

Robotic Tails for Enhanced Mobile Robot Performance

A thesis submitted to the University of Manchester for the degree of
Doctor of Philosophy
in the Faculty of Science and Engineering

2022

Damian Crosby
Department of Mechanical, Aerospace and Civil Engineering

Contents

Contents	2
List of figures	5
List of tables	10
List of publications	11
List of abbreviations	12
Abstract	14
Declaration of originality	15
Copyright statement	16
Acknowledgements	17
1 Introduction	18
1.1 Research Concepts	19
1.1.1 Mobile Robot Stability	19
1.1.2 Studies of Tails in Animals	24
1.2 Aims and Objectives	25
1.3 Thesis Organisation	26
2 A Literature Review of Terrestrial Robots with Robotic Tails and Their Functions	27
2.1 Introduction	28
2.2 Literature Review Methodology	28
2.3 Tail Functions of Terrestrial Robots	29
2.3.1 Categorisation System	29
2.3.2 Examples from each Category	33
2.4 Discussion	45
2.4.1 Performance Comparisons in the Selected Publications	45
2.4.2 Bio-Inspiration	46
2.4.3 Use of Multiple Segment Tails	47
2.5 Conclusion	47

3 A Novel Triad Twisted String Actuator for Controlling a Two Degrees of Freedom Joint: Design and Experimental Validation	49
3.1 Introduction	50
3.1.1 Twisted String Actuator	51
3.1.2 Antagonistic Triad	52
3.2 Control System	54
3.2.1 Actuated Universal Joint Position PID Controller with Acceleration Feedforward	54
3.2.2 Inverse Dynamics	54
3.2.3 Twisted String Actuator Force Optimisation Algorithm	54
3.2.4 Twisted String Actuator Force Proportional Controller	55
3.3 Simulation & Experimental Results	57
3.3.1 Experimental Setup	57
3.3.2 Windup & Tracking States	57
3.3.3 Results	57
3.4 Performance Comparison	58
3.4.1 Twisted String Actuator	58
3.4.2 Leadscrew	59
3.4.3 Direct Drive	59
3.4.4 Comparison	60
3.5 Conclusion	60
4 Optimisation Study for Multi-Segment Tails for COM Control	64
4.1 Introduction	65
4.1.1 Existing Study for Other Tail Function	65
4.2 Mathematical Definition of the Model	66
4.3 Inverse Kinetics	66
4.4 Inverse Kinematic Algorithms	67
4.4.1 Damped Least Squares	67
4.4.2 Weighted Quadratic Programming	67
4.5 Simulation Setup	68
4.5.1 Model Parameters	69
4.5.2 COM Trajectory	69
4.6 Results	69
4.7 Conclusion and Discussion	70
5 Creating a Configurable Payload for Instability Experiments	72
5.1 Introduction	73
5.2 Mathematical Design of the Configurable Payload	73
5.2.1 Mass and COM Configuration Space	73
5.2.2 Mass Limitation	74
5.2.3 Test Point Sets	74
5.2.4 Search Methods for Finding the Nearest Configuration to a Target	76

5.3	Implementation of the Configurable Payload	78
5.3.1	Container Design	78
5.3.2	Selected Search Method on the Configuration Space for the Cube Set and Balanced Set	79
5.3.3	Material Selection	80
5.4	Resulting Mass and COM Configurations of the Physical Payload	81
5.4.1	Configuration Space	81
5.4.2	Test Point Results	81
5.5	Discussion and Conclusion	87
5.5.1	Future Work	87
5.5.2	Conclusion	89
6	Investigating the use of a 2DOF Pendulum Tail for Compensating for Insta- bility when Carrying a Payload	90
6.1	Introduction	91
6.2	Measuring the COP using Four Load Cells	91
6.2.1	Theoretical Center of Mass (COM) Derivation	93
6.3	Test Sequence	94
6.4	Simscape Multibody™ Simulation	94
6.4.1	Simulation Construction	95
6.4.2	Simulation Control System	96
6.5	Experimental Apparatus	98
6.5.1	Base	99
6.5.2	Robot Arm	100
6.5.3	Robot Tail	100
6.5.4	Control System	101
6.6	Results	102
6.6.1	Minimum COP Deviation	103
6.6.2	Cumulative COP Deviation 2-Norm	104
6.7	Discussion	105
6.7.1	Payload and Tail Angle Regression Analysis	105
6.7.2	Vibration Analysis	106
6.8	Conclusion	107
References		108
Appendices		115

