $F(t)$ and the viscous friction was

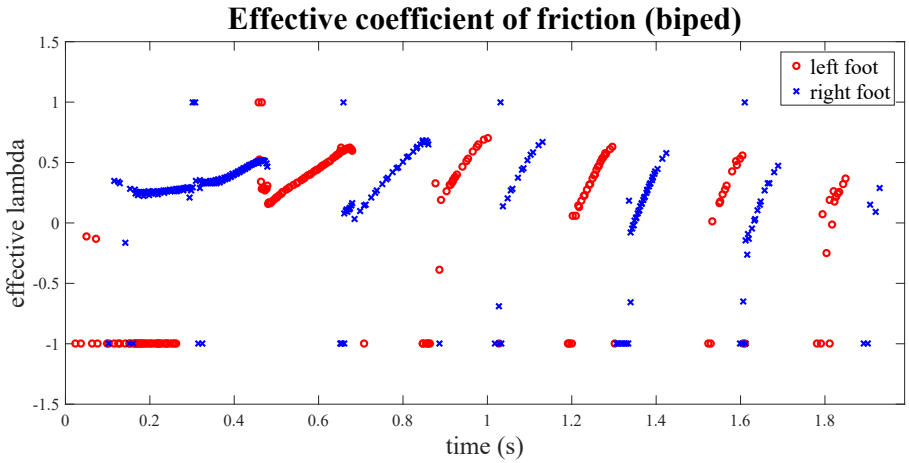

Figure 4.6: Note how the friction is often clipping at ± 1 (the coefficient of friction). At this point the foot can begin to slip, proving the need for these complex through-contact methods that enable the foot to slip.

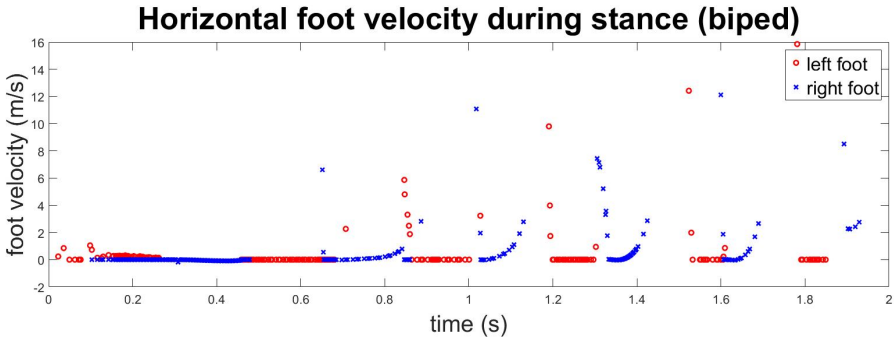

Figure 4.7: The magnitude of the horizontal foot velocity while the foot is in stance. Note how it tends to slip when the friction cone limit is reached.

represented by c. $x(t)$ was the position with $\dot{x}(t)$ the velocity and $\ddot{x}(t)$ the acceleration of the mass. The velocity profile of the long-time-horizon trajectory was compared to that of the sliding mass and is shown in Figure 4.8. It can be seen to match fairly accurately (considering the sliding mass model is a 1 DoF model).

In terms of robotics, this is an interesting result that shows that a simple model can be used to generate fairly accurate set points for more complex controllers based on the full DoF of the robot. Therefore, a simple sliding mass model can be generated in real time and be used as a set point for higher level planners to include these acceleration and deceleration motions. As the results match what is observed in nature [40, 128], this further reveals that animals are inherently minimising some form of CoT when performing these complex long-time-horizon trajectories.

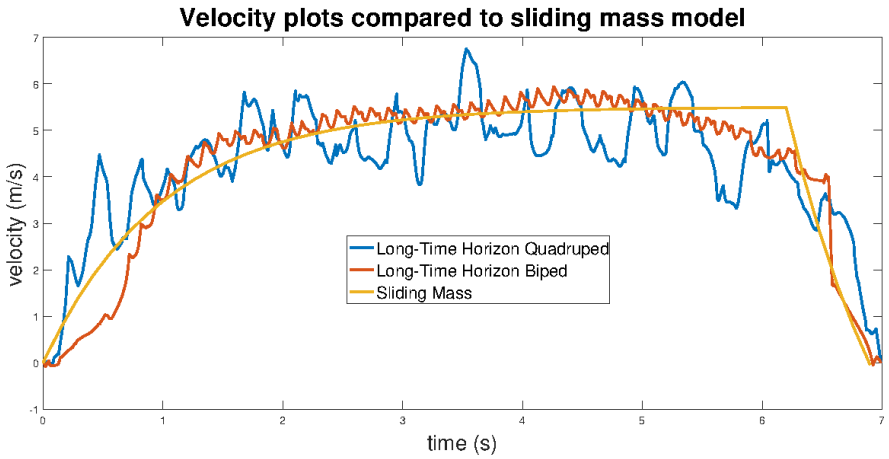

Figure 4.8: Note how the simple sliding mass 1 DoF model accurately tracks the forward velocity of the acceleration and deceleration manoeuvre of the biped and quadruped robot. This model can be used as a heuristic for higher level online control planning.

4.4 Summary of Discussion

These results show that it is optimal to launch straight into the desired gait or velocity when rapidly accelerating or decelerating for the current set of parameters and model. From these trajectories, a sliding mass template emerged which matches the hypothesis made in [43]. Multiple trends emerged across the different legged robots, namely, multiple gaits were not required, only a slow walk and fast run. When time was not a limiting factor, the slow walk was preferred, when time was crucial, the robots launched straight into the desired gait. These results match observations made in nature.

Chapter 5

What is the Optimal Spine Morphology for Rapid Acceleration and Deceleration Manoeuvres?

In nature, the spine has been seen to improve locomotion, however, it has only recently become popular in the robotics literature. These spines have generally been focused on steady-state locomotion and energy efficient manoeuvres [22, 79]. The effect of spine morphology on transient manoeuvres such as acceleration and deceleration has not been explored. This is partly due to the complexities involved in studying and modelling these manoeuvres and morphologies.

The design question of interest in this chapter is: ***When designing a legged quadruped robot to perform transient motions (such as rapid acceleration and deceleration) what spine morphology should be employed?*** In this chapter, a number of experiments were performed to answer this question. These experiments included the following:

- Generate 100 random robot parameter sets (scaled versions of one-another).
- Performed large-scale Monte Carlo simulations.
  - The long-time-horizon trajectory was optimised.
  - Three spines of interest were compared, namely the rigid, revolute and prismatic spine.

These results were analysed using Bootstrapping statistical methods to formulate general assumptions, which aimed at answering the proposed question. Special focus was given to the GRF, stride-length and stride time to gain an understanding of the results. The image in Figure 5.1 shows an overview of this chapter. The "X" type two link leg configuration was used for these Monte Carlo simulations.

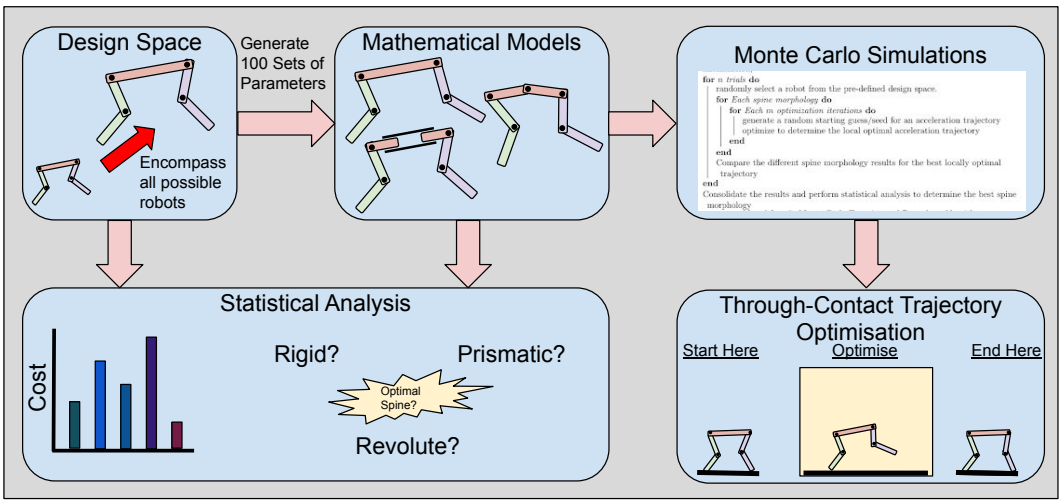

Figure 5.1: An overview of the main research question for this chapter. First 100 random robot parameters (scaled versions of one-another) were generated, each parameter set containing a rigid, revolute and prismatic spine. Large-scale Monte Carlo simulations were then performed for the long-time-horizon task. The results were compared and Bootstrapping techniques were applied to make general assumptions.

The methods and results described below have been published/submitted in [132, 134].

5.1 Determining the Optimal Spine Morphology

In order to determine the optimal spine morphology for rapid manoeuvres such as acceleration and deceleration, large-scale Monte Carlo simulations were performed. To date only a small number of large-scale Monte Carlo simulations have been performed and used to make general assumptions, such as the optimal leg bend direction [1, 73, 74]. These studies however, have either focused on simple models [73] or focused on short periodic trajectories [74], and have not analysed a full trajectory. For this study, the task being optimised was the long-time-horizon task which involved the robots starting and ending at rest and travelling a fixed distance of 30 body lengths. The long distance ensures a relatively steady-state velocity emerged (the velocity was seen to level off after the acceleration phase to a near steady-state velocity).

Three spine morphologies were investigated, namely the rigid, revolute and prismatic spine, as seen in Figure 5.2. Due to the infinite number of possible robot parameters, it is intractable to analyse them all, or to even analyse a fine grid search of the parameter space. Therefore, it was decided to randomly generate 100 random robots (scaled versions of one-another) from the parameter set (see Table 5.1), each parameter set containing the rigid, revolute and prismatic spine. The large-scale Monte Carlo Simulation algorithm can be seen in Algorithm 1.

These simulations were performed on 12 identical computers over a period of 3 months. In order to improve the chance of generating feasible robots, the parameter bounds were scaled

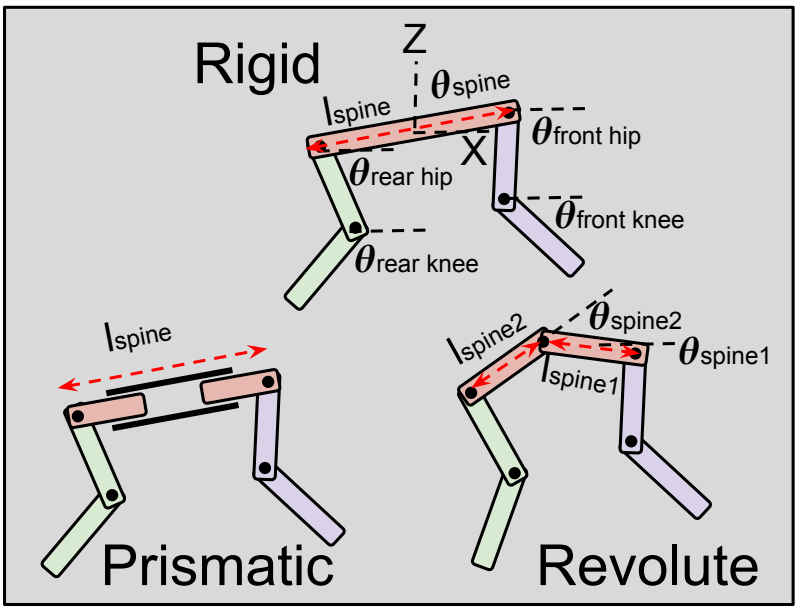

Figure 5.2: The three spines of interest are shown. Note all three spine morphologies use the same leg configuration with the same mass and power for the legs. The horizontal and vertical position of the robot was described by X and Z which was located at the COM of the spine (for the revolute spine, X and Z are located on the spine link connected to the front legs). For the revolute spine, θ_{spine1} was an absolute angle whereas θ_{spine2} was a relative angle. A torque actuator, τ, connects the two spine links. This was an active spine with no passive elements. For the prismatic spine, the spine length can lengthen and shorten through the applied force, F.

according to the spine length, l_{spine}, and mass, m_{spine} (which was randomly generated between its bounds). Therefore, robots with a small spine were not extremely heavy or had large legs. The mass was then linearly scaled to a spine length between the bounds (maximum spine mass resulted in the maximum spine length). These parameters were then used to generate the remaining parameters according to Table 5.1.

Once a set of parameters was generated, the equations of motion (see Section 3.1.1) for each spine morphology were generated using Euler-Lagrange dynamics. The generalised coordinates can be seen in Figure 5.2. The trajectory optimisation method can be found in Chapter 3.1.2. The specifics for this section include the following:

1. $N = 300$ nodes with 30 seeds
2. As the expected total time for the trajectory was unknown, 10 seeds with $T = 7$ s, 10 seeds with $T = 6$ s, 10 seeds with $T = 5$ s were optimised, where T was the expected time the trajectory would take
3. Start and terminal conditions: long-time-horizon conditions (start and end at rest, see Section 3.1.2.7)
4. Cost (minimum time): J_1

Data: Robot design space, n=100, m=30
Result: Optimal spine morphology
initialisation;
for *n trials* **do**
  randomly select a robot from the pre-defined design space.
  for *Each spine morphology* **do**
    for *Each m optimisation iterations* **do**
      generate a random starting guess/seed for an acceleration trajectory
      optimise to determine the local optimal acceleration trajectory
    end
  end
  Compare the different spine morphology results for the best locally optimal trajectory
end
Consolidate the results and perform statistical analysis to determine the best spine morphology

Algorithm 1: Monte Carlo Experimental Procedure Algorithm

5.2 Optimal Spine Morphology Results and Discussion

In total 9 000 trajectory optimisations were performed with an average convergence rate of 19 % which was comparable to previous studies [74, 132]. Of the 100 robots, 6 had no convergences for any of the spine morphologies. Each successful optimisation took between 4 to 5 hours, with unsuccessful seeds exiting after approximately an hour. Animations of the results can be found here: `https:/youtu.be/ElBvS13a6pw`.

Of the 30 seeds run per spine morphology per robot, the best (in terms of the cost function) was chosen as the optimal result. Figure 5.3 shows a velocity trajectory of a randomly sampled result, revealing that the robot accelerates to steady-state and then decelerates back to rest (a common trend among all the results). This shows that the results start and end in a feasible configuration (due to the constraints of the long-time-horizon task) and can, therefore, be implemented on a physical platform.

5.2.1 Bootstrapping Results

Bootstrapping techniques [135] (statistical analysis of the results using 10 000 samples with replacement) were performed on the optimal seeds and the results can be seen in Table 5.2. The trajectories were also split in two (15 body lengths each) to analyse the optimal spine morphology for the acceleration and deceleration phases of the trajectories.

These results revealed that the prismatic spine morphology was the optimal spine for the long-time-horizon trajectories 79 % of the time. Animations of these results were analysed for trends. These trends were found to match observations made in nature. Revolute robots were observed

Table 5.1: Design Space

Parameter	Value	Parameter	Value
Rigid Spine		Prismatic Spine	
m_{spine}	[1 30] kg	$l_{spine_{rest}}$	l_{spine} m
l_{spine}	0.06 m_{spine} m	$l_{spine_{min}}$	0.5 l_{spine} m
$l_{spine\ COM}$	0.5 l_{spine} m	$l_{spine_{max}}$	1 l_{spine} m
General		$l_{spine\ COM}$	0.5 l_{spine} m
l_{femur}	0.02 m_{spine} m	m_{spine}	m_{spine} kg
$l_{femur\ COM}$	0.5 l_{femur} m	F_{spine}	10 m_{spine} N
m_{femur}	0.05 m_{spine} kg	Revolute Spine	
l_{tibia}	0.04 m_{spine} m	l_{spine_1}	0.5 l_{spine} m
$l_{tibia\ COM}$	0.5 l_{tibia} m	$l_{spine\ COM_1}$	0.5 l_{spine_1} m
m_{tibia}	0.05 m_{spine} kg	m_{spine_1}	0.5 m_{spine} kg
τ_{hip}	5 m_{spine} Nm	l_{spine_2}	0.5 l_{spine} m
τ_{knee}	5 m_{spine} Nm	$l_{spine\ COM_2}$	0.5 l_{spine_2} m
ω_{max}	3 m_{spine} rad/s	m_{spine_2}	0.5 m_{spine} kg
τ_{max}	5 m_{spine} Nm	τ_{spine}	5 m_{spine} Nm

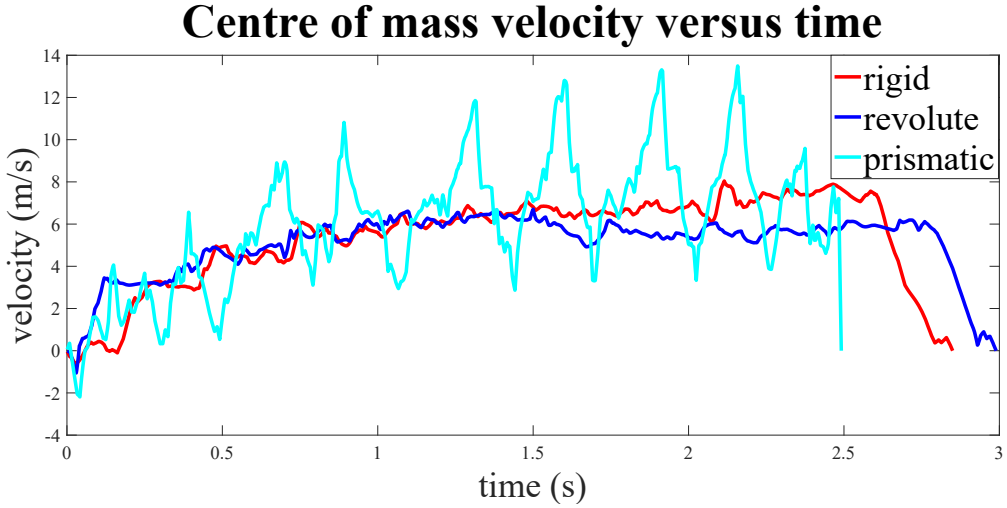

Figure 5.3: An example velocity profile for a successful trajectory. Note how the robots reach steady-state before decelerating back to rest. These results were generated from a random seed point. Note how the prismatic robot was the fastest for this trajectory as it reaches a higher average top speed. The reason why the prismatic result was more oscillatory was due to the fact that the CoM velocity was from the centre of the spine, which can lengthen and shorted due to the prismatic actuator.

to slightly bend their spines (pulling in their rear legs) before beginning to accelerate, whereas prismatic robots were observed to first contract their spine and pull in their rear leg before beginning to accelerate forwards. This added delay for the prismatic robot was one of the hypotheses as to why it was only 51.5 % the optimal morphology for the acceleration phase. The Author hypothesized that if these simulations were re-run with the spine starting fully contracted as opposed to the rest length (half way point), this percentage would increase in favour of the prismatic spine. Another alternative was to re-run these simulations and only optimise the acceleration portion of the trajectory. These avenues were not explored due to the limited availability of computational resources. During the deceleration phase, the prismatic

Table 5.2: Bootstrap Results (Minimum time results)

Morphology	Convergence Rate		
Rigid	13%		
Revolute	11%		
Prismatic	35%		
Bootstrap Results			
Morphology	% best	% best acceleration	% best deceleration
Rigid	3.0 ± 3.0	3.0 ± 3.0	3.0 ± 3.0
Revolute	18.2 ± 6.7	45.5 ± 8.6	3.0 ± 3.0
Prismatic	78.8 ± 7.1	51.5 ± 8.7	94.0 ± 4.1

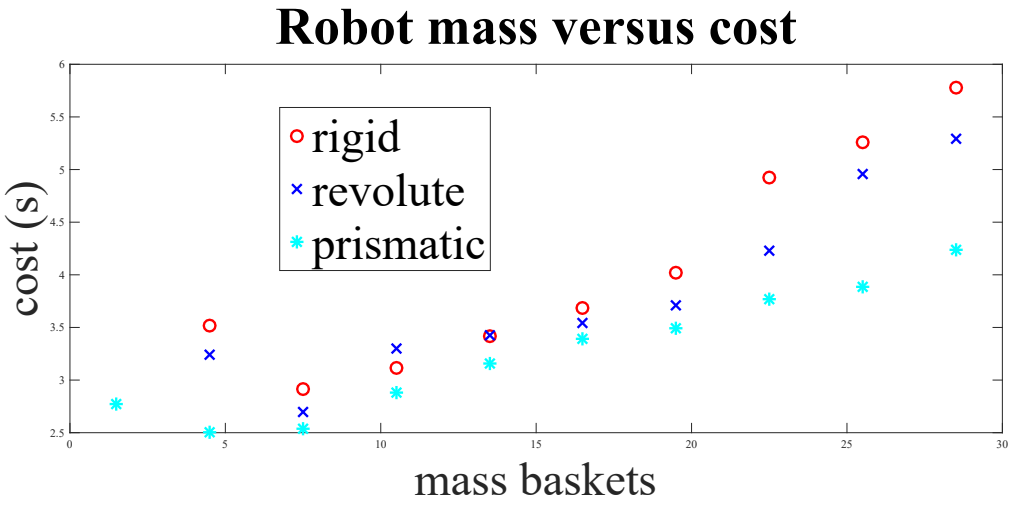

Figure 5.4: The mass versus cost (minimum time) for the optimal results reveal an interesting trend. As can be seen the prismatic spine was the optimal spine, which becomes more apparent as the robots got heavier.

spine was observed to contract and absorb the momentum of the rear half of the robot while keeping the front leg firm on the ground, bringing it to a stop quicker than the other spines. The rigid and revolute spines were observed to slide to rest, proving that our methods require the ability to allow the foot to slide.

