Control of Rapid Acceleration in a Planar Legged Robot

by

Christopher Mailer

Submitted to the Department of Electrical Engineering
in fulfilment of the requirements for the degree of

Master of Science in Electrical Engineering

at the

UNIVERSITY OF CAPE TOWN

January 2023

Thesis Supervisor
Dr Amir Patel
Associate Professor

Thesis Co-Supervisor
Dr Reuben Govender
Senior Lecturer

Control of Rapid Acceleration in a Planar Legged Robot

by

Christopher Mailer

Submitted to the Department of Electrical Engineering
on December 27, 2022, in fulfilment of the
requirements for the degree of
Master of Science in Electrical Engineering

Abstract

This thesis details the hardware and control design of Kemba: a planar legged robot intended for investigating bounding and explosive, agile manoeuvres. The robot incorporates both pneumatically actuated knees for powerful, compliant, and impact resistant actuation, and proprioceptive electric actuators at the shoulder and hip for high bandwidth torque control and foot placement. Kemba is capable of bounding at up to 1.7 m/s with a full flight phase, jumping just under 1 m high (2.2 times it's nominal leg length), and accelerating from rest into a top speed bound in only 2 strides and under half a second, demonstrating its agility. Stable bounding and acceleration is achieved using a discrete body oscillation stabiliser, and the more dynamic jumping and somersault motions are generated using offline nonlinear trajectory optimisation. The optimal jumping motion was executed on the physical robot while the somersault is currently still limited to simulation.

Due to the unique design and actuator combination, contact implicit trajectory optimisation served as a vital tool for motion identification and controller design. In addition to the robot dynamics and unilateral contact constraints, a more tractable pneumatic actuator model was developed which enabled the numerically stiff, discontinuous air dynamics and discrete valve switching to also be incorporated into the trajectory optimisation formulation. Trajectories resulting from optimisation were accurate enough to be implemented directly on the hardware in the case of the jump motion, and also crucially inform the design of the accelerate from rest controller.

The results presented in this work indicate that Kemba is a robust and agile platform, well suited for future work in understanding dynamic manoeuvres and optimal control.

Thesis Supervisor: Dr Amir Patel
Title: Associate Professor

Thesis Co-Supervisor: Dr Reuben Govender
Title: Senior Lecturer

Declaration of Authorship

I, Christopher Mailer, declare that this thesis, titled *Control of Rapid Acceleration in a Planar Legged Robot* and the work presented in it are my own. I confirm that all the work in this thesis, save for that which is properly acknowledged, is my own. Where I have consulted the published work of others, this is always clearly attributed. This work was done wholly while in candidature for a research degree at the University of Cape Town.

Christopher Mailer
June 23, 2023

Acknowledgements

A number of people have helped support and make this research possible.

In particular I would like to thank my supervisor, Amir Patel, for introducing me to the exciting field of dynamic legged robots and control, and for providing the encouragement and resources to experiment and explore. I have learnt so much from our discussions, and I am incredibly grateful for the opportunity and freedom to design, and experiment with other legged machines in the lab.

I would also like to thank my co-supervisor, Reuben Govender, for the tremendous amount he has taught me about mechanical design and the many hours reviewing and refining designs. His guidance throughout this project has been invaluable.

Stacey Shield, for teaching me about trajectory optimisation and for all of her help with the paper writing and the vision behind it.

Pierre Smith, Grant Springle, and the machinists in the workshop, for their great work on all of the components which went into this robot. They were always willing to help and I have learnt a great deal as a result.

Everyone in the African Robotics Unit, thank you all for the creative discussions, friendship, and technical advice. Working alongside all of you has been inspiring.

Finally, to my family. This would not have been possible without your support and guidance, and I am incredibly grateful to have you all in my life.

Funding

I would also like to extend my sincerest thanks to the Harry Crossley Foundation, as well as the Ada & Bertie Levenstein Bursary Program, and the Vice Chancellor Research Scholarship for their generous financial support. This research was also supported by the National Research Foundation (NRF) of South Africa grant number 137762.

Contents

List of Figures

List of Tables

Acronyms

ADC Analogue to Digital Converter.

CAN Controller Area Network.

DOF Degrees of Freedom.

EOM Equations of Motion.

FOC Field-Oriented Control.

GRF Ground Reaction Force.

IMU Inertial Measurement Unit.

LTI Linear Time-Invariant.

MPC Model Predictive Control.

NLP Nonlinear Program.

OD Outer Diameter.

PD Proportional-Derivative.

PI Proportional-Integral.

PWM Pulse Width Modulation.

RL Reinforcement Learning.

SEA Series Elastic Actuator.

STL Standard Tessellation Language.

TO Trajectory Optimisation.

Chapter 1

Introduction and Literature Review

This thesis documents the design, development and control of Kemba - a planar quadruped robot which combines both powerful, compliant and simple pneumatic actuators with quasi-direct drive actuators into a robust platform capable of rapid acceleration from rest manoeuvres.

Recently legged robots have begun performing more dynamic high acceleration manoeuvres like recovering from disturbances, jumping and somersaults in addition to the well understood steady-state motions. Understanding these motions is crucial in creating robots which can match or exceed the dynamic capabilities of humans and animals. This has prompted investigations into actuation strategies and control techniques better suited to these operating regimes.

For more dynamic, high acceleration manoeuvres, actuators which produce large forces, with little inertia, and can handle high impact loads are essential. Pneumatic actuators have a number of characteristics which make them ideally suited to this dynamic high acceleration task. They are mechanically simple with minimal moving parts making them inexpensive and easy to actuate without sophisticated electronics. They can produce high forces in relatively compact forms without the need for gear reductions, making them inherently robust to impact loads as the forces are distributed throughout the working fluid rather than concentrated at small points on gear teeth. Furthermore, the compressibility of air makes them good analogues for certain aspects of muscles, with slow internal dynamics [1] and inherent passive compliance [2]. However, it is also because of these complex internal dynamics that pneumatic actuators are typically avoided in modern robots, with traditional control

techniques unable to accommodate the additional unactuation this introduces. Additionally, air compressors with the required flow rates are typically large and noisy making them poorly suited for an agile mobile legged robot.

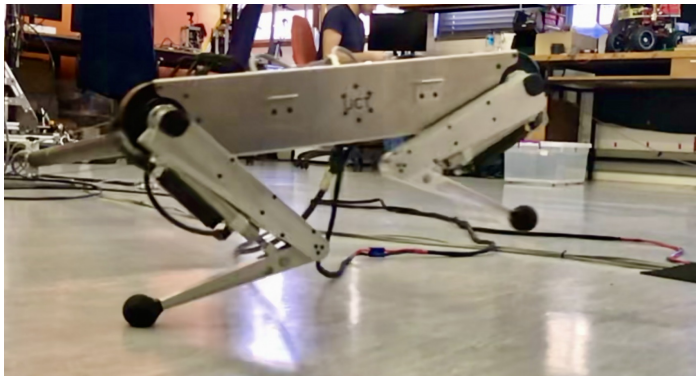

Figure 1-1: A photo of Kemba pushing off of with its hind leg and accelerating into a 1.5 m/s bounding gait.

To this end, Kemba was developed - a planar legged robot which combines off-the-shelf low bandwidth pneumatic pistons with high bandwidth force controlled electric actuators to explore this actuator configuration for complex agile manoeuvres, without the constraints of onboard power or computing. Through characterisation of the internal dynamics, simplifications, and integration with nonlinear optimal control techniques Kemba is able to exhibit bounding, jump, accelerate and somersault motions on par with the dynamic capabilities of existing legged robots, and exceeding the jump height of the previous agile legged robot in the lab. These motions incorporate the discrete switching, delays and nonlinear dynamics characteristic of pneumatic actuators. The same optimal control techniques are also used to inform the design of a controller which enables the robot to rapidly accelerate from rest in two strides and transition directly into a high speed bounding gait as shown in Figure 1-1 - a first for a legged robot.

1.1 Actuation in Legged Robots

Actuating legged robots presents a unique challenge as there are a number of competing performance characteristics. These required characteristics are also shared with the muscle actuators in their biological legged counterparts. Actuators should possess a high level of resilience when subjected to impacts to handle the frequent non-zero

velocity foot ground collisions, particularly for more dynamic manoeuvres. High force/torque output is also required without adding substantial mass to the robot, or inertia to the joint, and high velocities are also necessary for fast foot placement with high speed locomotion, and to reject disturbances. However, for the same locomotion task, robot actuators and biological muscles also differ significantly. Most legged robot control approaches leverage high bandwidth force control, needing frequencies in excess of 1 kHz to work effectively. This presents a notable disparity compared to the neuromechanical delays observed in cat-sized animals which typically exceeding 30 ms for running stance phases with a duration of approximately 100 ms [1]. Animal muscles are also compliant, enabling the storage and release of energy through passive spring-like elements [2].

1.1.1 Hydraulic Actuators

Hydraulic actuators offer incredibly high force density output in very small sizes, enabling the remarkably agile and dynamic behaviours exhibited by the WildCat, Cheetah, and Atlas robots from Boston Dynamics [3], and HyQ [4] from the Dynamic Systems Lab at IIT, shown in Figure 1-2. They can be mounted directly to joints and add very little inertia. Hydraulics are also incredibly robust to impacts as the impact force is distributed evenly within the working fluid. An incompressible working fluid eliminates any passive compliance, with a sealed chamber behaving as a rigid link. Energy storage and release is therefore also not possible, and force and compliance need to be actively controlled by high speed servo valves [5]. Hydraulic fluid is typically at a very high pressure and requires additional return lines to maintain a closed system, adding design complexity.

1.1.2 Electric Actuators

Electric actuators for legged robots can be split into two groups; proprioceptive actuators and Series Elastic Actuator (SEA), which differ predominantly based on their method of torque sensing and control.

Proprioceptive actuators were first used in the MIT Cheetah robots [6, 7]. These actuators feature a simple construction with a brushless electric motor coupled to a low reduction planetary gearbox. Impact loads are partially mitigated by carefully

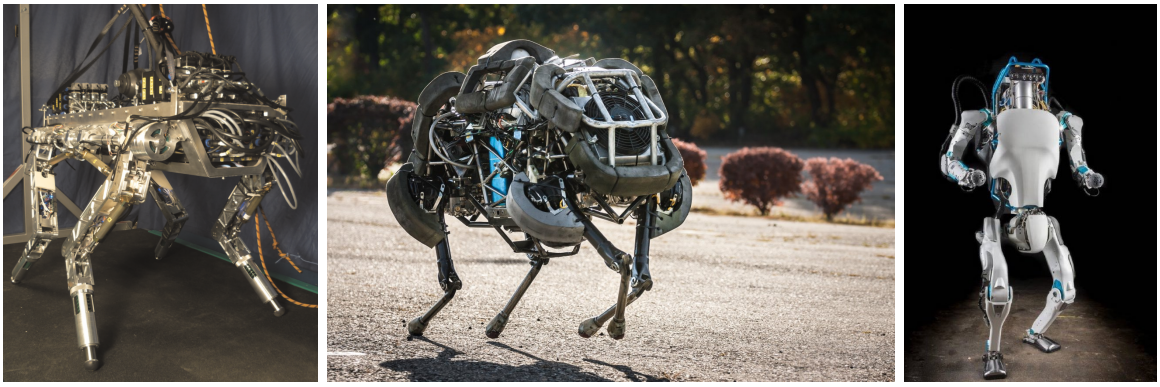

Figure 1-2: Hydraulically actuated robots HyQ [4] (Left), WildCat (Middle), and Atlas (Right) [3].

sizing the transmission (typically 6:1 to 8:1) to keep reflected inertia of the rotor low and not excessively load gear teeth [8]. Open loop high bandwidth torque control is achieved simply by controlling the motor phase currents with a high frequency internal control loop. This enables active spring and damper compliance control. These actuators can also store energy through regenerative braking.

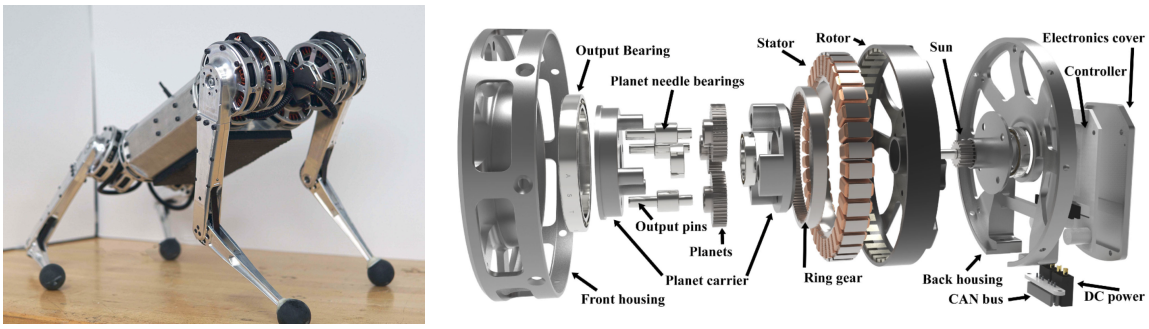

Figure 1-3: Mini Cheetah robot (Left) and exploded view of its proprioceptive actuators (Right) [7].

Series elastic actuators are used most notably on the AnyMal [9] and StarlETH [10] robots from the Robotic Systems Lab at ETH. These actuators combine a highly geared (typically a harmonic drive) electric motor with a series torsional spring element. The torsional spring provides passive output compliance, energy storage, protects the transmission from impact loads, and provides a means for measuring and controlling output torque via the spring deflection. Due to torque being regulated by motor position control, force control bandwidth is significantly lower than proprioceptive actuators. The actuator also requires more mechanical components and sensors, making proprioceptive actuators attractive for robots with lower torque requirements.

1.1.3 Pneumatic Actuators

Pneumatic actuators were first used in the early dynamic hopping robots in the Leg Lab [11]. Air in the cylinder could be compressed, serving as a series compliant element and storing energy. The locomotion algorithms used this compliance and the system's passive dynamics to their advantage enabling remarkably dynamic behaviours such as the first biped somersault [12]. These actuators exhibit a straightforward mechanical design with only two moving parts, making them very robust. They can produce high forces in relatively compact forms without any transmissions, and can be mounted directly to the joint adding very little inertia. Similar to hydraulics, impact forces are transferred through the working fluid and distributed over the large internal surface area making them well suited to high impact forces. As a consequence of the compliance in the air (working fluid), the actuators have very low control bandwidth even with high speed proportional valves [13]. Accurate position and force control is thus challenging, even with computationally demanding Model Predictive Control techniques [14]. An additional interesting characteristic of pneumatic actuators is that stiffness can be varied by controlling the air pressure in opposing piston chambers [13] similar to antagonistic muscle configurations.

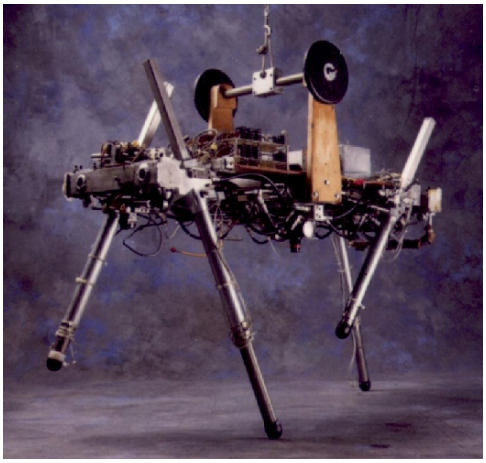

Figure 1-4: Quadruped with pneumatic piston leg actuators [11].

1.2 Legged Robot Control Approaches

The control of legged robots is a particularly challenging problem due to the underactuation and unique constraints imposed by contact. During the extended flight

phase of bounding the robot has no feet on the ground and therefore cannot influence its trajectory in any meaningful way. When feet are on the ground they are limited to only pushing on the ground, and any attempt to pull the body closer will simply result in a lifted foot. From a classical control perspective the model and available control inputs are changing with the contact state. A number of methods have been used to address these challenges which can be broadly combined into the following categories.

1.2.1 Stabilised Oscillations

Early control strategies in the 1980s relied on treating steady-state gaits as coupled oscillations excited by periodic thrusts from the springy legs and maintained by the body's natural dynamics [11, 15]. This decomposed locomotion into simple discrete control laws which would maintain this coupled oscillation while also controlling body velocity, height, and attitude. These control actions, rather than being functions of time, were functions of state, sequenced by an event based state machine. Stability of the oscillation ensured that even with disturbances such as pushes or varying terrain height, the system would converge exponentially back to its periodic orbit, even without this being explicitly enforced [16]. Remarkably dynamic locomotion capabilities were possible in both 2D and 3D with this control approach and robots have been able to exhibit hopping, trotting, pacing, bounding, and pronking gaits. One of the robots could even perform a somersault when commanded by the operator with additional modifications to the controller [12]. This control approach was also used on the relatively recent MIT Cheetah 2 robot to enable fast and highly efficient bounding [17]. Explicit control laws make this controller adaptable to different hardware configurations, surprisingly robust, and relatively intuitive to tune for lower Degrees of Freedom (DOF) legged systems.

1.2.2 Optimisation-based Control

As the system complexity increases with greater degrees of freedom, more complex terrains, varying gait patterns, and more agile and complex motions, hand crafting physically feasible trajectories around the nonlinear body dynamics, actuator characteristics, and contact constraints becomes prohibitively difficult. This has resulted in

locomotion being formulated as an optimisation problem, instead using a gradient-based solver such as IPOPT [18] to find control actions which produce trajectories that adhere to physics, nonlinear body dynamics, unscheduled foot contacts [19], and arbitrary footholds [20]. This approach is known as Trajectory Optimisation (TO) [21] and it is an incredibly powerful tool for synthesising remarkably natural movements for arbitrary robot morphologies [22] given only high level motion constraints. Trajectory Optimisation has enabled a variety of complex acrobatic legged robot manoeuvres [7, 23, 24]. Its fundamental shortcoming is that only a single trajectory is found rather than an optimal control policy. By simplifying the optimisation problem to where it can be run multiple times per second for a short time horizon, any deviation from the current trajectory will be corrected by the subsequent trajectories. This has been termed Model Predictive Control (MPC) and has enabled a variety of complex gaits on legged robots such as galloping [25, 26]. Boston Dynamics used a combination of skilled choreographers, accurate offline Trajectory Optimisation, and online Model Predictive Control to make their robots perform remarkable dance routines [27].

1.2.3 Reinforcement Learning

An alternative approach to tackling the legged locomotion problem is the application of Deep Reinforcement Learning (RL), which has recently shown remarkable progress. Reinforcement learning is a subclass of optimisation algorithms where an agent selects actions to transition to states which maximise some notion of a cumulative reward [28]. For legged locomotion, the transition to a new state as a result of an action is not often known. Model-free reinforcement learning solves this by learning and approximating the action and state transitions, referred to as a policy, with a neural network. Policies can be learnt in simulation and then transferred and evaluated on a real robot. This approach is the most general and has enabled highly robust walking over challenging unseen terrain on the ANYmal quadruped [29] and the Cassie bipedal robot [30]. An issue with reinforcement learning is that policies learnt in simulation often do not transfer to reality due to modelling discrepancies. Policy robustness is typically improved through domain randomisation [29, 30], better actuator models, or online optimisation. Structuring of the networks to a task is challenging, as legged

locomotion remains a complex task and requires significant experience, intuition, and computing resources to achieve even reasonable walking performance.

1.3 Acceleration and Gait Initiation

1.3.1 Animals

When accelerating from rest, quadrupedal animals such as dogs typically use a bounding gait as they can push off evenly with both back legs [31]. Bounding is symmetrical about the sagital plane with the two back legs moving together, out of phase with the front legs. For maximal movement initiation, greyhounds typically go through a number of body posture events as they transition from rest into a top speed gait. The first of these is the counter-movmement, characterised by a brief free-fall from the standing position into the crouched set position. During this movement body weight is transferred entirely onto the rear leg, storing elastic potential energy in the muscle fibres and tendons, and lifting the front legs off of the ground [32]. Animals do not waste time and energy with intermediate gaits when accelerating maximally, pushing off as hard as possible with their hind legs [31] and transitioning directly into the desired gait [33]. For racing greyhounds, the primary limitation to linear acceleration at low speeds was shown to be the prevention of nose-up pitching [34], governed by the equation:

$$a \leq \frac{g l_{spine}}{2 l_{leg}} \tag{1.1}$$

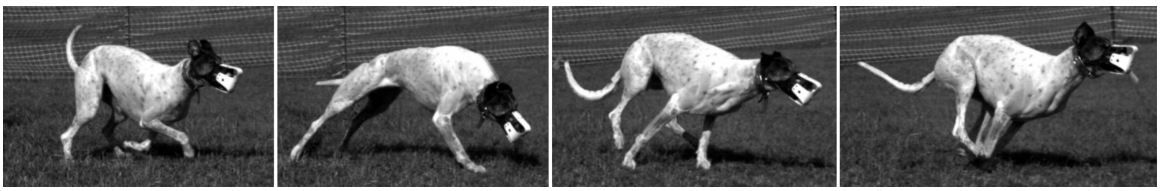

Figure 1-5: Greyhound executing a rapid acceleration manoeuvre from [35].

1.3.2 Legged Robots

Rapid acceleration and gait initiation in a legged robot is the combined result of robot morphology [22, 33], actuators [36], and the controller [37]. Current state-of-the-art MPC gait controllers can produce a variety of fast and agile gait patterns,

but to keep the optimisation problem convex for real-time execution they require a prescribed contact sequence and fixed foot swing and stance times [25, 26]. This is a reasonable simplification for symmetrical and periodic steady-state motions, which has largely been the focus of existing legged locomotion controllers. This, however, does not apply to aperiodic and asymmetrical acceleration manoeuvres which require different body poses, stance times, and ground reaction forces in animals [32]. The limitation of this control assumption is evident when considering the Mini Cheetah - an excellent benchmark for quadruped robot agility capable of performing backflips. Figure 1-6 shows that body velocity increases linearly even when large accelerations are commanded, with a maximum of acceleration of $1.8\,\mathrm{m/s^2}$. This behaviour does not agree with the minimum time sliding mass dynamics predicted in simulation without these controller constraints [16, 37], and it is also significantly lower than the $6.5\,\mathrm{m/s^2}$ predicted by the nose-up pitch limit (1.1) for the robot's dimensions.

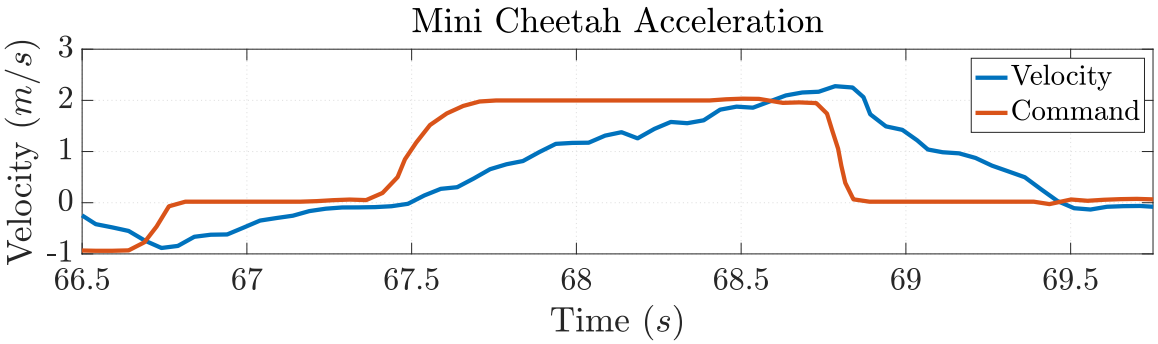

Figure 1-6: Mini Cheetah robot velocity tracking for large commanded accelerations with cMPC controller [7].

1.4 Research Question

This thesis attempts to answer the following research question:

Can low bandwidth, robust pneumatic actuators combined with high-bandwidth proprioceptive actuators be used to control rapid acceleration into a bounding gait for a legged robot?

Inspired by the remarkable agility exhibited by humans and animals, this question aims to utilise the best of two actuation schemes; powerful, robust pneumatic actuators with muscle-like characteristics, and high bandwidth proprioceptive electric actuators to perform a complex dynamic manoeuvre that has not yet been investi-

gated on a physical legged robot. This is done with the aim of creating legged robots that are more agile, and challenging the assumption that control of complex dynamic manoeuvres requires precise high bandwidth force control.

1.5 Approach

First, a small planar legged robot was designed with pneumatically actuated knees and performance characteristics comparable to another similarly sized legged robot, the MIT Mini Cheetah [7]. This is detailed in Chapter 2. To better understand the acceleration motion and how the robot should move given its unique actuator dynamics, trajectory optimisation was used to synthesise an accelerating from rest motion and other dynamic manoeuvres. This provided insight into how the robot should bound and accelerate into this bound, and is detailed in Chapter 3. Following this a bounding and acceleration controller was designed using the insight from these optimisation results and a discrete state-sequenced control approach. The design of the controller, implementation on the hardware, and results of stable bounding and acceleration are described in Chapter 4.

1.6 Contributions

In answering the research question this thesis has produced three key contributions;

1. A robust legged robot with pneumatic actuators at its knees, and high bandwidth torque control motors at its hips, capable of a variety of dynamic and agile manoeuvres such as fast bounding, jumps, accelerating from a crouch and theoretically also somersaults in simulation.

2. A novel pneumatic model and characterisation procedure which captures dominant actuators dynamics, and can be used with modern optimisation-based control. This enables the synthesis of a variety of dynamic manoeuvres which incorporate the unique actuator dynamics. [1]

[1] This contribution has been published as a conference proceeding to the International Conference on Robotics and Automation (ICRA 2023)

3. The identification of three discrete phases in the crouched acceleration motion; namely the *set*, *pop-up*, and *push-off* phases, which enable controller design and integration of the acceleration motion into a stable bounding controller.

Aside from these contributions, the legged robot produced in this work has proven to be robust, which alongside the simple and easy to modify controller, provides a reliable platform for future work investigating agile behaviours with high specific force pneumatic actuators.

1.7 Scope and Limitations

To simplify the control problem, the robot will be constrained to only move in a plane with forward and backwards translation, vertical translation, and body pitching, with no capability for yaw (turning left and right). Additionally, the robot will be assumed to be symmetrical about the sagital plane, inline with the planar approximation, with only a single front and back leg. This dramatically simplified design and manufacture. The legged robot is designed to be the minimum hardware configuration needed to test the desired motions, and other considerations, like onboard power or compressed air, were not necessary as the robot would be permanently tethered. The choice to use pneumatic and proprioceptive electric actuators was imposed as a constraint, informed by previous work in the lab with these actuators and the availability of existing infrastructure [37].

Chapter 2

Planar Robot Design

The following chapter presents the design of the planar legged robot Kemba, along with the planarising support boom and system electronics. Kemba, shown in Figure 2-1, is 0.52 m long, with four actuated joints, weighing 4.5 kg, and standing about 0.3 m tall. It features integrated pneumatic actuators at the knees capable of producing 19 N m of joint torque, and high bandwidth proprioceptive force controlled electric motors at the hips with a peak torque of 25 N m. The robot is supported and constrained by a 2.5 m long hollow carbon fibre boom which routes all of the necessary power, pneumatic, and sensor cables required for operation, while also providing body state information. The boom additionally keeps the robot vertical and its horizontal motions follows a large radius arc, approximating planar motion.

Figure 2-1: Kemba, a hybrid pneumatic and electric actuated legged robot.

2.1 Mechanical Design

Legged robotics is a challenging design space with hard to anticipate loading cases (predominantly during unexpected falls) and a tight coupling between control and the mechanical system. This is probably best conveyed by the design philosophy at Boston Dynamics; "*Build It. Break It. Fix It.*" [38]. Mechanical designs and their controllers typically evolve together in an iterative refinement, however, in the case of Kemba this needed a starting point. The MIT Mini Cheetah robot [7] built by Benjamin Katz is a small, incredibly agile and robust legged robot which utilises almost identical proprioceptive actuators to the ones intended for Kemba's hips, and was also approximately the desired size. This legged robot (parameters summarised in Table 2.1) was therefore used as a rough guideline for sizing of the robot dimensions, knee actuators, and many of the structural leg components.

Mass	9 kg
Maximum Knee Torque	17 N m
Maximum Hip Torque	26 N m
Body Length	0.38 m
Leg Upper Link Length	0.21 m
Leg Lower Link Length	0.18 m

Table 2.1: Mini Cheetah robot parameters [7].

2.1.1 Legs

The design of the legs is an important factor in determining the performance of a legged robot. For Kemba the legs need to be able to support multiple times its body weight for explosive manoeuvres, and provide the rigid structure and mounting for the pneumatic pistons. The design of the leg also largely determines the range of motion and performance characteristics of the robot. To enable quick repositioning of the leg during high speed running using minimal torque the leg should be lightweight and low inertia. Many popular legged control algorithms also rely on being able to ignore the effects of the leg dynamics on the body [6, 25], requiring the legs to be sufficiently low inertia relative to the body. The links will be modelled using rigid body algorithms, and therefore should also be as stiff and rigid as practical while keeping inertia to a

minimum. The pneumatic pistons can only be loaded along their axis, and so the leg must serve to support the piston and constrain forces to only act in this direction. Additionally, any unintended deformation or play in the mechanism effectively adds additional degrees of freedom, resulting in underactuation and positioning errors - a severe hindrance to control. The two main objectives guiding this leg design were therefore to:

- Maximise stiffness and rigidity
- Minimise leg mass and inertia

while also tightly integrating the pneumatic pistons. The design was kept as simple as possible with only 2 rigid links, aiding robustness and modelling. From the onset it was also decided to use an 'X' leg configuration which has been shown to be better suited for rapid acceleration manoeuvres [22] and also ensures the robot can operate identically in either direction - particularly useful for the planar motion.