List of figures

1.1	Boston Dynamics Spot® working with the Royal Air Force. <i>Public Domain</i>	18
1.2	Boston Dynamics Handle loading a pallet. © <i>Boston Dynamics</i>	19
1.3	Examples of loss of stability in bipedal and quadrupedal robots. <i>Left Image: Public Domain, Right Image: © Google Robotics</i>	20
1.4	Examples of loss of stability (static or dynamic) in a human and forklift truck.	20
1.5	2D representation of the static stability of a legged robot, with the support region defined by the contact points of the legs with the ground. Notice how the orientation of the robot with respect to the gravity axis can move it in and out of the static stability region.	21
1.6	2D representation of the ZMP of a legged robot under horizontal acceleration, with the support region defined by the contact points of the legs with the ground. Notice how increasing the horizontal acceleration of the robot can make it dynamically unstable.	23
1.7	2D representation of the static stability of a legged robot picking up a payload, with the support region defined by the contact points of the arm and legs with the ground. Notice how the payload acts as a contact point until it is lifted off the ground, preventing loss of stability even as the COM is translated due to the mass of the payload.	23
1.8	2D representation of the static stability of a legged robot with a weighted tail picking up a payload. By changing the angle of the tail, the position of its COM can be adjusted in order to compensate for the payload.	24
1.9	Charts from [8] showing how the cat's tail is used to maintain balance on the beam when it is shifted, and how impairing it causes a major loss of stability. Dark bars are a 2.5 cm displacement, cross-hatched bars are a 5 cm displacement.	25
1.10	Image from [9], showing the body angle of a lizard during a jump, before and after tail removal. Section A shows still frames of the lizard at various stages when jumping, with the spinal column highlighted. Section B shows the resulting body angle at these snapshots.	25
2.1	PRISMA Flowchart of the publication selection process.	29
2.2	Abstract graphs of the different functions of the tail, with the magnitude representing some kind of change in the robot's position and/or orientation.	31
2.3	Tree diagram of all the categorisations found in the publications with accompanying visual diagrams.	32

2.4	Image from [13] showing how the tail moves during the gait in order to correct for heading drift.	33
2.5	Data from [13] showing the effects of a static and dynamic tail on maintaining robot heading in a trotting gait.	34
2.6	Simulation Data from [14] showing the how the body pitch would be reduced when accelerating or braking with an active tail.	34
2.7	Images from [14] showing the robot performing a rapid acceleration test with the tail.	35
2.8	Experimental Data from [14] showing the maximum acceleration achieved with and without an active tail.	35
2.9	Images from [17] showing the robot performing a turn with the tail.	35
2.10	Experimental Data from [16] (left) and [17] (right) showing the maximum lateral acceleration achieved with and without an active tail. Note the right graph manages similar results to the left graph.	36
2.11	Images of the hexapodal and octopodal insect like robots in [18] (left) and [19] (right).	36
2.12	Data from [19] comparing the robot path on an XY plane for a 90° differential drive turn, open loop tail turn, and closed loop tail turn. Note the vastly increased sharpness of the turn when a tail is used.	37
2.13	Data from [18] showing the gain in yaw rate for various open loop tail trajectories and different frequencies and amplitudes.	37
2.14	Images from [19] showing the robot using its tail to make a turn.	37
2.15	Data from [20] showing the normalised change in COM for a static and dynamic walk.	38
2.16	Data from [20] showing the trajectory of the COM and ZMP for a static and dynamic walk.	38
2.17	Images from [21] showing how the tail or body deflect when hit by the wrecking ball, depending on if the tail is active.	39
2.18	Data from [21] showing the difference in hip displacement between a fixed and controlled tail.	39
2.19	Data from [22] showing the optimal spring constant for maximising jumping height.	40
2.20	Images from [22] showing the motion of the tail during a jump.	40
2.21	Data from [23] showing the increase in forward velocity and/or reduction in body pitch for the different tail lengths and masses. Note that a) and c) correspond to a 35° tail amplitude, while b) and d) are a 65° amplitude.	41
2.22	Images from [23] showing the motion of the body and tail over a single stride cycle.	42
2.23	Images from [24] showing the a single stride for the robot, and an image of the robot just before landing.	43
2.24	Diagrams from [26] outlining the two reorientation experiments.	44
2.25	Images from [26] showing the wall reorientation experiment.	44