5.2.2 Mass Related Results

To identify trends, the results were grouped into mass buckets and shown in the mass versus cost graph in Figure 5.4. The graph reveals that the prismatic spine was optimal over all masses, however, it becomes more apparent as the robots become heavier.

A paired t-test was performed on the data to determine its statistical relevance. The results are shown in Table 5.3. From the results, it can be seen that the null hypothesis was rejected, showing that the mean difference between the two data sets was not equal to zero. A single tailed t-test was performed showing that the prismatic spine had a lower mean than the rigid

and revolute spine (meaning it had a lower cost, minimum time, with respect to mass).

Table 5.3: t-test results

	Two-tailed	Left tail	Right tail
t-test(Rigid,Revolute)	1	0	1
t-test(Rigid,Prismatic)	1	0	1
t-test(Prismatic,Revolute)	1	1	0

5.2.3 Stride-Length and Frequency Observations

Observations made in nature [77, 78, 76] have shown that the spine increases the stride-length of animals. The stride-length and frequency of the results were analysed using Equation (5.1). Where $\#contacts$ was the number of times the feet touch the ground, $\sum_{i=1}^{N} h_i$ was the total time for the trajectory (the cost) and x_N was the distance travelled in the trajectory (30 spine lengths). These results are shown in Figure 5.5. As can be seen, the frequency remained roughly constant across the mass of the robots, whereas the stride-length increased with mass. It is interesting to note that the prismatic spine tends to have the lowest stride-frequency but the highest stride-length.

$$\begin{aligned} freq &= \frac{\#contacts}{2\sum_{i=1}^{N} h_i} \\ length &= \frac{2x_N}{\#contacts} \end{aligned} \tag{5.1}$$

In order to be the optimal spine, in terms of the minimum time cost, the robots either have to have the highest stride-averaged acceleration (to get to their top speed the quickest), highest stride-average deceleration (come to rest the quickest), the highest top speed, or some combination of the three. Therefore, the stride-average acceleration and deceleration, $\Delta\dot{x}/\Delta t$, was analysed (first and last 7 body lengths of the 30 body length trajectory) and shown in Figure 5.6. This figure shows that the prismatic spine clearly out performs the other spine morphologies and achieves a much higher stride-average acceleration and deceleration. This is partly due to the prismatic spines ability to increase the stride-length during acceleration and to compress and absorb energy during the deceleration phase.

5.2.4 Applied Work Results

The work done to achieve the forward motion [132] was calculated using Equation (5.2) and revealed that the prismatic spine robots exerted, on average, 37 % more forward work than the

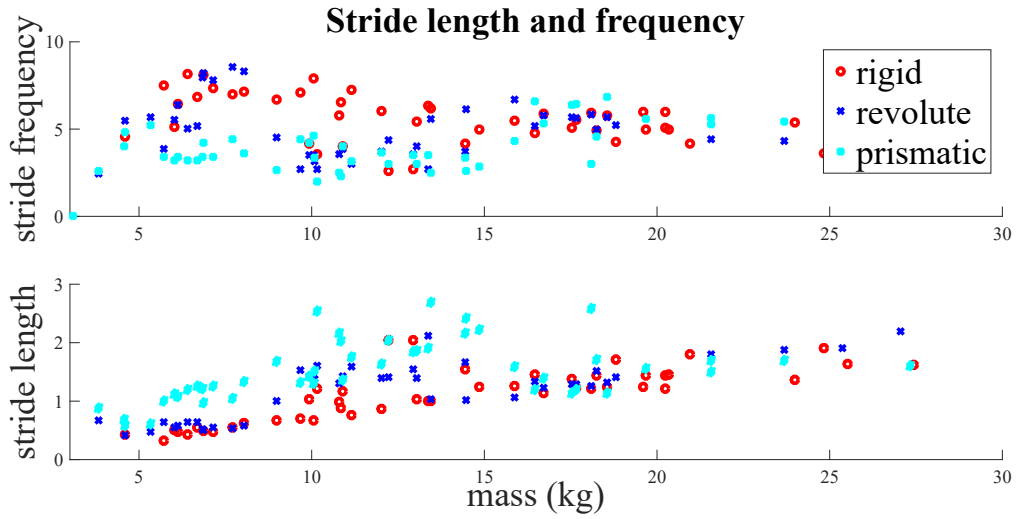

Figure 5.5: A comparison of the stride-frequency and stride-length for the three spine morphologies. It is interesting to note how the frequency remains relatively constant across the mass of the robots, whereas the stride-length increased. The prismatic spine tends to have the lowest stride-frequency with a larger stride-length. These results match observations made in nature [77, 78, 76].

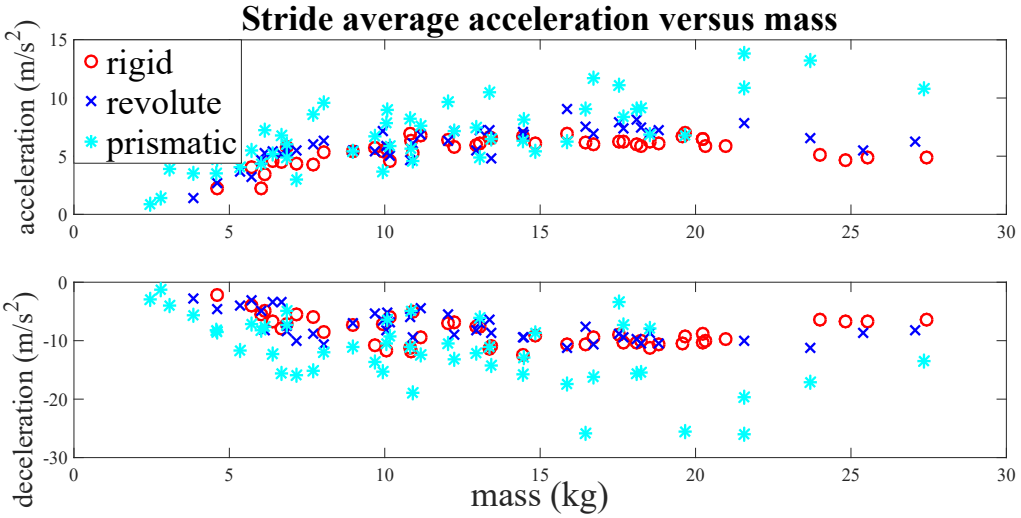

Figure 5.6: A comparison of the stride-average acceleration and deceleration of the three spines of interest. Note how the prismatic spine outperforms the other spine morphologies by achieving the highest stride-average acceleration and deceleration.

rigid robot and 28 % more than the revolute robots. This shows the advantage of the prismatic spine, it is capable of applying work in the forward direction, assisting with rapid acceleration manoeuvres.

$$\chi = \sum_{i=1}^{N} \frac{(\mid \lambda_x(i)\dot{X}(i) \mid)}{(\mid F(i)\dot{l}(i) \mid + \mid \tau(i)\dot{\theta}(i) \mid)} h(i) \tag{5.2}$$

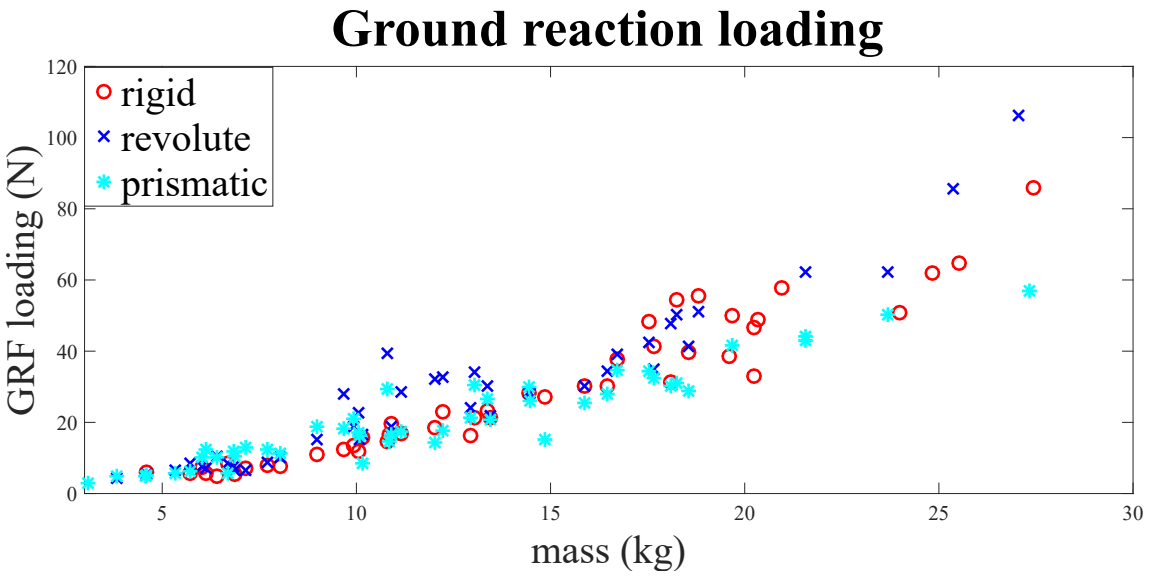

Figure 5.7: The graph shows the average total GRF loading for the optimal trajectories versus mass. It is interesting to note that the prismatic spine successfully lowers its average GRF loading [4].

5.2.5 GRF Observations

Previous nature studies [4] have shown that animals use their spines to help reduce the peak GRF loading. The average GRF loading experienced by the robots was calculated as follows:

$$loading = \frac{\sum_{i=1}^{N}(\lambda_{z1}(i) + \lambda_{z2}(i))h(i)}{2} \tag{5.3}$$

where $\lambda_{z1}(i)$ and $\lambda_{z2}(i)$ were the rear and front vertical ground reaction forces respectively for the i^{th} node and $h(i)$ was the time period for that applied force at node i. The average GRF loading is shown in Figure 5.7 and it is clear that the average loading for the prismatic spine morphology tended to be less over the whole trajectory. This is vital for protecting the skeletal structure and joints of the animal/robot when performing these manoeuvres. The average landing time (contact duration) was observed to be longer for the prismatic robot compared to other spine morphologies, Figure 5.8. This ties in with observations made in nature where the cheetah has a longer ground contact time compared to the slower greyhound [4]. This is partly due to the prismatic spine being able to compress and absorb energy during the impacts, which results in a lower peak GRF and, therefore, a larger contact time was required to restore the energy to the system.

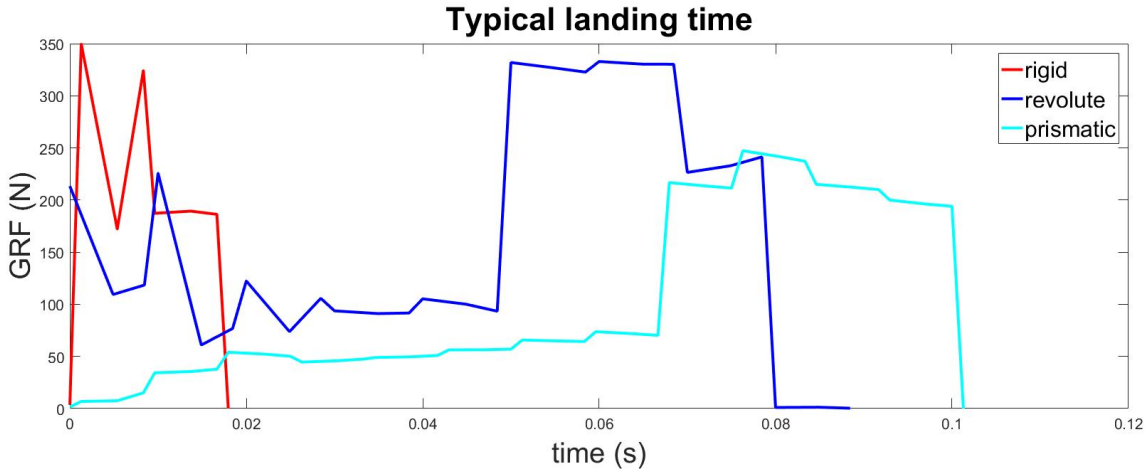

Figure 5.8: An example of the GRF profile, revealing that the prismatic spine robots have a lower peak force and a longer contact time compared to the other spine morphologies.

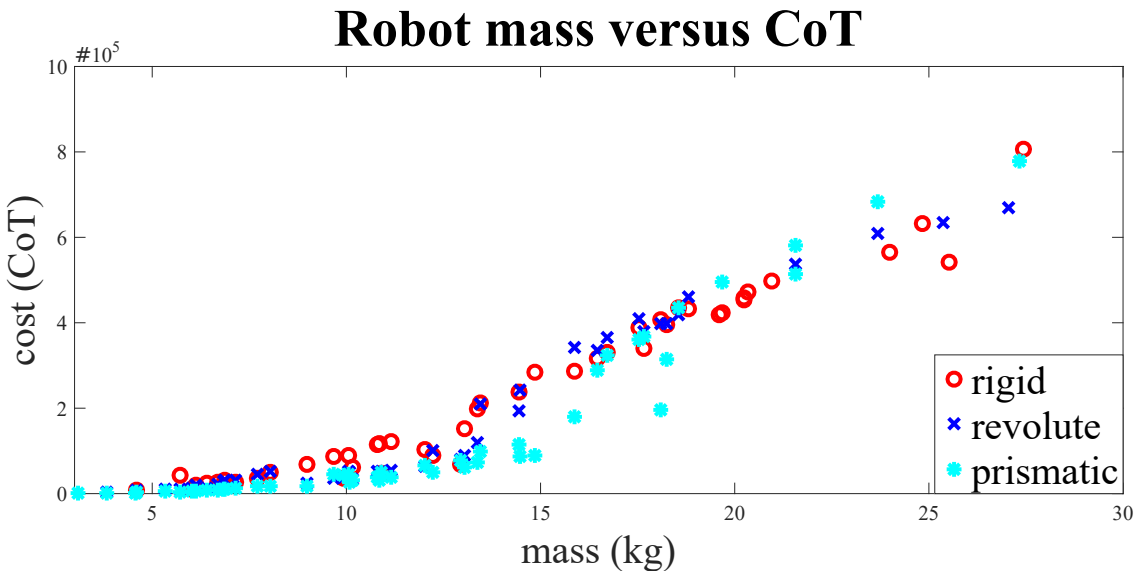

Figure 5.9: Graph showing the average cost of transport for each robot. Note how the prismatic spine performs well for light robots but worse for heavier robots. This is not a huge concern as we are focused on rapid acceleration and deceleration manoeuvre, but is an interesting observation for future robots.

5.2.6 Energy Efficiency Comparison

As energy efficiency is an important measure in robotic platforms, the CoT for each robot was calculated and can be seen in Figure 5.9. A noticeable trend emerges where for light robots, the prismatic spine was more efficient, however, as the robots got larger, this no longer held and the prismatic spine was worse in terms of CoT. One hypothesis for this unique result has to do with the location of the prismatic actuator with respect to the ground. For smaller robots, the actuator is closer to the ground, resulting in less torque experienced on the leg. Larger robots have larger legs, resulting in more torque on the leg (force of prismatic spine multiplied by height above ground), which could result in more slipping experienced by the robot. As this research focuses on rapid transient manoeuvres, energy efficiency was not a concern.

5.3 Summary of Discussion

It is evident that there is a discrepancy between robots and nature when looking at the spine of quadruped robots. Using large-scale Monte Carlo simulations, three spines were compared with the results revealing that the prismatic spine was the optimal spine for these long-time-horizon trajectories. The prismatic spine matched observations made in nature, where the flexible spine on animals was observed to pull in the feet when the front legs were on the ground and vice versa. Additionally the prismatic spine reduced the GRF loading, protecting the limbs.

Chapter 6

Can Optimisation Methods Inspire Controller Design?

For robots to leave the confines of our laboratories, dynamic manoeuvrability will need to be mastered. However, little attention has been given to transient, non-periodic manoeuvres. Currently, to implement these manoeuvres on a legged robotic platform requires accurate models with complex controllers or motion planning algorithms. This results in a computationally expensive problem/system that often suffers from robustness issues, making controller design an intractable task.

The aim of this chapter was to take a trajectory optimisation-inspired approach [136, 137] (akin to "bioinspiration") to controlling a legged robot performing a transient locomotion task. Initially complex mathematical models of a typical two link monopod leg were utilised to generate long-time-horizon trajectories to analyse for trends and heuristics. These experiments included the following:

- Steady-state steps were generated.
- Acceleration trajectories were generated from rest to the apex of the steady-state step.
- Deceleration trajectories were generated from the apex of the steady-state step to rest.
- These results were stitched together and analysed.

Special focus was given to the position of the foot with respect to the neutral point as well as the GRF profile. From these results, a hybrid pneumatic-electric monopod robot was proposed and developed. Additional simulation experiments were performed (same as above, just with a different model) to determine if this pneumatic-electric robot could perform these long-time-horizon trajectories. Further analysis of these trajectories revealed a common controller

emerged, which was tested under disturbances and validated on the physical platform. An overview of the chapter is shown in Figure 6.1.

The methods and results described below have been published in [127].