The primary kinematic purpose of the knee joint for legged robots during locomotion is to shorten the distance between the hip and the foot so that the leg can be swung forward without colliding with the ground. At a high level, a linear pneumatic cylinder in a leg can be arranged in one of two ways to serve this purpose. The first arrangement uses the whole cylinder as the leg, like a pogo stick, where retraction of the cylinder brings the foot closer to the body. This configuration was tested in previous work [37], however, it was found that the additional components required to prevent the significant bending loads on the cylinder rod negated the apparent simplicity of the configuration. Additionally, the minimum length of the leg remains quite long as additional length in the piston is required to house the sintered rod bearings, seals, and pneumatic fittings. This restricts crouching poses which are key to acceleration manoeuvres as they increase the time the foot spends pushing off of the ground. The alternative arrangement is for the leg to be serially actuated with the piston attached in a closed-kinematic chain configuration around the knee similar to many hydraulic piston arrangements as shown in Figure 2-2. This configuration provided some freedom to adjust the torque produced about the knee and also increased the leg's extended and retracted length, with the closed chain arrangement essentially serving as a transmission. The leg linkages also serve as the structural

elements, ensuring the piston rod is only loaded axially.

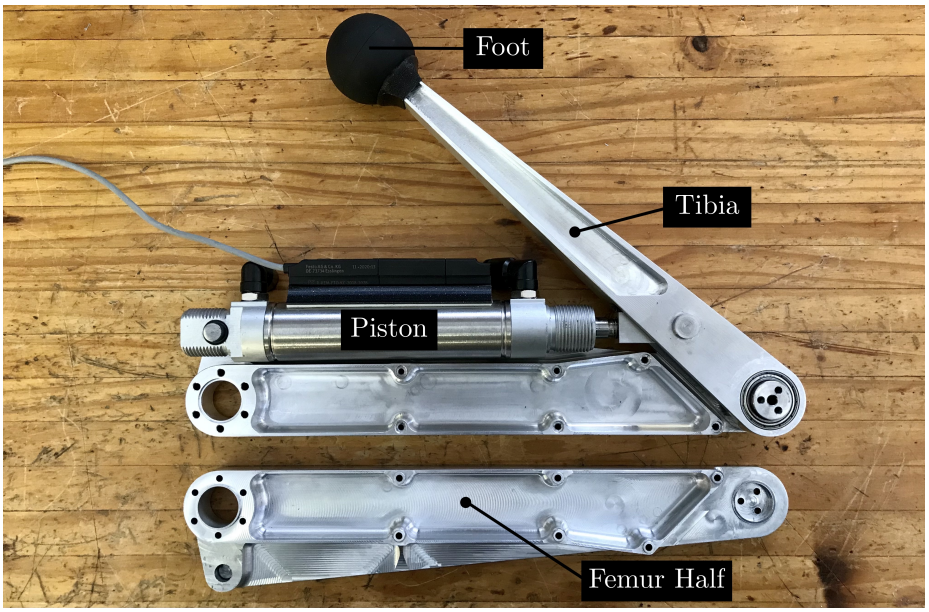

Figure 2-2: Partially disassembled leg with second femur half removed showing piston closed kinematic chain arrangement.

The leg was designed around a stock Festo DSNU-25-70 piston. The 25 mm bore was the largest piston available in the compact lightweight form, providing the highest specific force. Piston bores above 25 mm transition to a significantly more bulky design possibly due to the additional strength required at the bearing and end cap, which would make designing a lightweight compact leg challenging. A 25 mm bore at a nominal pressure of 0.7 MPa also provided enough force to produce sufficient peak knee torque (19 N m) even with relatively small lever arm lengths.

Together the piston stroke, upper link length, lower link lever arm length, and overall leg size all affect the maximum knee torque and knee range of motion. Furthermore, their effects are coupled as a larger piston stroke produces a greater range of motion, but typically requires a longer upper link length, increasing the overall leg size. A longer lever arm will increase the maximum knee torque, but it also reduces the knee's range of motion. Two requirements helped navigate this trade-off and constrain the design space.

Piston must limit range of motion The DNSU series pistons come with air cushioning end stops which are rated for large impact loads and are self-adjusting. These are ideally suited to serve as the knee range of end-stops. This will also

ensure the full piston stroke is utilised and eliminates the need for the design of additional joint limits.

Maximise leg contraction Crouching poses are important for acceleration manoeuvres as they serve to extend the time the foot can be in contact with the ground. The leg should therefore be able to contract as much as possible without any collisions between links or pneumatic fittings.

Additionally, plotting the leg extended and contracted length r (Equation provided in Appendix B) as a function of the permissible link lengths shows that there is an optimal point to simultaneously minimise the contracted leg length and maximise the extended leg length. This is shown in Figure 2-3. A large portion of the link

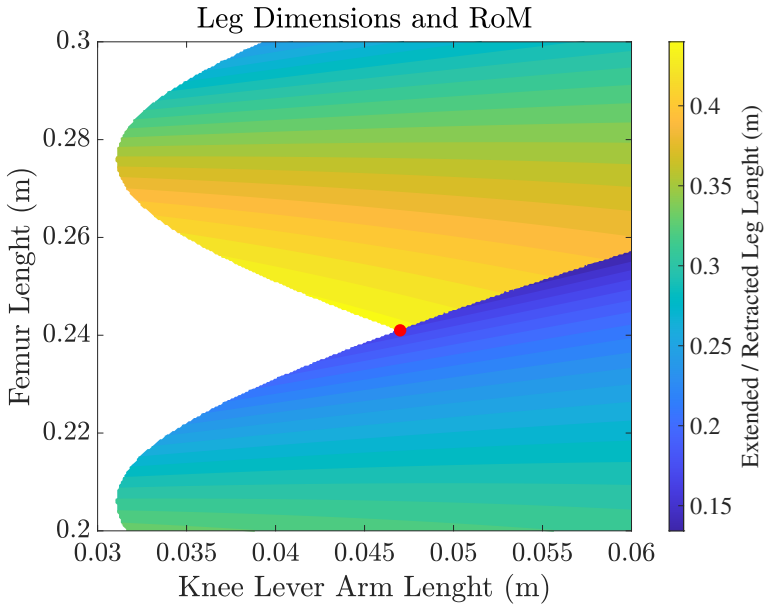

Figure 2-3: Contour plot showing the extended (top) and retracted (bottom) leg length for various lever arm and femur lengths and a 70 mm piston stroke. The optimal point used in the final design which maximises extension and minimises contraction is shown by the red marker.

length combinations are invalid as these would result in the piston intersecting with the femur geometry. The optimal link length combinations for three piston stroke options are tabulated in Table 2.2.

A piston stroke of 70 mm produced a satisfactory compromise between these requirements without over complicating design or manufacture, and kept the leg and overall body size to a reasonable scale. A stroke of 60 mm had lower than desired

Piston Stroke	Femur Length	Lever Arm Length	Peak Knee Torque
60 mm	226 mm	42 mm	16.6 N m
70 mm	242 mm	48 mm	19.0 N m
80 mm	256 mm	51 mm	20.1 N m

Table 2.2: Leg dimensions.

knee torque, and a stroke of 80 mm produced a leg that was going to be sightly too large and a robot bigger than desired without a substantial increase in knee torque.

This design is by no means optimal according to any metric; however, it has demonstrated remarkable capability across a range of motions, making it a favourable initial foundation upon which subsequent designs can improve. This configuration results in a peak knee torque of approximately 19 N m at maximum pressure, a minimum hip to foot length of 170 mm and a fully extended length of 440 mm. While the full range of motion is not utilised during bounding, it proved very useful during the jumping motions where the robot was able to crouch low and extend the push-off duration.

The links in the leg were designed to be light, and more importantly, very stiff in the directions they will experience loads. The lower link, or tibia, only experiences bending loads as a result of the spherical foot, and was therefore designed to have a mostly I-beam cross-section as shown in Figure 2-4 with very thin walls of 1.5 mm to 2.0 mm similar to the MIT Cheetah robots, giving it a mass of only 168 g. The overall size and wall thickness tapers down towards the foot to account for the decreasing magnitude of the bending moment. The design was largely determined by machining capabilities, with anticipated stresses being well below the yield limit (see Appendix A). Thinner walls would have increased machining time and difficulty with only marginal mass savings. A stainless steel pin, in a hollowed out section, serves as the attachment and pivot point for the rod-end clevis which screws onto the end of the piston rod. The bearings for the knee joint are press fit into the tibia, and the foot simply press fits over the end. More complex foot fastening isn't necessary as the ground reaction forces will never be pulling the foot off of the tibia. The upper link, or femur, was designed in two halves with a separate hollow stainless steel shaft for the knee. These two halves clamp together around the piston and tibia holding them

Figure 2-4: Tibia with press fit bearings and foot (Left) and section view showing I-beam cross section (Right).

in place and forming a rigid upper link. Like the tibia, the femur experiences predominantly bending loads, but it also experiences torsional loads from any transverse forces on the foot caused by the circular path on the boom. To maximise stiffness in these loading cases the two halves are designed to form a closed box cross section. Each of the halves has a wall thickness of only 1.5 mm keeping the weight of a single half to 122 g. Again, anticipated stresses are well below the material yield limit (see Appendix A) with wall thicknesses governed by machining time and difficulty for the thin features. Side walls extending from the closed cross-section further support the predominant bending loads and also enclose the piston, forming the upper link and piston into a single unit. Ribs on either side support these extended thin walls. The holes all have bosses to keep the two halves aligned, and to transfer loads ensuring the fasteners are not loaded in shear as shown in Figure 2-5 The steel knee shaft

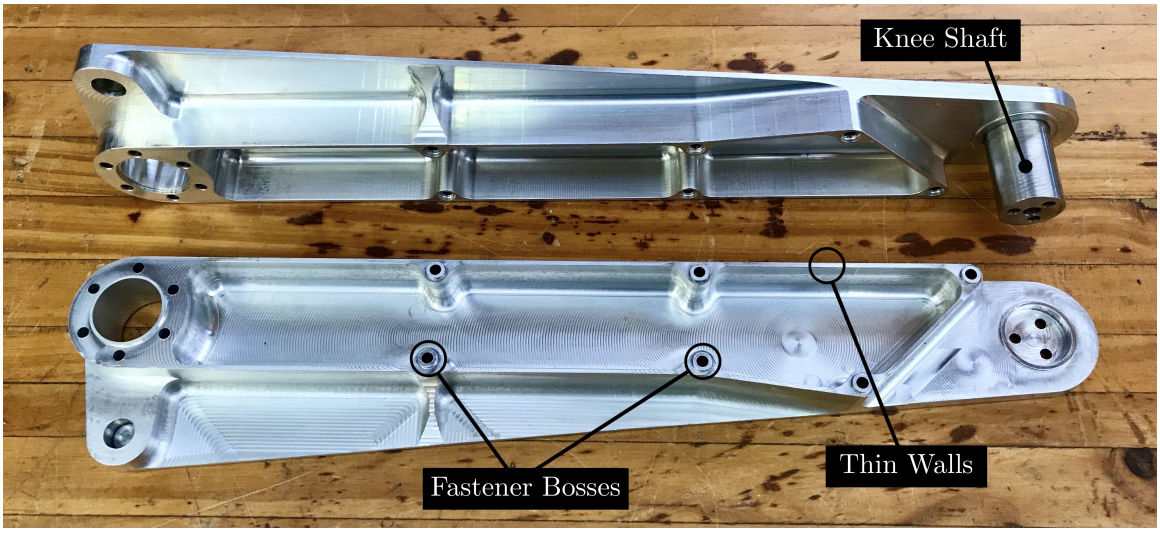

Figure 2-5: Separated femur halves with inserted steel knee shaft showing internal thin walled box shape and fastener bosses.

inserts through one half of the femur and is fastened into a recess in the other femur half shown in Figure 2-6. This clamps the two halves together onto the inner race

of the tibia bearings in a back-to-back arrangement, eliminating any axial play and dramatically increasing the rigidity of the whole knee joint. The two femur halves also

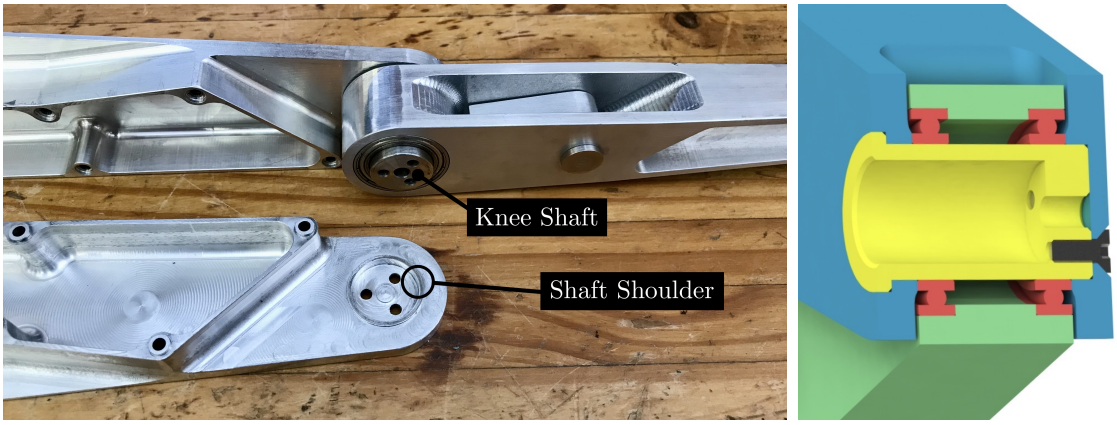

Figure 2-6: Exposed knee joint (Left) and cross section (Right) showing the bearings in red, tibia in green, femur in blue, and hollow steel knee shaft in yellow.

clamp around a steel dowel pin on which the polymer swivel bearing at the base of the piston can rotate. The dowel pin holes are tolleranced to ensure they will provide additional locating and structural support and even load transfer from the piston to both halves. Axial clearance of 0.5 mm is provided at the piston mounting pins so the piston can self-align, and that minor misalignment will not cause bending loads on the piston rod.

2.1.2 Body

The purpose of the body shown in Figure 2-7 is to provide a lightweight rigid mounting for the hip motors, the solenoid valves, the planarising support boom, and to house the wiring for the motors and solenoid valves. A secondary requirement was to keep the body simple and inexpensive to fabricate as it will be modified in future iterations. The body is constructed from two halves of 2 mm thick 6061 aluminium sheet which is laser cut and bent to form a closed rectangular section, providing rigidity in both bending and torsion. The body is 0.522 m long, which is just long enough to ensure that the knees cannot collide. The motors could also be offset so the legs can pass next to each other, allowing for a shorter body, but this would decrease the rotational inertia resulting in faster oscillations during bounding. The motors mount to the front side on either end, with the output shaft shoulder located and seated in a semicircular cutout. The solenoid valves are stacked and mounted on either end of the back side

shown in the bottom of Figure 2-7. Their mounting position ensures the minimum length of pneumatic tube is needed to limit dead volume, without restricting hip range of motion. If the hips extend too far past the body, this can result in the pneumatic swivel elbow connectors being pulled out. Simple 90° end-stops on the front of the body ensure the hips cannot rotate beyond this point when the motors are powered off or disabled. Fasteners through the body secure the robot to the pivoting end of the

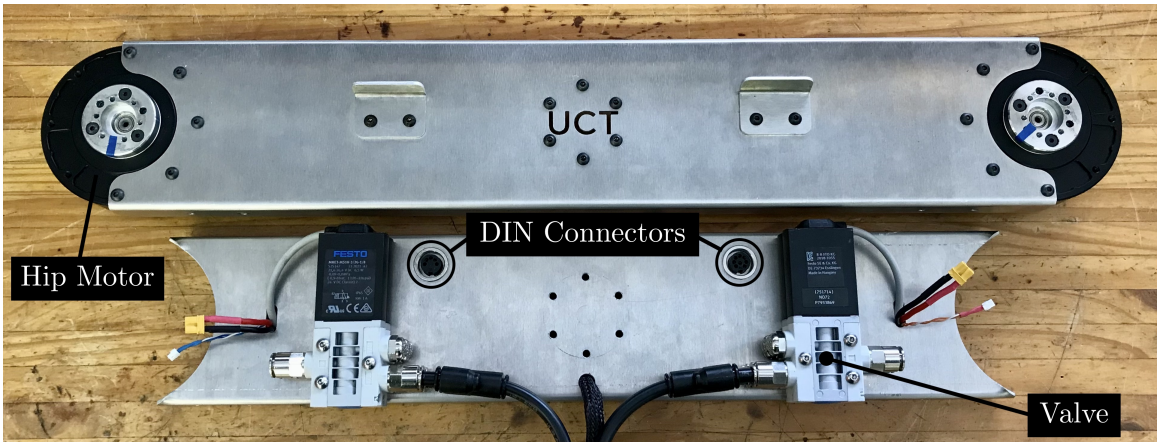

Figure 2-7: Sheet metal halves which make up the front (Top) and back (Bottom) of the body with mounted hip motors, solenoid valves, and signal DIN connectors.

planarising boom. An internal plastic insert inside the body distributes the force from these fasteners to the aluminium sheet metal and supports the two halves when the fasteners are tightened. The body also features female panel mount DIN connectors on either side of the boom mounting point for communication to the motors and solenoid valve signals. These needed to be removable connectors as the boom is used by other legged robots in the lab and so the robot must be easy to disconnect. Screw locking ensures that the connectors will not come loose even with the large shocks and vibrations experienced during bounding.

2.1.3 Pneumatics

The pneumatic system is powered by an air compressor which produces a nominal pressure of about 0.7 ± 0.1 MPa. Four high speed Festo MH3 3/2 200 L/min solenoid valves control the supply of compressed air to the two knee pistons. Two stacked 3/2 solenoid valves were used for each piston as opposed to a single 5/2 valve as this enabled independent control of pressure in each piston chamber providing a total of four discrete actuation states. Crucially this configuration enables the unactuated

or passive state for the robot where both chambers are at atmospheric pressure and not producing any force. The valves also default to this state which serves as a useful safety feature. With a single 5/2 solenoid valve the piston is limited to full force extension or full force retraction without the option for an unactuated/passive state. Proportional valves were not considered for controlling the pistons with bang-bang actuation being the desired operating regime. Proportional valves are also significantly heavier and more bulky. Mesh mufflers shown in Figure 2-8 are used on the venting ports to reduce noise and more importantly to stop fine debris from entering the valve. The length of pipe from the solenoid valve to the piston chamber

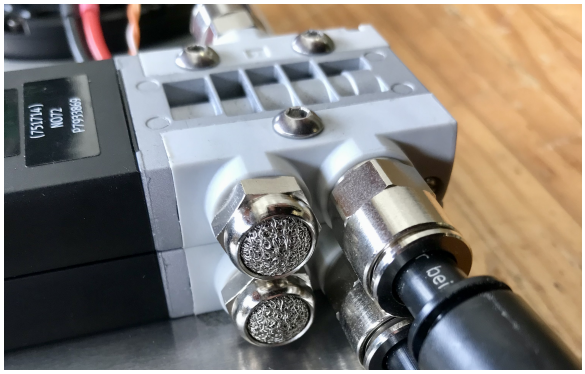

Figure 2-8: Stacked solenoid valves with vent port mufflers.

was kept to a minimum without restricting the hip range of motion. A longer pipe effectively makes the piston chamber volume appear larger, requiring more air per compression and taking longer to fill. Swivel elbow connectors are used to connect the pneumatic tubes to the knee pistons to ensure the tubes are free to move as the leg moves.

2.1.4 Motors

The actuators used in the MIT Mini Cheetah combine a low KV brushless drone motor with a 6:1 planetary gear reduction. This provides a reasonably high output torque without compromising backdrivability. Additionally, the relative simplicity of the design, use of off-the-shelf components, and impressive performance characteristics has resulted in a number of manufactures attempting to produce similar or even identical versions of the actuator. The AK70-10 motor from T-Motor was initially selected for its very similar performance characteristics and higher maximum torque

capability of 24 N m. To produce this higher torque, a slightly larger 10:1 planetary reduction was used. The gearbox of the AK70-10 unfortunately failed after only a few physical tests. Rather than the sun gear being press-fit into the aluminium rotor like in the MIT actuator, torque is instead transferred with a slightly unconventional dowel pin slot arrangement shown in Figure 2-9. After several early jump manoeuvres, this significant stress concentration resulted in a tearing forming in one of the motors. A similar failure mode was apparent in the second motor where a hairline crack had begun to form in the same place. To mitigate this issue a collar was machined from 304 stainless steel and press-fit over the aluminium rotor shaft to provide additional support and prevent it opening up further. This modification was also done preemptively to three other motors in the lab and none of them have shown the same failure. The second failure experienced with the T-Motor AK70-10 gearbox was

Figure 2-9: AK70-10 sun gear failure showing torn open pin slot (Left) and hairline crack developing in the same position on the second motor (Middle). Press fit steel collar to reinforce pin slot (Right).

the hardened steel planet and sun gear teeth shearing off. This was initially thought to be due to incorrect sizing of the gearbox for impact loads, a crucial consideration for proprioceptive actuators [8]. However, these were only partial breaks towards the ends of the teeth, suggesting poor tooth engagement along the width. Figure 2-10 shows that the planet gears do not ride on bearings and simply rotate on steel dowel pins with nylon spacers. The clearance needed for smooth rotation results in a visible degree of axial play, and is most likely the cause of the uneven tooth engagement and failure. This was a fundamental limitation of the actuator's design, and could not be repaired. The motor was instead replaced with the GIM8115-6 motor from Steadywin which has since proven significantly more reliable and robust. This motor

Figure 2-10: AK70-10 planetary gearbox with front cover removed (Left), and exposed planet gears on carrier with their steel dowel pins, nylon spacers, and broken gear teeth circled in red (Right).

is more closely based on the MIT actuator design with an almost identical planetary 6:1 gearbox. Needle roller bearings are press fit into the planet gears, and these ride on hardened dowel pins which should ensure a much more even engagement along the tooth width. The 3-phase driver also runs the same open-source firmware as the MIT actuator, simplifying debugging and opening up the potential for future customisation. The GIM8115-6 uses a brushless motor with a stack length of 19 mm shown in the middle of Figure 2-11 which is more than double the 8.2 mm stack length of the MIT actuator. This results in higher rotor inertia, higher mass, and 8 mm width increase, but significantly increases the maximum output torque from 17 N m to 25 N m without requiring a higher gear reduction and compromising impact resistance. The challenges and failures faced with the AK70-10 motor show that while

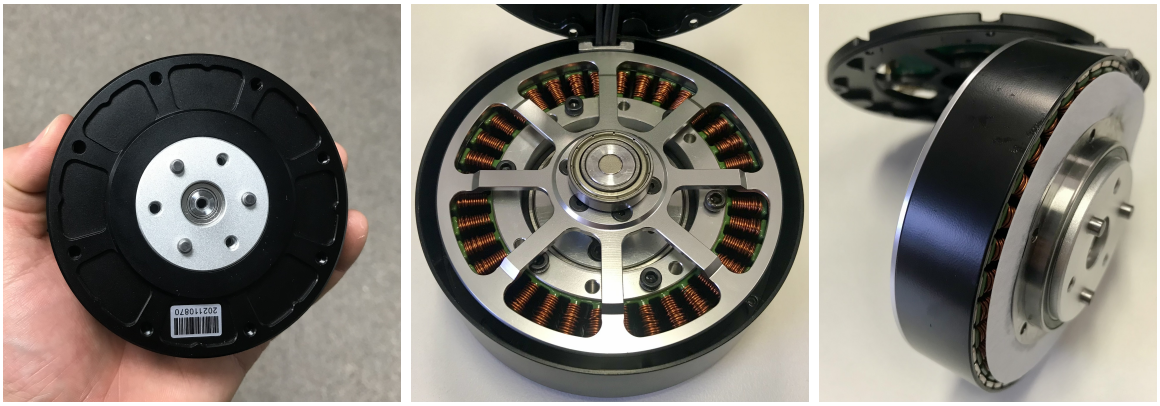

Figure 2-11: GIM8115-6 actuator (Left) with casing removed to show the back of the rotor (Middle) and it's large 19 mm stack length (Right).

these are very capable actuators, they are also incredibly complex electro-mechanical assemblies with multiple points of failure. This makes a good case for the mechanical simplicity and robustness of pneumatic actuators. The parameters of the three motors are summarised in Table 2.3.

Parameter	MiniCheetah Drive	T-Motor AK70-10	Steadwin GIM8115-6
Mass	480 g	620 g	705 g
Gear Ratio	6:1	10:1	6:1
Maximum Torque	17 N m	24 N m	25 N m
Maximum Speed	40 rad/s	35 rad/s	40 rad/s
Output Inertia	$0.0023\,\mathrm{kg\,m^2}$	$0.0091\,\mathrm{kg\,m^2}$	$0.0043\,\mathrm{kg\,m^2}$
Dimensions	⌀96 mm × 40 mm	⌀89 mm × 50 mm	⌀96 mm × 48 mm

Table 2.3: Hip actuator parameters.

2.1.5 Feet

The foot design is based off of that on the Mini Cheetah, comprising of a modified squash ball filled with an expanding urethane foam (Smooth-On FlexFoam-It 25) and a 3D printed insert which press-fits over the end of the tibia. A section view of this foot is shown in Figure 2-12. The squash ball provides a very durable and high grip surface which can interact with the smooth lab floor, while the high density urethane foam serves to dampen impacts and reduce the chance of the foot bouncing in and out of contact. Due to the force of foot impacts the flexible casting foam would partially disintegrate around the contact patch. This could possibly be mitigated by filling the foot with a slightly higher density foam.

2.2 Planarising Boom

The purpose of the planarising boom is to constrain the motion of the robot to an approximate plane whilst keeping the mass and inertia added to the robot to a minimum. It also provides the position, velocity, and acceleration of the robot for control. In addition to Kemba, there are other legged robots in the lab which need to use this planar support boom. It was therefore designed to also interface and work

Figure 2-12: Squash ball foot cut in half exposing functional layers and 3D printed insert. Holes in the insert provide a place for excess foam to expand.

with these robots.

A common configuration in legged robotics labs is to have the robot running in place on a treadmill and constrained by linear rails. This imposes the least additional mass on the robot as the linkages between the robot and linear rails can be made very short and light. However, this configuration is not realistic for investigating manoeuvres involving large accelerations as the robot does not need to accelerate its mass relative to the inertial frame. A much more realistic configuration would be to have the robot move over the stationary lab floor. However, if the robot moves in a straight line for the planar approximation, it will very quickly collide with a wall in the Lab. We therefore chose to have the robot run in a large circle, constrained and supported by a long boom arm. The larger this circle, the better the planar approximation. Aside from the benefit for acceleration manoeuvres, the boom arm configuration allows for longer distance runs and also provides a means to route power and data cables to the robot. A downside is that for bigger robots, the boom arm needs to be relatively long to provide a reasonable planar approximation, resulting in it adding significant mass.

Two boom arms have previously been built in the lab. The first used a parallelogram arrangement with two 2.2 m and 30 mm OD aluminium tubes and weighed 1.5 kg [39]. The parallelogram configuration ensured the end remained vertical irrespective of the boom angle. While relatively light, this design suffered from very poor torsional rigidity. For some of the single legged hopping robots we often need the rotation axis about the end to be fixed, limiting motion to only the vertical and

horizontal axes. This requires sufficient torsional rigidity to keep the robot upright. The latest design shown in Figure 2-13 used a single 2.1 m long 25 mm aluminium square tube with tensioned steel cables as stays to increase rigidity whilst keeping the mass under 2.5 kg [40]. While the tensioned cables did well to support lateral loads, they provided very little resistance to torsional loads resulting in similar poor torsional rigidity. The position of the end of the boom was estimated from the angle of the boom arm with two encoders at the base. Due to the long span the thin aluminium arm would deflect and vibrate with accelerations at the end. Small angle errors at the base produced very large position errors at the robot end thus making state estimation and control difficult. Vibration of the thin tensile members at their natural frequency also made velocity estimation from the encoders challenging.

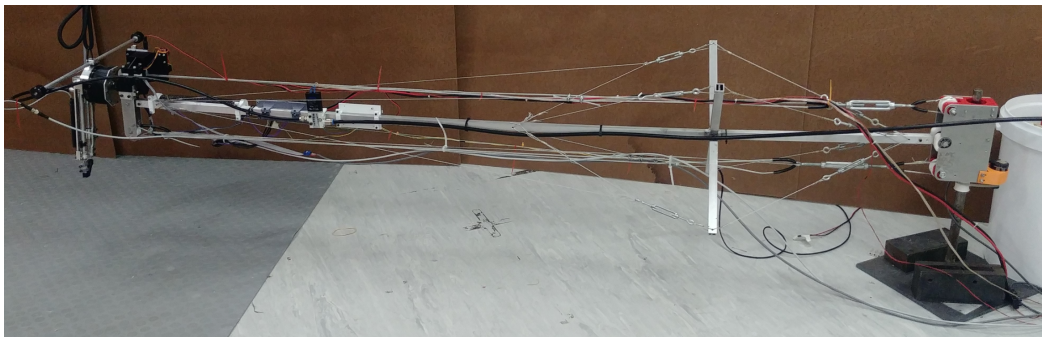

Figure 2-13: Previous boom design with square aluminium tubes and tensioned cable stays [40].

For all of the planar robots in the lab, a planarising boom is a crucial component of the system. A well designed robot will be near impossible to control if the boom cannot provide accurate state data, or is continuously introducing unmodelled dynamics. To this end, a new planarising boom was designed which could be used for all of the legged robots in the lab. A rendering of the boom is shown in Figure 2-14. The boom is 2.5 m long, adds approximately 1.5 kg of effective mass to the robot, has a positioning sensing accuracy of 1 mm, and has proved to be an integral part of controlling four of the legged robots in the lab.

2.2.1 Boom Arm

To maximise both torsional and lateral rigidity, while remaining as light as possible, the boom arm is a large diameter thin walled tube, maximising the polar moment of

Figure 2-14: CAD rendering of the boom.

inertia. The arm is 2.5 m long and is constructed from 3 pieces of 60 mm outer diameter carbon fibre tube shown in Figure 2-15, each with a wall thickness of 1.5 mm. The tubes are bonded to aluminium joining inserts with a high strength structural epoxy (Spabond 340LV). A toleranced shoulder on the joining inserts was used to maintain the recommended 0.2 mm bond gap for the epoxy while maintaining concentricity. The carbon fibre tubes are a stock configuration, roll wrapped with a combination of 0°, ±45° and 90° fibre orientations and finished with a 2x2 twill weave on the outer surface, which should provide a good balance of both bending and torsional strength. To protect the numerous data cables and simplify routing, the boom arm was intentionally kept hollow throughout with adequate space at either ends for the cables to exit.