2.26	Images from [26] showing the slope reorientation experiment.	44
2.27	Diagram from [26] showing the obstacle impact test.	45
2.28	Images from [26] showing the slope obstacle impact test without a tail, with a static tail, and with a dynamic tail with closed loop feedback.	45
2.29	Graphs of various publications that varied tail design or control, normalised to the best performing variation (= 100%) for each metric, for a unitless comparison.	46
2.30	Hierarchy of bio-inspiration with examples from the selected literature, from least to most influential on the research.	47
3.1	Twisted String Actuator (TSA) string length against motor angle with coefficients from table 3.1.	52
3.2	Kinematic diagram of the antagonistic triad, where the universal joint rotation is defined by $\theta_{1,2}$ on the y and x axes respectively, and the actuator lengths are defined by $\lambda_{1,2,3}$ for the “top”, “left” and “right” actuators. r and $l_{1,2}$ define the anchor points of the actuators.	53
3.3	Surface plots of each element of the vector function $\Lambda(\theta)$, assuming coefficient values from table 3.1. Note that λ_2 and λ_3 are symmetric.	53
3.4	Block diagram of the complete experimental control system, excluding the hardware velocity controllers for the motors.	54
3.5	The initial startup of the mechanism showing the transition between “windup” and “tracking” states.	58
3.6	Plots of the setpoint and response for three different trajectories, one on only the roll axis (column 1), one on only the pitch axis (column 2), and one on both axes (column 3). Plots include Actuated Universal Joint (AUJ) orientation, forces at the top, left and right TSA, and the motor positions. . .	61
3.7	Performance comparison of the TSA configuration using coefficients from table 3.1 to various leadscrew configurations with different d_m and λ , and the direct drive, where $\mu = 0.1$ for the leadscrews.	62
3.8	Annotated photograph of the experimental model, with the roll θ_1 and pitch θ_2 axes marked.	63
4.1	Chart from [55] showing how the net quadruped rotation increases with the number of segments.	65
4.2	Diagram of a Two Dimensional (2D) tail, with all parameters annotated. . .	66
4.3	Diagram of the 1,2,3 and 4 segment tails, showing the gravity direction. . .	69
4.4	Gaussian membership function with $c = 5$, $\sigma = 1$	69
4.5	70
4.6	70
5.1	Concept drawing of the configurable payload, with swappable cubes that allow the COM to be varied relative to the geometric center.	73

5.2	Various temperature cooling profiles for simulated annealing, assuming 1000 steps.	78
5.3	Schematic of the container, including body and lid.	79
5.4	Labelled photograph of the final container design, along with the material cubes.	79
5.5	Simulated annealing output for the target $\begin{bmatrix} 1.5 & 0.001 & 0.001 & 0.001 \end{bmatrix}$ with $\alpha = 0.997$ and even weighting $w_m, w_r = 0.25$ for 4.3×10^4 steps.	80
5.6	Masses for each set of eight 35 mm cubes of the configurable payload for each of the four materials, with the mean mass \bar{x} and standard deviation σ of each set.	81
5.7	Scatter graph of all 62969 unique mass and COM vectors in the configuration space, each point represents a unique permutation.	81
5.8	Mass and COM coordinates for each test point set.	82
5.9	Example configuration space for the new configurable payload design, where $z = 0$	88
5.10	Renders of a concept design for the new configurable payload, with a single rod and dual rod configuration.	88
5.11	Diagram on the COM dynamics of a fluid filled container. If the container is disturbed, a standing wave forms inside the container which causes the COM to oscillate.	89
6.1	Diagram of a typical force plate setup.	91
6.2	Free body diagram of the static test setup.	93
6.3	Trajectory sequence of the robot arm and payload.	94
6.4	Graph of joint trajectories q_2 and q_3 during the test sequence (all other joints remain static), and the position of the grippers relative to the fully open position and the maximum closed position to grip the payload. Hatched area indicates the “carry” interval, where the tail will be compensating for the payload mass.	94
6.5	Simulink model of the simplified force plate using the Simscape Multi-body™ library.	95
6.6	Simulink block diagram of the robot arm using the Simscape Multibody™ library.	96
6.7	Simulink block diagram of the robot tail using the Simscape Multibody™ library.	97
6.8	Simulation block diagram.	97
6.9	Simulink block diagram of the control system using the Simscape Multi-body™ library.	98
6.10	3D preview of the dynamic model of the robot using the Simscape Multi-body™ library.	98
6.11	System diagram of the static rig.	99
6.12	Labelled photograph of the static rig.	99

6.13	Exploded schematic of the base of the static rig, with important components labelled and clamping points marked.	100
6.14	Visualisation of the bevel gear system. When both driving gears turn in the same direction, the driven gear rotates around the driving gears axis. When the driving gears turn in opposite directions, it rotates around its own axis.	101
6.15	Block diagram of the experimental control system.	102
6.16	Graph of the mean COP during the <i>carry</i> interval of the experiment.	103
6.17		105
6.18		106
6.19	Resonant frequency of the tail calculated from accelerometer data captured during an impact with a small steel hammer. The left graph is the accelerometer data, and the right graph is the real part from a DFT on the data.	107