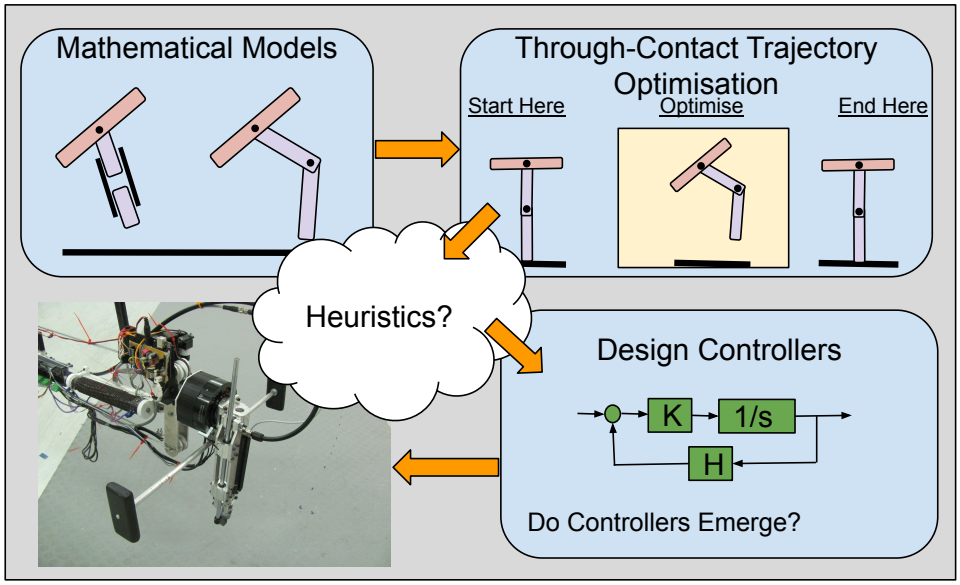

Figure 6.1: A graphical representation of the chapter is presented, where trajectory optimisation methods were used to inspire the design and control of a legged robot. First, using a general leg morphology, trajectories were generated and analysed for trends. This lead to the development of a hybrid pneumatic-electric robot, as a bang-bang like force was observed by the virtual-leg. This lead to the generation of trajectories for the pneumatic-electric robot which were analysed to inspire controllers. These were then tested in simulation and verified on the physical platform.

6.1 Generating Two Link Monopod Trajectories

In order to determine if any trends or heuristics emerge for a legged monopod robot, trajectories were generated for a two link generic monopod robot performing the long-time-horizon task. As the literature has mainly focused on steady-state locomotion, the trajectory was split into three phases (acceleration, steady-state and deceleration), as done in previous chapters. The following modifications for the steady-state step were made for the trajectory optimisation method described in Chapter 3:

1. Number of nodes: $N = 100$
2. Expected total time: $T = 0.5\ s$
3. Cost: J_3
4. Fixed contact order was enforced
5. Average velocity was enforced (5 seeds per velocity)

6. Periodicity was enforced.

A total of four different velocities were generated for the steady-state step ($0.6, 0.8, 1$ and $1.2\ m/s$). Following this, acceleration and deceleration trajectories were generated to the top ($1.2\ m/s$) and bottom ($0.6\ m/s$) velocities. For the acceleration and deceleration trajectories, the following modifications were made to the trajectory optimisation method described in Chapter 3:

1. Number of nodes: $N = 100$
2. Expected total time; $T = 0.5\ s$
3. Cost:J_3
4. 5 seeds per steady-state velocity
5. Accelerate to/decelerate from apex of steady-state velocity.

Importantly, through-contact methods were implemented, allowing the optimiser to determine the optimal number of steps for each transient manoeuvre. The optimal acceleration and deceleration trajectories were stitched to the steady-state step to generate the long-time-horizon task. The individual phases at the different velocities were then analysed for any emerging trends or heuristics.

6.2 Analysis of Two Link Monopod Results

A graph showing the motion of the robot for the fastest ($1.2\ m/s$) and slowest ($0.6\ m/s$) can be seen in Figure 6.2. The results are of the stitched trajectory (the three phases stitched together with three steady-state steps) to form the long-time-horizon task. From analysing the velocity profiles, common trends emerged between the two different velocities. The initial acceleration to the steady-state step was different, as the robot required two steps to reach the $1.2\ m/s$ steady-state step and only one step to reach the $0.6\ m/s$ step. Looking at the vertical velocity, it can be seen that the higher speed steady state-step experienced a higher vertical acceleration (the slope of the vertical velocity was steeper). It is also interesting to note that both trajectories hop at roughly the same constant height, with the higher velocity trajectory bottoming out more. The rapid changes in the horizontal velocity (for example at 0.6 seconds) are due to touch-down contact events that result in the forward velocity rapidly decreasing.

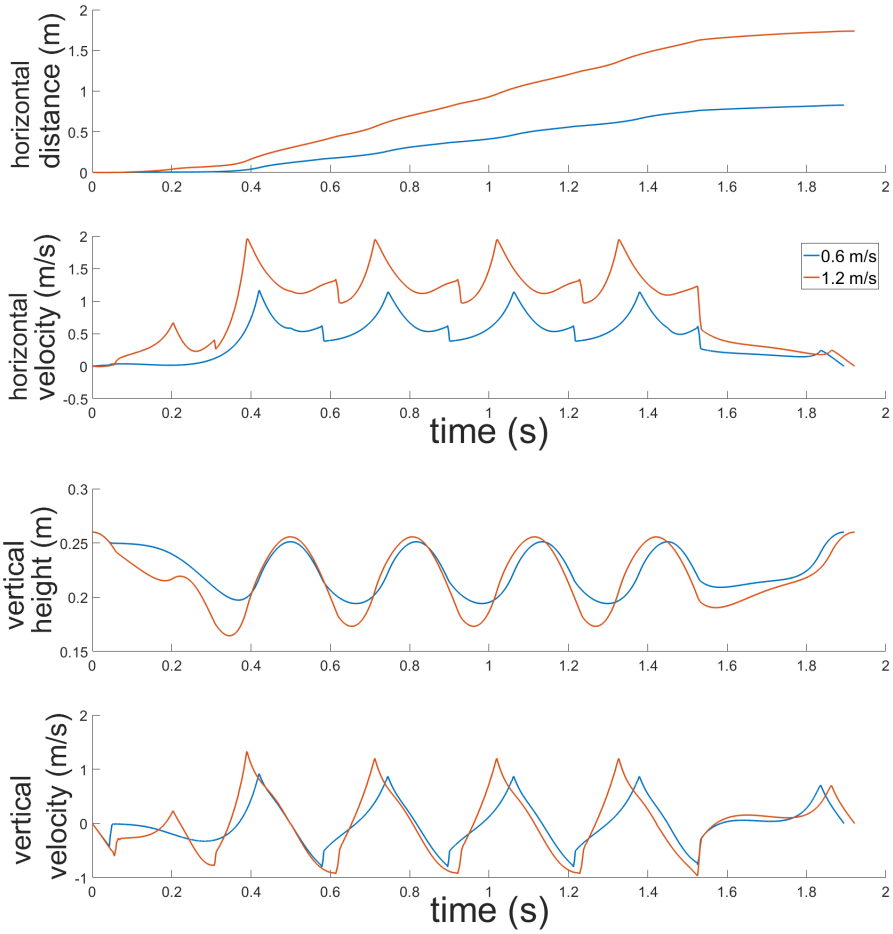

Figure 6.2: Graphs showing the horizontal and vertical position and velocity for the long-time-horizon task (using 3 steady-state steps) for the optimal result for the top speed, 1.2 m/s, and low speed, 0.6 m/s. Note how the robots hop at a constant height for the different velocities and the common pattern emerging in the velocity profiles.

6.2.1 Neutral Point Analysis

Analysing the touch-down position of the foot with respect to the neutral point, Figure 6.3 (acceleration on the left, steady-state in the middle and deceleration on the right), it was found that the Raibert controller naturally emerged. Further details explaining the graphs are presented in the caption of the figure. One observation from the graph was that for the acceleration and deceleration trajectories at the higher velocity, the foot was placed further behind (to achieve acceleration) or in-front (to achieve deceleration) than the lower velocity. The interesting point is that the Raibert controller was not prescribed, yet emerged naturally from the optimisation results, showing that these controllers are optimal for the current robot parameters and optimisation model.

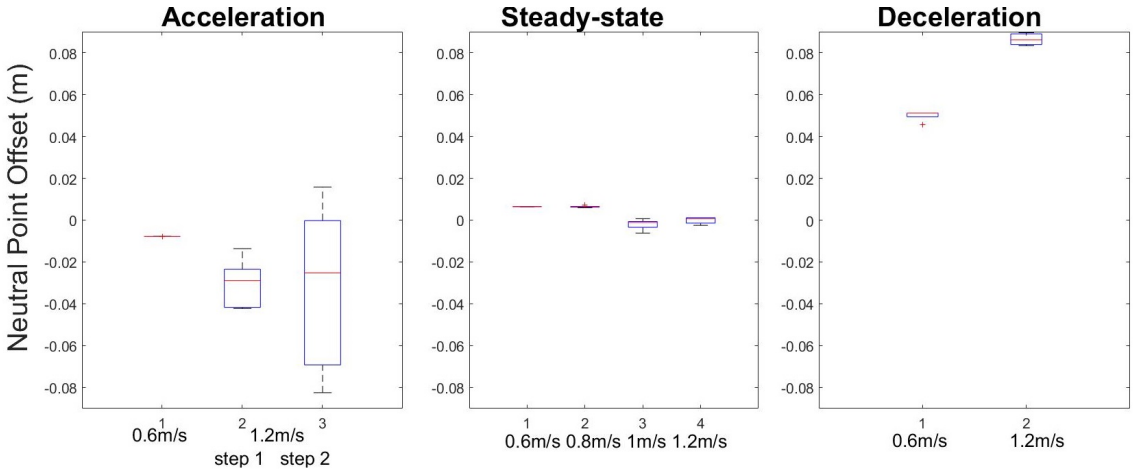

Figure 6.3: The foot touch-down position (in the form of a box and whisker plot) with respect to the neutral point plotted for the three phases of motion. On the left, the acceleration trajectory is shown. The first box and whisker is for acceleration to 0.6 m/s. The second and third box and whisker represent the first and second step taken to accelerate to 1.2 m/s. The foot lands short of the neutral point, resulting in acceleration according to the Raibert controller. The middle figure is for the steady-state steps (first box and whisker is the 0.6 m/s step, the second is for the 0.8 m/s, third for 1 m/s and fourth for the 1.2 m/s step). As can be seen the foot lands close to the neutral point, resulting in no acceleration. The graph on the right is for the deceleration trajectory (the first box and whisker is from the 0.6 m/s step, with the second from 1.2 m/s). As can be seen, the foot is located in front of the neutral point, resulting in deceleration according to the Raibert controller.

6.2.2 GRF Analysis

When rotating the GRF onto the virtual-leg for the steady-state steps, Figure 6.4, a fairly flat profile emerged with an increase in force near the lift-off point. The GRF for the optimal seed for each of the velocities was plotted on the same axis in Figure 6.5. As one can see the profile is very similar with a slight increase in magnitude as the velocity of the steady-state step increases.

For the steady-state step, a common pattern emerged for the GRF profile across the different velocities, however, does this trend emerge for the acceleration and deceleration trajectories? These rotated GRF profiles are shown in Figure 6.6. As can be seen, these trends also hold for the transient trajectories, with a similar magnitude as well. Looking at the acceleration trajectory to the higher velocity steady-state step, the magnitude of the GRF increases as the robots speed increases. Therefore, it is reasonable to assume a common controller can control both the acceleration, steady-state and deceleration phases of motion.

6.2.3 Proposed Robot

From these observations, a hybrid pneumatic-electric monopod robot was developed in the Mechatronics laboratory at the University of Cape Town by Joshua Van Zyl, Figure 6.7. The aim of the remainder of this chapter is to take the above heuristics and insights and apply them to the physical system. First, a pneumatic model of the cylinder was investigated in order to

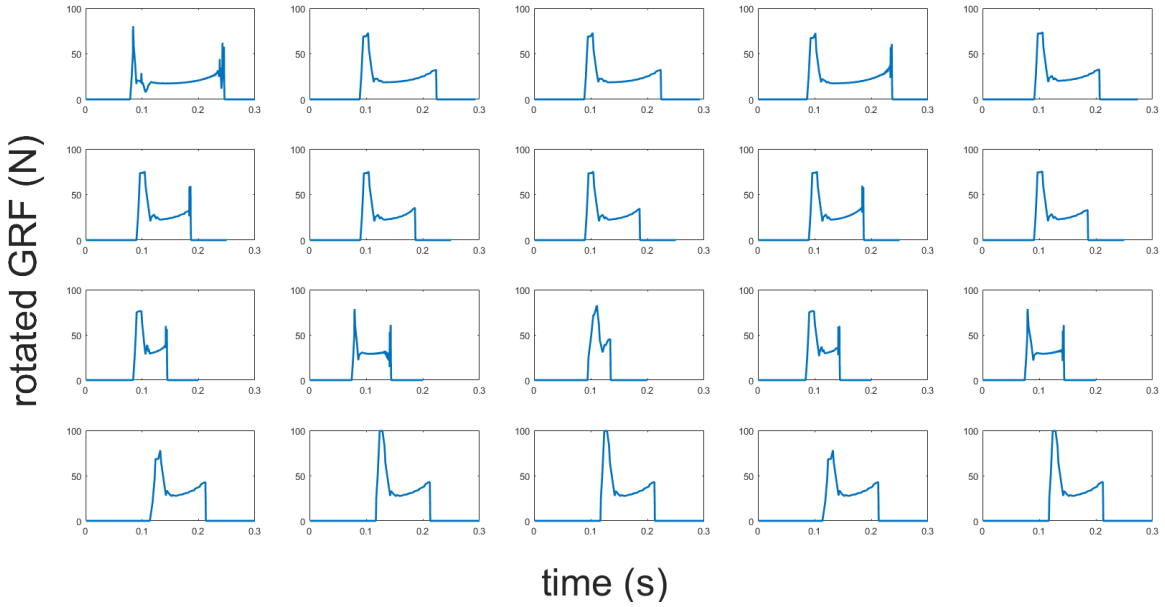

Figure 6.4: The rotated GRF profile for the four different velocities (rows) and five seed points (columns). As can be seen a common shape emerges in the profile and is shared across the different velocities. The GRF profile is relatively flat with spikes at the touch-down and lift-off of the foot, with the magnitude roughly constant across the different velocities.

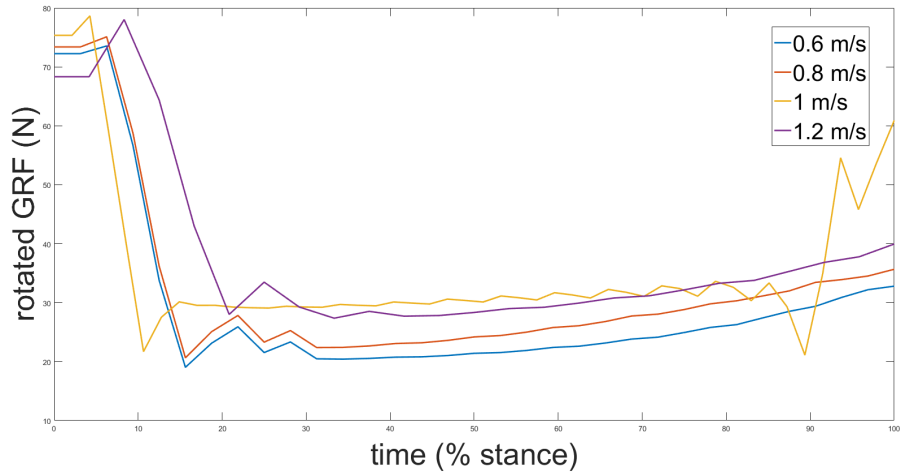

Figure 6.5: The rotated GRF of the optimal seed for each of the four steady-state velocities is shown and overlaid in this figure, with respect to stance duration percentage. As can be seen the profile of the GRF is the same and the magnitude across the different velocities is very similar.

generate trajectories to analyse to aid in the design of the final controllers.

6.3 Development of a Pneumatic Actuator Model

The development of the simplified model of a pneumatic actuator was performed by Joshua Van Zyl, under supervision of A/Prof. Patel and the Author at the University of Cape Town. Initially the cylinder was attached to a mechanical rig that only allowed vertical motion.

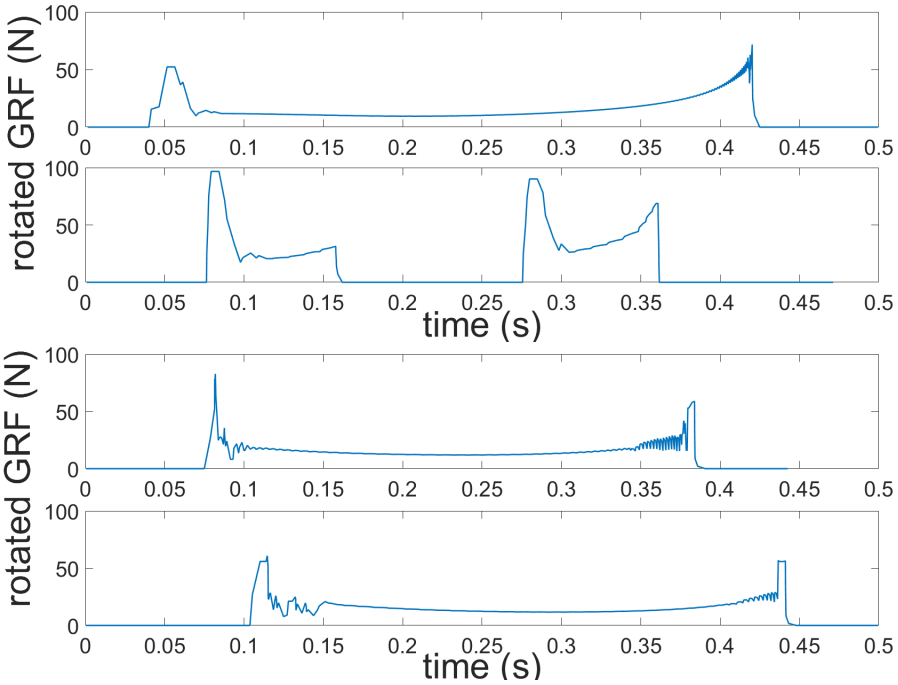

Figure 6.6: The rotated GRF for the acceleration trajectories (top two) being stitched to the 0.6 m/s steady-state step (top) and 1.2 m/s steady-state step (bottom). The bottom two graphs are for the deceleration step. Notice how two steps were required to accelerate to the 1.2 m/s steady-state step and the common trend amongst the GRF profiles and how the magnitude remains relatively constant for the different phases of motion and different target velocities.

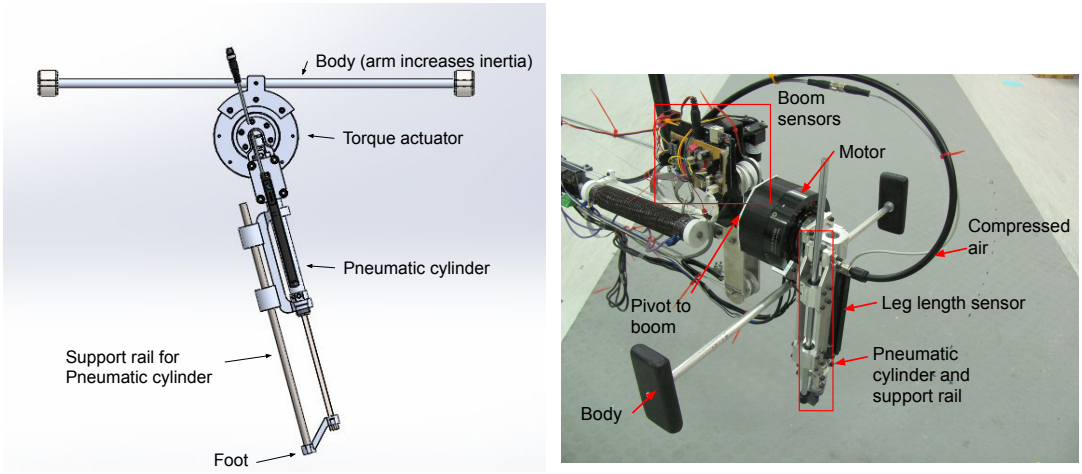

Figure 6.7: The robot being developed at the University of Cape Town by Joshua Van Zyl under supervision of A/Prof. Patel. Accurate models were developed for the robot and controllers were tested on the physical platform.