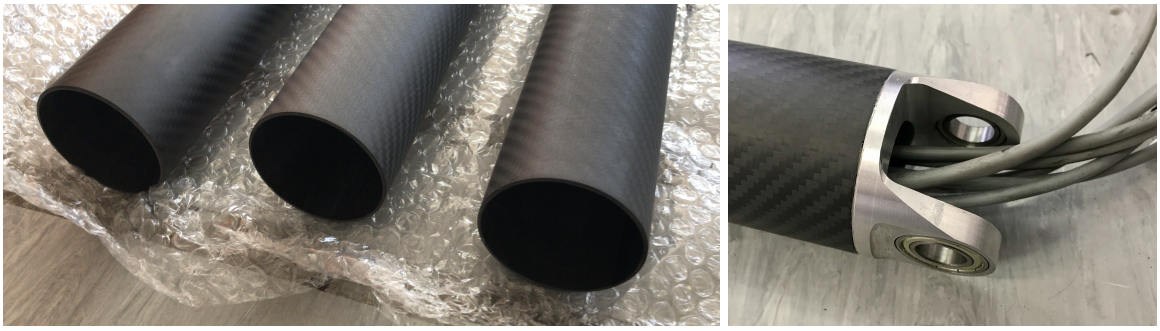

Figure 2-15: Carbon fibre tubes used in the boom arm (Left) and data cables routed through the boom (Right).

2.2.2 Mounting End

The mounting end of the boom provides an easy attachment point for robots and contains the final roll r degree of freedom which allows the body to pitch up and down about the end of the boom. Not all the robots in the lab require the third roll degree of freedom, so an additional fixed mounting end without this axis was designed which can easily be interchanged. Both of these boom mounting ends are

shown in Figure 2-16. Additional ends can also be designed if a robot has specific mounting requirements. For the roll enabled mounting end, the robot mounts to a flange on the roll axis shaft. The robot mounted to the end is kept vertical by two tensioned steel cables which run parallel to the boom arm between the mounting end and the centre pivot. A 500 pulses per revolution incremental optical encoder is

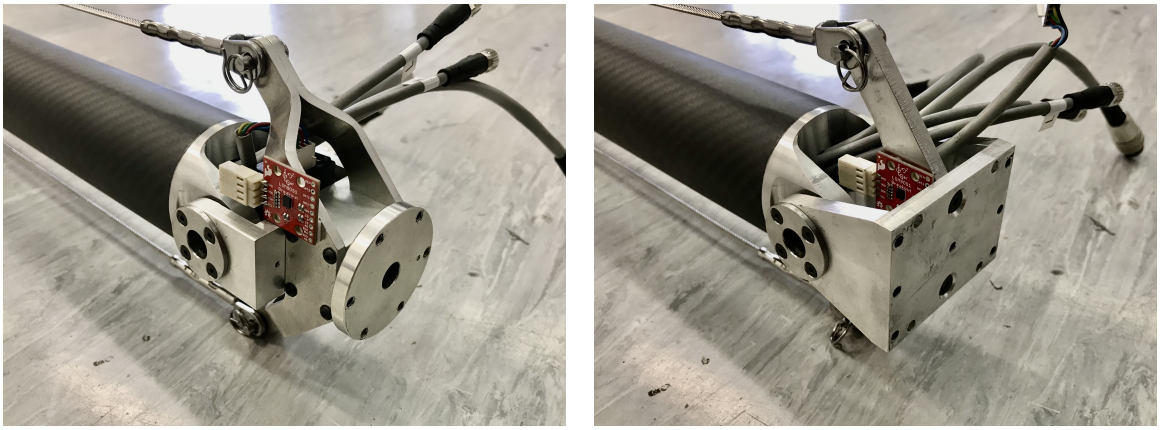

Figure 2-16: Interchangeable pivot (Left) and fixed (Right) boom end configurations each with their own accelerometer.

incorporated into the roll enabled end, to measure the angular position of the roll axis. Additionally, an accelerometer is also mounted to both boom ends to supplement the encoder measurements. The hollow stainless steel shafts for the elevation axis on the end insert from either side to preload the bearings in a back-to-back configuration and remove any axial play, as shown in Figure 2-17.

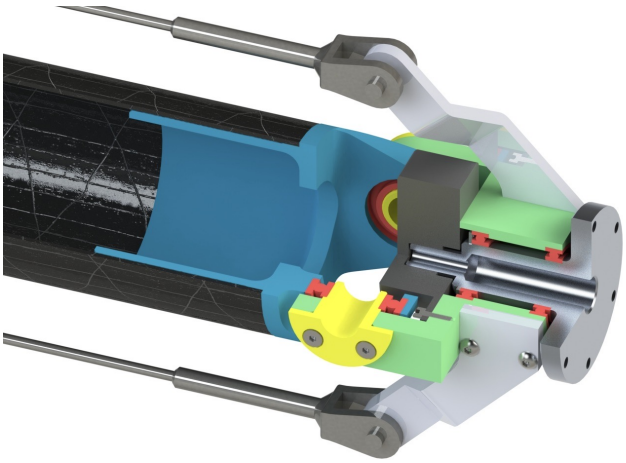

Figure 2-17: Section view of the elevation and roll axes on the boom mounting end. Bearings are shown in red and the hollow stainless steel shafts are shown in yellow.

2.2.3 Centre Pivot

The centre pivot of the boom shown in Figure 2-18 serves a stable base which the robot will run around, as well as the primary two degrees of freedom. These two degrees of freedom are the elevation axis θ and the azimuth angle ϕ in a spherical coordinate system. A flanged steel shaft serves as the azimuth axis. This is mounted to a steel base plate which will not move under the weight of the robot, but can still be positioned around the lab. Alternatively, this can also be mounted to anchor bolts in the floor for a more rigid and permanent fixture if the need arises. A 1024 pulses per revolution incremental quadrature encoder is mounted to each of the axes, geared with a 6 mm wide MXL profile timing belt for a 4:1 gear ratio to further increase angular resolution. This results in about 0.022° per encoder count, which translates to a position resolution at the end of approximately 0.9 mm. These encoders were selected for their availability in the lab. Higher resolution encoders were deemed unnecessary, as these sensors could resolve the 4 mm deformation in the rubber feet, which is more than sufficient for legged robot control. The stainless steel shafts for the elevation axis insert from either side of the yoke. This ensures that when tightened these two shafts preload the bearings and eliminating any axial play in the joint with the back-to-back bearing arrangement.

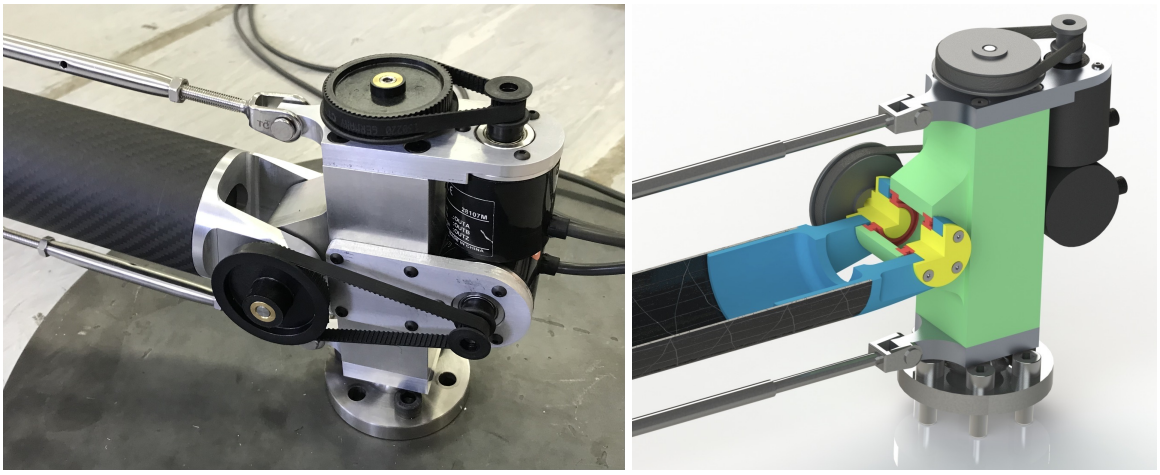

Figure 2-18: Centre pivot with encoders and 4:1 timing belt reduction (Left), and sectioned view (Right) of the elevation axis joint showing stainless steel shafts in yellow and bearings in red.

2.3 Electronics and State Estimation

The following sections describe the robot subsystems and their interfaces. Figure 2-19 provides a graphical representation.

2.3.1 System Architecture

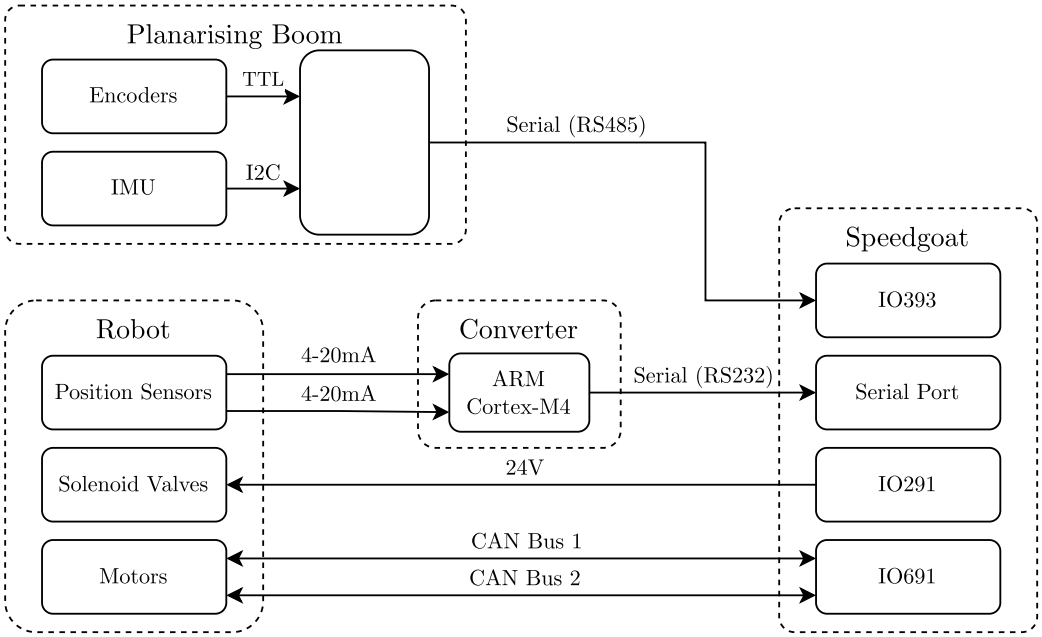

Figure 2-19: Diagram of the each of the subsystems making up the robot and their communication interfaces.

A Speedgoat baseline real-time target machine serves as the primary control computer for this robot. It has a variety of external interface modules which enable it to control the 24V solenoid valves and communicate with the boom and motors. It has a 2 GHz quad core Intel Celeron enabling it to run all of the controllers tested on the robot at 1 kHz on its Simulink Real-Time operating system.

2.3.2 Planarising Boom

While the boom and robot are integral to one another, the boom was designed as a separate subsystem to allow legged robots in the lab to be easily disconnected and interchanged. The boom receives robot state data from 3 encoders and an Inertial Measurement Unit (IMU). This sensor data is fused with a Kalman filter on an ARM Cortex M7 Teensy microcontroller to provide a full state estimate which is then sent

over a 1 Mbd RS485 serial interface to the Speedgoat real-time control computer at 1 kHz. The RS485 differential signalling helps improve noise immunity for the high communication rate as the cable between the boom and Speedgoat is reasonably long and can often run alongside the motor power cable on the floor. Information about the position of each of the three axes on the boom state is provided by three incremental encoders. As these are incremental encoders the boom will wait until the user has moved the boom past its index points on the elevation and roll axes before it begins sending state information, ensuring accurate absolute positioning. If the fixed mounting end (no roll axis) is being used and the roll encoder is disconnected, this is detected and the boom continues to function normally. This avoids a firmware flash each time a different boom configuration is used which would be extremely cumbersome and error prone. In addition to the three encoders, a 3-axis accelerometer mounted to the end also supplements the state estimation. Each time on startup, while the boom is stationary, the orientation of the accelerometer is checked and each of the axes are zeroed. The orientation check is necessary as with the tensioned cable arrangement, it is not guaranteed that the mounting end of the boom is perfectly vertical, so the gravity vector is used to identify the vertical axis of the world frame and compensated for this angular offset.

$$\phi = \text{atan2}(a_y, a_x) \tag{2.1}$$

Once the angle of the gravity vector in the x-y plane ϕ is known, the output coordinate frame of the accelerometer can be rotated to transform from the accelerometer frame to the world frame.

$$\mathbf{a}_w = \mathbf{R}_z(-\phi)\mathbf{a} \tag{2.2}$$

Where $\mathbf{R}_z$ is a counterclockwise rotation matrix about the z axis. The y-axis of the accelerometer now corresponds to the vertical movement of the boom mounting end, irrespective of the accelerometer placement or mounting end angle.

The accelerometer readings are combined with the encoder readings using a Kalman Filter [41] to provide more accurate estimates of state than if just the encoders were used. Each of the azimuth and elevation axes are orthogonal and therefore each runs

state estimation independently. To avoid the assumption that $\ddot{x}_{n+1} = \ddot{x}_n$ and resulting phase lag, the acceleration readings a were instead incorporated into the state extrapolation equation as a control input u rather than a measurement, producing the following state transition model:

$$\mathbf{x}_{n+1} = \mathbf{F}\mathbf{x}_n + \mathbf{G}u \tag{2.3}$$

$$\begin{bmatrix} x_{n+1} \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ dt \end{bmatrix} a \tag{2.4}$$

The acceleration variance σ_a^2 is discrete, and is included via the process noise matrix $\mathbf{Q}$. This is projected onto the dynamics model using the control matrix $\mathbf{G}$.

$$\mathbf{Q} = \mathbf{G}\sigma_a^2\mathbf{G}^\intercal = \sigma_a^2 \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix} \tag{2.5}$$

The acceleration variance σ_a^2 was simply calculated from 10 s of stationary accelerometer readings at 1 kHz shown in Figure 2-20 and found to be approximately $1.7 \times 10^{-4}\,\mathrm{m^2/s^4}$. The main form of uncertainty associated with the encoders is due their

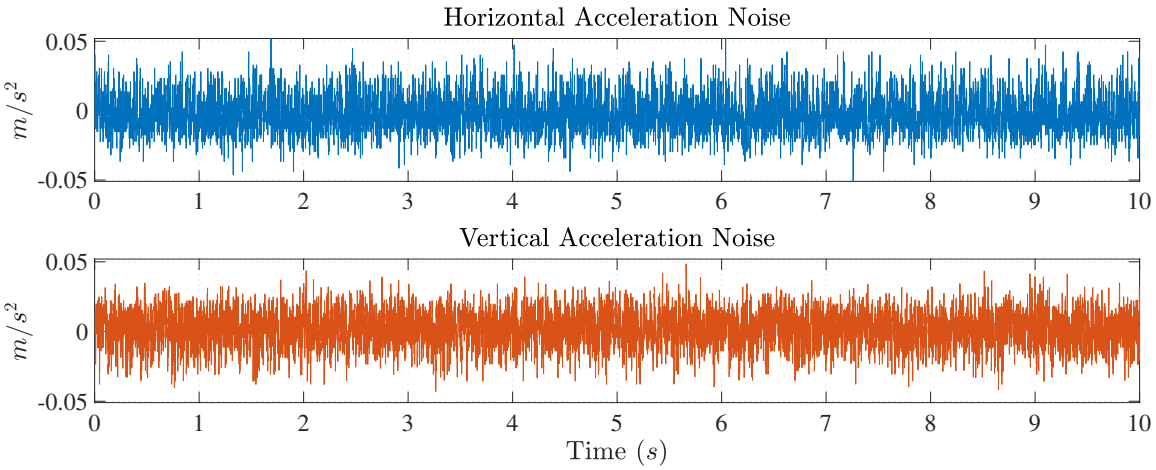

Figure 2-20: Accelerometer noise used to calculate the variance for the process noise matrix.

signal quantisation. At each edge the optical encoders provide a very accurate position measurement, apart from very small timing errors and manufacturing discrepancies. However, between edges it is not possible to tell whether the encoder is at the current

edge or the next edge. This position uncertainty can be reasonably approximated by a uniform probability distribution with a width of the encoder resolution. While this violates the requirement that sensor noise be normally distributed, it still provides a reasonable estimate of the sensor noise. Using this, the variance can be calculated from the encoder resolution δ_e.

$$\sigma_e^2 = \frac{\delta_e^2}{12} = 8.2 \times 10^{-8}\,\mathrm{m}^2 \tag{2.6}$$

This variance is included in the measurement noise matrix $\mathbf{R}$ for the Kalman filter. Combining both position and acceleration measurements greatly improves the accuracy of the velocity estimates shown in Figure 2-21. For the roll axis, the encoder is

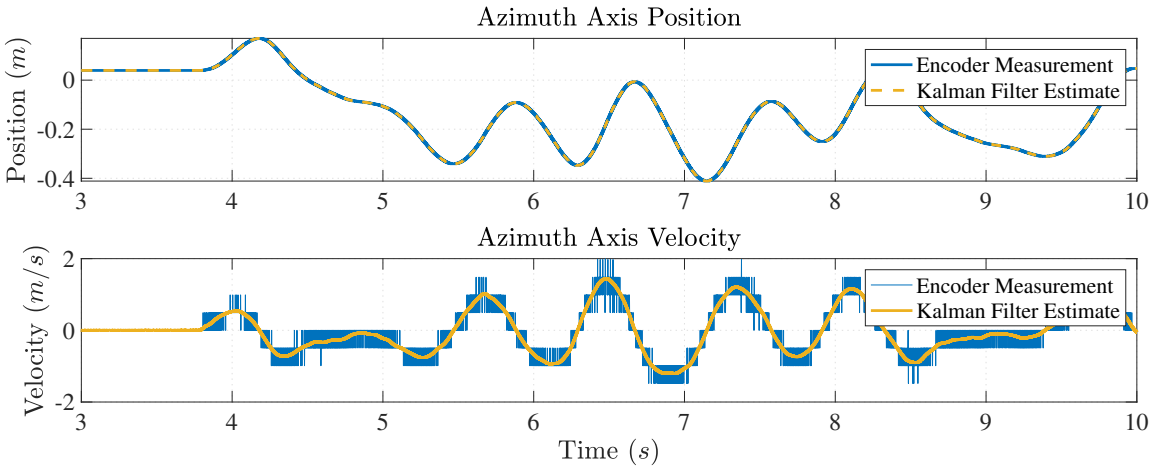

Figure 2-21: Boom state estimation performance for random motion along the azimuth axis. The Elevation axis performed similarly.

the only means for estimating velocity. Simply differentiating the encoder position measurements produces very noisy velocity estimates due to quantisation evident in Figure 2-22. A finite bandwidth PI feedback loop was used to track discrete encoder measurements and provide filtered estimates of angular position and velocity [42]. This is described in more detail in Appendix C. Tracking responsiveness and velocity estimate noise can be adjusted with the bandwidth parameter based on the encoder resolution and anticipated encoder speed. The estimator also provides a smooth extrapolation between discrete encoder steps which can be seen in Figure 2-22. The code for running these state estimators on the boom microcontroller can be found in Appendix E.

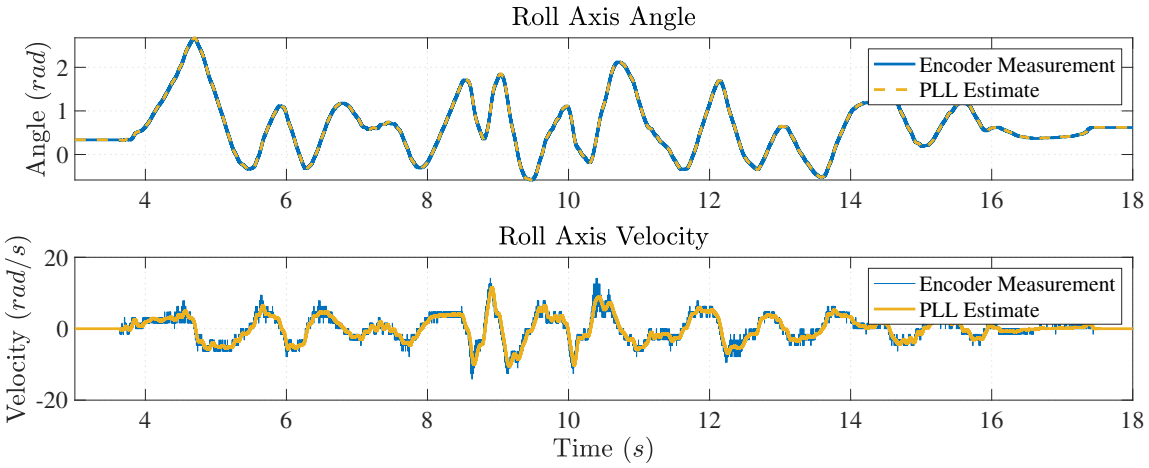

Figure 2-22: Boom encoder r axis position and velocity estimates.

2.3.3 Knee Position Sensors

The position of the knee pistons is measured with a Festo 80 mm SDAT-MHS Hall effect position sensor with a sensing resolution of approximately 0.05 mm. The position of the knee can be calculated from this measurement of the piston position using the leg's forward kinematics in Appendix B. The sensor outputs this position as a 4-20 mA current signal. Figure 2-23 shows the circuit used to convert the 4-20 mA current signal from the Hall effect sensor into a voltage signal for the microcontroller's ADC pin. The microcontroller sends the position measurements of the two pistons over an RS232 serial interface to the Speedgoat at 1 kHz. Current to voltage con-

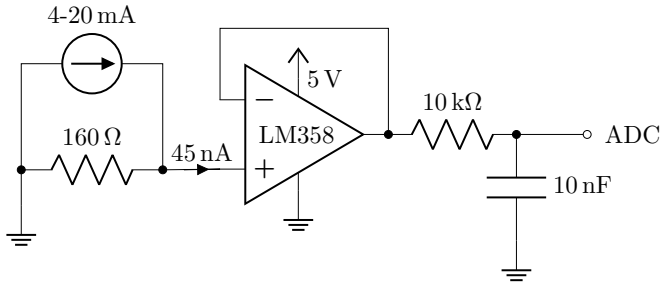

Figure 2-23: Knee position sensor current to voltage conversion circuit and ADC input buffer.

version is accomplished with a resistor, and a simple buffer amplifier circuit which protects the ADC pin from 24 V spikes on startup as the current loop stabilises.

2.3.4 Motors

The Speedgoat communicates with each of the two GIM8115-6 motors on the robot over separate CAN busses allowing simultaneous communication to both motors at 1 kHz. Each motor also runs an internal 40 kHz PD control loop with the structure shown in Figure 2-24. Position and velocity tracking errors are used to compute

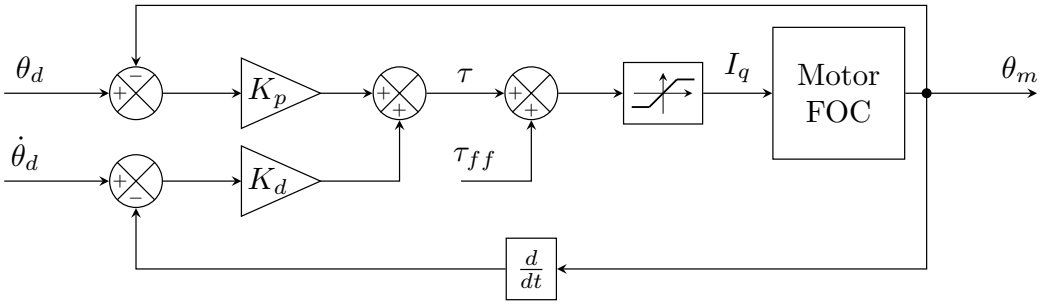

Figure 2-24: Motor 40 kHz internal PD control loop.

desired torques based on the K_p and K_d gain values. These produce behaviour equivalent to a rotary spring and damper with a higher K_p or K_d corresponding to a higher spring or damping constant. This desired torque τ can also be modified directly with a feed-forward torque command t_{ff} or used for pure torque control if K_p and K_d are both zero. The selection of the gain values was largely experimental. To track position and velocity commands as closely as possible it is desirable to have K_p and K_d set as high as possible, however setting them too high increases the gain crossover frequency and will cause the PD controller to go unstable. For a reasonable trade-off between stability and tracking performance K_p was typically set to 200 N m/rad and K_d to 4 N m s/rad.

The quadrature current I_q is calculated from the desired torque using the motor's torque constant, and this current is controlled in the stator's windings with Field-Oriented Control (FOC). For this control loop, the motor has inputs of desired position, desired velocity, position gain, velocity gain, and feed-forward torque command. Following a command, it responds with three state measurements: current position, current velocity, and estimated torque calculated from $K_T \cdot I_q$. Running a separate high-bandwidth internal controller ensures a closed-loop current and thus torque control bandwidth in excess of 1 kHz; far greater than could be achieved with only high level 1 kHz communication rates, enabling very fast torque responses.

The motors use a high resolution Hall effect digital encoders on the rotor. These are absolute sensors eliminating the need for indexing the hip at turn-on. However, as they are placed on the rotor before the planetary reduction, absolute position at turn-on of the output is only known to within 60°, so the legs are always started in the kneeling position to keep the hips within this range. More recent motor designs at the time of writing have improved on this absolute positioning limitation by adding a second encoder on the output of the planetary reduction.

2.3.5 Shunt Regulator

During braking the motors behave like generators and will back-feed the power supply, charging up the filtering capacitors on the output and raising the supply voltage. This typically triggers the over-voltage protection cutting power to the motors, causing the robot to collapse dramatically. To avoid this, a shunt regulator was built which will dump the energy from the motors into a 50 W 0.47 Ω power resistor. The schematic for this shunt regulator is shown in Figure 2-25. A microcontroller monitors the

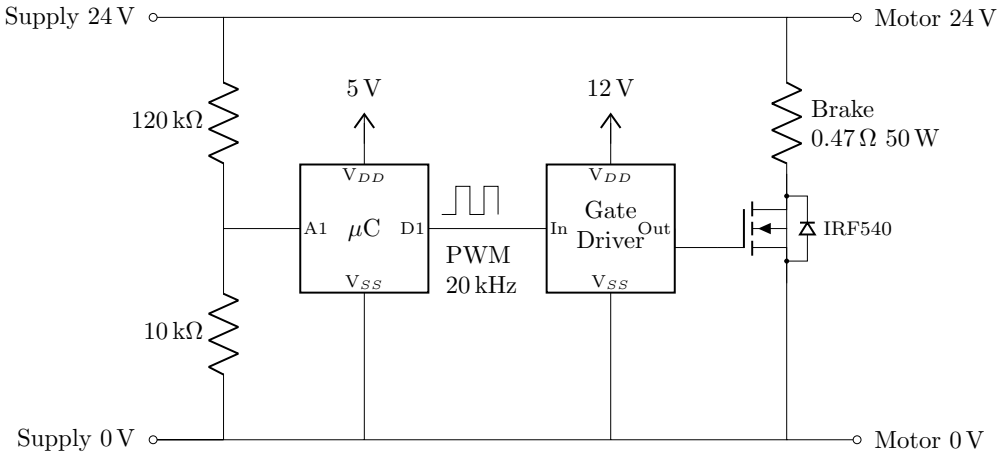

Figure 2-25: Simplified shunt regulator circuit.

supply voltage with a voltage divider and regulates the amount of power going into the brake resistor by varying the duty cycle of a 20 kHz PWM signal switching a MOSFET. The gate driver quickly charges the gate capacitance and ensures the MOSFET is turned on hard, minimising power dissipation during conduction. The PWM duty cycle remains at 0 % until the supply voltage exceeds 28 V, after which the duty cycle linearly ramps up to 100 % at 36 V. This linear ramp ensures the amount

of power dissipation is proportional to the rise in bus voltage, providing a smooth fade in or out as opposed to instantly dumping the maximum possible power, over 1.2 kW, through the brake resistor for even just small voltage spikes. It is difficult to gauge the amount of power dissipation the circuit should be rated for as this is largely dependent on the motions and the amount of braking required. However, this initial design has performed well, only getting slightly warm with extended runs. The code for the shunt regulator microcontroller can be found in Appendix E.

Chapter 3

Modelling and Trajectory Optimisation

This chapter details the modelling of Kemba and its actuators for use with trajectory optimisation. This approach to control incorporates the full body dynamics and also the actuator dynamics. Jumping, bounding, accelerating from rest, and a somersault motion are all explored to test the robot's limits, achievable motions and inform the subsequent bounding and accelerate controller design.

3.1 Trajectory Optimisation

Control of dynamic legged robots is an incredibly challenging task due to constraints imposed by the feet and frequent periods of limited control authority and permanent underactuation. These challenges are all exacerbated by the fact that for a motion such as accelerating from rest, it is still not clear at a high level how the robot should ultimately move. Trajectory optimisation is a useful tool which, in spite of these challenges, can be used to design physically feasible open-loop motions for this legged robot. Fundamentally trajectory optimisation involves transcribing the robot dynamics, control inputs, and constraints into a constrained boundary value problem

of the form:

$$\underset{\mathbf{x}[\cdot],\mathbf{u}[\cdot]}{\text{Minimize:}} \quad J(\mathbf{x}[n],\mathbf{u}[n]) \quad \forall n \in [1,N] \tag{3.1}$$

$$\text{subject to:} \quad \dot{\mathbf{x}}[n] = f(\mathbf{x}[n],\mathbf{u}[n]) \quad \text{(System dynamics)} \tag{3.2}$$

$$g(\mathbf{x}[n],\mathbf{u}[n]) \leq 0 \quad \text{(Inequality constraints)} \tag{3.3}$$

$$h(\mathbf{x}[n],\mathbf{u}[n]) = 0 \quad \text{(Equality constraints)} \tag{3.4}$$

$$\mathbf{x}[0] = \mathbf{x}_0 \quad \text{(Initial conditions)} \tag{3.5}$$

$$\mathbf{x}[N] = \mathbf{x}_N \quad \text{(Final conditions)} \tag{3.6}$$

This can then be solved by a gradient-based non-linear solver. There are multiple approaches and methods for formulating the trajectory optimisation problem as an NLP. The approach and methods used in this work are summarised in Table 3.1. The selection and justification of these components is discussed in the following sections.

Transcription Method	Direct Transcription
Integration Method	Implicit Euler
Timesteps	Variable ($\pm 20\,\%$)
Angle Formulation	Absolute [43]
Contact Model	Implicit Frictionless Impact [44]
Objective Function	Heat-based Cost & Penalty
Solver	IPOPT (MA86) [18]
Optimiser Tolerance	1×10^{-6}
Variable Initialisation	None (See Section 3.3)

Table 3.1: Summary of trajectory optimisation problem formulation and approach

3.1.1 Robot Dynamic Model

The model depicted in Figure 3-1 shows the planar dynamics model used in the trajectory optimisation problem formulation. This model is composed of nine rigid links, each with mass values measured from the actual robot and inertia properties estimated in SolidWorks. The body of the robot has three degrees of freedom, namely x_b and y_b translation, and θ_b pitch, keeping with the planar approximation enforced by the boom. The angles for each of the child links attached to the body are defined

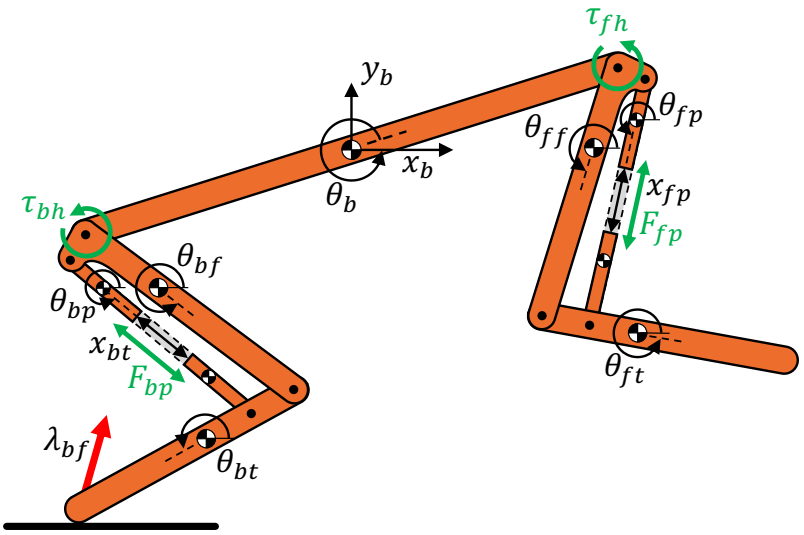

Figure 3-1: Robot rigid body arrangement with absolute angle formulation of the 11 generalised coordinates.

in absolute coordinates relative to the inertial frame, rather than the more common relative angle convention. This has the benefit of dramatically simplifying both the sparsity and the number of operations in the Coriolis matrix of the manipulator equation and thus improving solve times [43].

The Equations of Motion (EOM) for the nine links is derived using the method of Lagrange and the computer algebra system SymPy with the generalised coordinates shown in Figure 3-1. The system dynamics are enforced with the manipulator equation:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q}) = \mathbf{J_a}(\mathbf{q})\mathbf{u} + \mathbf{J_c}(\mathbf{q})\lambda \tag{3.7}$$

$\mathbf{M}$, $\mathbf{C}$ and $\mathbf{G}$ represent the inertial, Coriolis and gravitational matrices respectively, while and $\mathbf{J_a}$ and $\mathbf{J_c}$ are the control and constraint Jacobians. The generalised coordinates $\mathbf{q}$ are given by:

$$\mathbf{q} = \begin{bmatrix} x_b & y_b & \theta_b & \theta_{ft} & \theta_{ff} & \theta_{bf} & \theta_{bt} & \theta_{fp} & x_{fp} & \theta_{bp} & x_{bp} \end{bmatrix} \tag{3.8}$$

These symbols correspond to those shown in Figure 3-1.

The pistons make up a significant portion of the mass of the leg so the rigid body dynamics of the rod and piston body are also included in the leg arrangement. This closed-kinematic chain formed by the piston is no longer a minimal coordinate parameterisation of the robot, so the dynamics were first derived using an uncon-

strained configuration with the position constraint later imposed during optimisation and enforced via constraint forces.

The control inputs $\mathbf{u}$ are the two torques at the hips τ_{bh} and τ_{fh}, and the forces F_{fp} and F_{bp}, which correspond directly to the torques and forces produced by the motors and pistons respectively on the physical robot. The foot ground reaction and friction forces are described by λ_{bf} and λ_{ff}.

Not shown in the rigid body diagram, but also included in the modelling of the dynamics was the inertial effect of the boom. The boom arm was simply modelled as a rigid link with state defined as a function of the body position and velocity, and incorporated into the dynamics with additional potential and kinetic energy terms.

3.1.2 Contact Model

Foot-ground interactions are modelled as unilateral contact constraints using the complementarity approach from [44]. This eliminates the need for a prespecified contact schedule, allowing the solver to essentially decide when to make or break contact with the ground. A degree of timestep variability ensures that these contact events occur at transcription nodes. The actual feet on the robot are rubbery and will not behave as an inelastic collision. Small amounts of wear or dirt on the contact patch of the foot can also result in very different friction coefficients between runs. It was therefore undesirable for the solver to find motions which exploited the boundary of the friction cone and required a precise slipping behaviour. Underestimating the friction coefficient and enforcing a no-slip constraint kept the resulting trajectories away from the boundaries of the friction cone, and made them significantly more applicable to the real hardware. In addition to providing the knee actuation, the piston also serves as the range of motion limits. These hard end stops share a number of similarities with foot contacts and so are implemented in a similar way to the unilateral contact constraints. All of the discontinuous switching contact terms are formulated into complementarity equations enforced using the following constraint:

$$\alpha \cdot \beta \leq P \tag{3.9}$$

To simplify the numerical difficulties associated with complementarity constraints we relax the inequality with a penalty slack variable P which is minimised to zero in the cost function, similar to [45]. The foot contact interactions are therefore defined by the following minimisation:

$$\text{Minimize:} \quad \mathbf{P_s} + \mathbf{P_c} \tag{3.10}$$

$$\text{subject to:} \quad \lambda_{\mathbf{y}}[n] \cdot \mathbf{y_f}[n+1] \leq \mathbf{P_c}[n] \quad \text{(GRF)} \tag{3.11}$$

$$\lambda_{\mathbf{y}}[n] \cdot \dot{\mathbf{x}}_f \leq \mathbf{P_s}[n] \quad \text{(No-slip)} \tag{3.12}$$

$$\lambda_{\mathbf{x}}[n] \leq \mu \lambda_{\mathbf{y}}[n] \quad \text{(Friction cone)} \tag{3.13}$$

Where $\lambda_{\mathbf{x}}$ and $\lambda_{\mathbf{y}}$ are the foot horizontal and vertical Ground Reaction Force (GRF), μ is the friction coefficient, $\mathbf{y_f}$ is the feet height above the ground, $\dot{\mathbf{x}}_\mathbf{f}$ is the feet horizontal velocity, and $\mathbf{P_s}$ and $\mathbf{P_c}$ are the complementarity penalty terms.

3.1.3 Actuator Models

The two actuators on the model are the motors at the hips and the pistons at the knees. The current, and thus torque, controller in the motor has a bandwidth of approximately 1 kHz. This is almost an order of magnitude faster than the timesteps used to discretise the continuous trajectories, which are typically in the range of 10 ms. At these timescales, delays associated with producing a commanded torque will be negligible and the current controller dynamics can be ignored. A more significant consideration is the maximum rated torque for the motor, and the reduction in torque resulting from back-EMF generated at high velocities. Together these two effects limit the maximum achievable torque and velocity of the actuator, constraining its operating region to the shaded area of Figure 3-2. This is enforced with the following constraints in the optimisation formulation.

$$\tau = \begin{cases} min(\tau_0 - k\dot{\theta}, \tau_{max}), & \text{if } \dot{\theta} > 0 \\ max(-\tau_0 - k\dot{\theta}, -\tau_{max}), & \text{if } \dot{\theta} \leq 0 \end{cases} \tag{3.14}$$

Another motor behaviour requiring modelling is the effects of the rotor inertia on

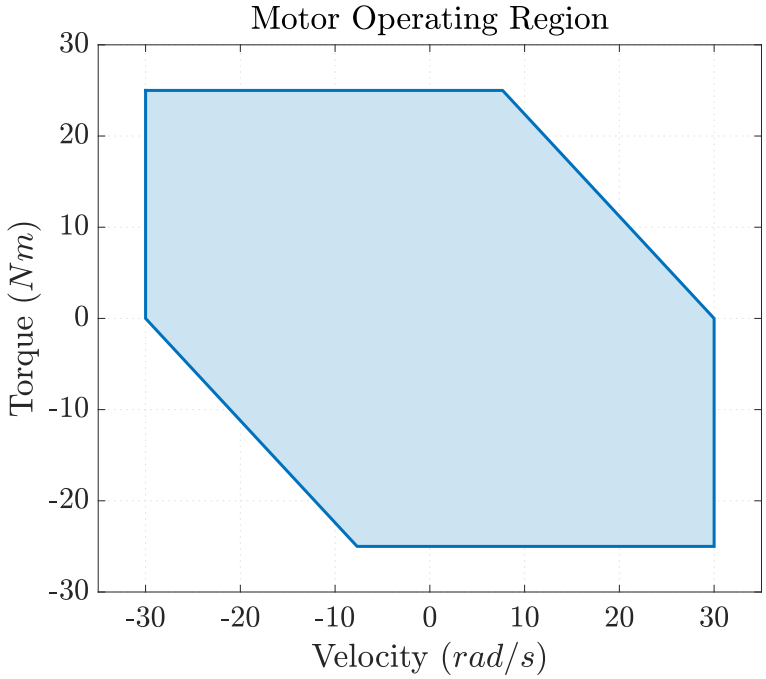

Figure 3-2: Motor torque and velocity operating region.

the system dynamics. In a planar model the rotor has two predominant effects:

Reaction torque The rotors produce a reaction torque on the link they are mounted to when they accelerate.

Reflected inertia The inertia of the rotor is also felt through the output shaft by the link the rotor is geared to. Even though the rotor is designed to be lightweight with a low rotational inertia, because of the gear ratio, the inertia felt at the output shaft is equal to the rotor inertia multiplied by the gear ratio squared, which ends up being a relatively large inertia term.

Ignoring these effects would result in trajectories where the actuator accelerates faster than achievable in reality. The effect of the rotors is included in the body dynamics with an additional kinetic energy term for each rotor.

$$T_{rot} = \frac{1}{2} I_r \dot{\theta}_r^2 = \frac{1}{2} I_r (\dot{\theta}_i + N(\dot{\theta}_{i+1} - \dot{\theta}_i))^2 \tag{3.15}$$

The angular velocity of the rotor in the inertial frame $\dot{\theta}_r$ is composed of the angular velocity of the base link $\dot{\theta}_i$ and the relative link velocity $(\dot{\theta}_{i+1} - \dot{\theta}_i)$ multiplied by the gear ratio N. Potential energy from the rotor is already accounted for in the total motor's mass, and due to the planar simplification, Coriolis torques from what is

effectively a reaction wheel can also be ignored. Similar to the motors, the pistons are by no means an ideal force producing element. Air is characterised by relatively slow dynamics and is also compressible, and the valves have flow restricting elements which can all have quite a complex effect on the force produced by the actuators. Section 3.2 deals with developing a model for these double acting pneumatic actuators which can be incorporated into the trajectory optimisation formulation.

3.1.4 Cost Function

A cost function is used to find the optimal solution within the feasible region based on a single metric. For example, to force a motion to be fast the time taken can be minimised by including a time duration value in the cost function. This however has a number of caveats, with multiple local minima for non-convex problems and term scaling for compound cost functions.

To keep the cost function simple and explicit, and avoid complex weighting of multiple terms, it is comprised of only the complementarity penalty terms, and a secondary heat-based cost to regularise motor torques and reduce chattering [33]. The complementarity penalty term is scaled ρ times higher to prioritise it over the motor's heat-based cost as shown in the following equation:

$$J = \sum_{i=1}^{N} \boldsymbol{\tau}_i^T \boldsymbol{\tau}_i + \rho \boldsymbol{P}_i \tag{3.16}$$

Where $\boldsymbol{\tau}_i$ and $\boldsymbol{P}_i$ are the hip torques and complementarity penalty terms at the i^{th} node. The weight ρ is set to 1×10^4 to prioritise penalty minimisation over torque regularisation. To minimise an aspect such a time for a motion, rather than including it in the cost function, it is enforced as a constraint. This constraint is then iteratively reduced until it cannot be reduced any further due to the problem becoming infeasible [33]. Without rigorous investigation, this approach was anecdotally found to produce more consistent solutions compared to leaving the problem less constrained and including the variable in the cost function.

3.2 Pneumatic Actuator Characterisation

This section details the development of a tractable model of a generic double-acting pneumatic actuator through simplification of an established physics-based model described in [13] combined with characterisation experiments. This is necessary as while the physics-based model fully describes the air dynamics, the equations are discontinuous and numerically stiff making integration into high level motion optimisation impossible. This model therefore aims to capture the dominant convergent dynamics of the actuators, simplifying the model and making it tractable for motion optimisation.

3.2.1 Physics-based Model

While the physics-based model cannot be used directly in trajectory optimisation, it serves as an essential guide to the development of the simplified model for optimisation. A number of the governing equations are described to provide context, however, for a complete background and description of the model and parameters see [13].

At its simplest, a double acting pneumatic cylinder is a device with two opposing air chambers separated by a sliding bore. Differences in pressure in the two chambers produces a resultant force on the sliding bore and connected piston rod. By controlling the pressure in each of these chambers the output force of the actuator can be determined. There are two mechanisms by which the pressure in a chamber can change: due to air exiting or entering the chamber, and due to the movement of the bore which changes the volume and therefore pressure of the air in the chamber. These two mechanisms are described by the following differential equation:

$$\dot{P} = \frac{\kappa R T}{V}\dot{m} + \frac{\kappa P}{V}\dot{V} \tag{3.17}$$

The first term describes the pressure change resulting from mass flow of air $\dot{m}$ into or out of the chamber, and the second term encapsulates the pressure change resulting from a change in volume $\dot{V}$ caused by movement of the bore. Air flows into the chamber through a small orifice in the solenoid valve. The flow of air is therefore governed by the valve command $u \in \{0, 1\}$, the cross-sectional area of the orifice a,

and the thin-plate flow function $\phi()$.

$$\dot{m} = \begin{cases} u \cdot a \cdot \phi(P_u, P_d) & \text{if } P_u \geq P_d \quad (3.18a) \\ -u \cdot a \cdot \phi(P_d, P_u) & \text{if } P_u < P_d \quad (3.18b) \end{cases}$$

The flow direction is determined by the upstream P_u and downstream P_d pressure difference. Due to the small orifice and Venturi effect, the flow of air very quickly reaches sonic velocity, transitioning to a choked flow regime. This means that the mass flow through the solenoid valve orifice and into or out of the chamber becomes independent of the downstream pressure. This flow transition is described by the following thin-plate flow function:

$$\phi(P_u, P_d) = \begin{cases} \alpha P_u \sqrt{(\frac{P_d}{P_u})^{\frac{2}{\kappa}} - (\frac{P_d}{P_u})^{\frac{\kappa+1}{\kappa}}} & P_u/P_d \leq \theta \quad (3.19a) \\ \beta P_u & P_u/P_d > \theta \quad (3.19b) \end{cases}$$

Where α, β, θ, and κ are air constants defined in [13]. The first term in (3.17) contains complex piecewise mass flow dynamics which require timesteps on the order of microseconds to integrate, far shorter than is feasible with high level trajectory optimisation optimisation. A single actuator also has two opposing chambers meaning (3.17) for one chamber is coupled to (3.17) for another chamber through the bore and chamber volumes, adding to the modelling complexity.

Equation (3.17) shows that by holding the bore and piston rod stationary, the changing volume term can be eliminated allowing the complex mass flow dynamics and resulting force response to be isolated and characterised independently. This insight is incredibly beneficial as with the piston stationary this test is essentially equivalent to a maximum torque test for a motor, simplifying the number of states for the system.

Through previous work in the lab with pneumatic actuators [37, 40], two key principles were identified which largely describe the more dominant behaviour exhibited by the pistons:

1. Output force does not respond instantly with a change in valve state

2. Steady-state output force decreases with an increased piston speed

The basis for these anecdotal observations is evident when looking at the physics-based model. Air flow into or out of a chamber is restricted by the small orifice in the valve producing a slower pressure and therefore force response for a change in valve state.

3.2.2 Force Response

When the state of a solenoid valve changes, air enters or leaves a chamber of the piston producing a particular pressure response and therefore force response on the piston rod. Air flow in the system is restricted and so this response is far from instantaneous. To characterise this force response, an Axia80-M20 force transducer was fixed to the rod of the piston and held stationary. The force transducer was sampled at 1 kHz and has a resonant frequency of 2.6 kHz along its axis, which are both significantly higher than the approximately 10 ms force response being measured. By keeping the piston bore stationary, the second term in (3.17) is eliminated and the pressure dynamics become a function of only the chamber volume V and air mass flow $\dot{m}$ through the solenoid valves, with κ, R, and T all being constants. The force produced by a single chamber is therefore a function of only the chamber volume and the valve state.

In total, five repeated test were performed at three supply pressures 0.6 MPa, 0.7 MPa, and 0.8 MPa for each of the two compression or venting valve states at the maximum and minimum chamber volumes. The recorded output force response from these 60 tests is shown in Figure 3-3 with the valve commands being set at $t = 0$.

Figure 3-3 shows that firstly the force response time is independent of the supply pressure with the three tested pressures having almost indistinguishable normalised force responses. As expected the response is dependent on chamber volume, with the maximum chamber volume taking longer to compress or vent and therefore showing a slower response. Additionally, venting also has a marginally slower response compared to compressing a chamber. All of these responses also exhibit identical 6 ms dead time. They can all be reasonably well approximated by fitting a first-order plus dead time model, shown by the dashed lines in Figure 3-3. While more complex approximations might provide a marginally better fit, this is unnecessary as at these small timescales trajectory optimisation will be a greater source of inaccuracy, with integration timesteps around 10 ms. The chamber force dynamics can therefore be

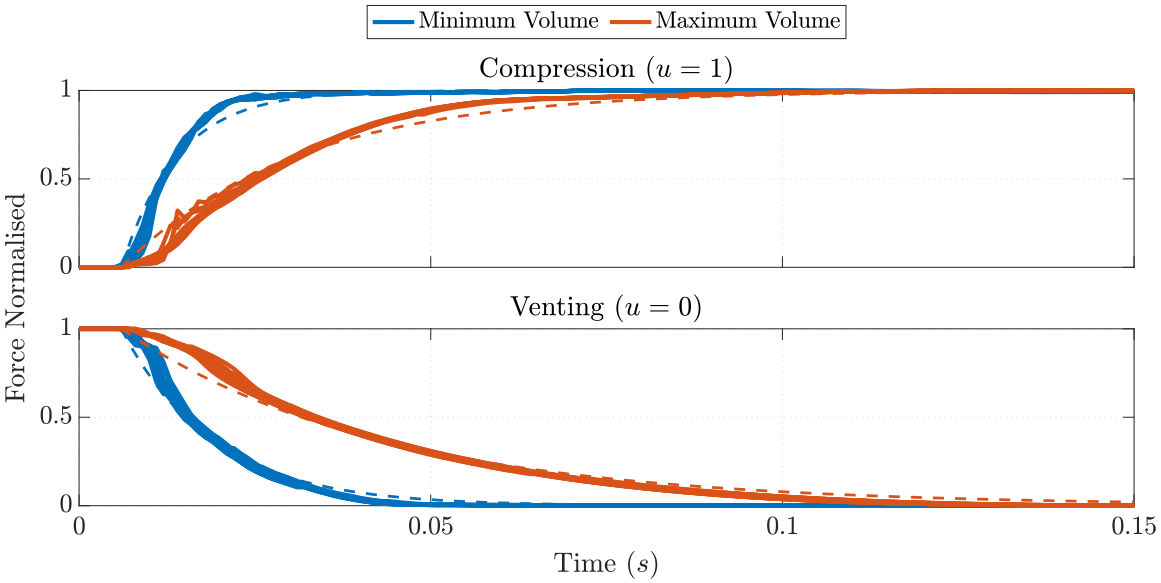

Figure 3-3: Chamber normalised force responses for compression and venting valve commands at the minimum and maximum chamber volumes. The dashed lines show the best fit first-order plus dead time approximation for these responses. Lines of different pressure are kept the same colour to simplify the plot as they are overlapping and almost indistinguishable.

approximated with the following differential equation:

$$F(t) = F_{static} \cdot u(t - t_0) - \tau(u, x)\dot{F}(t), \tag{3.20}$$

Where F_{static} is the maximum steady-state force produced by the chamber, controller by valve command $u \in \{0, 1\}$ delayed by dead time t_0, and with rise time τ. The rise times for the approximations shown in Figure 3-3 are summarised in Table 3.2.

Table 3.2: Chamber Force Rise Time.

Chamber Mode	**Rise Time (τ)**	
	$x = 0\,mm$	$x = 70\,mm$
Compression ($u = 1$)	8 ms	25 ms
Venting ($u = 0$)	13 ms	37 ms

While only the maximum and minimum chamber volume responses were recorded to limit the number of tests, Figure 3-4 shows that the rise time has an approximately linear relationship to chamber volume and therefore piston position based on the theoretical model. If the chamber volume doubles, it will take twice as long to fill

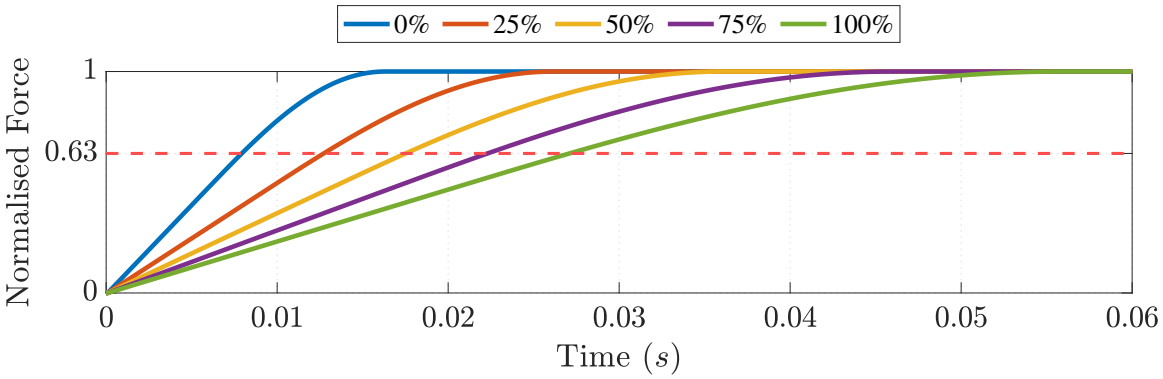

Figure 3-4: Chamber force response time for different chamber volumes showing linear relationship. The red dashed line shows the 63% point where rise time is estimated.

given that the mass flow of air spends the majority of the time in the choked flow regime where it is independent of downstream pressure. The rise time between these maximum and minimum chamber volumes can therefore be linearly interpolated to find the rise time for all chamber volumes and the pistons full range of motion.

While the rise time is linearly dependent on chamber volume, the valve commands u are discrete. To ensure the model remains continuous for gradient based methods we also linearly interpolate the rise time for the chamber modes with the discrete states being imposed during optimisation. The rise time can therefore be expressed as the following linear function:

$$\tau(u, x) = \tau_0 + u\tau_1 + x\tau_2 + ux\tau_3 \tag{3.21}$$

Where x is the extension of the actuator and the four time constants τ_0, τ_1, τ_2, τ_3, are fit directly from the measured data in Table 3.2 using a bilinear interpolation.

3.2.3 Force-Velocity Relationship

Another observation with the actuators was that steady-state output force is speed dependent, decreasing for higher speeds. This is most likely due to the second term in (3.17) which describes how a change in volume produces a proportional change in the chamber pressure. Additionally, both chambers are coupled via this term with an increase in volume of one chamber having a corresponding decrease in volume of the opposing chamber. Rather than considering each chamber in isolation, the combined effect of both chambers was considered for the whole double acting piston simplifying

the model and testing procedure. To characterise this behaviour the two knee pistons were arranged opposing each other with the Axia80-M20 force transducer mounted between their output rods as shown in Figure 3-5.

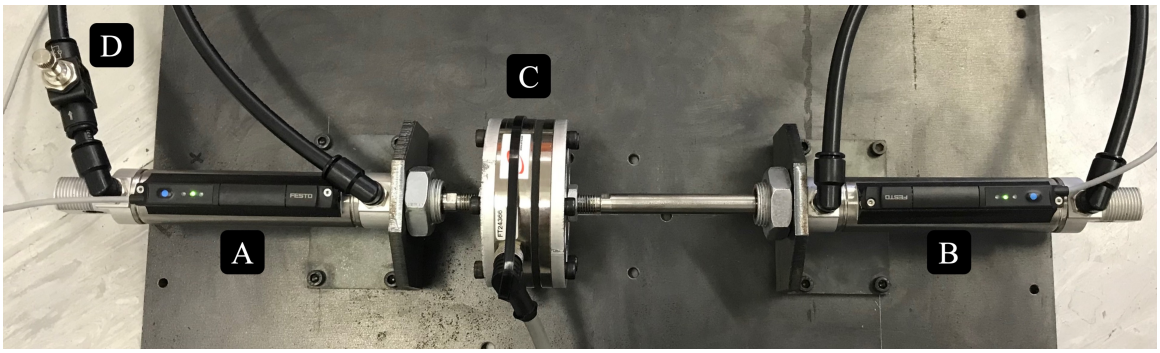

Figure 3-5: Piston characterisation setup showing the resistance piston (A), the piston being characterised (B), Axia80-M20 force transducer (C) and flow restricting valve (D).

This allows a full range of movement whilst simultaneously allowing the axial force between the output rods to be measured. Each piston also includes its Hall effect position sensor for measuring output rod position and velocity. The left piston (A) in the configuration shown in Figure 3-5 served as the variable resistance piston with a flow restricting valve to adjust the load on the piston being characterised (B). If the resistance load is decreased, the piston being characterised will reach a higher steady-state velocity. This configuration is analogous to a motor dynamometer arrangement with a torque transducer between a motor and variable load. Figure 3-6 shows the output force of the piston over its full velocity range for all of the four possible actuation states for 0.6 MPa, 0.7 MPa, and 0.8 MPa. Each chamber in the piston can also be compressed or vented independently resulting in four actuation states; extending, retracting, both compressed, unactuated. Each pressure and valve state data set is the combination of approximately 25 tests at different resistance levels to fully explore the achievable velocity range, resulting in a total of about 275 tests, each containing multiple force and velocity measurements.

These data show that the output force of the piston has an approximately linear inverse dependence on the output rod speed. As the speed of the rod increases, the force produced by the actuator drops proportionally. The linear fits for each of these test cases is summarised in Table 3.3 along with the R^2 value to show the strength of the fit.

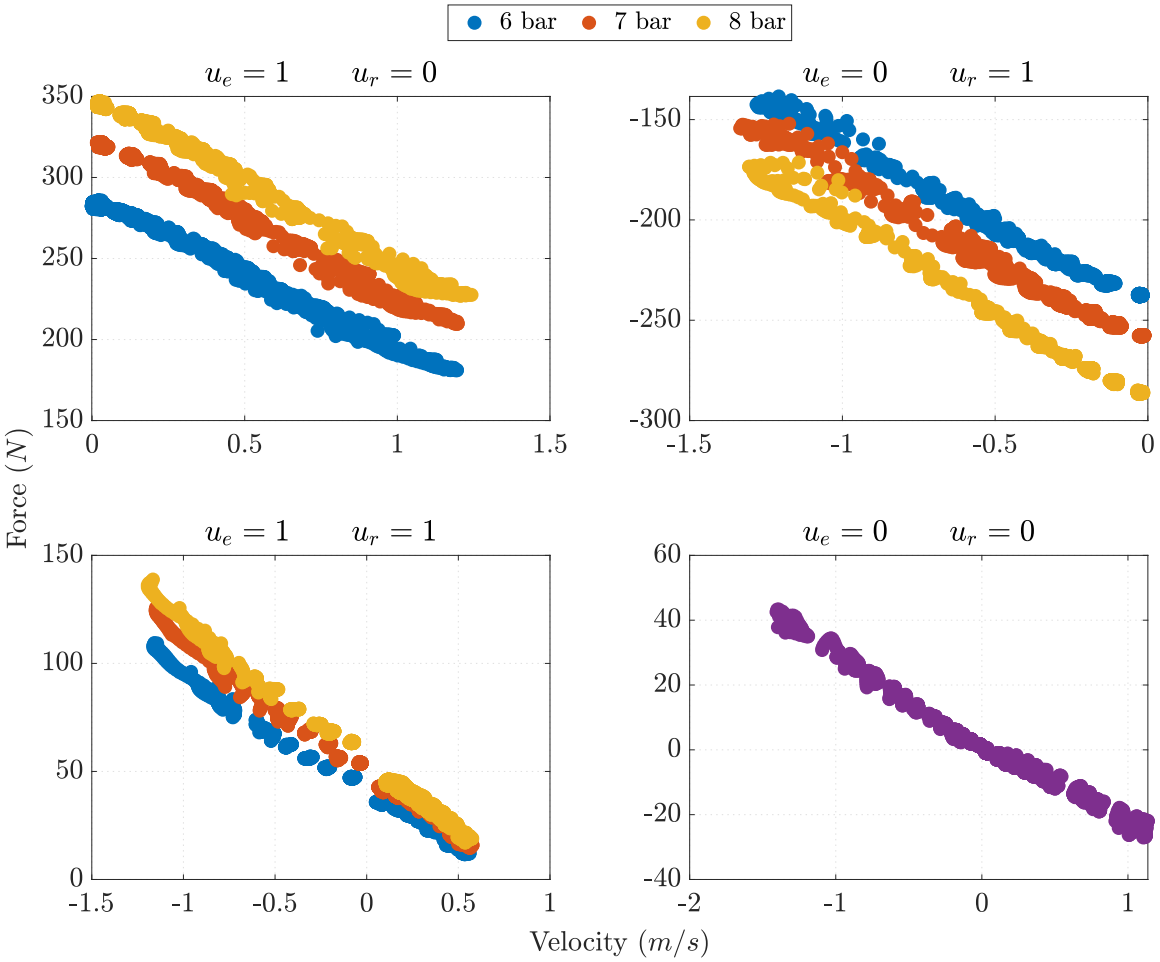

Figure 3-6: Relationship between piston force and rod velocity for the four discrete actuation modes.