The cylinder top chamber was connected to a solenoid and compressor charged at 800 kpa . Parameters such as the GRF, solenoid input and cylinder height and length were logged at 1 kHz.

Initial tests involved starting the cylinder on the ground and fully compressed. The solenoid was then triggered, pressurizing the top cylinder, launching the robot into the air. A number of phases were observed during the test:

1. A delay from when the solenoid was triggered to when the cylinder started moving due to the inherent mechanical delay in solenoids.
2. The cylinder piston started accelerating due to the applied pressure in the top cylinder.
3. The cylinder piston reached steady-state velocity.
4. The robot entered a ballistic stage.
5. The piston stopped as it reached the end stops.

During the acceleration phase, as the piston velocity increases, air cannot escape out of the bottom cylinder fast enough. This causes back pressure, ultimately reducing the applied force and resulting in steady-state velocity of the piston.

The main driving force was the differential pressure between the upper and lower chamber. As the back pressure increased, the net force approaches zero as the cylinder reaches steady-state velocity. A curve was fitted to this data, resulting in a reduced order model of the cylinder that comprised of an applied force, F_{max}, and a damping component, C. The force was calculated using the applied pressure and bore diameter of the cylinder, the damping was numerically determined from experimental results. The velocity of the piston was $\dot{l}_{piston}$. The resulting model was:

$$F_{net} = F_{max} - C\dot{l}_{piston} \tag{6.1}$$

F_{max} was calculated as 160 N and C was numerically calculated as 33 Ns/m. When the solenoid was off (no pressure to the top chamber), the damping effect was still active. Experimental data was captured and compared to the reduced order model and can be seen in Figure 6.8. Due to the bang-bang nature of the pneumatic actuator, a state machine was implemented that turned the applied force, F_{max}, on or off. The different states of the state machine were toggled depending on the leg angle and if the foot was on the floor or not.

6.4 Generating Trajectories for the Pneumatic Robot

The above pneumatic model needed to be incorporated into the robot dynamics and trajectory optimisation method in order to generate trajectories for the hybrid pneumatic-electric monopod robot. The generalised coordinates of the robot can be seen in Figure 6.9.

The first setback was that the pneumatic model operated on a state machine. As the robot was a monopod, it was easy to estimate the contact order for these transient trajectories. Using

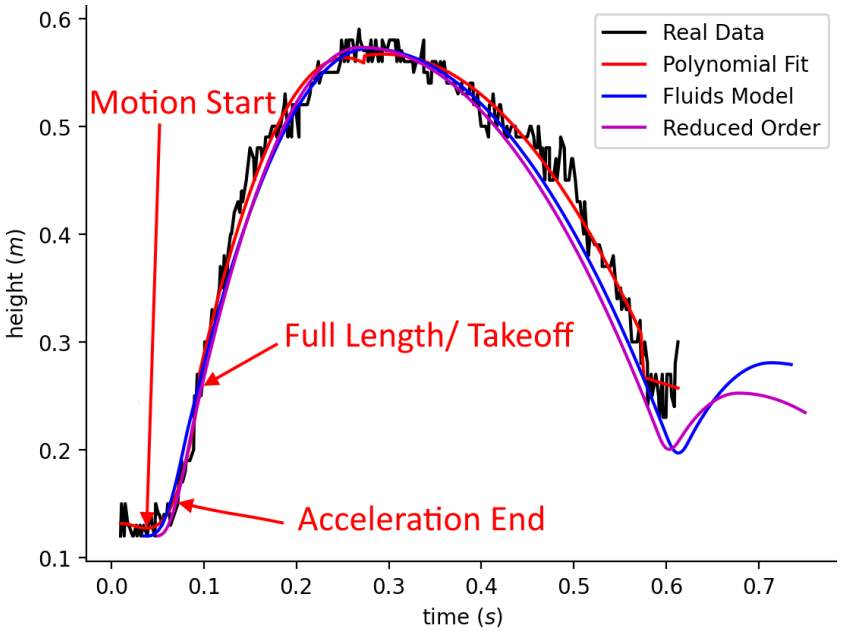

Figure 6.8: Image generated by Joshua Van Zyl, University of Cape Town. Note how the reduced order model tracked the real data. The reduced order model consisted of an applied force and a damping component.

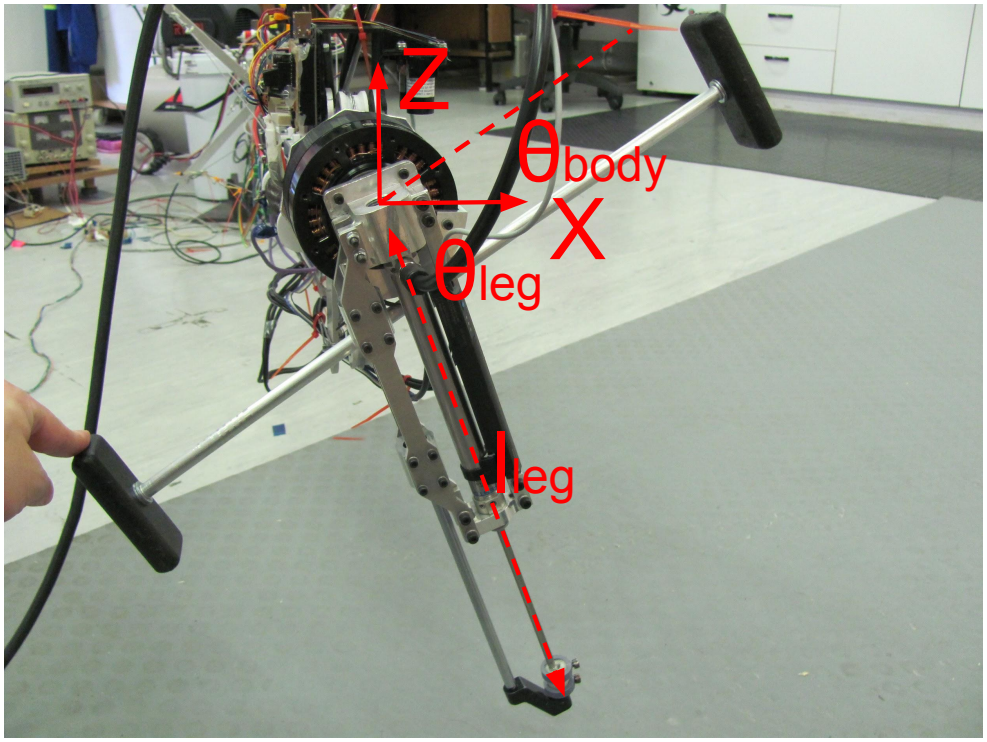

Figure 6.9: The generalised coordinates of the robot are shown.

the above results for the two link monopod, the resulting contact order was assumed for the pneumatic monopod and the desired state machine for the pneumatic cylinder was enforced (the contact order was not strictly enforced, however, a predictable contact order emerges due to the pneumatic state machine). This resulting contact order and state machine can be seen

in Figure 6.10.

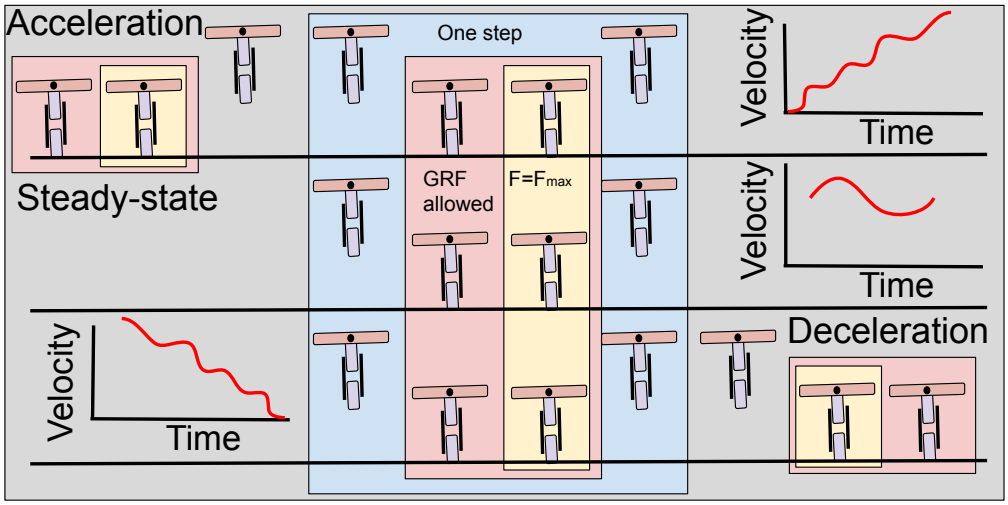

Figure 6.10: The state machine that was enforced for the acceleration (top), steady-state (middle) and deceleration (bottom) trajectories. The state machine determined when the pneumatic cylinder was applying a force (yellow block). A contact order was not strictly enforced, but due to enforcing when the pneumatic cylinder fires, a predictable contact order emerged (GRF was allowed during the red blocks, but not enforced).

To enforce the state machine, a number of additional constraints were added to the trajectory optimisation method. First the total number of nodes for the trajectory was divided between the phases (for example a trajectory that had 7 phases and 70 nodes, results in 10 nodes per phase). Constraints were added for each phase that either set the force of the pneumatic actuator to zero or to the maximum value. These constraints were as follows:

$$\begin{aligned} F_{actuated\ nodes} &= F_{max} \\ F_{unactuated\ nodes} &= 0 \end{aligned} \tag{6.2}$$

As there was no control on the force, this lead to the actuator extending past the maximum leg length (or contracting past the minimum length). Therefore, hard stops were added to stop the actuator violating these limits. These constraints were implemented using complementarity constraints and are shown in Chapter 3.

One of the properties of the pneumatic cylinder (according to the above model) was that damping was always present in the cylinder. Therefore, a damping term was added to the EoM of the robot for the leg length (the coefficient of damping was experimentally determined on the robotic platform). Trajectories were then generated for the hybrid pneumatic-electric monopod robot. Once again the long-time-horizon trajectory was split up into the three phases. Initially steady-state trajectories were generated to match the top and low speed of the two link monopod robot above. The following modifications to the trajectory optimisation method

presented in Chapter 3 are as follows:

1. Number of nodes: $N = 100$
2. Expected total time: $T = 0.4\ s$
3. Cost: J_2
4. State machine to control force was implemented (4 states, 25 nodes each)
5. Damping on cylinder
6. Hard stops on the cylinder
7. Enforced average velocity.

A total of 10 steady-state trajectories were generated for 0.6 m/s and 1.2 m/s (to match the top speed and bottom speed of the two link leg monopod). Of the 10 trajectories for each velocity, the trajectory with the lowest cost was chosen as the optimal trajectory. Acceleration and deceleration trajectories were then generated, with the following modifications to the trajectory optimisation method:

1. Number of nodes: $N = 70$
2. Expected total time: $T = 0.5\ s$
3. Cost: J_2
4. State machine to control force was implemented (7 states, 10 nodes each)
5. Damping on cylinder
6. Hard stops on the cylinder.

The state machine used can be seen in Figure 6.10. These trajectories were then stitched together and analysed for trends to inspire controller design.

6.5 Analysis of Pneumatic-Electric Monopod Results

The analysis of the performance of the pneumatic-electric monopod robot follows the same order as the two link monopod robot shown above. Firstly, the plots of the motion of the robot (of the stitched results) are shown in Figure 6.11. As can be seen, there are trends that emerge between the two different velocities as well as similar trends to the two link monopod shown

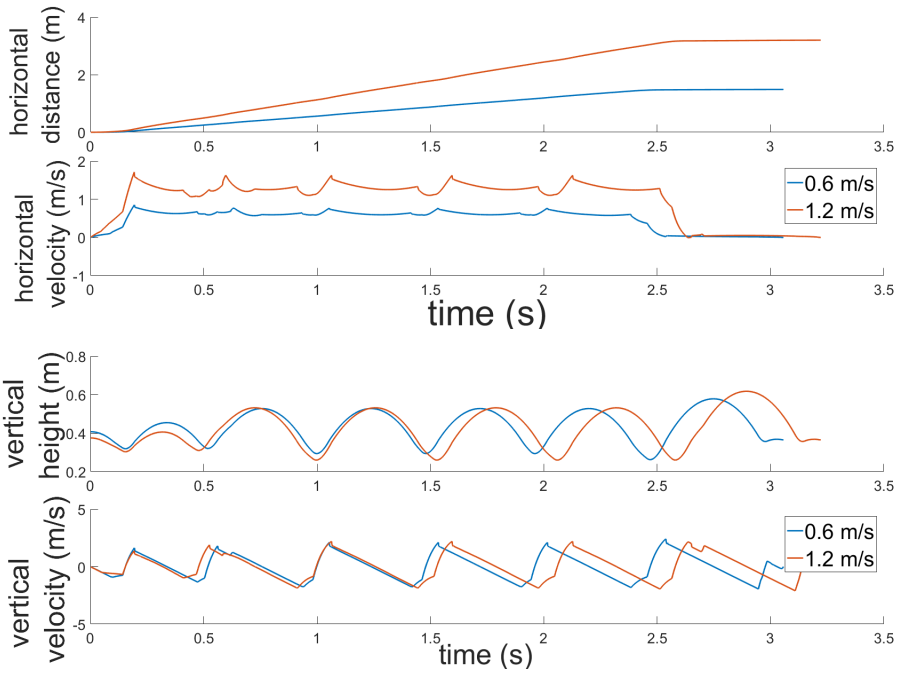

Figure 6.11: The horizontal and vertical position and velocity are shown for the two steady-state velocities (0.6 and 1.2 m/s). Similar trends emerge between the two velocities. The graphs show the stitched results of the acceleration, three steady-state steps and the deceleration trajectories. When comparing to the two link robot, the pneumatic robot jumps, on average, higher as the two link robot can regulate the applied force and contract its leg to get ground clearance, whereas the pneumatic robot could not.

in Figure 6.2. It is interesting to note that the pneumatic-electric robot hops higher (constant hopping height even at different forward velocities) than the two link monopod, as there was no fine control on leg force.

6.5.1 Neutral Point Analysis

For the two link robot, the Raibert controller emerged. To determine if it emerged for the pneumatic robot, the foot position with respect to the neutral point was plotted for the three phases of motion and can be seen in Figure 6.12. From these results it was clear that the Raibert controller in terms of neutral point naturally emerged from these results. As there was no control, only bang-bang, on the pneumatic cylinder, it can be seen that for the acceleration to the slower steady-state step, two steps were required. The first step accelerates the robot forward with a low hopping height, whereas the second step tends to slow it down (or achieve no acceleration for some of the results) and increased the vertical hopping height to link up to the apex of the steady-state step. When looking at the deceleration trajectories, deceleration from a higher speed resulted in the foot being placed further in-front of the neutral point, achieving larger decelerations.

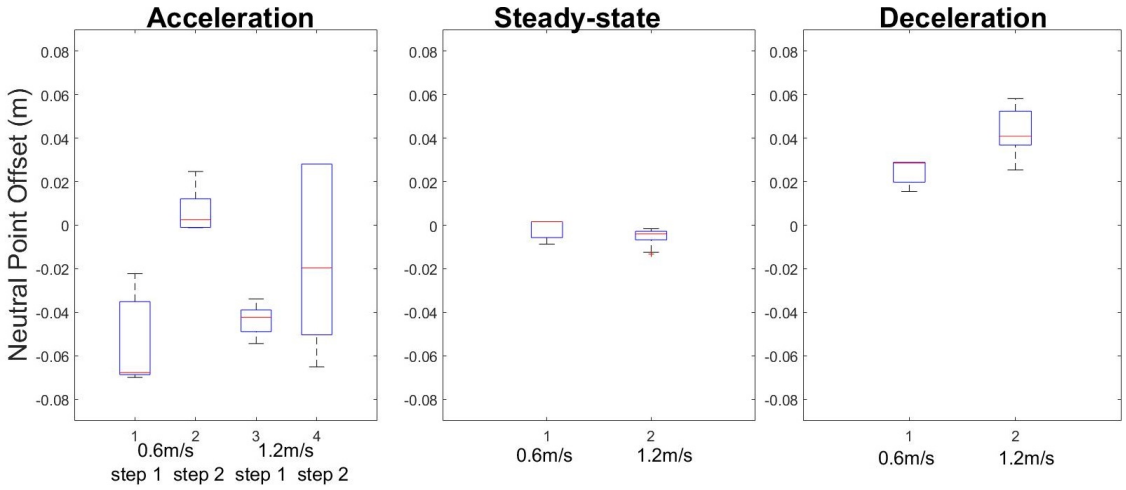

Figure 6.12: The foot touch-down position with respect to the neutral point is shown for the acceleration (left), steady-state (middle) and deceleration (right) trajectories. For the acceleration trajectories, two steps were required to accelerate to 0.6 m/s (first two box and whisker plots) and 1.2 m/s (the remaining box and whisker plots). For the steady-state graph, the first box and whisker plot is for the 0.6 m/s step with the second for 1.2 m/s, the same for the deceleration graph. When the foot lands behind the neutral point, the robot will tend to accelerate (seen on the left) and when it lands in-front of the neutral point (as seen on the right) the robot will tend to decelerate.

6.5.2 Torque Analysis

Analysing the torque profiles using LSQ (least square) methods, evidence of a PD controller emerged (using the Matlab *lsqcurvefit* function). Initially a PFL (partial feedback linearisation) controller was investigated, however, due to the simplicity of the model, it reduced to a PD controller (see Appendix B). A controller was developed for the different phases and was in the form:

$$\tau = Kp(\theta_{body}) + Kd(\dot{\theta}_{body}) + K \tag{6.3}$$

where Kp was the proportional gain, Kd the differential gain and K a constant. The LSQ method was run for each stance phase for each trajectory. A single controller was then chosen for each phase (acceleration, steady-state and deceleration) with different K values for each trajectory. The resulting torques from the PD controller can be seen in Figure 6.13.

6.5.3 GRF Analysis

Finally the rotated GRF can be observed, the three phases of motion can be seen in Figure 6.14. The GRF profile was different compared to what was observed in the two link monopod, however, a common trend emerged between the different velocities for the pneumatic-electric robot. The spike in GRF half way through the stance phase indicates where the pneumatic

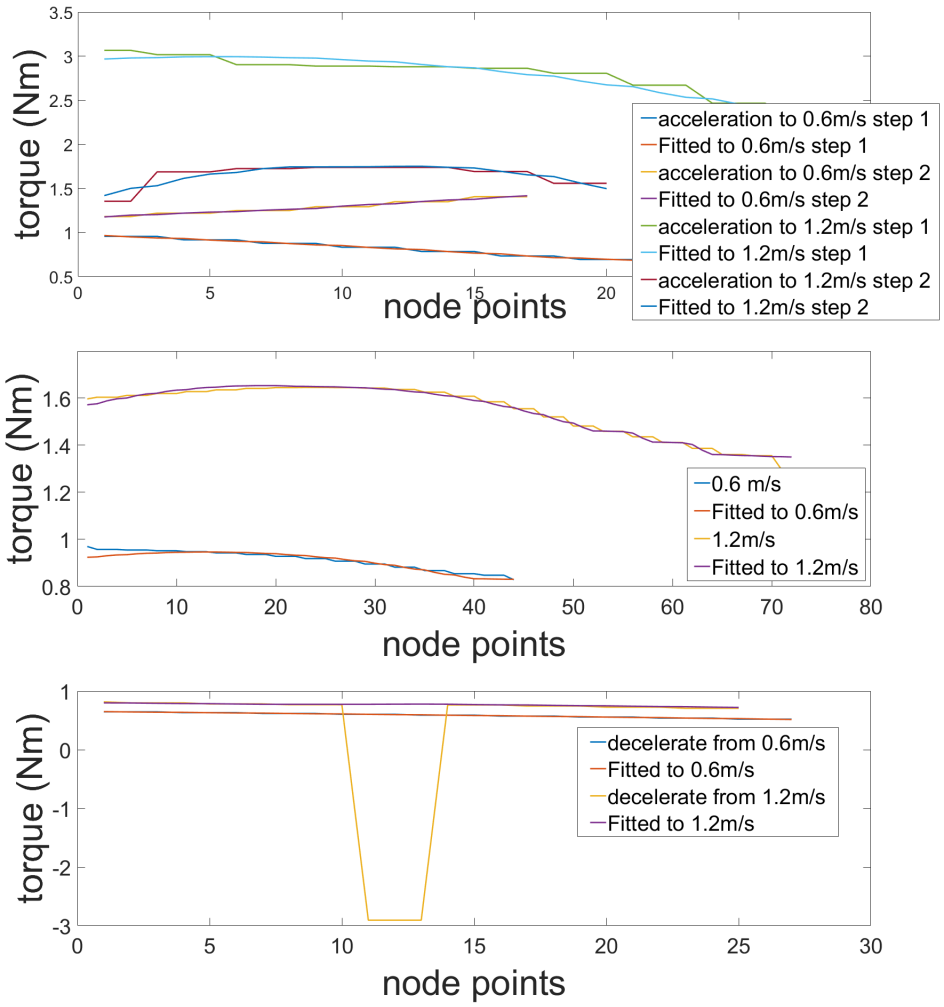

Figure 6.13: Three PD controllers for the three phases of motion (acceleration on the top, steady-state in the middle and deceleration at the bottom) were designed using LSQ methods. As can be seen, these simple controllers accurately track the desired torques. For the deceleration graph from 1.2 m/s, the negative torque was used to slow down the angular rate of the body, however, this behaviour was not captured by the proposed PD controller.

cylinder fires (applies a constant force for the remainder of the phase). Before the cylinder fires, the force was due to the damping acting on the cylinder when it compressed under the weight of the robot.