Table 3.3: Pneumatic steady-state linear fit results.

Mode	**Supply (MPa)**	$\boldsymbol{F_{static}}$ **(N)**	$\boldsymbol{c_d}$ **(Ns/m)**	$\boldsymbol{R^2}$
Extend	0.6	287	91	0.992
	0.7	324	97	0.994
	0.8	349	103	0.995
Retract	0.6	-239	78	0.995
	0.7	-260	84	0.989
	0.8	-291	89	0.994
Both	0.6	41	52	0.993
	0.7	50	59	0.989
	0.8	55	64	0.991
Unactuated	0.0	0	25	0.980

The simplicity of these relationships is surprising when considering that it is the combination of two opposing chambers, each with complex pressure and mass flow dynamics. Consider just the simple unactuated state. For the chamber with a rapidly decreasing volume, pressure will rise while simultaneously air is being forced out through the small orifice in the solenoid valve. In the opposing chamber, the rapidly increasing volume causes a pressure drop while air is again simultaneously entering through the solenoid valve orifice. The pressure felt at the output is the complex combined effect of both of these chamber dynamics, and yet, the dynamics of the system converges to a simple linear relationship for the entire velocity range.

From Table 3.3 it is apparent that the linear force-velocity relationship depends on both the supply pressure and actuation mode. For the following proposed model we only consider the dependence on actuation mode, assuming a nominal supply pressure of 0.7 MPa. The damping constant c_d changes depending on the actuation mode. To describe this as a single function which is both smooth and continuous for gradient based optimisation, a bilinear fit was used, with the discrete actuation modes imposed later during optimisation. The damping term c_d is described by the following equation:

$$c_d(u_e, u_r) = c_0 + u_e c_1 + u_r c_2 + u_e u_r c_3 \tag{3.22}$$

Where u_e and u_r are the extend and retract chamber valve commands, and c_0, c_1, c_2, c_3 are the constants found by fitting the bilinear surface to the data in Table 3.3. The force-velocity relationship characterised in this section and the force response from the previous section are combined to produce the full double acting pneumatic cylinder model.

3.2.4 Simplified Model

The full actuator model is the combination of the transient force response for the two chambers with the overall steady-state force-velocity behaviour. The force output for the actuator can therefore be written as:

$$F = F_e(u_e, x_e) - F_r(u_r, x_r) - c_d(u_e, u_r)\dot{x}, \tag{3.23}$$

where $F_e()$ and $F_r()$ are the extend and retract chamber first-order dynamics and $c_d()$ is the combined damping effect. This model provides a significant simplification on the physics-based model whilst still retaining the dominant force dynamics. To verify the performance of this proposed actuator model its output force can be compared to the output force predicted by the physics model. Both models are sent the same random Bernoulli distributed binary valve commands. The piston and velocity state are also randomly generated from a uniform distribution to ensure adequate exploration of the full state space. Both the valve commands and state values are sent to the models at 10 Hz. The predicted piston force output from both the physics-based and approximate model for 800 different command and state combinations is shown in Figure 3-7.

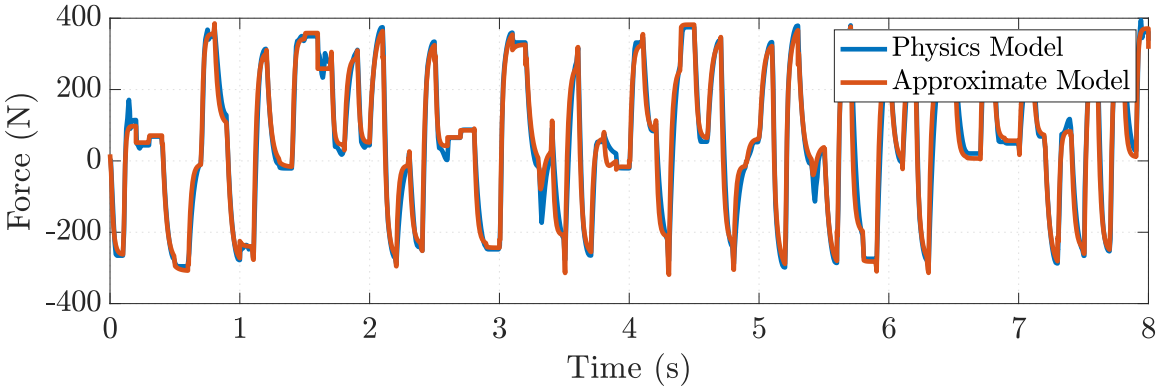

Figure 3-7: Plot showing the predicted piston output force for the physics-based and approximate model for random rod states and valve commands.

There are slight differences between the models at the command and state transitions, however, they both very quickly converge on the same force prediction. This is expected as with these major simplifications it is not possible to capture all of the higher-order transience. Furthermore, the goal of this approximate model was to capture only the dominant dynamics whilst remaining as simple as possible to balance accuracy and complexity for the optimisation. This model should therefore provide a good approximation in trajectory optimisation for what is achievable on the real hardware.

3.2.5 Integration with Optimisation

The approximate piston model includes a differential equation for the chamber force response. An additional constraint was included to enforce Implicit Euler integration of these dynamics in the optimisation formulation. This is the same integration method used for the integration of the body dynamics. The approximate model accepts a continuous valve command between 0 and 1 to keep all of the functions smooth for gradient-based optimisation. In reality, the valve command can only ever be 1 or 0, so the following constraint squeezes the valve command to those discrete states.

$$u_i(1 - u_i) \leq P \tag{3.24}$$

By minimising the penalty term P and forcing it closer to zero, the valve command u_i will be squeezed to the discrete state of either 0 or 1. This is identical to the relaxed complementarity method [45] and the penalty term is included in the cost function penalty sum. As with contacts, it is therefore possible that a feasible solution is found which satisfies all of the constraints but does not result in sufficient minimisation of the valve penalty term, so this is verified for all of the solutions.

3.3 Motion Experiments

A number of motions were generated for the planar robot using the trajectory optimisation model. These motions aim to demonstrate the dynamic capabilities of the robot, show the benefit of using trajectory optimisation to generate complex motions, and also to provide insight into the design of the locomotion and accelerate controllers. The motion experiments include vertical jumps, bounding, accelerating from rest, and a somersault. All of the motion experiments were written in the Python optimisation library Pyomo [46] and solved using the Nonlinear Program (NLP) solver IPOPT [18] with its linear solver MA86. This solver combination has shown excellent performance for these high-dimensional optimisation problems with a large number of variables. While initialisation of the variables close to a known solution or even with random values has been shown to help the optimiser find a feasible solution, no variable initialisation was used in any of these experiments to keep the problem formulation as simple and as repeatable as possible. Code showing the calculation

of the robot dynamics, optimisation formulation, and motion results can be found in Appendix E.

3.3.1 Bounding

A bounding motion was of particular interest as one of the primary aims of this work was to make the robot bound, however, initially it was very unclear how this motion would look with this particular leg and actuator configuration. The motion from trajectory optimisation provided insight into how the pistons should be actuated and how fast the robot could be expected to bound. A steady-state gait is inherently a

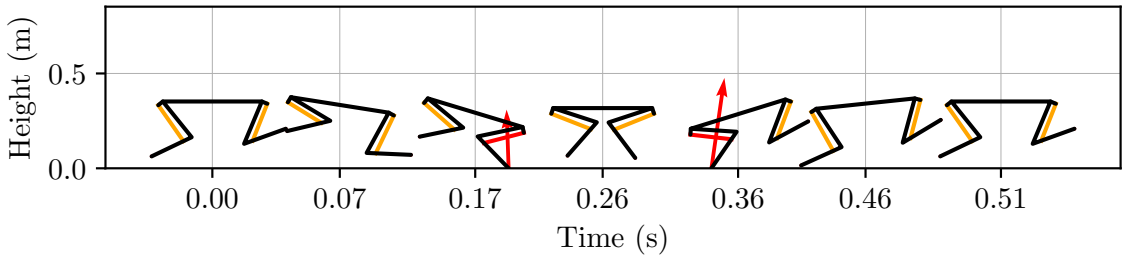

Figure 3-8: Evenly spaced frames of a periodic 2 m/s bound.

periodic motion, repeating between strides. To enforce this periodicity the positions and velocities for the body and joints were constrained to be equal at the start and end, except for the body's horizontal position which must increase as the robot moves forward. At the start and middle the body was constrained to be the horizontal points of the body's back and forth pitching. It was particularly challenging for the to solver to converge on a feasible solution, so foot contacts were prespecified by fixing the foot height and GRF for stance and flight phases to guide the solver to a realistic solution. The front foot was constrained to be on the ground from 20 % to 40 % of the bounding period, and the back foot on the ground from 60 % to 80 %. To provide sufficient flexibility around these contact durations the maximum timestep variation was increased from the typical ±20 % to ±50 %. A single bound took approximately 0.51 s irrespective of the speed with a fairly regular body pitching frequency. To enforce a particular speed, a constraint was set on the final body position. For this approximately 2 m/s bound, the body was constrained to finish at least 1 m from the starting point. Feasible solutions for slightly faster bounds were found. However, these typically had very irregular hip torque profiles and valve switching, indicating

that 2 m/s was most likely close to the maximum speed for this robot. This bounding trajectory was trasncribed with 100 nodes and is shown in Figure 3-8. A video of the motion can be found in Appendix D. The actuator commands for this motion

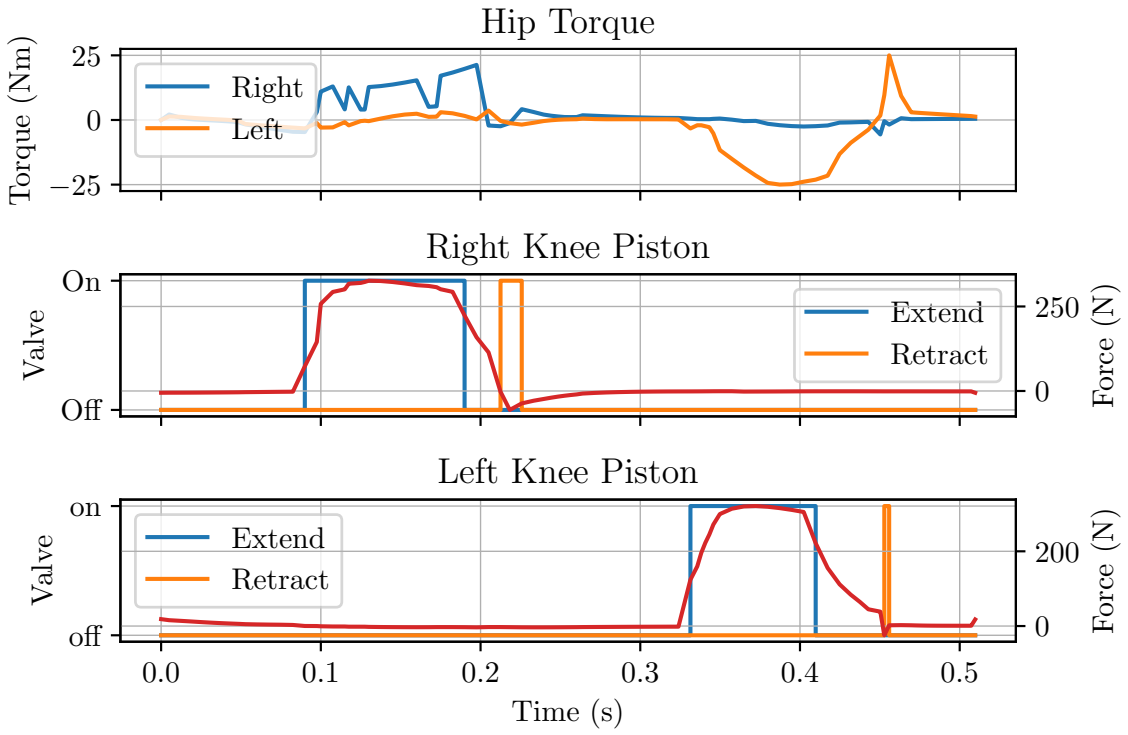

Figure 3-9: Actuator commands for a 2 m/s bound.

shown in Figure 3-9 provide some useful insight into how the pistons could be used. Interestingly, careful ground reaction force regulation with piston pulsing does not appear necessary as the pistons are simply providing their maximum force while the feet are on the ground. The duration of the piston force is most likely the means by which vertical momentum is regulated. Quick retraction pulses are also used to retract the legs so they can be swung forwards without colliding with the ground. When not pushing off the ground or retracting the pistons simply remain unactuated. This insight proved invaluable for the design of the locomotion controller in the following chapter. It must also be emphasised that these simple switching profiles are in no way specified or incentivised in the optimisation formulation and result merely from the motion and intrinsic actuator dynamics.

3.3.2 Accelerating From Rest

Another motion of primary interest for this thesis was accelerating from rest into a top speed bound. As this is effectively a gait transition it was not clear what the footfall pattern should be or how the actuators should be used. Producing this motion

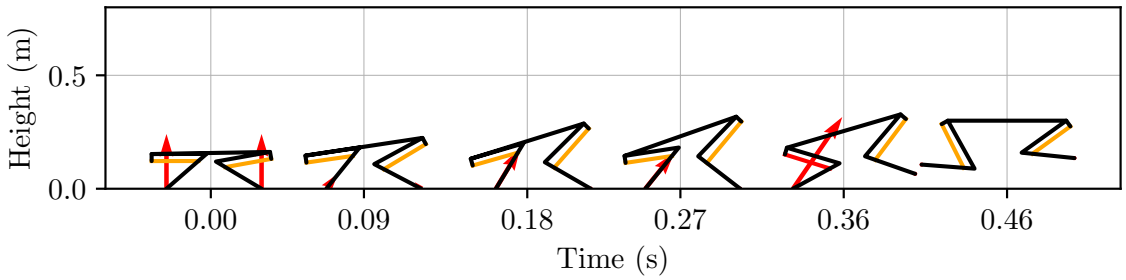

Figure 3-10: Evenly spaced frames of an accelerate from rest motion into a 2 m/s bounding gait.

with trajectory optimisation was straightforward with only the initial and final poses specified. The robot was constrained to start in the rest pose: pistons fully retracted, both feet on the ground, and completely stationary. For the final pose the robot was constrained to end in the flight phase of a 2 m/s bounding motion. This had the body constrained to be horizontal, 0.3 m above the ground, pitching forward at about 3.0 rad/s, with both feet off the ground, and travelling forward at 2 m/s. These parameters were taken from a prior bounding motion. For this motion to be executed as quickly as possible the duration constraint was iteratively decreased to a value of 0.4 s. For a shorter duration the solver could not converge on a feasible solution. This trajectory was transcribed into 100 nodes and is shown in Figure 3-10. A video of the motion can also be found in Appendix D. This motion elucidates an interesting sequence of states that initially, was rather counter-intuitive. Instead of immediately trying to start moving forwards, the front leg is used to pop-up and rock the body onto the back leg. This results in the robot's centre of mass moving backwards slightly. However, from this point the robot is in a configuration almost identical to the crouched set position for an accelerating dog [32]. From here it can simply propel itself forward, pushing off with a large left hip torque. The back hip is also preemptively positioned right up against the body at its joint limit. Similar to bounding this motion also only requires a simple piston firing scheme with the front pop-up and rear push-off having discrete extending periods, followed by short

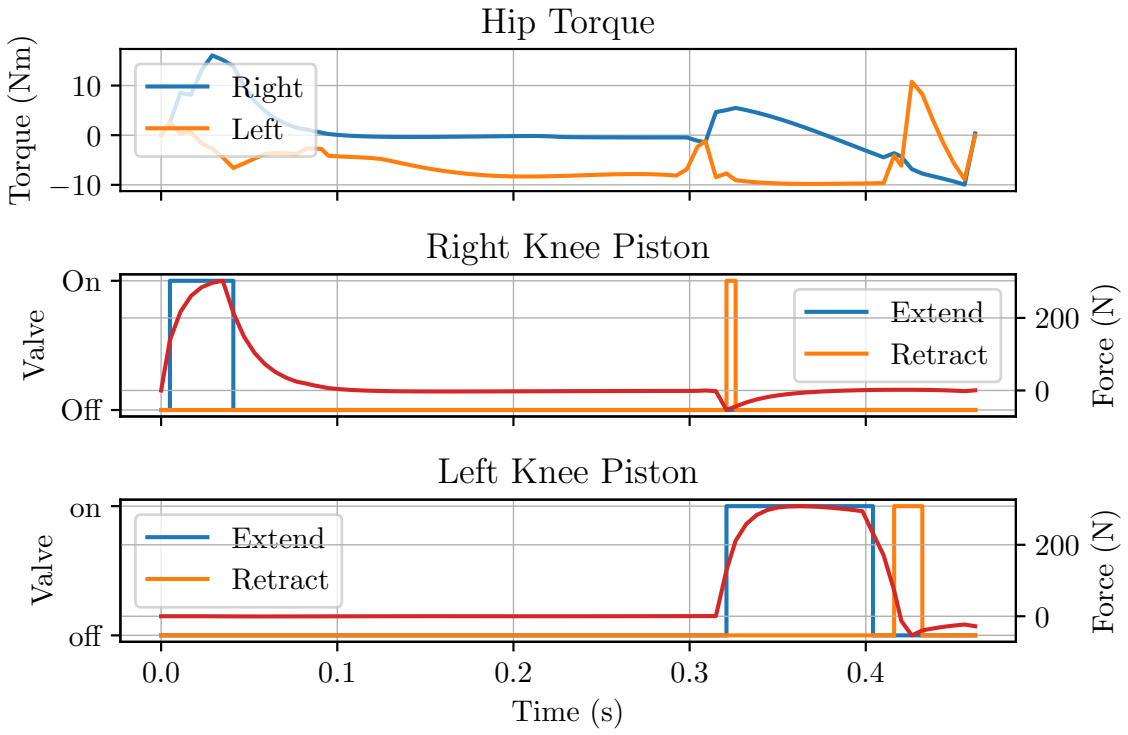

Figure 3-11: Actuator commands for an accelerate from rest motion.

retraction pulses to lift the feet.

3.3.3 Somersault

A somersault has become quite a popular motion for demonstrating the dynamic capabilities of a legged robot. It requires powerful legs to launch the body both high enough, and with sufficient rotational velocity. It is also an interesting control problem as whilst in the air, control of the body rotation is limited. The first legged robot to execute a flip from a running gait was the Leg Lab 3D-Biped in 1992 [12]. This robot used a simple state-based controller and adjusted how much the legs were tucked in to control the angular rate of the body whilst in the air. Following this Boston Dynamics' biped robot Atlas performed a backflip off of a raised box in 2017, but unfortunately there is no published work on the controller. MIT's Mini Cheetah quadruped robot performed a backflip from standing in 2019 [7]. This somersault motion was generated offline using trajectory optimisation and then simply executed on the robot with only joint feedback and high leg damping for landing to handle body orientation errors. Unfortunately executing the flip on this robot would not be possible in reality as all of the sensor, power, and pneumatic cables would become

wrapped around the rotating end of the boom. However, it is still useful to see that the robot is capable of executing a dynamic manoeuvre like this in simulation and is not restricted by its particular actuator configuration. The motion is shown in Figure 3-12 and a video can be found in Appendix D. To produce the motion the

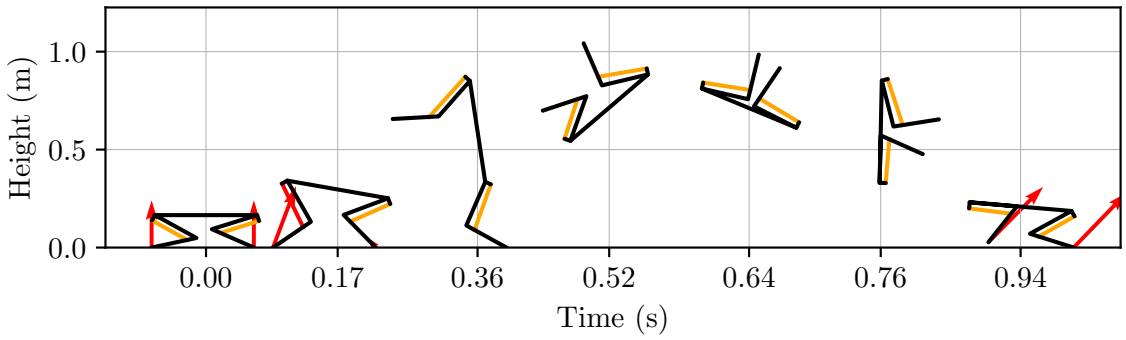

Figure 3-12: Evenly spaced frames of somersault motion.

robot was constrained to start in the rest pose with pistons fully retracted, feet on the ground, and completely stationary. For the apex the body was constrained to be upside down with a body angle of $-\pi\,rad$ and both feet off of the ground. Rather than being in the middle of the motion the apex was around 70 % of the way through with the launch phase taking quite a bit longer than the landing. For landing the body was constrained to finish at an angle of at most $-1.9\pi\,rad$, with both feet on the ground, and the knee pistons retracted. The whole motion requires approximately 1 s for a full rotation and was transcribed with 100 nodes. The somersault motion is shown in Figure 3-12. The robot reaches a height of about 0.7 m and is rotating at approximately 9.6 rad/s (550 °/s) before landing. Most interesting is how the legs are tucked in during the flight phase, possibly to reduce the body's rotational inertia, similar to how a gymnast would tuck in their legs. The right leg is also tucked in quickly with a sharp change in torque direction after the foot leaves the ground to impart additional rotation. Surprisingly for such a dynamic manoeuvre, only 40 % of the maximum torque at the hips is used with most of this being for the right leg during the push-off and tuck portion.

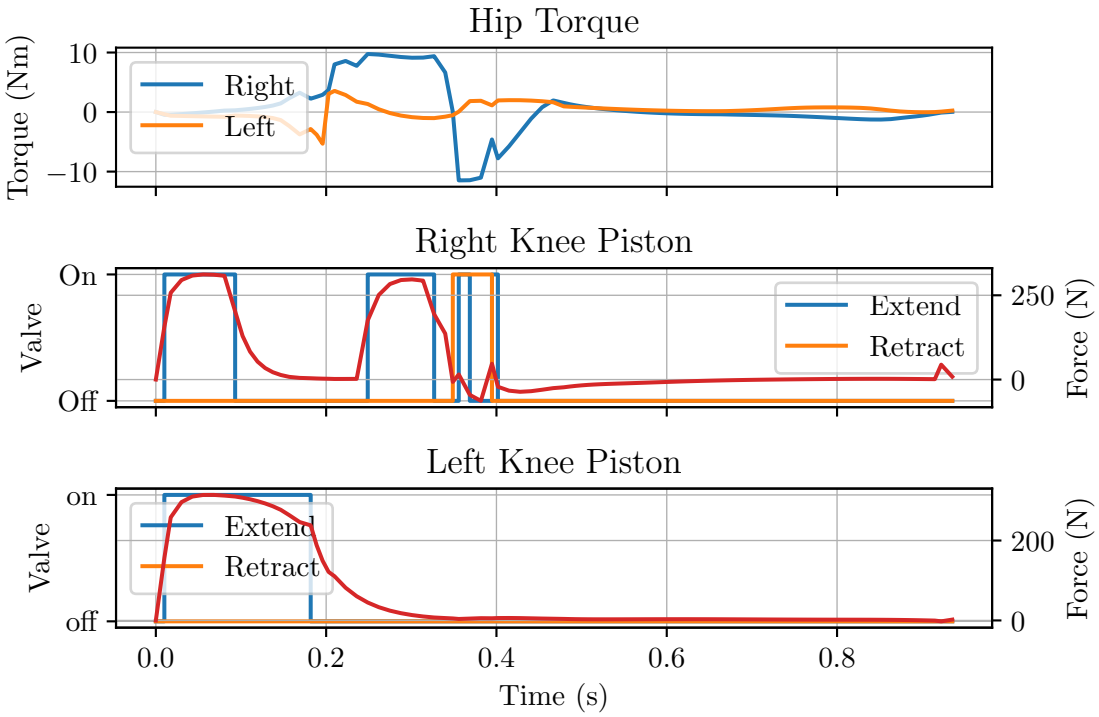

Figure 3-13: Actuator commands for somersault.

3.3.4 Limitations

While these motions are all considered optimal by the solver, due to their non-convexity as a result of the contact formulation and nonlinear dynamics, there is no formal guarantee for their optimality. It is therefore possible that the solver has found a local minimum, and that a marginally *better* solution could be found with some slightly different variable initialisation or collection of constraints. These motions therefore shouldn't be viewed as optimal in a strict sense, but rather as feasible trajectories. Most importantly, these are all realistic motions for the robot that adhere to actuator constraints, as well as momentum and energy conservation within the rigid body framework.

3.4 Hardware Validation

To verify that the model of the robot and actuators used with trajectory optimisation produces realistic and physically feasible results we executed a prior generated 1 m vertical jump trajectory from optimisation directly on the hardware similar to other approaches [7, 23]. Modelling discrepancies will show up as significant tracking errors

and most likely improper timing and execution of the trajectory. Small modelling discrepancies are inevitable. However, these should not be so significant that even this simple motion cannot be executed. A 1 m vertical jump was used as it demonstrates the dynamic capabilities of the platform, makes full use of the knee pistons, and has a very clear jump height metric, all while being a relatively *safe* manoeuvre to execute with minimal body pitching. At 1 m it would be the highest jump performed by a legged robot in our Lab with *Baleka* jumping to a maximum height of 0.92 m [36]. The majority of the vertical impulse for the jump comes from the knees and pistons so any discrepancies in the piston model will show up clearly as the robot not achieving the height anticipated in the generated trajectory. Determining actuator commands for a precise jump height is also not a trivial task, requiring specific timing on valve commands to produce sufficient impulse. While it is certainly possible for the robot to jump and land, we chose to only consider the launch and flight phases and catch the robot just after the apex. Small variations in actual jump height produce very large contact timing errors upon landing. For example, if the robot jumps slightly below its desired height it will hit the ground much earlier than expected resulting in the valves activating too late and the pistons bottoming out under the full weight of the falling robot. Up to this point, the hardware had proven to be very capable and robust. However, it was deemed too risky to test this further with a potentially damaging 1 m drop. Previous work [23] got around this contact timing issue with a dedicated landing controller that would work at a range of heights and would adjust based on foot touchdown which can be explored in future work.

3.4.1 Trajectory Tracking

To track the trajectory from optimisation, the desired hip position, velocity, and feed-forward torque commands are sent to the PD controllers running in the hip motors. The transmitted trajectories are linearly interpolated to transform the variable 200 Hz trajectories from optimisation to the fixed 1 kHz control rate used on the hardware. Valve commands from optimisation are sent directly to the solenoids shifted forward by 6 ms to account for the dead-time which cannot be included in optimisation. It is worth noting that unlike with the motors, there is absolutely no feedback on the pistons. Any position tracking error will be solely due to modelling discrepancies.

The commands from optimisation being sent to the actuators are shown in Figure 3-14. The piston dynamics can be clearly seen on the green piston force plot with the

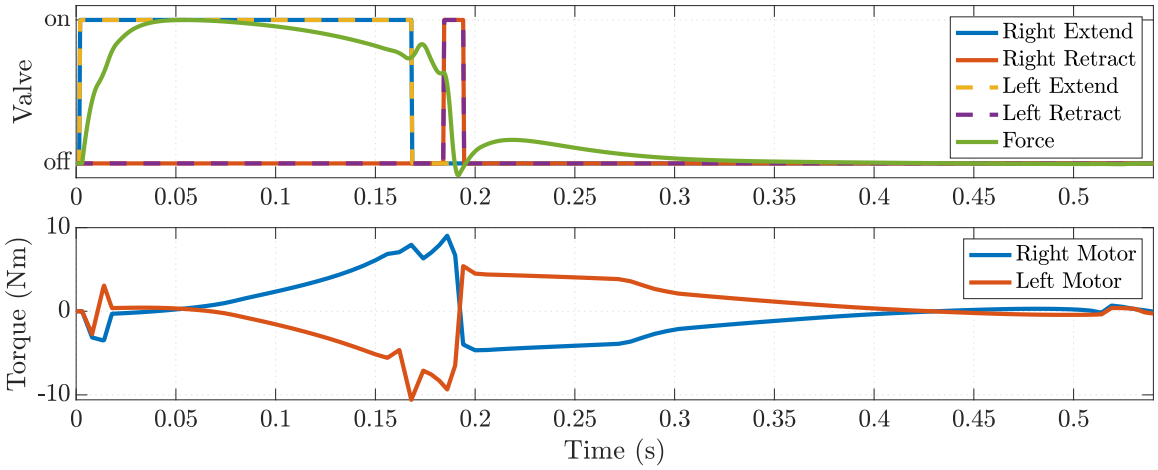

Figure 3-14: Valve and motor torque commands from optimisation for 1 m jump trajectory. The estimated piston force is also shown in green with the valve commands.

force dropping as the knees extend faster. The residual force is also accounted for in the jump with the extend valve turning off before the feet have even left the ground.