The above results show that even with the use of a "bang-bang" pneumatic actuator, the robot was still capable of performing these long-time-horizon trajectories with comparable speeds compared to the two link monopod. Now that these controllers have been developed, they were tested for robustness and implemented on the platform.

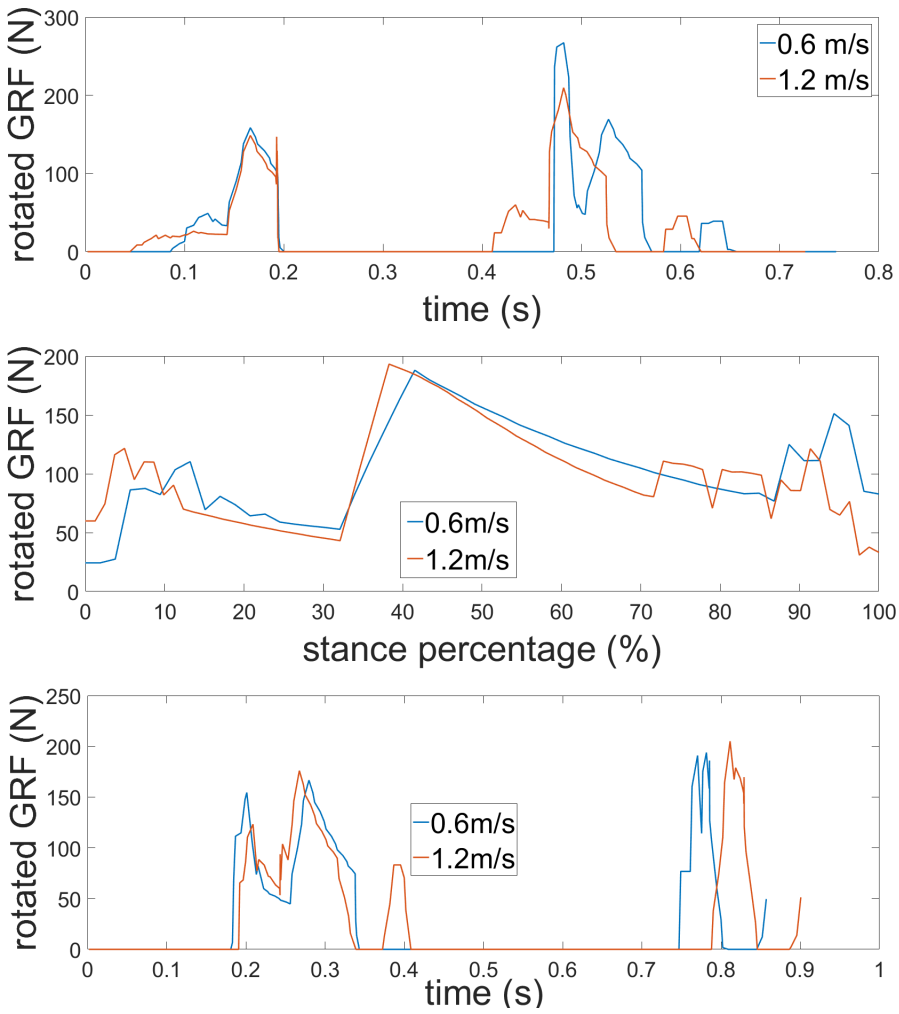

Figure 6.14: Trends emerged throughout for the acceleration (top), steady-state (middle) and deceleration (bottom) trajectories. The spike in the GRF occurring half way through the stance phase indicating where the cylinder fires. The magnitude of the GRF was higher than that of the two link robot and as a result the pneumatic-electric robot jumped far higher during the trajectory compared to the two link robot, as well as the pneumatic-electric robot weighing more due to the added boom parameters.

6.6 Controller Simulation Tests

In order to determine if the above designed controllers would work on the platform, they were first tested in simulation. This was done by modifying the above trajectory optimisation method as follows:

1. Number of nodes: $N = 260$ with an enforced state machine for the pneumatic actuator and stance controllers.
    - 7 acceleration phases (10 nodes each)
    - 12 steady-state phases (10 nodes each, with an enforced average velocity of 1.2 m/s for the entire steady-state phase, periodicity was not enforced).

- 7 deceleration phases (10 nodes each)

2. The above stance torque PD controllers were implemented in the stance phases
3. Task: Long-time-horizon task
4. Expected total time: $T = 1.8\ s$
5. Cost: constant value (purely looking for a feasible trajectory and not optimising anything)
6. Distance constraint ($3.2\ m$) to ensure simulation solution matched the above stitched result.

The same state machine as mentioned above was implemented in this simulation, the only difference being that the whole long-time-horizon task was optimised instead of dividing it into smaller trajectories. As the state machine determines when the foot was on the floor, it also determined when the PD stance controllers were active.

Trajectories were generated and compared to the above stitched results. First the position and velocity for the horizontal and vertical motion were compared and shown in Figure 6.15. The horizontal position and velocity follow the same trend for the simulation trajectories and the stitched results. Both trajectories launched straight into their desired velocity. The stitched results were more periodic during the steady-state section, as periodicity was enforced. There was a difference when the robots decelerate which can be further seen in the vertical height and velocity plots. The stitched robot tended to launch straight into the air (forward velocity of zero) to stop whereas the simulation robots tended to stop more naturally. Looking at the acceleration phase, the simulation results show the robot hopping at a constant height, whereas the stitched results have the robot perform a short hop before entering the steady-state constant hopping height.

6.6.1 GRF Analysis

A comparison of the GRF was also done and shown in Figure 6.16. It is interesting to note that the GRF spikes were higher during the simulation trajectory compared to the stitched results. This may be due to the robot applying a hip torque due to the PD controller as the foot contacts the ground, as well as the fact that the average duration of the contact for the simulation was shorter than that of the stitched results. During the acceleration and deceleration portion of the trajectory, the GRF profiles were fairly consistent between the different trajectories. During the steady-state phase, the profiles were different, in the simulation results, the foot looked to almost lift-off the ground momentarily before the pneumatic actuator fires (this was also observed in the stitched results during the acceleration phase). It must also be noted that the stitched results (multiple optimisation results stitched together) were more accurate as they

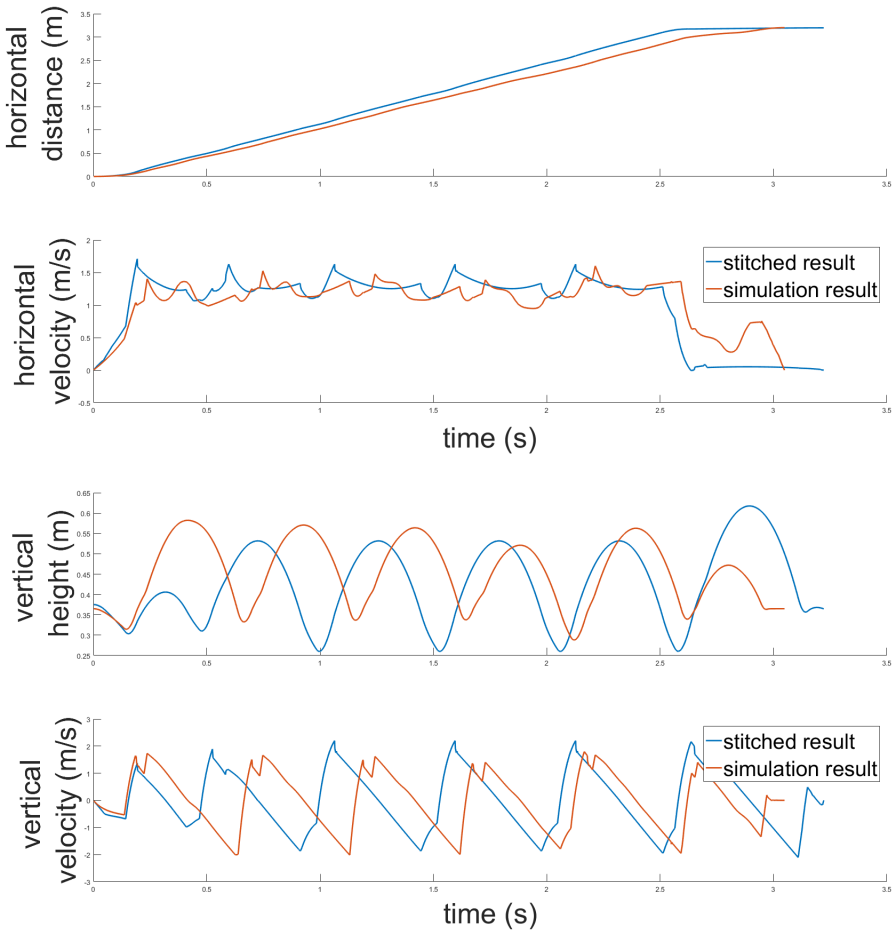

Figure 6.15: The horizontal and vertical position and velocity are shown for the stitched result and simulation result. Trends can be observed between the trajectories. It is interesting to note the difference in the acceleration and deceleration phases. The simulation robot was observed to hop higher than the stitched result during acceleration and to stop differently. The stitched result was observed to stop and launch itself into the air (with zero forward velocity), whereas the simulation results were observed to stop more naturally.

had more nodes compared to the simulation trajectories (which optimised the entire trajectory at once).

6.6.2 Neutral Point Analysis

The foot placement with respect to the neutral point for the simulation was plotted and shown in Figure 6.17. As can be seen the robot started and ended at rest, hence the first and last step lie on the neutral point. The second step was far behind the neutral point causing a large acceleration (in the same range as the stitched results). The second last step was far in-front of the neutral point, causing the robot to rapidly decelerate. During the steady-state step, the foot was slightly behind the neutral point, however, still in the same range as the above stitched result (can be compared to Figure 6.12).

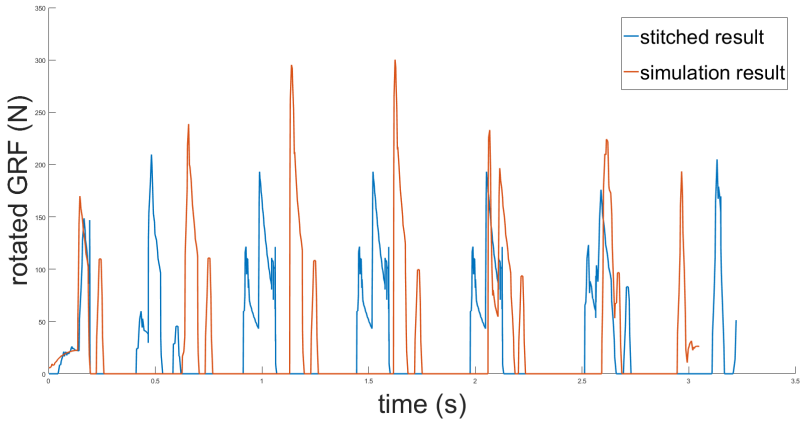

Figure 6.16: A comparison of the GRF from the stitched result and simulation result. Note how the simulation result had a higher spike and shorter average contact duration. The acceleration and deceleration GRF profiles have a similar profile between the trajectories.

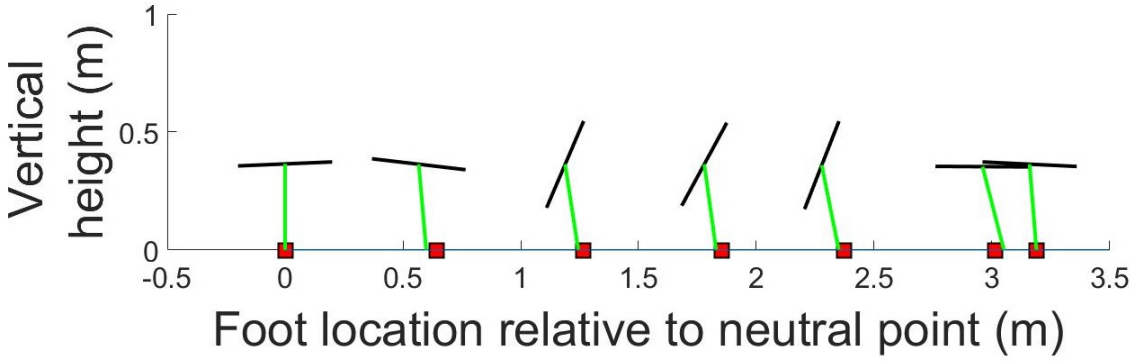

Figure 6.17: The foot touch-down position with respect to the neutral point (red blocks) is shown for the simulation results. Note how the robot starts and ends at rest with the initial step used to launch the robot into the steady-state motion and the final step used to rapidly decelerate it to rest.

6.6.3 Disturbance Analysis

As can be seen the controllers enabled the robot to perform the desired long-time-horizon task in simulation. Before these controllers can be tested on the physical platform, robustness tests were performed in simulation. This was achieved by sampling the simulation result (once during the acceleration phase, once during the deceleration phase). At the sample points, a disturbance force was added to the CoM of the body (similar disturbances were applied in [138]). First a positive force was applied and then a negative force was applied. The disturbance force was simulated by multiplying the horizontal velocity at the sample point by 1.2 or 0.8, resulting in an instantaneous 20 % change in forward velocity. The simulation was then started from the sample point and simulated to the end of the long-time-horizon trajectory.

In total, four disturbance forces were applied to the robot. Two positive forces (one during the acceleration phase, one during the deceleration phase) and two negative forces (one during the acceleration phase, one during the deceleration phase). A comparison of the horizontal velocity

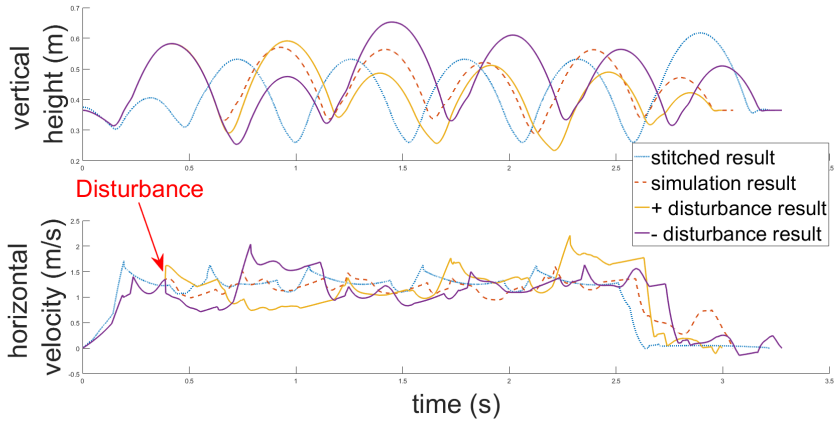

Figure 6.18: A comparison of the stitched result, simulated result and disturbances (both positive and negative) applied to the simulation result during the acceleration phase. The robot took about a step to correct itself and regain its steady-state motion. Even with the disturbances, the robot was able to complete the long-time-horizon trajectory.

and vertical height for the disturbances during the acceleration phase can be seen in Figure 6.18. As can be seen the robot still completed the long-time-horizon trajectory even with a ± 20 % disturbance during the acceleration phase. This does result in the robot taking a step to regain the steady-state velocity, however, it remained stable with no alteration to the stance controllers.

Due to these disturbances, the leg was seen to compress more in the next contact phase, with the leg compressing the most for the negative disturbance. For the positive disturbance, the robot was seen to hop roughly the same height as the simulation results, however, the forward velocity was decreased for the step after the disturbance and then increased afterwards (to get the average velocity right). For the negative disturbance, the robot was seen to hop lower in the next step and then hops at a normal height. The next step also had a big increase in velocity which goes back to normal in the subsequent steps. Therefore, the robot takes approximately two steps to recover from the disturbance. The limit cycle for the disturbance trajectory can be seen in Figure 6.19, revealing that after the disturbance, the trajectory managed to converge to the simulated results.

A similar comparison was performed for the disturbances that were applied during the deceleration phase, and can be seen in Figure 6.20. The disturbance on the deceleration does not seem to make much of a difference, with the robots still managing to complete the long-time-horizon trajectory. The negative disturbance does result in the robot jumping higher during the deceleration phase, however, not as high as the stitched result.

The positive disturbance results in the robot stopping quicker (more aggressive deceleration) and the robot hops lower than the simulation result. For the negative disturbance, the robot hops higher and has a higher velocity till the end (to make up the remaining distance to satisfy the task). A comparison of the limit cycle for the disturbance trajectory and simulation results

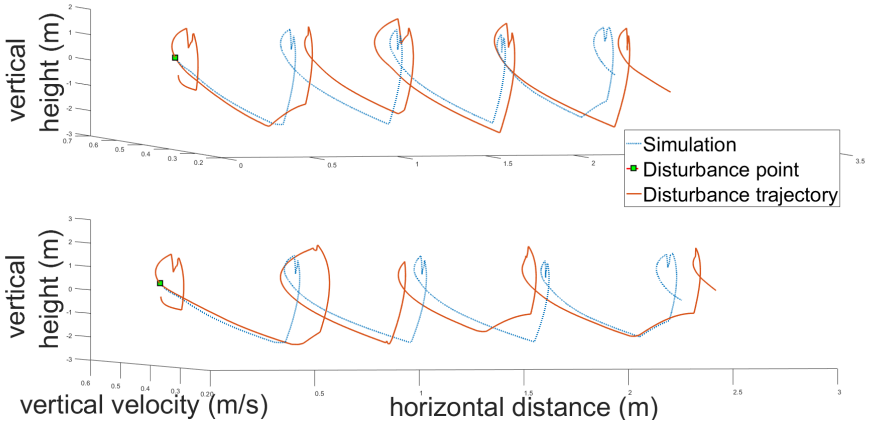

Figure 6.19: A comparison of the limit cycles for the simulated trajectory and the disturbance trajectory. The top image is of the positive disturbance, the bottom image is the negative disturbance. Note how the disturbance trajectory deviates from the simulated results, however, still manages to converge to the limit cycle after two steps.

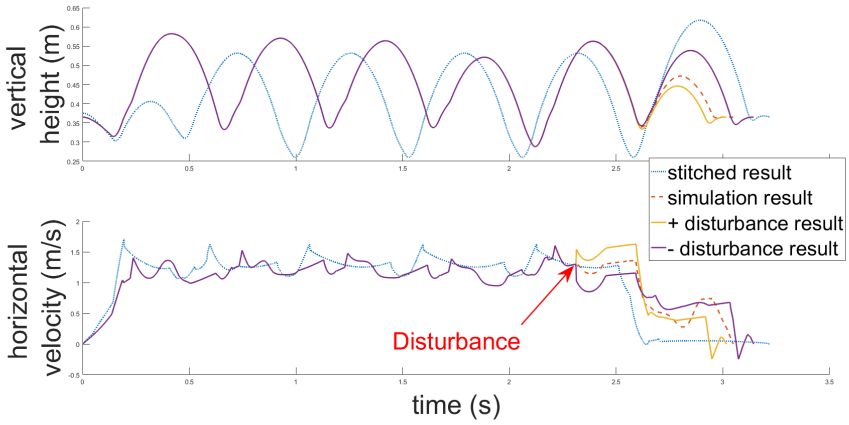

Figure 6.20: A comparison of the stitched result, simulated result and disturbances (both positive and negative) applied to the simulation result during the deceleration phase. Note how even with the disturbances, the robot was able to complete the long-time-horizon trajectory.

are shown in Figure 6.21. Note how the disturbance trajectory deviates from the simulation result, however, the robot still manages to stabilise and stop in time.

6.7 Physical Experiments

In order to conduct the experiments, the robot was attached to a mechanical boom, supporting the robot in the sagittal plane, as seen in Figure 6.22. The robot contained a number of sensors along with a KF (Kalman Filter) to estimate the position and velocity with encoder feedback providing body angle and leg angle values. Initial vertical and "fixed body" experiments were performed and are shown in Appendix C, along with the relevant robot and boom parameters in Chapter 3.2. This section details the final free body experiments. The above state machine and controllers were coded, however, due to space constraints of the testing area, only the

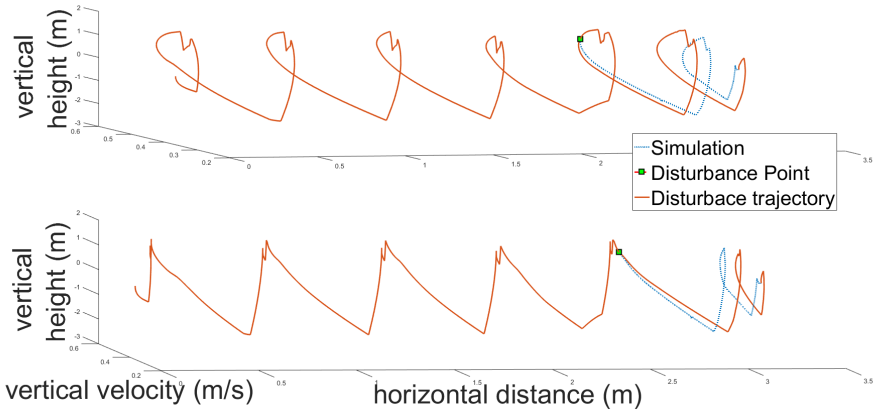

Figure 6.21: A comparison of the disturbance trajectory to the simulated results. The top image is of the positive disturbance, the bottom image is of the negative disturbance. Note how the disturbance limit cycle remains stable for the remainder of the trajectory.

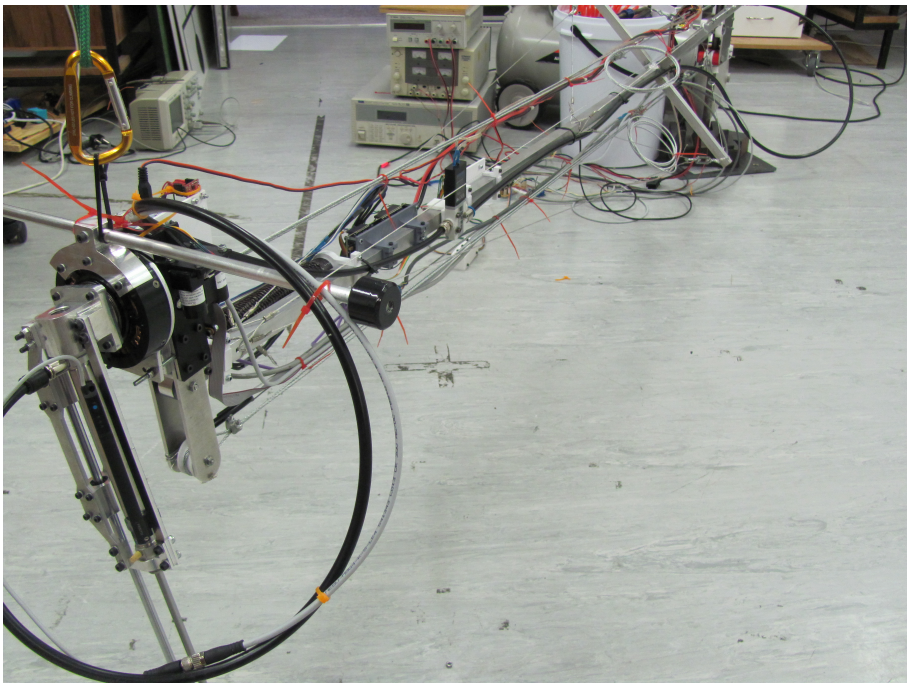

Figure 6.22: An image taken of the monopod attached to the mechanical boom providing support in the sagittal plane. The compressor as well as the required power supplies can be seen in the background. The robot was developed by Joshua Van Zyl.

0.6 m/s long-time-horizon experiment was conducted. A video of the results can be found here: `https://youtu.be/4ryzEzSQftg`.

The results of the experiment are shown in Figure 6.23. As can be seen the simulation results align with the horizontal position and velocity data. A similar hopping height was achieved to the simulation results (approximately 0.4 m heigh). Static images of the robot performing the long-time-horizon task can be seen in Figure 6.24.

There were a number of differences between the simplified robot model and the physical platform,

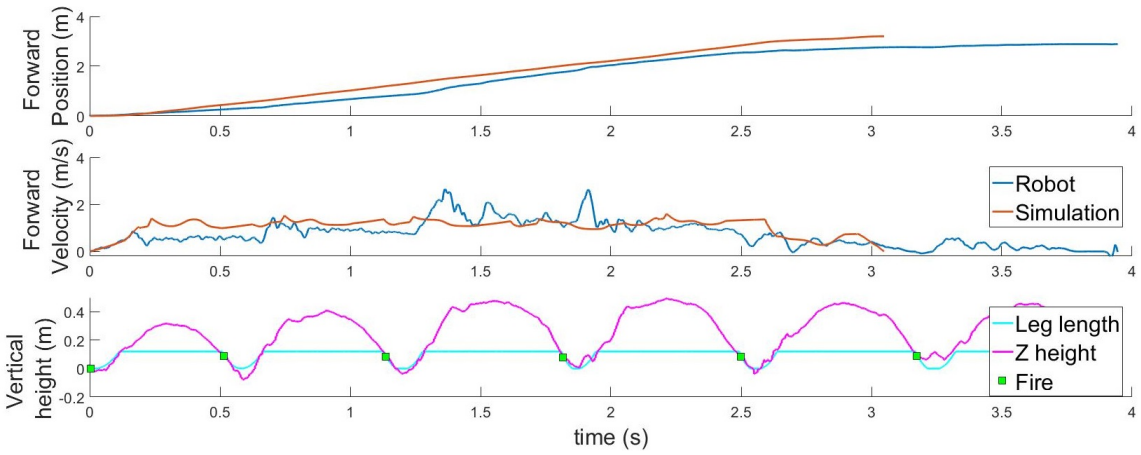

Figure 6.23: A comparison between the horizontal position (top) and velocity (middle) of the simulation results versus the physical experiments are presented. The vertical hopping height and leg length during the experiment are shown (bottom). The green square represents when the pneumatic cylinder was actuated.

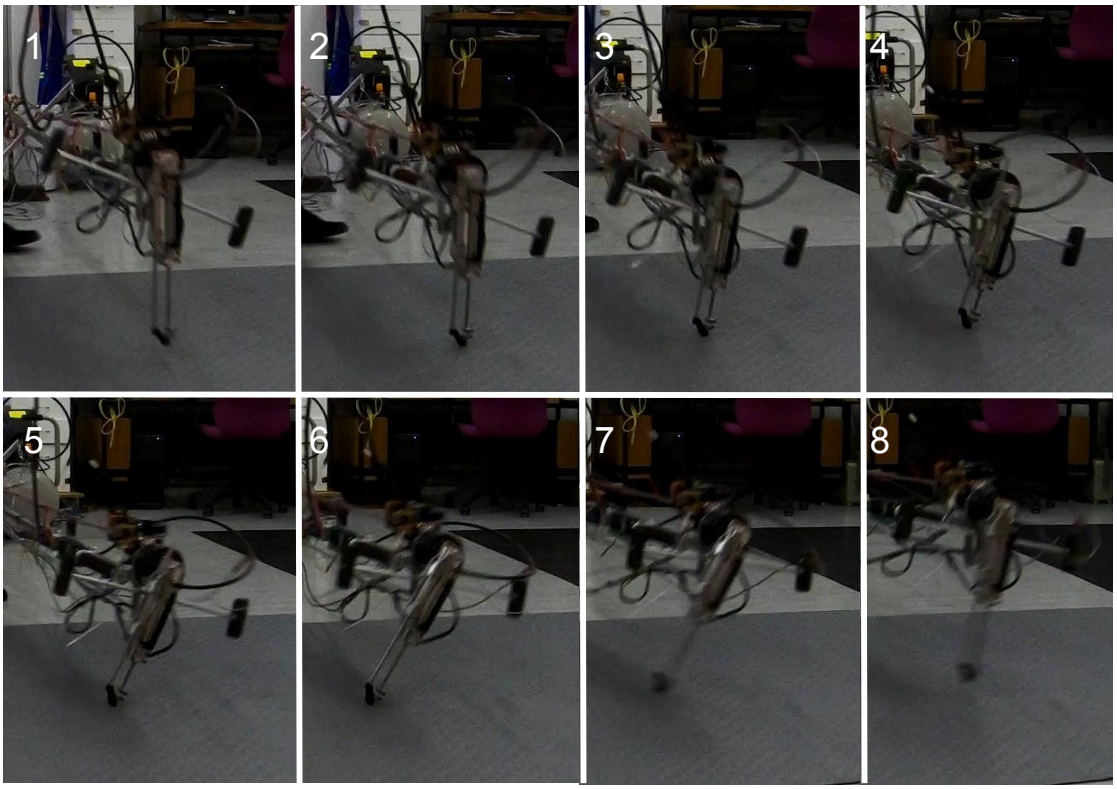

Figure 6.24: Static images of a contact event taken while the robot was performing the long-time-horizon task. Notice how the body was stabilised during the stance phase. Frames are 30 Hz apart.

however, the designed controllers were shown to work. Some of the differences include:

1. Friction coefficient was assumed to be 1 during simulation, however, it depends on how the foot lands and if it digs into the rubber mat or not.

2. The mechanical boom was modelled as a point mass, however, it has an inertia and other

dynamics due to flexing during the experiment.

3. The CoM of the body was offset and does not align with the pivot point of the leg and boom mounting, as assumed in the model.

6.8 Summary of Discussion

Initial trajectories revealed a number of trends and heuristics, which resulted in the development of a hybrid pneumatic-electric monopod robot. First a simplified model of the pneumatic actuator was developed, followed by generating long-time-horizon trajectories. These were analysed and it was found that a PD controller morphology emerged for each phase of motion (during stance), while when the robot was aerial, the Raibert controller emerged. These controllers were then simulated, with disturbances added and it was shown to stabilise and still perform the desired motion. These controllers were then experimentally tested and validated on the physical platform, which successfully performed the long-time-horizon task.

Chapter 7

Conclusions and Future Work

In recent years, legged robotics has become a focal point in the literature. Some platforms, such as ATRIAS [30, 61] and Atlas [2], have advanced far ahead of their field and can demonstrate complex manoeuvres in real time. However, the large majority of research groups still consider tasks such as walking and running unsolved and complex to implement, especially when taking a low-cost approach. Very few research groups have focused on transient behaviours, which further complicates things, and have limited their research to more tractable motions such as periodic locomotion.

This research has shown that using trajectory optimisation methods, the design and control of these legged platforms can be more tractable as opposed to hand tuning controller gains. Classical control methods are less effective for these manoeuvres as the desired motion is poorly understood. This method does, however, require fairly advanced and complex optimisation methods. In this research, a strong emphasis was placed on using accurate integrators (3-point collocation methods were implemented [112]) along with through-contact methods. It was stressed that when analysing transient motions, one cannot enforce a contact order, as it is unknown. Specifying the contact order will drastically narrow the solution space and therefore, the relevance of the results will be questionable. By implementing through-contact methods [111], the optimiser determined the optimal foot contact order and allowed the foot to slip if the friction cone was violated.

In this study, a thorough simulation investigation was conducted into how these legged robots should accelerate (positive and negative), what spine morphology was best for these transient motions, as well as how to control them, with experimental results validating the methods employed. This research was aimed at answering the three main research questions posed in Chapter 1.2 which were:

1. How should legged robots accelerate and decelerate?

2. What is the optimal spine morphology for quadruped robots performing rapid transient manoeuvres?

3. Can trajectory optimisation be used to inspire mechanical and controller design?

The literature review showed a gap in the available research, where it was unknown how robotic platforms should rapidly accelerate and decelerate under time constraints. In the literature, robots are often shown to slowly increase their speeds, or to perform discrete intermediate gait changes [9, 10, 11], whereas in nature, animals leap straight into their desired gait [40, 44, 128]. In Chapter 4, it was shown through the use of trajectory optimisation techniques, that the optimal method to rapidly accelerate and decelerate is to leap to the desired gait/speed without performing intermediate discrete gait transitions. Two different robot models were used, namely the planar biped and quadruped robot. Using two different cost functions (heat based and CoT), some interesting trends emerged from the results. When time was a limiting factor, both robots launched straight into the desired gait (for both cost functions) and when decelerating, they skidded to a halt, without utilising intermediate discrete gaits.

When time was not a limiting factor, for the heat based cost function, the robots remained stationary until the last moment and then launched straight into the desired gait/speed. The same was observed for deceleration, the robots skidded to a halt and remained standing for the remainder of the time. For the CoT cost function, the robots performed a slow walk, until the final moment and then launched into the desired gait. The same was observed for the deceleration phase. Long-time-horizon trajectories were also generated and compared. The robots launched straight into the desired gait, without utilising intermediate gaits.

Hubicki et al. [43] noted in their research on monopod robots, that the sliding mass template accurately tracked the forward velocity during acceleration and deceleration manoeuvres. They hypothesized that the sliding mass template would work for more complex models (such as the biped and quadruped). Therefore, the results presented in Chapter 4 were compared to the sliding mass template and it was shown that the simple one DoF model accurately predicted the velocity profile of the acceleration and deceleration trajectories, proving their hypothesis. It has been proposed that this template can be used to generate a set-point in real time for acceleration and deceleration controllers, along with online trajectory planning, forming part of the future work.

It was also noted that there is a discrepancy between animal spines in nature, and the spines commonly employed on robotic platforms. In Chapter 5, through the use of large-scale Monte Carlo-based trajectory optimisations, the optimal spine morphology was determined. In total, 100 robot parameters were randomly generated, each containing a rigid, revolute and prismatic spine. For each set of parameters and each spine, 20 seed points were optimised in order to thoroughly explore the solution space. The performance of the three spines to complete the long-time-horizon task were compared, with the optimal morphology being the prismatic spine

(79 % of the time).

The prismatic spine also helped reduce the GRF loading on the limbs by decreasing the magnitude of the GRF spikes and lengthening the resulting contact duration. These robots also experienced the highest stride-average acceleration and deceleration. This matches observations in nature, where it was identified that the flexible spine is crucial for agile transient motions. The spine of the cheetah has been observed to increase the stride-length by 5 %, with a resulting increase in top speed of 10 % [77, 78].

The final problem that was investigated was to use trajectory optimisation to identify trends and heuristics to inspire the controller design of a simple monopod robot (Chapter 6). First trajectories were generated for a general two link monopod robot. These were analysed and it was found that a Raibert controller emerged as well as the applied virtual force on the leg seemed to not require fine control and exhibited a bang-bang like nature. These heuristics were then applied to the hybrid pneumatic-electric robot that was developed by Joshua Van Zyl. Trajectories were generated for the pneumatic-electric monopod robot and analysed. It was found that with the use of a bang-bang actuator, the robot could still achieve the desired long-time-horizon task.

Analysing these results, it was found that PD controllers emerged for the applied torque during the stance phase (using LSQ methods). These controllers were then tested in simulation and the robot was still able to complete the task. To test the robustness of these emerging controllers, disturbances were applied during the acceleration and deceleration phases of the trajectory. The controllers successfully stabilised the trajectories and completed the desired task. These controllers were then tested on the physical platform, which successfully performed the long-time-horizon task. Due to space limitations, only the 0.6 m/s trajectory was experimentally tested.

From these results it can be concluded that trajectory optimisation is a vital tool for robotic design and control, even with the use of simplified models. Future work will involve extending the pneumatic model to not rely on a state machine, as well as extending the controller design to that of a planar quadruped with a prismatic spine. Online planning for these long-time-horizon trajectories using the sliding mass template will also be investigated, with an extension into 3D locomotion. Finally, an investigation into the effects of friction will also be considered.

Bibliography

[1] L. Raw, C. Fisher, and A. Patel, "Effects of limb morphology on transient locomotion in quadruped robots," *International Conference on Intelligent Robots and Systems*, 2019.

[2] N. Banerjee, X. Long, R. Du, F. Polido, S. Feng, C. Atkeson, M. Gennert, and T. Padir, "Human-supervised control of the atlas humanoid robot for traversing doors," *15th International Conference on Humanoid Robots (Humanoids)*, pp. 722–729, 2015.

[3] M. Senn, *More Parkour Atlas*, 2020 (accessed June 1, 2020), https://www.youtube.com/watch?v=_sBBaNYex3E.

[4] K. T. T. Kamimura, S. Aoi and F. Matsuno, "Body flexibility effects on foot loading in quadruped bounding based on a simple analytical model," *IEEE Robotics and Automation Letters*, 2018.

[5] Z. J. Z. Gan and C. Remy, "On the dynamic similarity between bipeds and quadrupeds: A case study on bounding," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, 2018.

[6] W. Xi, Y. Yesilevskiy, and C. Remy, "Selecting gaits for economical locomotion of legged robots," *The International Journal of Robotics Research*, vol. 35, no. 9, 2016.

[7] J. Reher, W. Ma, and A. D. Ames, "Dynamic walking with compliance on a cassie bipedal robot.," *In 2019 18th European Control Conference (ECC)*, pp. 2589–2595, 2019.

[8] E. Unplugged, *Cheetah vs Greyhound, World's Fastest Dog In Super Slow Motion*, 2020 (accessed July 1, 2020), https://www.youtube.com/watch?v=jc8Hno4M0Qs.

[9] D. Owaki and A. Ishiguro, "A quadruped robot exhibiting spontaneous gait transitions from walking to trotting to galloping," *Scientific reports*, vol. 7, no. 1, 2017.

[10] O. Kwon and J. Park, "Gait transitions for walking and running of biped robots," *IEEE International Conference on Robotics and Automation (ICRA)*, 2003.

[11] D. Hyun, J. Lee, S. Park, and S. Kim, "Implementation of trot-to-gallop transition and subsequent gallop on the mit cheetah 1," *The International Journal of Robotics Research*, vol. 35, no. 13, 2016.