3.4.2 Results

The robot jumped to a height of 0.95 m with minimal tracking error at the pistons. Figure 3-15 shows a sequence of overlaid frames of the jump motion and a video can be found in Appendix D. While not perfect, the similarity between the optimised trajectory and the performance on the actual hardware provides a degree of confidence that the system model and implementation of the pneumatic model in trajectory optimisation is a reasonable approximation of reality. The jump also sets a new height record for a legged robot in the lab whilst only utilising 40 % of its available 25 N m of motor torque. Figure 3-16 shows the tracking performance. Tracking of the pistons is not perfect and the positioning error of the left piston is slightly higher than the right piston, causing the body to leave the ground unevenly. This is most likely the cause for the 0.33 rad (19°) body pitch deviation, and possibly also contributes to the slightly lower final height. In reality, modelling errors are inevitable, especially considering the reduced-order approximations required for the pneumatic model. Tracking performance will therefore benefit from a feedback controller on the pistons which can provide small valve timing corrections around a nominal trajectory,

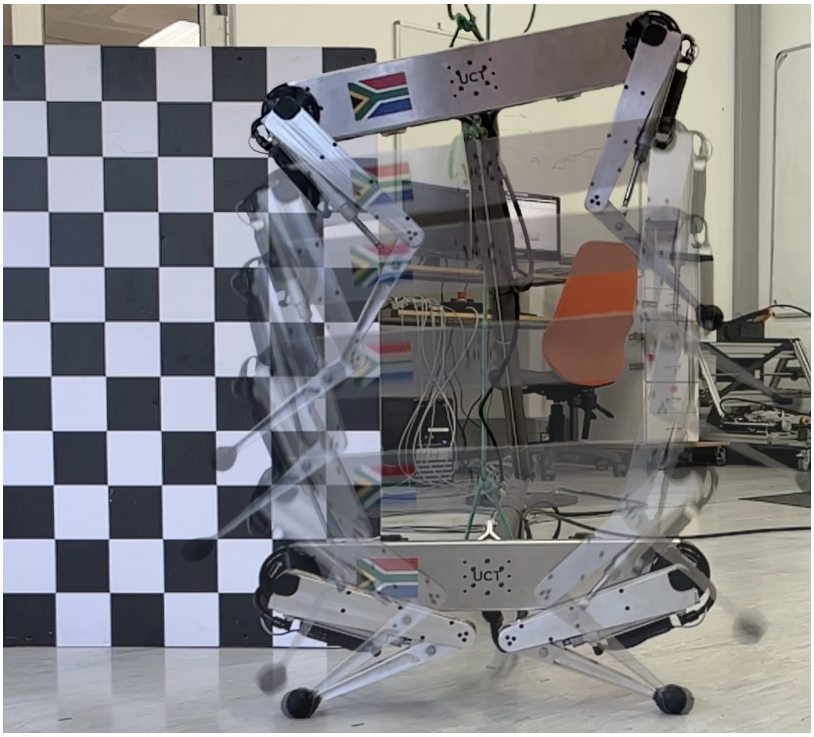

Figure 3-15: A sequence of overlaid frames showing the 1 m jump trajectory executed on the hardware. The squares on the checkerboard are 9 cm and the green rope is used to catch the robot after the apex.

similar to the motor's PD controller.

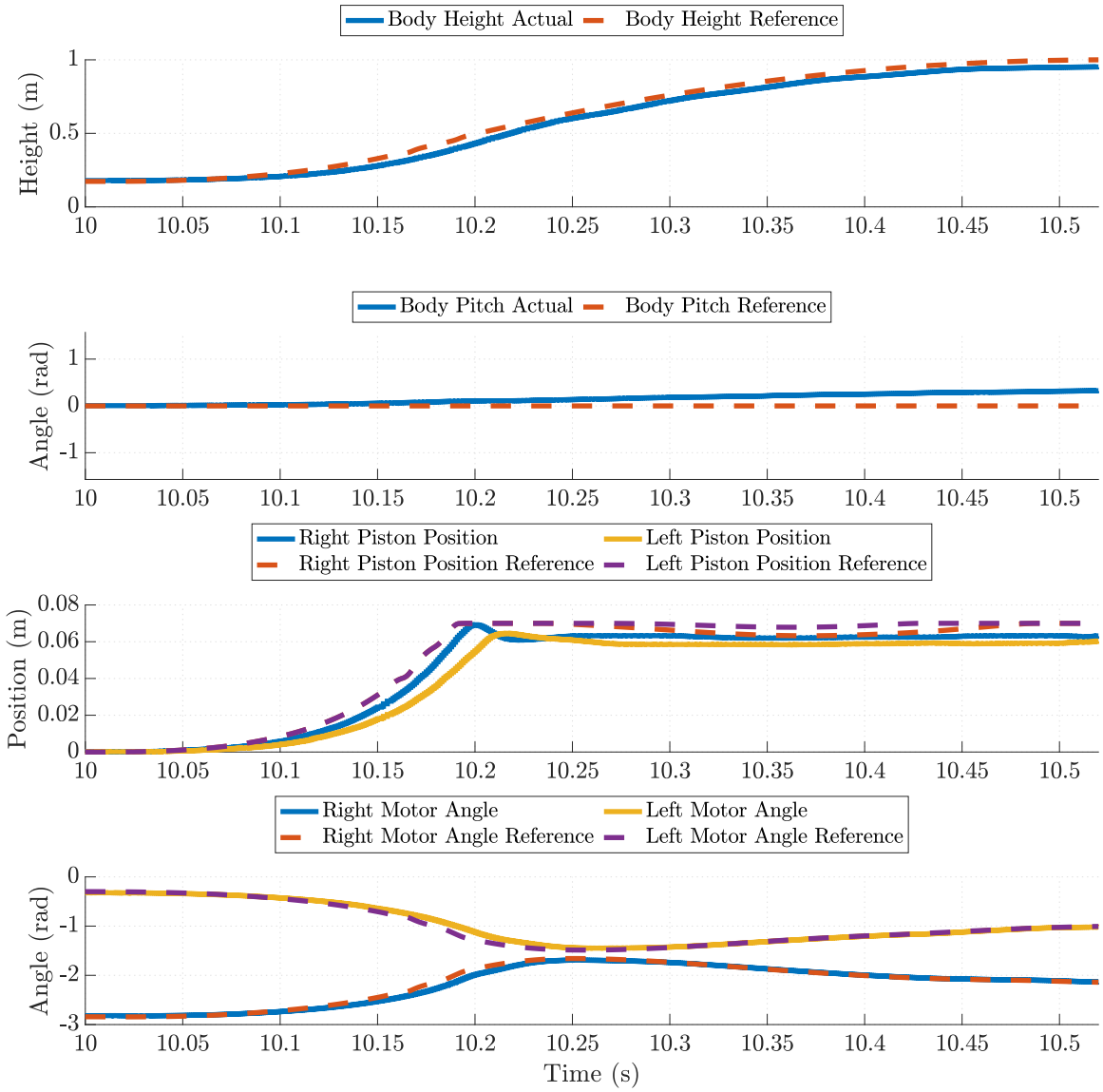

Figure 3-16: Tracking performance of the body and actuators when executing the 1 m vertical jump. The body reaches a height of 0.95 m, slightly below the desired 1 m.

Chapter 4

Bounding and Acceleration Control

This locomotion controller was inspired by the remarkable acceleration manoeuvres exhibited by cheetahs and greyhounds and the desire to try and make Kemba move in a similar way. The goal was to make Kemba able to bound at a desired velocity, and also accelerate as quickly as possible into this bounding gait. On its own, a bounding gait is particularly interesting as it has fully extended flight phases where both feet are entirely off the ground, and it requires the body to pitch back and forth as the front and back legs alternate between support phases. These make it a particularly dynamic motion and an interesting control problem. Steady-state motions like bounding however typically break down with large accelerations, motivating the design of an additional acceleration controller. This chapter details the development of bounding and acceleration controllers, along with a simulation used to test and debug the controllers before execution on the hardware.

4.1 Simulation and Visualisation

The simulation of the robot was a crucial part of control development. All the controllers presented here were first tested in this simulation before being transferred to the actual hardware. The simulator was also structured so that data logs from runs on the hardware could be replayed with the simulation serving as a visualisation tool. A rendering of the robot moving typically provided a much more intuitive means of identifying faults in the controller or areas for improvement. These visualisations could also be replayed at substantially slower than real-time speed due to the 1 kHz

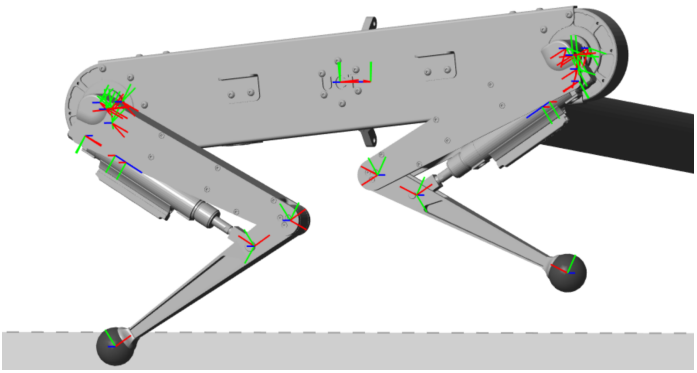

Figure 4-1: Rendering of robot and boom in simulation with link coordinate frames.

logging frequency. The selection and design of the simulation was guided by two main objectives:

- Ease of integration with hardware controllers
- Prioritise accuracy over simulation speed

The hardware controllers were built in Simulink to run on the Speedgoat from MathWorks. The simulation and visualisation were both built in Simscape Multibody to simplify reuse of the simulation and hardware controllers and to keeping everything within the MathWorks product suite. As a result the controller running on the hardware could be run directly in the Simscape Multibody simulation without any modifications, dramatically reducing development time and the potential for bugs. All of the simulation and controller code has been provided in Appendix E.

Typically, increases in simulation accuracy result in an increase in computational complexity and therefore decrease simulation speed. The purpose of this simulation is to test controllers before running them on the hardware. The simulation will not be used for time critical tasks such as real-time control. There is no benefit to sacrificing simulation accuracy for speed, so where possible the more accurate modelling approach was used.

The following description of the simulation modelling approach is the result of iterative refinement; comparing the simulation to the real system and modelling additional components as necessary.

4.1.1 Robot Dynamics

The robot is modelled in Simscape Multibody with a rigid body model, each with measured masses, and inertia tensors estimated from SolidWorks. The legs are modelled with the closed kinematic chain formed between the piston and knee joint. Simscape Multibody automatically computes the rigid body dynamics in 3D for an open kinematic tree structure of the robot and then adds additional constraints to maintain the closed loops.

In prioritising simulation accuracy, the four rigid bodies of the boom were also included in the dynamics model. This complicates dynamics, but provides a much more accurate representation of the real system. An additional length constraint serves the same purpose as the tensioned steel cables and keeps the end of the boom and robot vertical. The movement of the boom constrains the robot's movement.

In total the system (robot and boom) comprises of $N = 13$ rigid bodies, connected by $j = 13$ single degree of freedom ($f_i = 1$) joints. Each rigid body is visually represented with a Standard Tessellation Language (STL) file. Figure 4-2 shows the rendering of the robot.

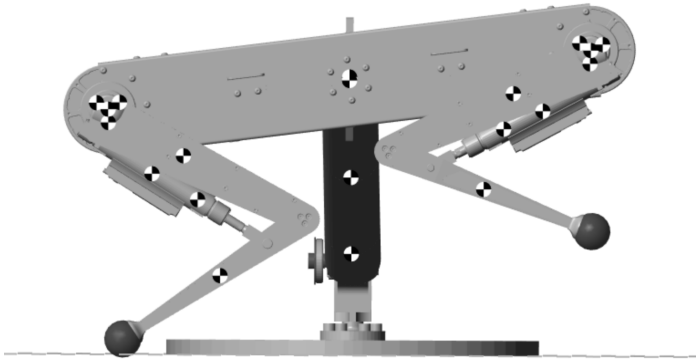

Figure 4-2: Rendering of robot and boom in simulation showing centre's of mass for rigid bodies.

Using the Chebychev–Grübler–Kutzbach criterion the number of degrees of freedom of the system can be calculated as follows.

$$M = 6 \cdot (N - 1 - j) + \sum_{i=1}^{j} f_i = 6 \cdot (13 - 1 - 13) + 13 = 7 \tag{4.1}$$

The system has a total of seven degrees of freedom. Four actuated degrees of freedom for the robots legs, and then three unactuated degrees of freedom for the body imposed by the planarising boom.

4.1.2 Motor Model

Previous work [8] has shown that the rotors in the quasi-direct drive actuators used at the hips play a significant role in the dynamics of the leg with their inertia multiplying by the gear ratio squared as described in Section 3.1.3. An additional rigid body for the rotor was included for each of the motors shown in Figure 4-3 to capture this effect. These are coupled to the output degree of freedom of the motor with a coaxial gearing constraint enforced by the planetary gearbox. The rotors are modelled from

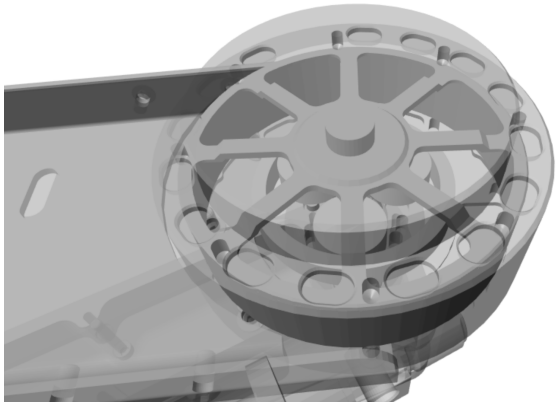

Figure 4-3: Rotor visible inside the slightly transparent motor casing. The ring and three planet gears in the epicyclic gearbox are also partially visible underneath the rotor.

a disassembled motor. Viscous friction is present in the real system as a result of the grease in the gearbox. However, the viscous forces can be considered negligible relative to the rotor inertia and actuator forces, and are therefore ignored. The motor has an internal current control loop which runs at 40 kHz producing a torque bandwidth of about 1 kHz. Similar to [47], this behaviour was approximated with a fixed frequency 1 kHz PD controller in simulation without inclusion of any of the fast current dynamics, assuming perfect torque control. Additionally, the same torque-speed curve from Section 3.1.3 is used with the motor torque and speed specifications. On the actual hardware the current controller goes unstable at very high position and velocity gain values, due to noise from the Hall effect position encoder. This is very

undesirable on the actual motors, so the velocity gain is limited in the firmware to 5 N m s/rad where the motor remains stable. The encoder noise and controller instability can therefore also be ignored.

4.1.3 Piston Model

To model the piston pressure dynamics the full physics-based model described in Section 3.2.1 is used with the additional 6 ms valve switching delay. All parameters are measured from the piston characterisation and provide sufficiently similar performance to the real system. The piston body and piston rod shown in Figure 4-4 are constrained in a prismatic joint configuration, only translating along a single axis. The piston serves as the range of motion limit for the knee joint and so the end-stop

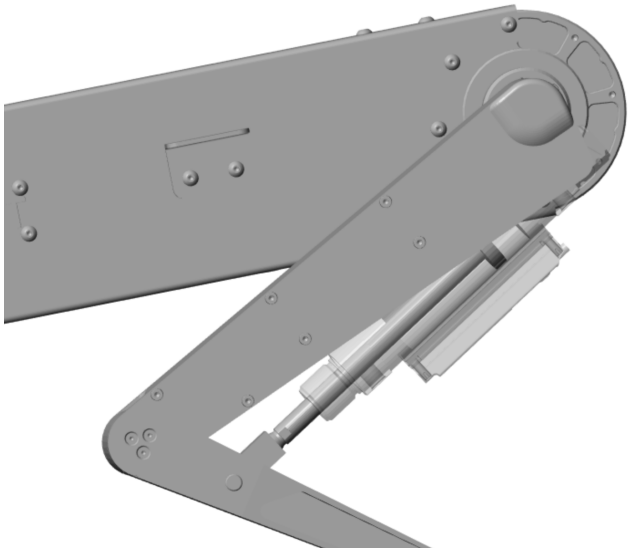

Figure 4-4: Piston rod visible inside transparent piston body.

limits inside the piston are modelled as a very stiff spring and damper system. This is the only contact method available in Simscape Multibody, and provides a reasonable approximation at small timesteps. There is no information on the mechanical structure or properties of Festo's proprietary automatically adjusting air cushioned end stops in the pistons, so these are left unmodelled.

4.1.4 Foot Contacts

Foot-ground contacts are modelled as a smoothed linear spring damper system. The ground is modelled as an infinite plane and the feet are modelled as spheres. The

ground was also assumed to be perfectly rigid. The spring constant for the feet was estimated from their deflection under the known weight of the robot and found to be approximately $1 \times 10^4\,\mathrm{N/m}$. This appears in simulation as the robot sinking into the ground slightly when the foot is loaded as shown in Figure 4-5. In reality the

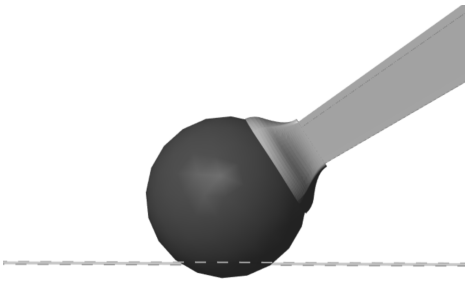

Figure 4-5: Slight foot-ground penetration as a result of the spring-damper contact model.

feet deform, flattening onto the ground. The feet are relatively well damped due to their expanding foam filling and so a damping constant of $1 \times 10^3\,\mathrm{N\,s/m}$ was used as this produced minimal foot contact oscillations. This stiff spring-damper contact model performed adequately, although the high stiffness typically required very small timesteps to solve accurately, with the simulation slowing down significantly at foot touchdown. More complex contact models exist which work for larger timesteps. However, this is the only contact model implemented natively into Simscape Multibody.

No other body contact points were implemented as self contact or any other link contact with the ground is clearly visible and only occurs when the controller fails or the robot falls over. Continuously checking for additional contacts also imposes further computational load, slowing down the simulation unnecessarily.

4.1.5 Limitations

While the simulation is an excellent approximation of the real system, it still has aspects which limit its accuracy and utility. The main one of these is the variable pressure source. The pneumatics on the physical robot are supplied with compressed air by a compressor of capacity 130 L/min, mounted on a 100 L accumulator tank. As air in the tank is used, the supply pressure drops affecting the force produced

by the actuators. The simulation assumes a constant supply pressure of 0.7 MPa, however the actual compressor supply varies between 0.6 MPa and 0.8 MPa. For this work assuming a nominal supply pressure was a sufficient approximation, however, later work might require a more accurate pressure estimate. Another limitation of the simulation was its speed. While speed was not a priority, Simscape Multibody was orders of magnitude slower than other common robotics simulators, with the small timestep required for the simplistic contact model and numerically stiff pneumatic dynamics being the main culprits. The simulations' slow speed was not an issue for this work. However, for control strategies which rely on numerous simulations to develop a control policy such as reinforcement learning, it will be severely limiting.

4.2 Bounding Controller

This section details the structure of the controller used for a bounding gait on the robot. The robot is capable of bounding to a top speed of 1.7 m/s [1].

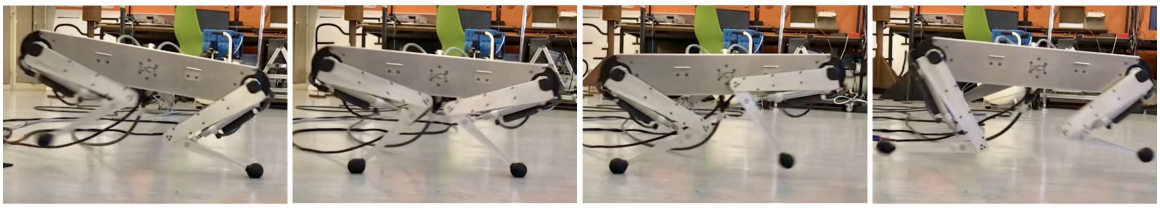

Figure 4-6: Several frames of Kemba bounding to the right at 1.5 m/s. Frames show a full gait cycle. From left to right; front foot touchdown and stance phase, rear foot touchdown, front foot lift-off and rear foot stance phase, rear foot lift-off and extended flight phase.

4.2.1 Approach

As this is the first gait implemented on the robot, and first bounding robot in the Lab, the general aim for the controller was simplicity and consistency. The bounding controller for a planar robot, presented in [11], decomposes bounding into discrete components such as height control and velocity control, and sequences control actions based on state changes. This controller, while simple, produces a relatively sophisti-

[1] This top speed is an absolute maximum on the verge of stability. It works best with air pressures around 0.7 MPa. Faster commanded speeds will most likely result in the robot tripping and falling.

cated bounding behaviour. We therefore build on this work, adapting it to Kemba's unique hardware and actuator configuration.

Fundamentally a legged robot can be thought of as a floating mass kept in the air by the forces from contacts between its feet and the ground. The forces can vary in magnitude and duration but their total impulse must ultimately remain the same to keep the mass above the ground. One approach used in [17] is to fix the stance time and regulate the vertical momentum by scaling the vertical ground reaction force. While this works well, precise vertical force scaling would be challenging with the pneumatic pistons and would most likely have a slow response. The trajectory optimisation results in the previous chapter showed that precise force control for fast bounding is not necessary with each piston having a clearly defined extend pulse during each stance phase. It must therefore be possible to control vertical momentum with a fixed vertical force by instead varying the duration of the stance phase. This is the approach we have adopted for the bounding controller.

4.2.2 Virtual Leg

Independently controlling each of the leg joints results in a relatively arbitrary movement of the foot. By coupling the degrees of freedom and controlling the position of the foot in polar coordinates the leg essentially becomes a virtual pogo stick leg shown in Figure 4-7.

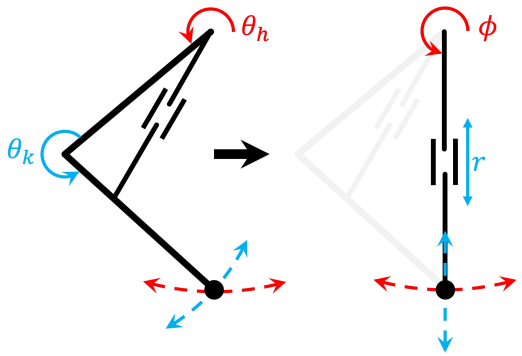

Figure 4-7: Virtual pogo stick leg simplification. The piston's degree of freedom shown in blue is constrained to lie along the length of the virtual leg.

This is achieved by using the hip motor to cancel out any movement caused by the piston along the ϕ axis, effectively restricting its degree of freedom to only the r axis. Vertical movement can now be controlled with the length or r axis of the

leg and horizontal movements can be controlled with the hip or ϕ axis of the leg, decomposing control of the leg.

The leg uses actuators with very different characteristics on each of its joints. The hip motor is very good for high fidelity position and velocity control with its internal PD controller while the knee piston can provide explosive bursts of force but remains largely unactuated most of the time. These characteristics will therefore also map to the foot coordinate system and dictate the control authority along the axes in the chosen coordinate system. This mapping is shown through the leg joint to polar Jacobian.

$$\mathbf{J}_p = \begin{bmatrix} \frac{\partial \phi}{\partial \theta_h} & \frac{\partial \phi}{\partial \theta_k} \\ \frac{\partial r}{\partial \theta_h} & \frac{\partial r}{\partial \theta_k} \end{bmatrix} = \begin{bmatrix} 1 & \cdots \\ 0 & \cdots \end{bmatrix} \tag{4.2}$$

The terms in the second column are left out for brevity. As $\frac{\partial r}{\partial \theta_h} = 0$, the hip θ_h is unable to affect any change in the r axis irrespective of the configuration of the leg. Therefore any vertical height r axis control is done entirely by the piston, with the hip free to control the ϕ axis and compensate for any movement in this axis as a result of the knee piston and nonzero $\frac{\partial \phi}{\partial \theta_k}$ term. It is worth noting that the the selection of the coordinate axes is not arbitrary as using a Cartesian coordinate system would not provide these simplifications.

The angle of the body is also included in control of the leg to keep the ϕ axis relative to the world frame, and so as the body pitches the hip will adjust to always maintain the same leg angle with respect to the ground. The kinematics for the leg can be found in Appendix B.

4.2.3 Vertical Hopping Control

For the robot to be able to move forwards or backwards, each of the legs must spend some time on the ground supporting the body and the remaining time in the air, free to move around independent of the ground. This is essentially a hopping motion. Previous work [11] found that with independent hopping of the front and back legs in a simulated planar dog, the body naturally settling into a stable coupled oscillation with the front and back legs 180° out of phase. Therefore all that is required is for

a single leg to bounce between a minimum and maximum height. Back and forth pitching of the body and centre of mass height control will all follow suit.

This bouncing is controlled with the piston in the leg along the virtual leg r axis. The piston should provide thrust when on the ground and supporting the weight of the body and turn off at the correct time to achieve the desired maximum height. By controlling the hopping heights of the two legs the body height can also be controlled.

A single leg hopping vertically can be simplified into a lumped mass m at the hip height y above the ground with weight W and acted on by leg radial axis force F_r which can be switched on or off. Trivially, for a mass moving vertically and acted on

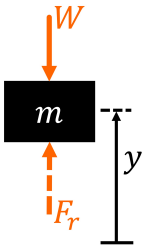

Figure 4-8: Leg lumped mass approximation.

by a net force:

$$F_{net} = m\ddot{y} \tag{4.3}$$

Integrating (4.3) twice and substituting the first integration into the second results in the following time independent state space trajectory for the vertical motion of the mass:

$$y - y_0 = \frac{m\dot{y}^2}{2F_{net}} \tag{4.4}$$

The net force on the mass can be expressed using the following piecewise function which switches depending on whether the leg is pushing off the ground (Thrust) or falling (Flight).

$$F_{net} = \begin{cases} -mg, & \text{Flight} \\ F_r - mg, & \text{Thrust } (F_r > mg) \end{cases} \tag{4.5}$$

With adequate air pressure the leg always produces sufficient F_r to overcome its weight and raise the hip, so the net vertical force during thrust is always positive. Substituting these values for F_{net} into (4.4) with appropriate constants and plotting the two trajectories on the phase plane produces Figure 4-9. The two parabolic flight

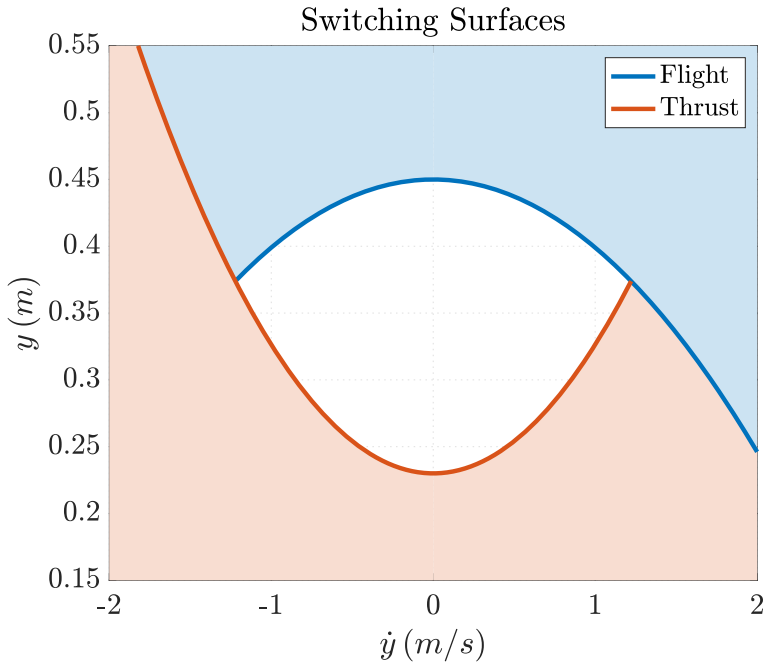

Figure 4-9: Phase plot of hip mass flight and thrust trajectories with switching regions to maintain a desired limit cycle.

and thrust trajectories form a neat switching surface which can be used to control vertical hopping and maintain a stable limit cycle. During the thrust phase the lumped hip mass will move counterclockwise along the bold red trajectory. When it intersects the bold blue trajectory the thrust force must be turned off. The hip mass will then follow this ballistic flight trajectory until it intersects the red trajectory where the thrust force needs to be turned on again to catch the hip mass before it hits the ground. The shaded red region represents where the piston thrust force needs to be on and the shaded blue region where the piston force must be off. This control law can be written as:

$$\text{Thrust} = \begin{cases} \text{On}, & \text{when } y \leq \frac{m\dot{y}^2}{2(F_r - mg)} + y_{min} \\ \text{Off}, & \text{when } y \geq \frac{\dot{y}^2}{2g} + y_{max} \end{cases} \tag{4.6}$$

The constants y_{max} and y_{min} provide an intuitive means for setting the maximum and minimum heights the hip will reach when hopping by shifting the flight and thrust switching surfaces in Figure 4-9 up or down. Another possibly more intuitive way of thinking about this controller is with energy, as disguised in the control law is the vertical kinetic energy of the hip $\frac{1}{2}m\dot{y}^2$. Essentially, the thrust force must turn on when it can do sufficient negative work to remove all of the kinetic energy from the

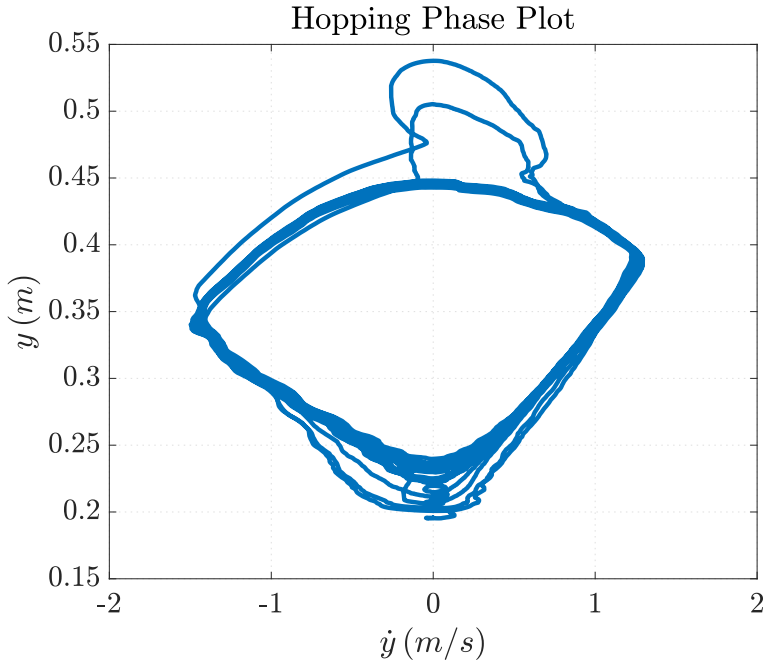

Figure 4-10: Hopping limit cycle phase plot for a single leg mounted to the boom. Deviation from the limit cycle at the top of the plot is due to the leg being lifted and dropped.

falling hip by the time it reaches the desired minimum height. Following this the thrust force must then turn off again once the hip has gained sufficient kinetic energy to reach the maximum or apex height on its ballistic trajectory. This ensures that even if the hip is dropped from an arbitrary height, it will still come right back to the desired limit cycle.