[12] Q. Liu, Y. Bao, W. Yu, J. Zhang, C. Li, and X. Xie, "Mechanism of spine motion about contact time in quadruped running," *Chinese Journal of Mechanical Engineering*, vol. 32, 2019.

[13] K. Koutsoukis and E. Papadopoulos, "On passive quadrupedal bounding with translational spinal joint," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[14] R. K. G. Haynes, J. Pusey and D. Koditschek, "Dynamic bounding with a passive compliant spine," *In Proceeding Dynamic Walking Conference*, 2012.

[15] Q. Cao and I. Poulakakis, "Passive stability and control of quadrupedal bounding with a flexible torso," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[16] M. Kani, M. Derafshian, H. Bidgoly, and M. Ahmadabadi, "Effect of flexible spine on stability of a passive quadruped robot: Experimental results," *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2793–2798, 2011.

[17] S. Bhattacharya, A. Singla, D. Dholakiya, S. Bhatnagar, B. Amrutur, A. Ghosal, and S. Kolathaya, "Learning active spine behaviors for dynamic and efficient locomotion in quadruped robots," *28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1–6, 2019.

[18] D. Kuehn, A. Dettmann, and F. Kirchner, "Analysis of using an active artificial spine in a quadruped robot," *International Conference on Control, Automation and Robotics (ICCAR)*, vol. 4, pp. 37–42, 2018.

[19] Y. Kajiwara, S. Ikemoto, and K. Hosoda, "Pneumatically-driven quadruped robot with biomimetic legs and flexible spine," *Procedure of International Symposium on Adaptive Motion in Animals and Machines*, vol. 23, 2017.

[20] M. Khoramshahi, A. Sprowitz, A. Tuleu, M. Ahmadabadi, and A. Ijspeert, "Benefits of an active spine supported bounding locomotion with a small compliant quadruped robot," *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[21] J. Pusey, J. Duperret, G. Haynes, R. Knopf, and D. Koditschek, "Free-standing leaping experiments with a power-autonomous elastic-spined quadruped," *International Society for Optics and Photonics in Unmanned Systems Technology XV*, vol. 8741, 2013.

[22] S. Seok, A. Wang, M. Chuah, D. Hyun, J. Lee, D. Otten, J. Lang, and S. Kim, "Design principles for energy efficient legged locomotion and implementation on the mit cheetah robot," *IEEE/ASM Transactions on Mechatronics*, vol. 20, no. 3, pp. 1117–1129, 2015.

[23] J. Duperret and D. Koditschek, "Empirical validation of a spined sagittal-plane quadrupedal model," *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[24] N. F. Steenkamp and A. Patel, "Minimum time sprinting from rest in a planar quadruped," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3866–3871, 2016.

[25] J. Luo, Y. Su, L. Ruan, Y. Zhao, D. Kim, L. Sentis, and C. Fu, "Robust bipedal locomotion based on a hierarchical control structure," *Robotica*, vol. 37, no. 10, 2019.

[26] C. Boussema, M. Powell, G. Bledt, A. Ijspeert, P. Wensing, and S. Kim, "Online gait transitions and disturbance recovery for legged robots via the feasible impulse set," *Robotics and Automation Letters*, vol. 4, no. 2, pp. 1611–1618, 2019.

[27] I. Spectrum, *Boston Dynamics' Atlas Robot Shows Off Parkour Skills*, 2020 (accessed July 1, 2020), https://spectrum.ieee.org/automaton/robotics/humanoids/boston-dynamics-atlas-robot-shows-off-parkour-skills.

[28] Z. Petr, T. Cnops, and C. Remy, "A detailed look at the slip model dynamics: Bifurcations, chaotic behavior, and fractal basins of attraction.," *Journal of Computational and Nonlinear Dynamics*, vol. 14, no. 8, 2019.

[29] J. Usherwood, "Inverted pendular running: a novel gait predicted by computer optimization is found between walk and run in birds," *Biology Letters*, 2010.

[30] A. Ramezani, J. W. Hurst, K. A. Hamed, and J. W. Grizzle, "Performance analysis and feedback control of atrias, a three-dimensional bipedal robot," *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 2, 2014.

[31] G. Bledt, P. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," *International Conference on Robotics and Automation (ICRA)*, pp. 1–8, 2018.

[32] M. Raibert, B. B. Jr, M. Chepponis, J. Koechling, J. K. Hodgins, D. Dustman, and W. K. Brennan, "Dynamically stable legged locomotion," *Massachusetts Institute of Technology*, 1989.

[33] J. D. Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," *In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–9, 2018.

[34] I. Poulakakis, J. A. Smith, and M. Buehler, "Modeling and experiments of untethered quadrupedal running with a bounding gait: The scout ii robot," *The International Journal of Robotics Research*, vol. 24, no. 4, pp. 239–256, 2005.

[35] J. A. Smith and I. Poulakakis, "Rotary gallop in the untethered quadrupedal robot scout ii," *In 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 2556–2561, 2004.

[36] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," *arXiv preprint arXiv:1804.10332*, 2018.

[37] G. Bledt, M. Powell, B. Katz, J. Carlo, P. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2245–2252, 2018.

[38] C. Hubicki, A. Abate, P. Clary, S. Rezazadeh, M. Jones, A. Peekema, J. V. Why, R. Domres, A. Wu, W. Martin, and H. Geyer, "Walking and running with passive compliance: Lessons from engineering: A live demonstration of the atrias biped," *IEEE Robotics and Automation Magazine*, vol. 25, no. 3, pp. 23–39, 2018.

[39] H. Park, P. Wensing, and S. Kim, "High-speed bounding with the mit cheetah 2: Control design and experiments," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 167–192, 2017.

[40] R. Gillette and T. Angle, "Recent developments in canine locomotor analysis: A review," *The Veterinary Journal*, vol. 178, 2008.

[41] D. W. Haldane, M. Plecnik, J. K. Yim, and R. S. Fearing, "Robotic vertical jumping agility via series-elastic power modulation," *Science Robotics*, 2016.

[42] S. Curran, D. Orin, B. Knox, and J. Schmiedeler, "Analysis and optimization of a series-elastic actuator for jumping in robots with articulated legs," *Dynamic Systems and Control Conference*, vol. 43352, pp. 287–294, 2008.

[43] C. Hubicki, M. Jones, M. Daley, and J. Hurst, "Do limit cycles matter in the long run? stable orbits and sliding-mass dynamics emerge in task-optimal locomotion," *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[44] L. Long and M. Srinivasan, "Walking, running, and resting under time, distance, and average speed constraints: optimality of walk-run-rest mixtures," *Journal of The Royal Society Interface*, vol. 10, no. 81, 2013.

[45] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane, "Computational co-optimization of design parameters and motion trajectories for robotic systems," *The International Journal of Robotics Research*, vol. 37, 2018.

[46] R. Margaria, "Biomechanics and energetics of muscular exercise," *Clarendon Press*, 1976.

[47] I. SPECTRUM, *RHex Does Parkour All Over UPenn*, 2020 (accessed July 1, 2020), https://spectrum.ieee.org/automaton/robotics/drones/rhex-does-parkour-all-over-upenn.

[48] N. Geographic, *Finding a New Monkey for East Africa*, 2020 (accessed July 1, 2020), https://blog.nationalgeographic.org/2015/02/05/finding-a-new-monkey-for-east-africa/.

[49] M. Hutter, C. D. Remy, M. A. Hoepflinger, and R. Siegwart, "Scarleth: Design and control of a planar running robot," *In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 562–567, 2011.

[50] H. Dai and R. Tedrake, "Optimizing robust limit cycles for legged locomotion on unknown terrain," *Decision and Control (CDC)*, 2012.

[51] D. Kim, J. D. Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, 2019.

[52] S. B. Williams, H. Tan, J. Usherwood, and A. Wilson, "Pitch then power: limitations to acceleration in quadrupeds," *Biology Letters*, vol. 5, no. 5, pp. 610–613, 2009.

[53] Na, *Horse Skidding.*, 2020 (accessed July 1, 2020), https://www.youtube.com/watch?v=Fcsi8MPICXM.

[54] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, and J. H. et al., "Anymal-a highly mobile and dynamic quadrupedal robot," *In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 38–44, 2016.

[55] C. Semini, N. G. Tsagarakis, E. Guglielmino, and D. G. Caldwell, "Design and experimental evaluation of the hydraulically actuated prototype leg of the hyq robot," *In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3640–3645, 2010.

[56] J. K. Hodgins, "Biped gait transitions," *International Conference on Robotics and Automation*, pp. 2092–2097, 1991.

[57] M. Buehler, R. Playter, and M. Raibert, "Robots step outside," *In Int. Symp. Adaptive Motion of Animals and Machines (AMAM)*, pp. 1–4, 2005.

[58] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of hyq–a hydraulically and electrically actuated quadruped robot," *Proceedings of the Institution of Mechanical Engineers, Part I Journal of Systems and Control Engineering*, vol. 225, no. 6, pp. 831–849, 2011.

[59] G. Kenneally, A. De, and D. Koditschek, "Design principles for a family of direct-drive legged robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, 2016.

[60] C. Fisher, A. Blom, and A. Patel, "Baleka: A bipedal robot for studying rapid maneuverability," *Frontiers in Mechanical Engineering*, 2020.

[61] J. A. Grimes and J. W. Hurst, "The design of atrias 1.0 a unique monopod, hopping robot," *In Adaptive Mobile Robotics*, pp. 548–554, 2012.

[62] M. Hutter, C. Gehring, A. Lauber, F. Gunther, C. D. Bellicoso, V. Tsounis, and P. F. et al, "Anymal-toward legged robots for harsh environments," *Advanced Robotics*, vol. 31, no. 17, pp. 918–931, 2017.

[63] K. Sreenath, H. Park, I. Poulakakis, and J. Grizzle, "Design and experimental implementation of a compliant hybrid zero dynamics controller for walking on mabel," *Conference on Decision and Control (CDC)*, no. 280-287, 2010.

[64] K. Sreenath, H. Park, I. Poulakakis, and J. Grizzle, "Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on mabel," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 324–345, 2013.

[65] K. Sreenath, H. Park, I. Poulakakis, and J. Grizzle, "A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1170–1193, 2011.

[66] C. Lee and S. Oh, "Development, analysis, and control of series elastic actuator-driven robot leg," *Frontiers in Neurorobotics*, 2019.

[67] J. Schmiedeler and K. Waldron, "Leg stiffness and articulated leg design for dynamic locomotion," *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 36533, pp. 1105–1112, 2002.

[68] Y. Ding and H. Park, "Design and experimental implementation of a quasi-direct-drive leg for optimized jumping," *International Conference on Intelligent Robots and Systems (IROS)*, pp. 300–305, 2017.

[69] A. De, G. Lynch, A. Johnson, and D. Koditschek, "Motor sizing for legged robots using dynamic task specification," *Conference on Technologies for Practical Robot Applications*, pp. 64–69, 2011.

[70] G. Bledt and S. Kim, "Extracting legged locomotion heuristics with regularized predictive control," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[71] S. Fengy, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization based full body control for the atlas robot," *IEEE-RAS International Conference on Humanoid Robots*, 2014.

[72] P. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim, "Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 509–522, 2017.

[73] J. Yu, D. Hong, and M. Haberland, "Energetic efficiency of a compositional controller on a monoped with an articulated leg and slip dynamics," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2221–2228, 2018.

[74] M. Haberland and S. Kim, "On extracting design principles from biology: Ii. case study-the effect of knee direction on bipedal robot running efficiency," *Bioinspiration & Biomimetics*, vol. 10, no. 1, 2015.

[75] A. Biewener and S. Patek, "Animal locomotion," *Oxford University Press*, 2003.

[76] P. E. Hudson, S. A. Corr, and A. M. Wilson, "High speed galloping in the cheetah (acinonyx jubatus) and the racing greyhound (canis familiaris): spatio-temporal and kinetic characteristics," *Journal of Experimental Biology*, vol. 215, no. 14, pp. 2425–2434, 2012.

[77] M. Hildebrand, "Motions of the running cheetah and horse," *Journal of Mammalogy*, vol. 40, no. 4, pp. 481–495, 1959.

[78] M. Hildebrand, "Further studies on locomotion of the cheetah," *Journal of Mammalogy*, vol. 42, no. 1, pp. 84–91, 1961.

[79] B. Pontòn, F. Farshidian, and J. Buchli, "Learning compliant locomotion on a quadruped robot," *IROS Workshop (Ed.), Compliant manipulation: Challenges in learning and control*, 2014.

[80] KOD*LAB, *KOD*LAB*, 2020 (accessed November 6, 2020), https://kodlab.seas.upenn.edu/group/jeff/.

[81] M. News, *MIT cheetah robot lands the running jump*, 2020 (accessed November 6, 2020), https://news.mit.edu/2015/cheetah-robot-lands-running-jump-0529.

[82] C. Gehring, S. Coros, M. Hutter, C. Bellicoso, H. Heijnen, R. Diethelm, and M. Bloesch, "Practice makes perfect: An optimization-based approach to controlling agile motions for a quadruped robot," *IEEE Robotics and Automation Magazine*, vol. 23, no. 1, 2016.

[83] P. Electronics, *For sale soon: Quadruped robot that picks up objects, climbs stairs, walks*, 2020 (accessed November 6, 2020), https://www.powerelectronics.com/markets/robotics/article/21864156/for-sale-soon-quadruped-robot-that-picks-up-objects-climbs-stairs-walks.

[84] ANYbotics, *ANYmal – Let Robots Go Anywhere*, 2020 (accessed November 6, 2020), https://www.youtube.com/watch?v=m1-s8iOJaI4&app=desktop&ab_channel=ANYbotics.

[85] M. Khoramshahi, H. Bidgoly, S. Shafiees, A. Asaei, A. Ijspeert, and M. Ahmadabadi, "Piecewise linear spine for speed-energy efficiency trade-off in quadruped robots," *Robotics and Autonomous Systems*, vol. 61, no. 12, 2013.

[86] Eureka!, *Quadruped robot does a backflip*, 2020 (accessed November 6, 2020), https://www.eurekamagazine.co.uk/design-engineering-news/backflipping-robot/212761/.

[87] K. Koutsoukis and E. Papadopoulos, "Comparing the effects of revolute and prismatic spinal joint on quadrupedal bounding," *School of Mechanical Engineering, National Technical University of Athens*, 2016.

[88] K. Weinmeister, P. Eckert, H. Witte, and A. Ijspeert, "Cheetah-cub-s: Steering of a quadruped robot using trunk motion," *IEEE International Symposium on Safety, Security and Rescue Robotics*, 2015.

[89] P. M. Wensing and D. E. Orin, "3d-slip steering for high-speed humanoid turns," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[90] V. Kurtz, P. Wensing, M. Lemmon, and H. Lin, "Approximate simulation for template-based whole-body control," *arXiv preprint arXiv:2006.09921.*, 2020.

[91] V. Kurtz, R. da Silva, P. Wensing, and H. Lin, "Formal connections between template and anchor models via approximate simulation," *IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pp. 64–71, 2019.

[92] J. Grizzle, J. Hurst, B. Morris, H. Park, and K. Sreenath, "Mabel, a new robotic bipedal walker and runner," *American Control Conference*, pp. 2030–2036, 2009.

[93] G. Haynes and A. Rizzi, "Gaits and gait transitions for legged robots," *Proceedings 2006 IEEE International Conference on Robotics and Automation*, 2006.

[94] M. Raibert, "Legged robots," *Communication of ACM*, vol. 29, no. 6, 1986.

[95] M. Qiao and D. Jindrich, "Task-level strategies for human sagittal-plane running maneuvers are consistent with robotic control policies," *PLoS One*, vol. 7, no. 12, 2012.

[96] P. M. Wensing and D. E. Orin, "Generation of dynamic humanoid behaviors through task-space control with conic optimization," *In 2013 IEEE International Conference on Robotics and Automation*, pp. 3103–3109, 2013.

[97] J. Degrave, M. Burm, T. Waegeman, F. wyffels, and B. Schrauwen, "Comparing trotting and turning strategies on the quadrupedal oncilla robot," *International Conference on Robotics and Biomimetics*, 2013.

[98] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Fast, robust quadruped locomotion over challenging terrain," *In 2010 IEEE International Conference on Robotics and Automation*, pp. 2665–2670, 2010.

[99] H. Hirukawa, S. Hattori, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, and M. Morisawa, "A universal stability criterion of the foot contact of legged robots - adios zmp," *IEEE International Conference on Robotics and Automation*, 2006.

[100] B. Dynamics, *Cheetah Robot runs 28.3 mph; a bit faster than Usain Bolt*, 2020 (accessed July 1, 2020), https://www.youtube.com/watch?v=chPanW0QWhA&ab_-channel=BostonDynamics.

[101] I. V. Caekenberghe, K. D. Smet, V. Segers, and D. D. Clercq, "Overground vs. treadmill walk-to-run transition," *Gait & posture*, vol. 34, no. 4, pp. 420–428, 2010.

[102] I. Spectrum, *Teaching Bipedal Robots to Step Across Discrete Terrain*, 2020 (accessed July 1, 2020), https://spectrum.ieee.org/automaton/robotics/robotics-hardware/dynamic-walking-on-stepping-stones.

[103] R. M. Alexander, *Optima for animals.* Princeton University Press, 1996.

[104] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *Society for Industrial and Applied Mathematics*, vol. 59, no. 4, pp. 849–904, 2017.

[105] M. Fevre, B. Goodwine, and J. Schmiedeler, "Design and experimental validation of a velocity decomposition-based controller for underactuated planar bipeds," *Robotics and Automation Letters*, pp. 1896–1903, 2018.

[106] P. Wensing, S. Kim, and J. Slotine, "Linear matrix inequalities for physically consistent inertial parameter identification: A statistical perspective on the mass distribution," *Robotics and Automation Letters*, vol. 3, no. 1, pp. 60–67, 2017.

[107] T. Lens, K. Radkhah, and O. von Stryk, "Simulation of dynamics and realistic contact forces for manipulatorsand legged robots with high joint elasticity," *International Conference on Advanced Robotics*, pp. 34–41, 2011.

[108] M. Khadiv, S. Moosavian, A. Yousefi-Koma, M. Sadedel, A. Ehsani-Seresht, and S. Mansouri, "Rigid vs compliant contact: an experimental study on biped walking," *Multibody System Dynamics*, vol. 45, pp. 379–401, 2018.

[109] L. Drnach and Y. Zhao, "Trajectory optimization over uncertain terrain using stochastic complementarity.," *Dynamic Walking*, 2020.

[110] A. Wachter and L. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[111] C. C. M. Posa and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[112] A. Patel, S. Shield, S. Kazi, A. Johnson, and L. Biegler, "Contact implicit trajectory optimization using orthogonal collocation," *Robotics and Automation Letters*, vol. 4, no. 2, pp. 2242–2249, 2019.

[113] D. Ralph and S. Wright, "Some properties of regularization and penalization schemes for mpecs," *Optimization Methods and Software*, vol. 19, no. 5, pp. 527–556, 2004.

[114] R. Fletcher and S. Leyffer, "Solving mathematical programs with complementarity constraints as nonlinear programs," *Optimization Methods and Software*, vol. 19, no. 1, pp. 15–40, 2004.

[115] T. Hoheisel, C. Kanzow, and A. Schwartz, "Theoretical and numerical comparison of relaxation methods for mathematical programs with complementarity constraints," *Mathematical Programming*, vol. 137, 2013.

[116] J. Duperret, G. Kenneally, J. Pusey, and D. Koditschek, "Towards a comparative measure of legged agility," *Experimental Robotics: The 14th International Symposium on Experimental Robotics*, 2016.

[117] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics*. Springer Science & Business Media, 2010.

[118] S. Safdarnejad, J. Hedengren, N. Lewis, and E. Haseltine, "Initialization strategies for optimization of dynamic systems," *Journal of Computers and Chemical Engineering*, vol. 78, 2015.

[119] S. H. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, p. 1082–1085, 2005.

[120] G. D. Corporation, "Generic algebraic modeling system (gams),24.4.6," *Washington, DC*, 2015.

[121] L. Biegler, "Nonlinear programming: concepts, algorithms, and applications to chemical processes," *SIAM*, vol. 10, 2010.

[122] A. Blom and A. Patel, "Investigation of a bipedal platform for rapid acceleration and braking manoeuvres," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 426–432, 2018.

[123] N. Doshi, K. Jayaram, B. Goldberg, Z. Manchester, R. Wood, and S. Kuindersma, "Contact-implicit optimization of locomotion trajectories for a quadrupedal microrobot," *Robotics: Science and Systems (RSS)*, 2018.

[124] A. Knemeyer, S. Shield, and A. Patel, "Minor change, major gains: The effect of orientation formulation on solving time for multi-body trajectory optimization.," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, 2020.

[125] R. Christensen, J. Hollerbach, Y. Xu, and S. Meek, "Inertial-force feedback for the treadport locomotion interface," *Presence Teleoperators & Virtual Environments*, vol. 9, no. 1, 2001.

[126] OriginFitness, *Treadmill Mechanical*, 2020 (accessed July 1, 2020), https://originfitness.co.za/commercial-gym-equipment/commercial-cardio/treadmills/mechanical-treadmill/.

[127] C. Fisher, J. van Zyl, R. Govender, and A. Patel, "Optimization-inspired controller design for transient legged locomotion," *International Conference on Robotics and Automation, Under Review*, 2020.

[128] R. Walter and D. Carrier, "Rapid acceleration in dogs: ground forces and body posture dynamics," *The Journal of Experimental Biology*, 2009.

[129] C. Fisher, C. Hubicki, and A. Patel, "Do intermediate gaits matter when rapidly accelerating?," *Robotics and Automation Letters*, vol. 4, no. 4, pp. 3418–3424, 2019.

[130] C. Santos and V. Matos, "Gait transition and modulation in a quadruped robot: A brainstem-like modulation approach," *Robotics and Autonomous Systems*, vol. 59, no. 9, 2011.

[131] youtube, *Cheeetah Running*, 2019 (accessed Feb 21, 2019), https://youtu.be/mufJyH7qRpI.

[132] C. Fisher, S. Shield, and A. Patel, "The effect of spine morphology on rapid acceleration in quadruped robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[133] M. Srinivasan, "Fifteen observations on the structure of energy-minimizing gaits in many simple biped models," *Journal of the Royal Society Interface*, vol. 8, no. 54, 2010.

[134] C. Fisher and A. Patel, "On the optimal spine morphology for rapid maneuvers in quadruped robots," *Under Review*, 2019.

[135] M. Haberland and S. Kim, "On extracting design principles from biology: I. method–general answers to high-level design questions for bioinspired robots," *Bioinspiration & Biomimetics*, vol. 10, no. 1, 2015.

[136] C. Hubicki and J. Hurst, "Running on soft ground: Simple energy-optimal disturbance rejection," *Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR)*, 2012.

[137] C. Hubicki and J. Hurst, "Optimization-inspired control policies for analytically and computationally intractable systems.," 2011.

[138] P. M. Wensing and D. E. Orin, "High-speed humanoid running through control with a 3d-slip model," *In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5134–5140, 2013.

[139] A. Shkolnik and R. Tedrake, "High-dimensional underactuated motion planning via task space control," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.

Appendix A

Equations of Motion

Each robot was modelled using rigid links. Each link had a mass, moment of inertia and centre of mass. Rigid links were connected via torque actuators. In order to generate the EoM, first the generalised coordinates were defined, $\mathbf{q}$, along with the velocity, $\dot{\mathbf{q}}$, and acceleration, $\ddot{\mathbf{q}}$.

Using the generalised coordinates, the position vector of the CoM, P_{CoM}, of each rigid link was defined. Using the Jacobian matrix, the velocity of the CoM was calculated:

$$\dot{P}_{CoM} = jacobian(P_{CoM}, \mathbf{q})\dot{q} \tag{A.1}$$

Bounds were placed on the leg angles (see Table A.1) in order to ensure the range of motion of each link was feasible for the robot to execute. These bounds enforced the leg configuration (for example the "X" configuration for the quadruped) and ensured the legs could not rotate past the body (from horizontal to horizontal with respect to the body). These bounds were enforced in the trajectory optimisation problem and not the EoM.

From these positions and velocities, the kinetic, T, and potential, V, energy was calculated for each link (summed to generate the total kinetic and potential energy of the system).

$$\begin{aligned} T_{linear} &= \frac{1}{2} m \times \dot{P}_{CoM} \times \dot{P}'_{CoM} \\ T_{rotational} &= \frac{1}{2} \omega' \times J_{MoI} \times \omega \\ V &= m \times g \times P_{CoM_z} \end{aligned} \tag{A.2}$$

Table A.1: Leg Morphology Bounds

Quadruped				
	Hip Angle		Knee Angle	
2 link leg	Lower Bound	Upper Bound	Lower Bound	Upper Bound
'X' rear	10°	170°	0°	150°
'X' front	10°	170°	−150°	0°
	Hip Angle		Leg Length	
	Lower Bound	Upper Bound	Lower Bound	Upper Bound
Prismatic leg	10°	170°	$0.125m$	$0.25m$
Biped/Monopod				
	Hip Angle		Knee Angle	
	Lower Bound	Upper Bound	Lower Bound	Upper Bound
2 link leg	10°	170°	0°	150°
	Hip Angle		Leg Length	
	Lower Bound	Upper Bound	Lower Bound	Upper Bound
Prismatic leg	10°	170°	$0.125m$	$0.25m$

where m was the mass of the link, g was the gravitational constant and P_{CoM_z} was the vertical height of the CoM. ω was the angular rate vector of the link, with J_{MoI} being the moment of inertia of the link.

To generate the EoM, the manipulator equation was used:

$$\mathbf{M}(\mathbf{q},\dot{\mathbf{q}})\ddot{\mathbf{q}} = -\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) + \mathbf{B}(\mathbf{q})\boldsymbol{\tau} + \mathbf{J}'_{foot}\boldsymbol{\lambda} \tag{A.3}$$

where $\mathbf{M}$ was the mass matrix, $\mathbf{C}$ the Coriolis matrix, $\mathbf{G}$ the gravitational potential matrix, $\mathbf{B}$ maps the applied forces and torques on the system and $\mathbf{J}'_{foot}\boldsymbol{\lambda}$ maps the ground reaction force at the foot to the generalised coordinates.

To calculate the mass matrix, the following formula was used:

$$\mathbf{M} = jacobian(jacobian(T_{total},\dot{\mathbf{q}})',\dot{\mathbf{q}}) \tag{A.4}$$

where T_{total} was the total kinetic energy of the system. Using partial differentiation, the gravitational potential matrix was calculated as follows:

```
for i = 1:length(q)
    G(i) = diff(V_total,q(i))
end
```

where V_{total} was the total potential energy of the system. Calculating the Coriolis matrix was slightly more complex and was done as follows:

```
for i = 1:length(q)
 for j = 1:length(q)
  for m = 1:length(q)
   temp=0.5(diff(M(i,j),q(m)) + diff(M(i,m),q(j)) - diff(M(j,m),q(i)))q̇(m);
   C(i,j) = C(i,j)+ temp;
  end
 end
end
```

In order to map the GRF to the generalised coordinates, the Jacobian of the foot position was calculated, this was done as follows:

$$\mathbf{J}_{foot} = jacobian(\mathbf{P}_{foot}, \mathbf{q}) \tag{A.5}$$

where P_{foot} was the position of the foot. Lastly the $\mathbf{B}$ matrix needed to be calculated. This matrix maps the actuators forces and torques to the appropriate generalised coordinates. This was different for each robot (and depends if absolute or relative coordinates were used). For example if there was a torque actuator acting on the second generalised coordinate and a force actuator acting on the third generalised coordinate then $\mathbf{B}$ would be:

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{A.6}$$

and $\boldsymbol{\tau}$ would be:

$$\boldsymbol{\tau} = \begin{bmatrix} torque\ actuator \\ force\ actuator \end{bmatrix} \tag{A.7}$$

Appendix B

PFL Mapping

In order to generate a PFL (partial feedback linearisation) mapping [139], the EoM were divided into actuated and unactuated equations as follows:

$$unactuated : \mathbf{M}_{11}\ddot{\mathbf{q}}_1 + \mathbf{M}_{12}\ddot{\mathbf{q}}_2 + \mathbf{h}_1 = 0 \tag{B.1}$$

$$actuated : \mathbf{M}_{21}\ddot{\mathbf{q}}_1 + \mathbf{M}_{22}\ddot{\mathbf{q}}_2 + \mathbf{h}_2 = \boldsymbol{\tau} \tag{B.2}$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11}\mathbf{M}_{12} \\ \mathbf{M}_{21}\mathbf{M}_{22} \end{bmatrix} \tag{B.3}$$

where subscript 1 refers to the unactuated generalised coordinates, $\mathbf{q}_1 = [X, Z, \theta_{body}, l_{leg}]$ and subscript 2 refers to the actuated coordinates, $\mathbf{q}_2 = [\theta_{leg}]$, with $\mathbf{M}$ representing the mass matrix. This holds for n generalised coordinates, with m unactuated coordinates and l actuated coordinates with $l = (n - m) > 0$. For the monopod robot, the leg force (the pneumatic actuator force) was considered a disturbance, D, as it was considered 'bang-bang' and therefore not controllable. This resulted in $\mathbf{h}$ as follows:

$$\mathbf{h} = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{D} - \mathbf{J}_{foot}.'\lambda \tag{B.4}$$

$\mathbf{h}$ was split between the unactuated, $\mathbf{h}_1$, and actuated, $\mathbf{h}_2$, coordinates. The desired task was to stabilize the body angle:

$$y = \theta_{body} = \mathbf{f}(\mathbf{q}) \tag{B.5}$$

Using the desired task, a Jacobian matrix was defined as follows:

$$\begin{aligned} \mathbf{J} &= [\mathbf{J}_1, \mathbf{J}_2] \\ \mathbf{J}_1 &= \frac{\partial \mathbf{f}}{\partial \mathbf{q}_1} = [0, 0, 1, 0] \\ \mathbf{J}_2 &= \frac{\partial \mathbf{f}}{\partial \mathbf{q}_2} = 0 \\ \dot{\mathbf{J}} &= \frac{\partial \mathbf{J}}{\partial \mathbf{q}} = [0, 0, 0, 0, 0] \end{aligned} \tag{B.6}$$

According to [139], if the actuated joints are commanded as follows:

$$\ddot{\mathbf{q}}_2 = \bar{\mathbf{J}}^{+}[\mathbf{v} - \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}_1\mathbf{M}_{11}^{-1}(\mathbf{h}_1)] \tag{B.7}$$

where $\bar{\mathbf{J}} = \mathbf{J}_2 - \mathbf{J}_1\mathbf{M}_{11}^{-1}\mathbf{M}_{12}$ and $\bar{\mathbf{J}}^{+} = \bar{\mathbf{J}}^{T}(\bar{\mathbf{J}}\bar{\mathbf{J}}^{T})^{-1}$, then $\ddot{\mathbf{y}} = v$. The desired task space trajectory can then be commanded as follows:

$$\mathbf{v} = \ddot{\mathbf{y}}_d + K_d(\dot{\mathbf{y}}_d - \dot{\mathbf{y}}) + K_p(\mathbf{y}_d - \mathbf{y}) \tag{B.8}$$

Substituting (B.1), (re-arrange and solve for $\ddot{\mathbf{q}}_1$), (B.7) and (B.8) into (B.2), the commanded torque was as follows:

$$\tau = K\mathbf{v} \tag{B.9}$$

where K was some constant, which effectively changed the gains in (B.8). Therefore a PD mapping, as seen in (B.8), was implemented instead.

Appendix C

Physical Experiments

In addition to the free body experiments shown in Chapter 6, vertical and "fixed body" experiments were also performed. The "fixed body" experiments had the body angle fixed and did not allow rotation. Therefore the robot could move forwards, up/down, rotate the leg and lengthen/shorten the leg. The "free body" experiments allowed the body to rotate as well.

C.1 Vertical Experiments

Initial tests were performed in order to validate the system that was developed in collaboration with Joshua Van Zyl. These tests included continuous vertical hopping (with the body and leg fixed in place), testing the accuracy of the KF (Kalman Filter) that was providing state estimation of the robot (position, velocity and acceleration). A state machine was implemented with the following states:

1. Touch-down: As soon as the robot foot touched the ground (using the height data from the KF and the length of the leg), the pneumatic cylinder was actuated.
2. Aerial positive: While the robot was travelling vertically up, the solenoid remain activated.
3. Aerial negative: While the robot was travelling vertically down, the solenoid was de-activated, allowing the leg to compress on touch-down.

An average height of 0.35 m was achieved, with the resulting vertical height of the body and foot shown in Figure C.1. It was noted that the leg fully compressed during the stance phase.

The vertical tests were repeated, except with a free body. The results are shown in Figure C.2. By adjusting when the cylinder fires, the leg did not bottom out during these tests and the

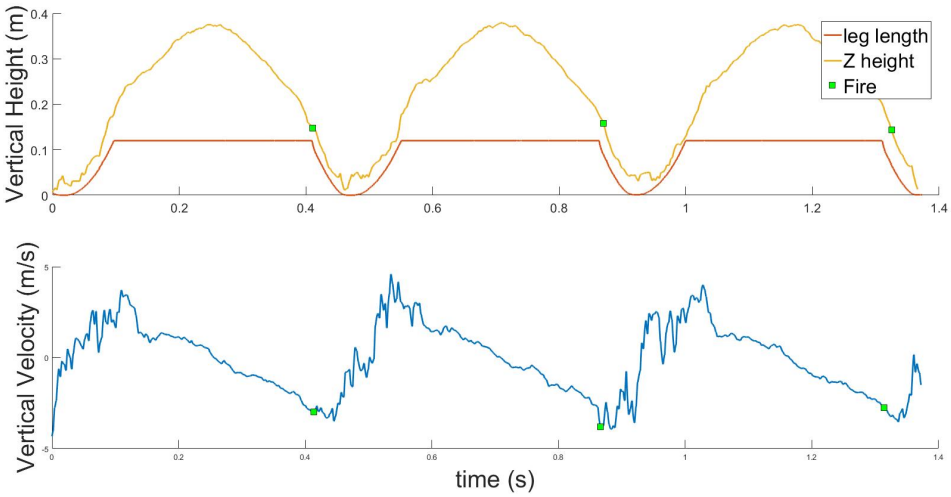

Figure C.1: Notice how the robot achieved an average vertical height of 0.35 m, with the leg fully compressing during the stance phase. The green squares represent when the solenoid was activated to apply leg force.

robot achieved a lower average height.

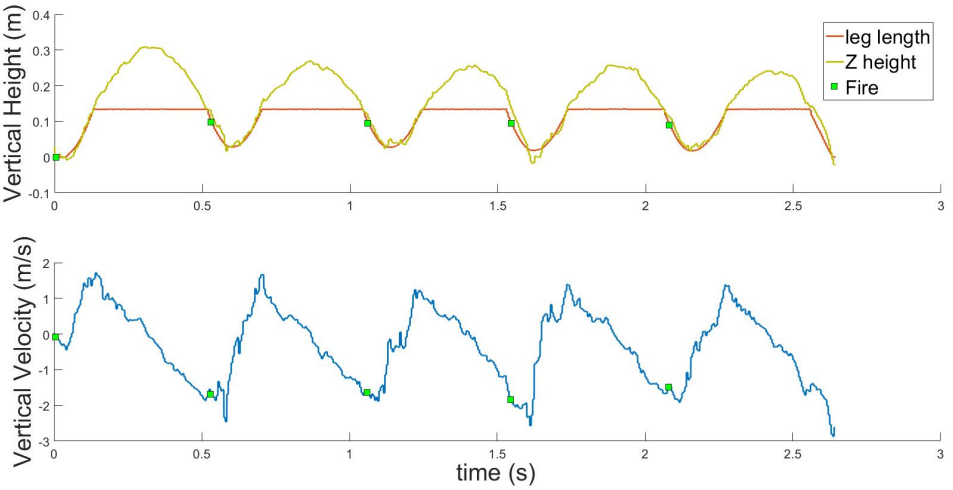

Figure C.2: Notice how the robot achieved an average vertical height of 0.3 m, with the leg not fully compressing during the stance phase. The green squares represent when the solenoid was activated to apply a leg force.

C.2 Fixed Body Experiments

The vertical experiments were used to test the reliability of the system, ensuring all data was transmitted, received and logged accurately and that the mechanical structure was robust to the impulsive contacts.

The next set of experiments involved performing the long-time-horizon task using the controllers

developed in a previous chapter. For these experiments the body was fixed in the horizontal position. A state machine was implemented with the following states:

1. Stance: The solenoid was activated on touch-down, with the stance controllers implemented.
2. Aerial (positive velocity): The solenoid remained activated with the swing controllers implemented. These controllers swung the leg forward to the desired touch-down angle.
3. Aerial (negative velocity): The solenoid was de-activated, allowing the foot to collapse on touch-down. The swing controllers remained activated, holding the foot in the desired touch-down angle.

The step number was also tracked. For the first two steps, the acceleration controller was implemented, for the next three steps, the steady-state controller was implemented and for the final step, the deceleration controller was implemented.

The stance controller consisted of the constant gain, K, calculated in the previous chapter (the Kd and Kp gains were not used as the body was fixed and could not rotate). For the aerial controller, the desired touch-down angle was sent to the O-Drive, and using the onboard position controller, the leg moved into the desired position. The onboard position controller was hand tuned to have limited overshoot when the leg was attached to the motor.

Results of the experiment can be seen in Figure C.3. The robot achieved an average height of 0.4 m and travelled a distance of 1.7 m. As the boom was not designed for the fixed body tests, severe flexing and oscillations were experienced, making the vertical and forward velocity very noisy. The leg fully compressed during these experiments.

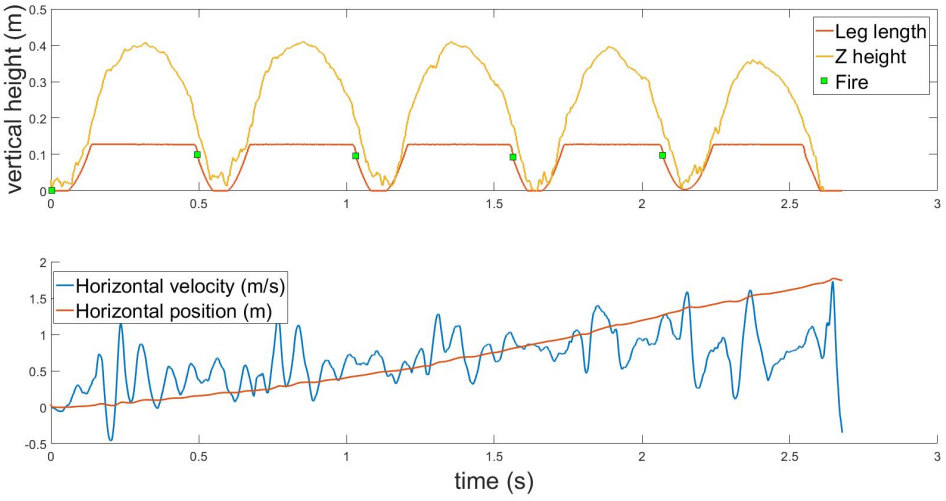

Figure C.3: Experimental results of the fixed body forward tests. The robot travels a distance of 1.7 m with an average velocity of 0.6 m/s. The leg bottomed out during the stance phase with the robot achieving 5 aerial phases, starting and ending at rest.