By regulating hip height, body height and vertical momentum are also implicitly regulated. The control law is independent of time so the body's vertical momentum will be regulated with a fixed force by simply keeping the leg thrust force on for longer until the correct height is reached, thereby scaling the vertical impulse delivered by the leg. This controller was tested on a single leg hopping on the end of the boom and the results are plotted in Figure 4-10. This plot is very close to the anticipated behaviour shown in Figure 4-9, reaching the desired maximum and minimum heights for hopping. Even when lifted and then dropped the leg quickly returns to the limit cycle at the thrust switching surface. The slight left asymmetry is a result of the assumption that the piston force is constant and turns on or off immediately. This is not the case for the real piston due to slow valve switching and pressure dynamics, yet even with these approximations the control law produces the desired behaviour.

This controller will ensure the legs hop between desired heights alternating between the stance and flight phases necessary for bounding. To pick the leg up off the ground after the thrust force has been turned off, the retract chamber is simply pulsed with air for a fixed time of 20 ms. This provides just enough pressure to fully retract the leg without fully pressurising the chamber, as only a small amount of force is required to retract the leg.

4.2.4 Velocity Control

Forward speed is controlled by using the foot to push on the ground and propel the body forwards during the stance phase. Torques about the hip produce this pushing force. To maintain a desired forward speed these torques are proportional to the velocity tracking error. The motor PD controller is used to regulate hip torque by tracking the virtual leg's angular speed $\dot{\phi}$ to produce a desired foot speed and move the body forwards. The desired horizontal speed of the foot $\dot{x}$ is related to the virtual leg's $\dot{\phi}$ by

$$\dot{\phi} = \frac{\dot{r}\cos(\phi) + \dot{x}_d}{r\sin(\phi)}. \tag{4.7}$$

As the virtual leg shortens or lengthens along r while compressing and pushing off the ground, this adjusts $\dot{\phi}$ of the leg to maintain the desired horizontal speed of the foot $\dot{x}_d$. The velocity of the foot is related to the joint velocities of the leg via the leg Jacobian in (4.2).

$$\dot{\mathbf{x}} = \mathbf{J}_p\dot{\mathbf{q}} \tag{4.8}$$

Where $\dot{\mathbf{x}}$ is the foot polar coordinates and $\dot{\mathbf{q}}$ the joint velocities. From this the virtual leg angular speed $\dot{\phi}$ is given by:

$$\dot{\phi} = \dot{\theta}_h + \frac{\partial\phi}{\partial\theta_k}\dot{\theta}_k \tag{4.9}$$

Rearranging this equation to instead make $\dot{\theta}_h$ the subject allows the hip motor velocity to be determined as a function of the desired virtual leg angular speed $\dot{\phi}_{des}$.

$$\dot{\theta}_h = \dot{\phi}_d - \frac{\partial\phi}{\partial\theta_k}\dot{\theta}_k \tag{4.10}$$

By combining (4.7) and (4.10) the hip motor can be used to control the horizontal foot velocity when on the ground, thereby producing a desired body velocity. The second term in (4.10) is the change of the virtual leg angle ϕ resulting from the movement of the knee $\dot{\theta}_k$ from the piston. The hip motor is compensating for this motion to ensure there is no unwanted movement in the virtual leg ϕ axis.

Another important consideration for velocity control is placement of the feet, as during the stance phase the feet have to move with the ground and can end up moving quite far relative to the body at high speeds. To attempt to keep the time averaged position of the foot directly below the hip the heuristic from [11] was used. When the robot is moving forwards at velocity $\dot{x}$ the foot must touchdown

$$\Delta x = \dot{x}\frac{t_s}{2} \tag{4.11}$$

in front of the hip with t_s being the stance duration, and the stance duration of the previous step being used for the current t_s value. This is accomplished by swinging the leg forwards to the following virtual leg angle setpoint before touchdown.

$$\phi = -\frac{\pi}{2} + \arcsin(\frac{\dot{x}t_s}{2r}) \tag{4.12}$$

4.2.5 Leg State Machine

Control of a single leg is coordinated by a finite state machine shown in Figure 4-11 which transitions between discrete control action states based on sensed events similar to previous work [11, 17]. These states combine the individual components required for bounding such as the virtual leg control, hopping thrust control, velocity control, and foot placement. Each leg runs an independent finite state machine without any knowledge of the state the other leg is in. Most interesting is that even with this decoupled control approach the robot naturally settles into a bounding motion where the front and back legs are out of phase. It is unclear why, but it is most likely that the body serves as a dynamic coupling between the front and back legs, driving the system to a stable limit cycle.

Each state and transition is described in sequence starting with the rests state. State names and transitions are in bold with the transitions italicised.

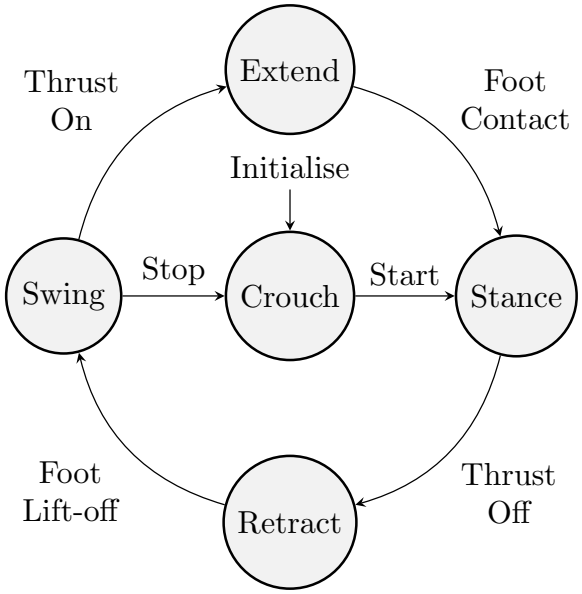

Figure 4-11: Finite state machine responsible for sequencing control actions of a single leg for bounding.

Crouch The robot starts up in a crouch state laying flat on the ground with knees down. Here control of the virtual leg is started to move the feet to directly below the hip in preparation for popping up into bounding.

Start The state machine transitions from the rest state to the stance state when the controller presses the start button. This transition happens immediately for the right leg, but left leg state machine waits for some initial body pitch in addition to the start button before transitioning. Leading with the right leg helps the robot to settle into the alternating limit cycle quicker, but is not crucial.

Stance During the stance state the knee piston is commanded to extend and push the foot against the ground. This is the thrust phase of vertical hopping control. Additionally, velocity control is done during this state as the foot is loaded and making good contact with the ground. The maximum motor velocity tracking gain of $5\,\mathrm{N\,m\,s/rad}$ is used to ensure the desired body velocity is tracked as closely as possible with large torques applied to correct $\dot{\phi}$ errors. Contact during this state is consistent so the height of the ground for estimating foot contact is also measured by looking at the body height and leg length. This is measured during each stance phase to accommodate varying terrain height.

Thrust Off Once the threshold specified in (4.6) is reached and the hip is on its

way to reaching the desired apex height, the piston can stop providing thrust and transition to the retract state.

Retract Here the piston thrust force is turned off and the retract chamber of the piston is quickly pulsed to contract the leg and lift the foot. The foot continues to move at the velocity commanded for velocity control as it is still partially making contact with the ground for much of this state and needs to 'follow through' until it has been properly lifted.

Foot Lift-off The foot is considered lifted and no longer making contact with the ground when it is above the ground height measured during the stance phase. This absolute foot height measurement for foot contact estimation is possible as the boom provides a body height measurement. For untethered legged robots this is often not available and alternative foot contact detection is necessary such as with sensors in the feet.

Swing With the foot no longer in contact with the ground the leg can be swung forwards in preparation for the next touchdown and stance phase. As with the other states the virtual leg is maintained throughout with the motor position and velocity tracking controller. The ϕ angle setpoint it is swung to is determined by the foot placement (4.12). Moderate position and velocity tracking gains, $100\,\mathrm{N\,m/rad}$ and $4\,\mathrm{N\,m\,s/rad}$ respectively, ensure the leg swings forward fast enough to get to the correct position in time, without flicking forward too violently.

Stop If the stop button is pressed the leg transitions back to the rest state, laying flat on the ground.

Thrust On This transition to start pushing with the piston is triggered by the thrust on threshold in (4.6). The hip is now falling back to the ground and a vertical thrust force needs to be applied to maintain vertical height control.

Extend In this state the piston thrust force is turned on and the leg is now extending to meet the ground with the extend chamber of the piston being pressurised. With the foot in the correct place for touchdown, velocity control is prematurely

activated. Even though the foot is not yet in contact with the ground, this ensure the foot matches the ground speed when it eventually does make contact.

Foot Contact Foot contact is detected with a large acceleration spike on the leg knee joint as the fast extending piston suddenly comes in contact with the stationary ground. This does not rely on calculating foot height and the assumption that the ground is level, allowing the robot to run over uneven terrain. It is however not the most robust detection method and a number of scenarios will cause a contact to not be detected, such as a spongy floor material, so an additional foot height check is used. Foot contact transitions the leg back to the stance state, and the state machine repeats producing bounding.

This remarkably simple control structure produces a surprisingly robust bounding motion at speed up to 1.7 m/s. Its robustness lies in its dependence on state rather than time for the sequencing of control actions. If a foot comes in contact with the ground earlier than expected this is handled, quite literally in its stride, as the controller is waiting for a ground contact rather than anticipating a particular swing duration for the leg.

4.3 Acceleration Controller

The design of the accelerate controller was largely motivated by the remarkable acceleration exhibited by cheetahs and greyhounds, and the desire to build a robot which could do the same. Only recently have legged robots begun to perform more dynamic and agile manoeuvres such as box jumps [23] and backflips [7], yet none have attempted accelerations into a top speed bounding gait.

Figure 4-12: Several frames of Kemba accelerating from rest into a 1.5 m/s bound. Frames show the full 0.5 s acceleration motion. From left to right; set pose, front foot popup, rear foot push-off, extended flight phase.

Most locomotion controllers fail with very large accelerations as the feet are improperly positioned and the bounding pattern breaks down [11]. This is also true for the bounding controller on Kemba, with accelerations needing to be limited to $0.5\,\mathrm{m/s^2}$ to stop it tripping and falling. Accelerating from rest presents an even more difficult case as the stable body pitching oscillation has not yet been initiated. With the following controller Kemba was able to accelerate from a crouched rest state into a $1.5\,\mathrm{m/s}$ bound in approximately $0.5\,\mathrm{s}$; an average forward acceleration 6 times that achievable with the bounding controller.

4.3.1 Approach

An accelerate from rest motion is not particularly useful if it cannot transition into a steady-state gait. The accelerate controller was therefore designed around the discrete states of the bounding controller to facilitate a coordinated gait transition. The trajectory optimisation results from Section 3.3.2 formed the basis of the motion and provided much of the insight required to design the controller. This insight can be summarised into the following:

- The rear hip is preemptively positioned for push-off
- The front leg pops the body up and loads the rear leg before push-off.
- There are three distinct phases to the motion

The first insight was that the rear hip is tucked right up against its limit, almost parallel with the body before the motion even begins. This preemptively places the back foot ahead of the hip in preparation for its backward movement when pushing off the ground and propelling the robot forwards. This is analogous to the foot placement required in the bounding controller to keep the stance averaged position of the foot below the hip. If the hip was not tucked in towards the body, then the back foot ends up too far behind the body. The next insight was that the front leg is used solely to pop the body up and load up the back leg. This was particularly interesting as it was very counter-intuitive. To get to a top speed as quickly as possible my expectation was that the robot should begin moving forwards immediately. This is however not the case with the front leg first pushing off the ground to pitch the body up, and then

retracting. This in fact even results in the body moving backwards slightly as it rocks onto the back leg. A similar behaviour is seen in greyhounds except it is typically executed from a standing pose. From standing the legs are suddenly lifted off the ground causing the body to pitch backwards and drop into a similar pose produced by the front leg pop-up [32]. Gravity is used to eccentrically load the rear legs, storing elastic potential energy in the muscles. Due to the currently limited position holding capabilities of the pneumatic actuators, Kemba cannot start in a standing pose. This is most likely the reason for the slightly different behaviour. The front leg pop-up does however, shift all of the body weight on the rear leg in preparation for the rear leg push-off phase, producing a similar result. This leads into the final insight which was that the motion can be broken into three distinct phases. This decomposition was crucial for aligning the accelerate controller with the structure of the bounding controller. We have termed these three phases set, pop-up, and push-off. They are shown in Figure (4-13). The set phase incorporates the preemptive tucking in of the

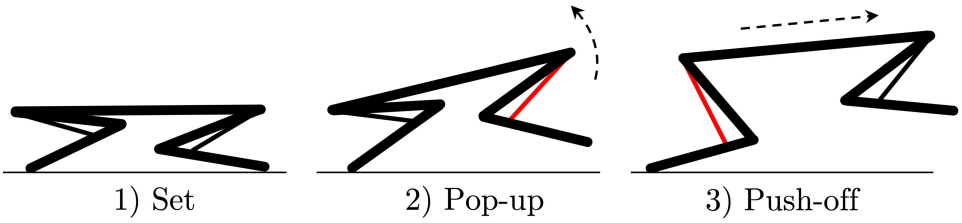

Figure 4-13: Discrete accelerate from rest motion phases.

rear leg. The pop-up phase uses the front leg to pitch the body back and load the rear leg, and the final push-off phase pushes with the back leg to accelerate the body to the desired forward speed. In total, the motion takes approximately 0.5 s with the phases executed in rapid succession. After the final phase the robot is effectively in the extended flight phase of bounding and can transition directly into steady-state bounding. With some minor modifications to only a few control actions and state transitions, the bounding state machines running on each of the legs can be used to produce the three phases of the accelerate motion. The crouch state can be use for the set phase, and the front and back leg stance phases can be used for the pop-up and push-off phases respectively. Integrating both motions into the same state machine not only simplified control, but also produces a seamless transition from acceleration into steady-state bounding that appears remarkably natural.

4.3.2 State Machine Modifications

The following are the changes made to the states and state transitions depicted in Figure 4-14 to enable the robot to rapidly accelerate from rest into stable bounding;

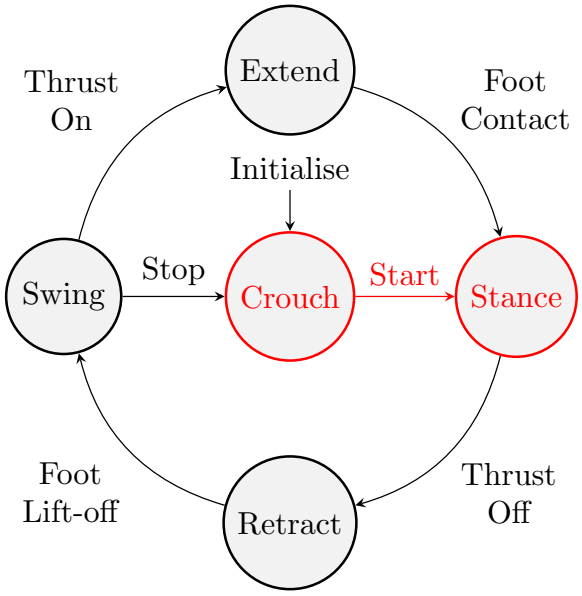

Figure 4-14: Modified bounding states and state transitions are shown in red.

Crouch The crouch state in the bounding controller becomes the set phase of the accelerate motion. Here, rather than keeping both feet directly under their respective hips, the back foot is preemptively moved forwards based on the desired final velocity. The amount it is moved forwards is determined by (4.12).

Start For the front leg, pressing of the start button still triggers the state transition. However, for the back leg, in addition to waiting for some initial body pitch it also only transitions once the body has reached its stationary pop-up apex where $\dot{\theta}_b \leq 0$. At this point the body is stationary and the full body weight is resting on the back leg. Triggering the transition too early results in the body not being high enough, and transitioning too late means the front hip has already begun falling back to the ground, often resulting in the front foot touching down far too early.

Stance During the unmodified stance phase a torque about the hip produces a foot velocity which aims to match the desired body velocity. This desired velocity is ignored for the first stance phase of the front leg, resulting in the leg just

pushing straight up irrespective of the desired velocity. This produces the front leg pop-up phase of the accelerate motion. For the back leg the stance state is left unmodified. The torque about the hip will accelerate the robot to match the commanded velocity, while the thrust from the piston will raise the back hip and pitch the body forwards. As a result of the crouch phase the foot is also already correctly placed ahead of the hip according to (4.12).

In addition to sequencing the accelerate from rest motion, these modifications do not interfere with the normal bounding state machine cycle, allowing the robot to seamlessly transition directly into its bounding gait. Unlike with bounding, the acceleration controller is asymmetrical requiring slightly different control actions for the legs depending on the direction of acceleration. Whether a leg is at the front or back is determined simply by looking at the direction of the commanded final velocity. This check enables the robot to accelerate in either direction based on the commanded velocity.

These few modifications to the bounding controller allow the robot to accelerate itself from rest directly into a maximum speed bounding gait. This motion also looks remarkably natural for a robot, considering the simplicity of the controller and minimal modifications required.

4.4 Results

4.4.1 Bounding

The bounding controller was tested at a range of commanded velocities in both directions. The velocity tracking performance is shown in Figure 4-15 and a video of the robot bounding can be found in Appendix D.

The robot is able to track velocity commands up to about 1.7 m/s. Any higher and the robot would typically trip and fall. Supply pressure also varied significantly during this 1.5 minute test, with the compressor turning on at about 0.6 MPa and turning off again at 0.8 MPa. This pressure variation did not noticeably affect velocity control. There is some slight variation in velocity around the commanded value due to the body bouncing back and forth as it alternates between feet, but similar variation

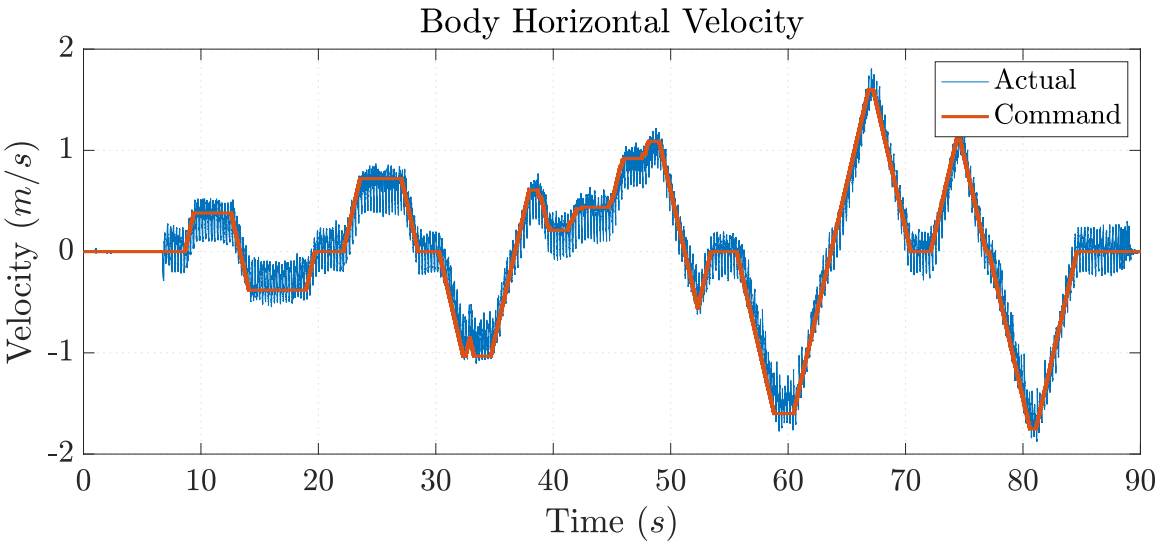

Figure 4-15: Velocity control of bounding at up to 1.7 m/s.

is also evident on much more sophisticated MPC bounding controllers [25]. The body pitching oscillation characteristic of bounding was also stable for a wide range of speeds, converging on a very clearly defined limit cycle shown in Figure 4-16. This

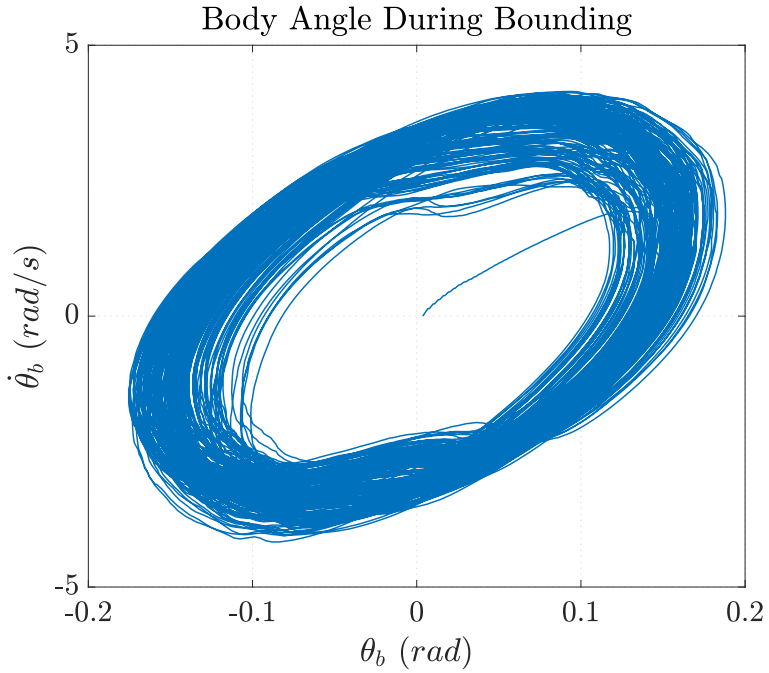

Figure 4-16: Body pitch limit cycle for a wide range of bounding speeds.

limit cycle is for the full test shown in Figure 4-15. The body oscillation is not specifically controlled and instead results from the body dynamics and bang-bang hip height control on each of the legs.

The controller state, hip velocities, hip torques, piston position, and piston com-

mands are shown in Figure 4-18 for 1 s of bounding at 1.5 m/s. Actuator command remain relatively consistent between gait cycles. A single gait cycle is approximately 0.3 s with stance times on the order of 0.1 s. The cyclic gait pattern becomes slightly asymmetrical with faster bounds due to the hip torques acting in opposing directions. The front hip torque acts with the natural body oscillation direction, while the back hip torque acts against it. Hip torques during stance phases remain well below 10 N m even with the motor velocity tracking gain at its maximum. Interestingly, repositioning of the right (rear) foot for touchdown during the swing phase produces the highest motor velocities and torques, with torques regularly hitting the motor's 25 N m limit. Lower swing phase position and velocity tracking gains would reduce this, but these would result in the foot not reaching the desired touchdown position causing the robot to trip and fall.

4.4.2 Limits to Faster Bounding

Interestingly, interrogation of the actuator data in Figure 4-18 suggests that the swing state for the rear leg during bounding appears to be the limiting factor for bounding speed. The hip actuator spends a significant portion of its time at its torque limit and the leg cannot be flicked forward any faster to position the foot correctly for the next touchdown. With a fixed stride duration, faster bounding speeds will result in larger foot displacements during stance requiring the foot to be swung forward even further in the same amount of time. This will not be possible if the actuator is at its limit and the foot will not be placed correctly. This is the reason for the robot tripping and falling for faster bounding speeds. Additionally, the back hip is also at its range of motion limit and cannot be brought any further forward during bounding as seen in Figure 4-17. This will also contribute to incorrect foot placement for faster bounding speeds and result in the robot falling. Higher vertical ground reaction forces are a potential solution as they will reduce the required stance time, shortening the distance the foot needs to swing back along and increasing the duration of the swing phase. Another solution is the addition of a flexible spine which bends along the plane of motion. This would behave like an additional actuator in series with the hip motor, providing additional speed and torque for the back leg repositioning during the swing phase.

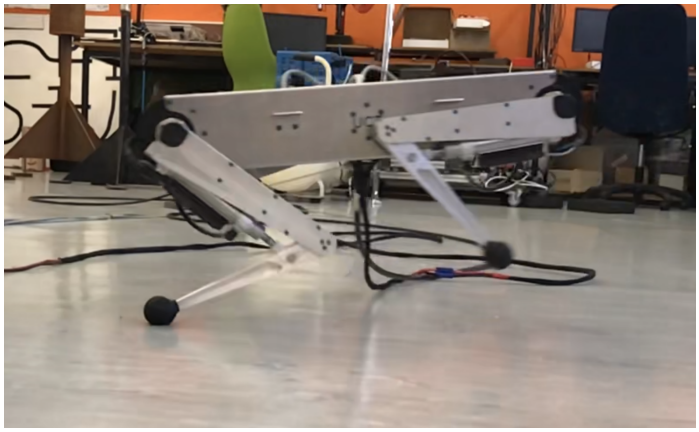

Figure 4-17: The back hip at its range of motion during fast bounding to the left.

4.4.3 Accelerating

The accelerate controller was tested on Kemba with a command to go from rest to 1.5 m/s. Figure 4-19 shows the body velocity, angle, and height during this acceleration manoeuvre, and a video of the robot accelerating can be found in Appendix D.

The commanded velocity is reached in about 0.5 s and a single stride from rear foot push-off. The transition into the steady-state bounding gait also happens smoothly with body angle and height stabilising in only a few strides. Figure 4-20 shows the controller and actuator data for the acceleration manoeuvre. Rear hip torque during push-off peaks at about 20 N m. This is near the actuator limit and will likely be the limiting factor for the acceleration manoeuvre. Average body acceleration is approximately $3\,\mathrm{m/s^2}$ for the whole motion, a 40 % improvement over the Mini Cheetah with its cMPC controller. During the rear leg push-off phase body forward acceleration remains consistently high at approximately $9\,\mathrm{m/s^2}$. This is close to the $10.2\,\mathrm{m/s^2}$ accelerate limit predicted by the nose-up pitch equation (1.1) with a spine length of 0.52 m and a nominal leg length of approximately 0.25 m. This is already very close to the theoretical limit for the rear foot push-off. Greater overall body accelerations could possibly be achieved in future work by modifying the controller to maintain a longer rear foot stance duration.

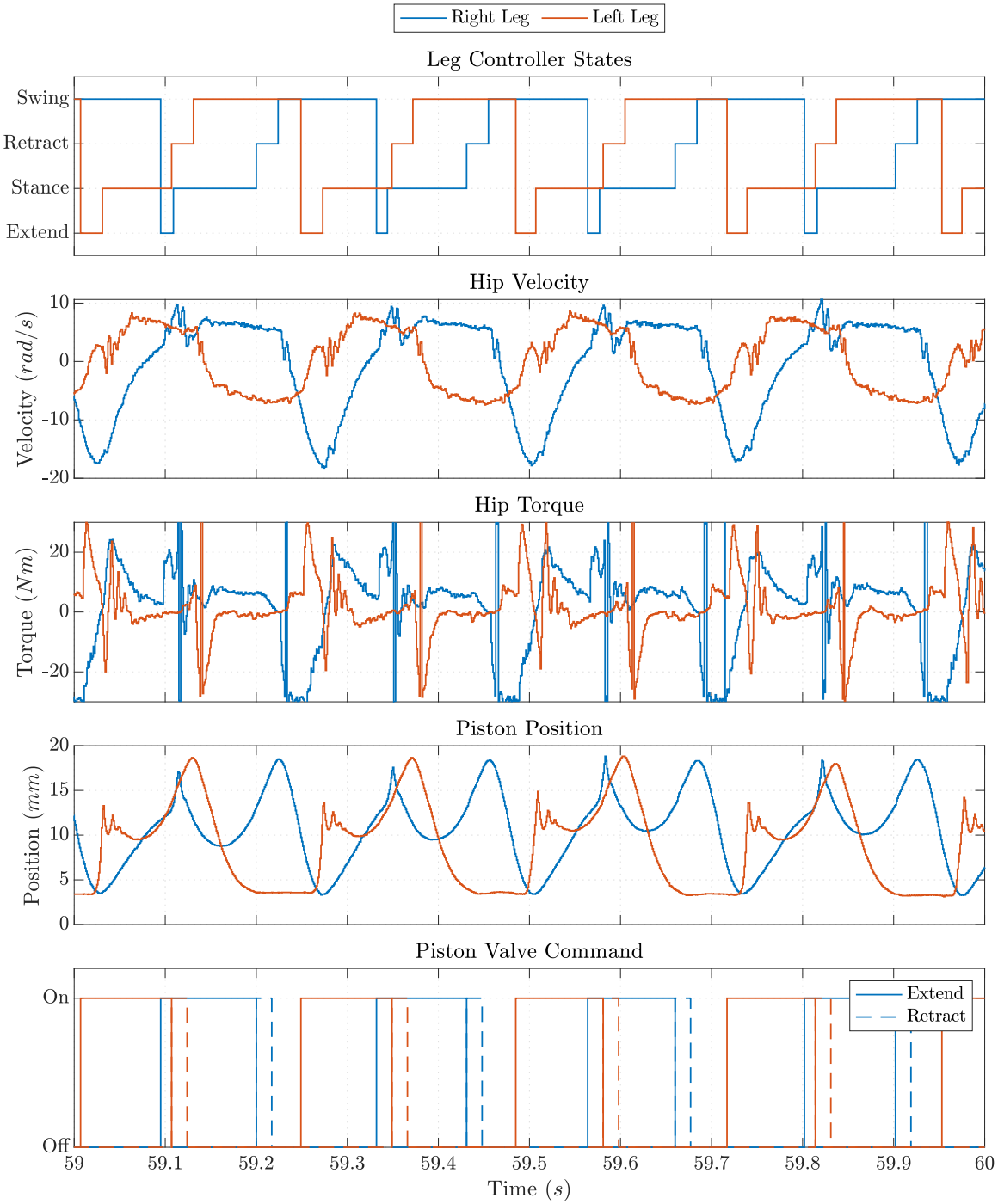

Figure 4-18: Controller and actuator data for 1 s of bounding to the left at 1.5 m/s.

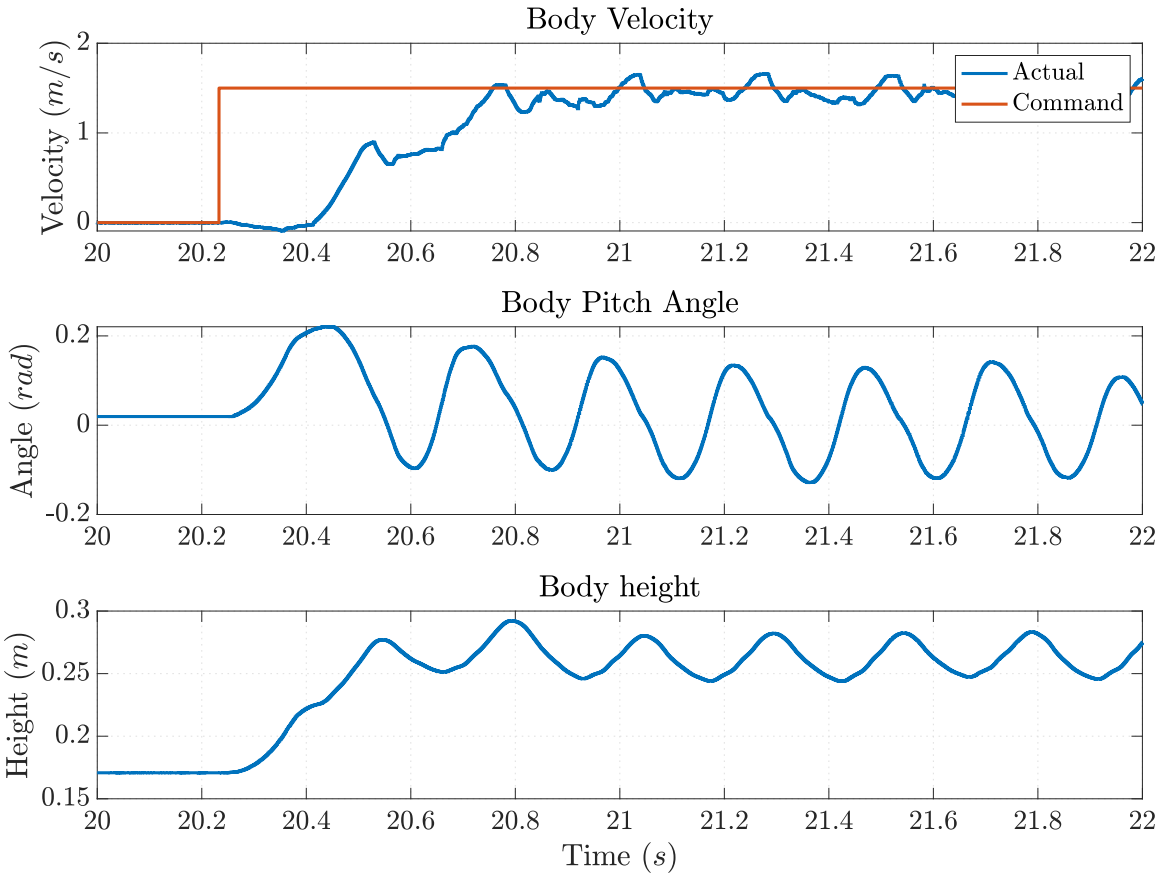

Figure 4-19: Body velocity, angle, and height for accelerating from rest into a 1.5 m/s bounding gait.

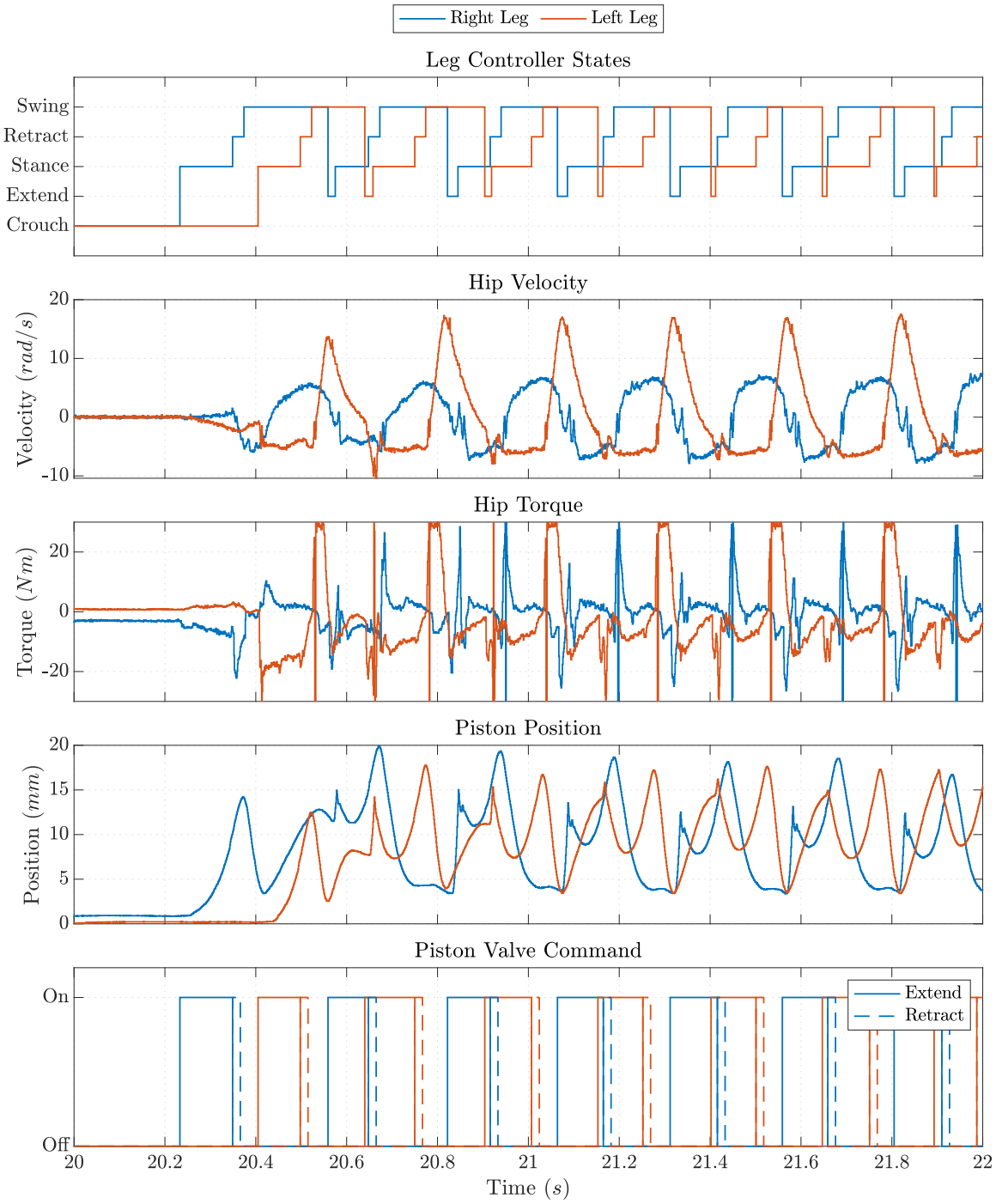

Figure 4-20: Controller and actuator data for accelerating from rest into a 1.5 m/s bounding gait.

Chapter 5

Conclusion and Recommendations

5.1 Conclusion

This thesis presents Kemba, a planar legged robot which combines high bandwidth proprioceptive actuators and pneumatic pistons and is capable of fast, agile, and dynamic manoeuvres.

To date, Kemba is able to jump 0.95 m high, holding the height record for two legged robots previously developed in the African Robotics Unit (ARU). This jump motion was made possible by integrating the pneumatic actuator dynamics directly into the optimisation formulation with a novel reduced-order model and characterisation procedure, enabling the use of a modern control tool with underutilised pneumatic actuators. The resulting open-loop jump trajectory was accurate enough that it could be replayed directly on the physical hardware, resulting in a height error of only 5 cm. This framework also enabled the synthesis of more complex bounding and crouched acceleration motions, as well as a somersault. In addition to demonstrating the agility of the platform, these motions crucially informed the design of the subsequent bounding and accelerating from rest controllers, showing three distinct motion phases for acceleration: *set*, *pop-up*, and *push-off*. Without this insight, designing the bounding and accelerate controller would not have been possible.

With its bounding and accelerate controller Kemba is able to accurately track speed commands and bound at a top speed of 1.7 m/s, approximately 3.3 body lengths per second. Kemba is also able to accelerate from rest directly into a 1.5 m/s bounding gait in 0.5 s, an acceleration 3 times greater than what is possible without the

acceleration controller. The acceleration manoeuvre also enables a 40% greater acceleration than the agile Mini Cheetah with its cMPC controller. Most importantly this is a motion which has been under-explored for legged robots, but is often exhibited in animals, and this thesis presents a first implementation on hardware. This work additionally demonstrates that when actuator dynamics are incorporated into control, dynamic and agile legged manoeuvres are not inhibited by the use of low bandwidth muscle-like pneumatic actuators.

Aside from these findings, this work has produced a capable, and robust planar legged robot platform with an actuator and control framework flexible enough to explore increasingly dynamic motions. It is the authors hope that this work presents a small contribution to producing more agile and capable legged robots.

5.2 Recommendations

To further explore the compliance and energy storage properties, the pneumatic actuators can be fitted with additional valves to control the venting of a chamber. This would allow a chamber to be blocked at a specific pressure forming a spring with variable stiffness, and store potential energy. This can be monitored with pressure sensors mounted between the valves and piston chamber to provide feedback on the piston's internal state.

Bounding could be significantly enhanced by the addition of a spine with a single revolute joint in the centre as described in 4.4.2. This would enable placement of the foot further ahead for faster bounds, and actuator torques and velocities from the hip and spine would act serially, resulting in greater effective hip torque and horizontal foot speed.

Finally, using the same approach presented in this thesis it would also be possible to investigate rapid stopping. The baseline bounding controller could be modified with additional states to perform a rapid stopping manoeuvre. This might provide insight into a motion often performed by animals and rarely explored in legged robots.

Appendix A

Leg Design Calculations

Tibia

The tibia only experiences bending loads as a result of the ball foot which cannot transfer torques. The maximum bending load occurs when the piston is actuating and the leg is pushing off of the ground. Forces can be assumed to be perpendicular for a worst-case loading scenario. When supplied with 0.8 MPa, the piston produces $F_p = 392\,\text{N}$ of force. The tibia is 234 mm long and the piston is mounted $L_1 = 48\,\text{mm}$ from the knee.

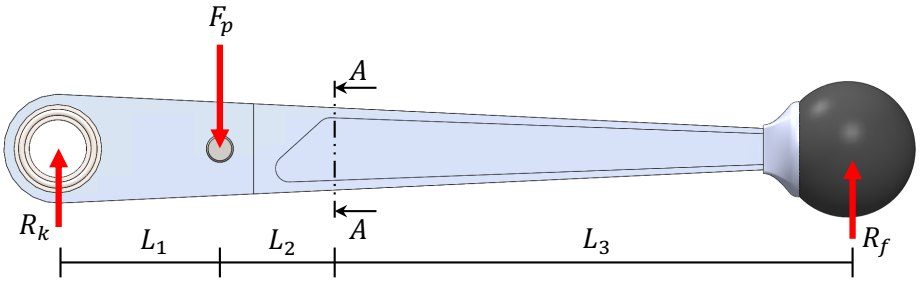

Figure A-1: Tibia loading diagram.

Taking moments about the knee, the ground reaction force at the foot R_f is:

$$\sum M = -F_p L_1 + R_f(L_1 + L_2 + L_3) = 0 \tag{A.1}$$

$$R_f = F_p \frac{L_1}{L_1 + L_2 + L_3} = 80.4\,\text{N} \tag{A.2}$$

The bending moment is highest and the cross section thinnest at section A-A, $L_3 =$

155 mm from the foot. At this section the internal bending moment is:

$$M = F_f L_3 = 80.4\,\text{N} \cdot 155\,\text{mm} = 12.5\,\text{N m} \tag{A.3}$$

Here the cross section is approximately a thin walled I-beam shown in Figure A-2. The wall thickness is $t = 2\,\text{mm}$, the web height is $H = 18\,\text{mm}$, and the flange width is $B = 19.5\,\text{mm}$.

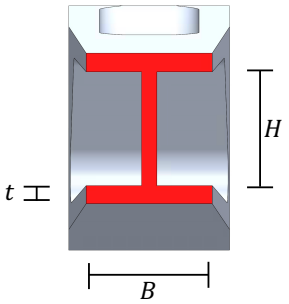

Figure A-2: Tibia I-beam section and dimensions.

The area moment of inertia for section A-A is:

$$I_{xx} = \frac{H^3 t}{12} + 2(\frac{t^3 B}{12} + \frac{tB(H+t)^2}{4}) = 8.80 \times 10^{-9}\,\text{m}^4 \tag{A.4}$$

Bending stress dominates over shear stress in this member. Tensile stress in the section is at its maximum in the bottom I-beam flange $y = 11\,\text{mm}$ from the neutral axis:

$$\sigma_{max} = \frac{My}{I_{xx}} = 14.3\,\text{MPa} \tag{A.5}$$

This is approximately an order of magnitude below the typical yield strength of 6061 aluminium billet.

Femur

The femur experiences primarily bending loads from the hip actuator torque, and torsion loads at the knee transverse friction forces at the foot, shown in Figure A-3. With maximum hip actuator torque the femur has an internal bending moment of $M_h = 24\,\text{N m}$. The structural part of the femur has an approximate hollow thin walled rectangle cross section along its length, shown in Figure A-4. This ignores the

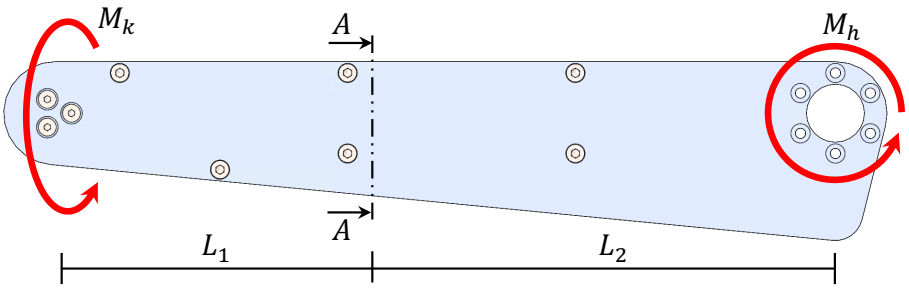

Figure A-3: Femur loading diagram.

additional thin walls for the piston. However, this is a worst case approximation to simplify calculation as these will provide additional strength and rigidity. The wall

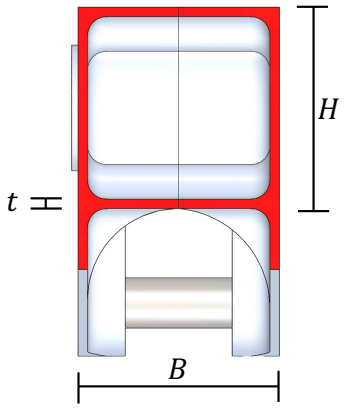

Figure A-4: Femur approximate square hollow section and dimensions.

thickness is $t = 1.5\,\text{mm}$, and the width and height are both $B = H = 32\,\text{mm}$.

The area moment of inertia for this section is approximately given by:

$$I_{xx} = \frac{BH^3}{12} - \frac{(B-2t)(H-2t)^3}{12} = 2.84 \times 10^{-8}\,\text{m}^4 \tag{A.6}$$

The maximum tensile stress occurs in the top or bottom walls 16 mm from the neutral axis, and is:

$$\sigma_b = \frac{M_h y}{I_{xx}} = 13.5\,\text{MPa} \tag{A.7}$$

The foot also experiences transverse loads as a result of the curved path constrained by the boom. This effectively pushes the foot sideways, twisting the knee and producing a torsion load in the femur. Assuming a best case friction coefficient of 1.0, and maximum foot normal force shown in (A.2), this transverse friction force will peak at 80.4 N. The tibia has a length of 234 mm resulting in a torsion load about the knee of $M_k = 18.8\,\text{N m}$. This torsion load produces the following average shear stress in

the thin walls [48]:

$$\tau = \frac{M_k}{2t(B-t)(H-t)} = 6.7\,\text{MPa} \tag{A.8}$$

Calculating the Von-Mises stress for the ductile 6061 aluminium is left out for brevity. The combined stresses resulting from these two loads is clearly also an order of magnitude below the typical yield strength of the 6061 aluminium billet from which the femur is machined.

Appendix B

Kinematics

Leg

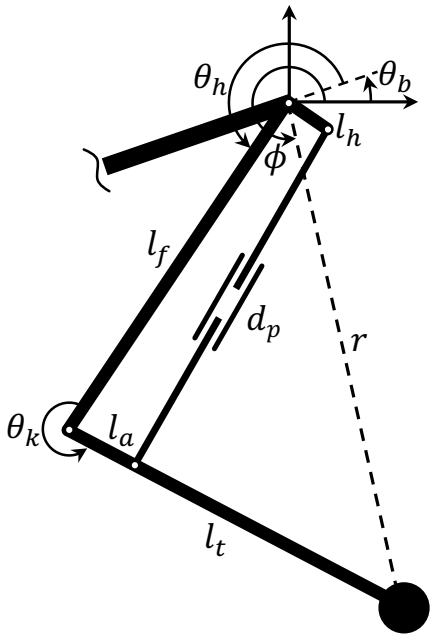

Figure B-1: Front leg kinematic diagram.

$$\theta_k = \arccos\left(\frac{d_p^2 - (l_h^2 + l_f^2) - l_a^2}{-2l_a\sqrt{l_h^2 + l_f^2}}\right) + \arctan\left(\frac{l_h}{l_f}\right) \tag{B.1}$$

$$r = \sqrt{l_f^2 + l_t^2 - 2l_f l_t \cos(\theta_k)} \tag{B.2}$$

$$\phi = \theta_h + \arcsin\left(\frac{l_t \sin(\theta_k)}{r}\right) + \theta_b \tag{B.3}$$

Dimension	Symbol	Value
Femur length	l_f	242 mm
Tibia length	l_t	234 mm
Piston attachment	l_a	48 mm
Hip offset length	l_h	31 mm
Piston length	d_p	206 mm to 276 mm

Table B.1: Leg dimensions.

Planarising Boom

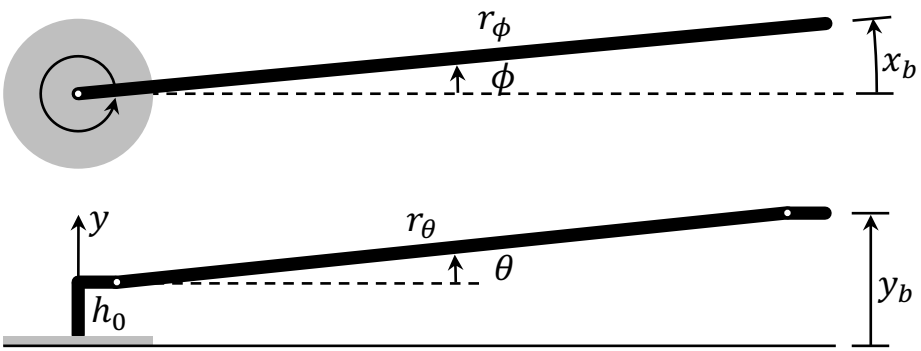

Figure B-2: Planarising boom kinematic diagram showing top view (Top) and side view (Bottom).

Dimension	Symbol	Value
Elevation angle	θ	-
Azimuth angle	ϕ	-
Radial length (ϕ)	r_ϕ	2575 mm
Radial length (θ)	r_θ	2493 mm
Height offset	h_0	101 mm

Table B.2: Boom dimensions.

$$x_b = r_\phi \cdot \phi \tag{B.4}$$

$$y_b = h_0 + r_\theta \cos(\theta) \tag{B.5}$$

Appendix C

Encoder State Estimator

This simple state observer is used to track states of encoder counts x and counts/s $\dot{x}$ with an adjustable bandwidth based on discrete encoder count measurements u. This state estimator is based on the implementation in the open source ODrive motor controller. The relationship between the states in the estimator is described by:

$$x_n = x_{n-1} + \Delta t \cdot \dot{x}_{n-1} \tag{C.1}$$

The observer error is calculated from the measured encoder count and the estimated encoder count. The floor function is used to match the continuous state to the discrete encoder counts.

$$e = u_n - \text{floor}(x_n) \tag{C.2}$$

The observer error is used to adjust the estimator state with a PI loop, where the error modifies the position and velocity state.

$$x_n = x_n + \Delta t \cdot Kp \cdot e \tag{C.3}$$

$$\dot{x}_n = \dot{x}_{n-1} + \Delta t \cdot Ki \cdot e \tag{C.4}$$

The gains for the PI loop can be computed using pole placement and the desired system response. First the controller is written in the Linear Time-Invariant (LTI) form.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{C.5}$$

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} -Kp & 1 \\ -Ki & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} Kp \\ Ki \end{bmatrix} \begin{bmatrix} u \\ 0 \end{bmatrix} \tag{C.6}$$

By looking at the eigenvalues of the **A** matrix we can place the poles and determine the natural response of the system.

$$poles = eig(\mathbf{A}) = \frac{-Kp}{2} \pm \frac{\sqrt{Kp^2 - 4Ki}}{2} i \tag{C.7}$$

For a critically damped response:

$$Ki = \frac{Kp^2}{4} \tag{C.8}$$

The proportional gain is set from the desired bandwidth ω_B of the PI loop.

$$Kp = 2\omega_B \tag{C.9}$$

The bandwidth is in units encoder counts/s. Higher bandwidth will improve tracking performance reducing phase lag, but will also result in a noisier velocity estimate depending on the encoder resolution.

Appendix D

Videos

Jump: https://youtu.be/nL0rYawcLIk

Accelerating: https://youtu.be/KZ1mVP8R52g

Bounding: https://youtu.be/u2Hn26uojoM

Trajectory Optimisation: https://youtube.com/playlist?list=PLHuPRft3DfC2witroNBBU2YTugbpRklmP

Appendix E

Code

Trajectory Optimisation Framework

https://github.com/chrismailer/kemba-trajectory-optimisation

Kemba Simulink Code

Controller: https://github.com/African-Robotics-Unit/kemba-controller

Simulation: https://github.com/African-Robotics-Unit/kemba-simulation

Peripheral Systems

Boom Firmware: https://github.com/African-Robotics-Unit/boom-firmware

Shunt Regulator: https://github.com/chrismailer/shunt-regulator

Bibliography

[1] H. L. More and J. M. Donelan, "Scaling of sensorimotor delays in terrestrial mammals," *Proceedings of the Royal Society B: Biological Sciences*, vol. 285, no. 1885, 2018.

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

[3] "Boston Dynamics." [Online]. Available: https://www.bostondynamics.com

[4] 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.

[5] C. Semini, V. Barasuol, T. Boaventura, M. Frigerio, M. Focchi, D. G. Caldwell, and J. Buchli, "Towards versatile legged robots through active impedance control," *International Journal of Robotics Research*, vol. 34, no. 7, pp. 1003–1020, 6 2015.

[6] G. Bledt, M. J. Powell, Katz Benjamin, J. Di Carlo, P. M. Wensing, and S. Kim, "MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2245–2252.

[7] B. Katz, J. Di Carlo, and S. Kim, "Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6295–6301.

[8] P. M. 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, 6 2017.

[9] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "ANYmal - A highly mobile and dynamic quadrupedal robot," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-November, 2016, pp. 38–44.

[10] M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, C. D. Remy, and R. Siegwart, "StarlETH: A compliant quadrupedal robot for fast, efficient, and versatile locomotion," in *Adaptive Mobile Robotics*. World Scientific, 2012, pp. 483–490.

[11] M. Raibert, *Legged Robots that Balance*. MIT Press, 1986.

[12] R. Playter and M. Raibert, "Control Of A Biped Somersault In 3D," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1992, pp. 582–589.

[13] Y. Tassa, T. Wu, J. Movellan, and E. Todorov, "Modeling and identification of pneumatic actuators," in *2013 IEEE International Conference on Mechatronics and Automation*. IEEE, 2013, pp. 437–443.

[14] H. Qi, G. M. Bone, and Y. Zhang, "Position Control of Pneumatic Actuators Using Three-Mode Discrete-Valued Model Predictive Control," *Actuators*, vol. 8, no. 3, p. 56, 7 2019.

[15] K. N. Murphy and M. H. Raibert, "Trotting and Bounding in a Planar Two-legged Model," in *Theory and Practice of Robots and Manipulators*. Springer, 1985, pp. 411–420.

[16] 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," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2015-June, no. June, 2015, pp. 5113–5120.

[17] H. W. Park, P. M. 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, 2 2017.

[18] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 5 2006.

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

[20] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and Trajectory Optimization for Legged Systems Through Phase-Based End-Effector Parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 7 2018.

[21] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review*, vol. 59, no. 4, pp. 849–904, 2017.

[22] L. Raw, C. Fisher, and A. Patel, "Effects of Limb Morphology on Transient Locomotion in Quadruped Robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 3349–3356.

[23] Q. Nguyen, M. J. Powell, B. Katz, J. D. Carlo, and S. Kim, "Optimized Jumping on the MIT Cheetah 3 Robot," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7448–7454.

[24] M. Chignoli, D. Kim, E. Stanger-Jones, and S. Kim, "The MIT Humanoid Robot: Design, Motion Planning, and Control for Acrobatic Behaviors," in *IEEE-RAS International Conference on Humanoid Robots*, vol. 2021-July. IEEE Computer Society, 2021, pp. 1–8.

[25] J. Di 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)*. IEEE, 2018, pp. 1–9.

[26] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly Dynamic Quadruped Locomotion via Whole-Body Impulse Control and Model Predictive Control," *arXiv*, vol. cs.RO, 9 2019. [Online]. Available: http://arxiv.org/abs/1909.06586

[27] E. Ackerman, "How Boston Dynamics Taught Its Robots to Dance," 1 2021. [Online]. Available: https://spectrum.ieee.org/automaton/robotics/humanoids/how-boston-dynamics-taught-its-robots-to-dance

[28] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.

[29] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, vol. 5, no. 47, 10 2020.

[30] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement Learning for Robust Parameterized Locomotion Control of Bipedal Robots," *arXiv*, 3 2021. [Online]. Available: http://arxiv.org/abs/2103.14295

[31] R. M. Walter and D. R. Carrier, "Rapid acceleration in dogs: Ground forces and body posture dynamics," *Journal of Experimental Biology*, vol. 212, no. 12, pp. 1930–1939, 6 2009.

[32] T. C. Angle, R. L. Gillette, and W. H. Weimar, "Kinematic analysis of maximal movement initiation in Greyhounds," *Australian Veterinary Journal*, vol. 90, no. 3, pp. 60–68, 3 2012.

[33] C. Fisher, C. Hubicki, and A. Patel, "Do Intermediate Gaits Matter When Rapidly Accelerating?" *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3418–3424, 10 2019.

[34] S. B. Williams, H. Tan, J. R. Usherwood, and A. M. Wilson, "Pitch then power: Limitations to acceleration in quadrupeds," *Biology Letters*, vol. 5, no. 5, pp. 610–613, 10 2009.

[35] N. F. Steenkamp and A. Patel, "Minimum time sprinting from rest in a planar quadruped," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2016-November. Institute of Electrical and Electronics Engineers Inc., 11 2016, pp. 3866–3871.

[36] C. Fisher, A. Blom, and A. Patel, "Baleka: A Bipedal Robot for Studying Rapid Maneuverability," *Frontiers in Mechanical Engineering*, vol. 6, 7 2020.

[37] C. Fisher, J. Van Zyl, R. Govender, and A. Patel, "Optimization-Inspired Controller Design for Transient Legged Locomotion," *2019 International Conference on Robotics and Automation (ICRA)*, 2021.

[38] C. Hennick, "Build It. Break It. Fix It." 8 2021. [Online]. Available: https://www.bostondynamics.com/resources/blog/build-it-break-it-fix-it

[39] A. Blom, "Design of a Bipedal Robot for Rapid Acceleration and Braking Manoeuvres," Master's thesis, University of Cape Town, Cape Town, August 2019.

[40] J. Van Zyl, "Rapid Acceleration of Legged Robots - A Pneumatic Approach," Master's thesis, University of Cape Town, Cape Town, February 2021.

[41] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 3 1960.

[42] R. Lorenz and K. Van Patten, "High-resolution velocity estimation for all-digital, AC servo drives," *IEEE Transactions on Industry Applications*, vol. 27, no. 4, pp. 701–705, 1991.

[43] 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, pp. 5331–5338, 2020.

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

[45] Z. Manchester and S. Kuindersma, "Variational Contact-Implicit Trajectory Optimization," in *Robotics Research*. Springer, 2020, pp. 985–1000.

[46] W. E. Hart, C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, and J. D. Siirola, *Pyomo-optimization modeling in python*. Springer, 2017, vol. 67.

[47] J. Di Carlo, "Software and Control Design for the MIT Cheetah Quadruped Robots," Master's thesis, Massachusetts Institute of Technology, Cambridge, January 2020.

[48] W. C. Young, R. G. Budynas, and A. M. Sadegh, *Roark's formulas for stress and strain*. McGraw-Hill Education, 2012.