List of tables

3.1	Model coefficients.	51
3.2	PID gains in the simulation and experiment.	57
5.1	The materials chosen for the cubes and their calculated density.	81
5.2	Table of the mass and COM vectors of \mathcal{E} , excluding \mathcal{E}_1	82
5.3	Table of the target and actual mass and COM vectors for \mathcal{C} , \mathcal{B} and \mathcal{E}_1 with the L2 norm error. * notation indicates “don’t care” and is excluded from the search algorithm.	83
5.4	Table of all test point configurations of the <i>Extrema Set</i> , showing the mass and COM and material permutation in matrix and 3D form.	84
5.5	Table of all test point configurations of the <i>Cube Set</i> , showing the mass and COM and material permutation in matrix and 3D form.	85
5.6	Table of all test point configurations of the <i>Balanced Set</i> , showing the mass and COM and material permutation in matrix and 3D form.	86
6.1	Table of the values of the coefficients from figure 6.2 used in the simulation and experimental apparatus.	95

List of publications

Publications go here.

List of abbreviations

AUJ Actuated Universal Joint

2D Two Dimensional

3D Three Dimensional

COM Center of Mass

DLS Damped Least Squares

DOF Degree of Freedom

PID Proportional Integral Derivative

PI Proportional Integral

PD Proportional Derivative

SHTP Sensor Hub Transport Protocol

MSB Most Significant Bit

LSB Least Significant Bit

I²C Inter-Integrated Circuit

SPI Serial Peripheral Interface

UART Universal Asynchronous Receiver-Transmitter

DIO Digital Input/Output

IMU Inertial Measurement Unit

GOOP Graphical Object Oriented Programming

CS Chip Select

DFT Discrete Fourier Transform

CPG Central Pattern Generator

ODE Ordinary Differential Equation

GRF Ground Reaction Force

ZMP Zero Moment Point

RMS Root Mean Square

COP Center of Pressure

ADC Analog to Digital Converter

NI National Instruments

PC Personal Computer

PLA Polylactic Acid

IMU Inertial Measurement Unit

PSU Power Supply Unit

TSA Twisted String Actuator

URDF Unified Robotics Description Format

RPM Rotations per Minute

DC Direct Current

AUJ Actuated Universal Joint

TSA Twisted String Actuator

DCC Direct Current Control

LVDT Linear Variable Differential Transformer

Abstract

This is abstract text.

Declaration of originality

I hereby confirm that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright statement

- i The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.
- ii Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made *only* in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.
- iii The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.
- iv Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see <http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=24420>), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see <http://www.library.manchester.ac.uk/about/regulations/>) and in The University’s policy on Presentation of Theses.

Acknowledgements

Acknowledgements go here.

Chapter 1

Introduction

In recent years, there has been a significant increase in the use of mobile robots for industrial, military and service applications, such as moving stock around in warehouses, inspecting equipment in hazardous or remote environments, performing search and rescue operations or acting as infantry support. With advances in battery technology, material fabrication and onboard processing power, mobile robots have been able to fill an increasing number of roles. Modern legged robots such as the *Boston Dynamics* Spot® pictured in figure 1.1 have allowed mobile robots into less rigidly controlled environments, able to navigate rough terrain and stairs. Some of these robots are also incorporating robot arms or other mechanisms that allow them to pick up, carry, and set down payloads. This can be useful for a number of reasons, such as collecting samples from a hazardous environment, clearing away obstructions that impede progress, moving products in a warehouse or providing vital supplies to trapped person. However, payload mass and COM offset is limited not just by the strength of the robot arm, but by the stability of the robot itself. An object that is too heavy, or that is highly unbalanced, will cause the robot to lose stability and fall over, in a similar fashion to a forklift truck or crane.

There are a number of ways this can be mitigated. The *Boston Dynamics* Handle pictured in figure 1.2 robot is a bipedal “wheel-leg” robot designed for warehouse tasks that uses a counterweight that can be swung around the robot body in order to maintain stability. This design, while reasonably effective, is unsuitable for most other mobile robot designs, which are closer to the ground and have four or more legs or wheels. It is also possible to use an adjustable mass on the robot body [1], however the range of compensation that can be pro-



Figure 1.1: Boston Dynamics Spot® working with the Royal Air Force. *Public Domain*



Figure 1.2: Boston Dynamics Handle loading a pallet. © *Boston Dynamics*

vided is necessarily limited by the size of the robot's torso.

One option that has yet to be explored is the use of a robotic tail, in a more traditional sense than the Handle, resembling something closer to what is found in the animal kingdom. Robotic tails of this kind have already been extensively studied for other areas of robot stability, as will be explored in chapter 2.

1.1 Research Concepts

The following section gives an overview of important concepts necessary to understand the issue of mobile robot stability when carrying a payload and the proposed solution.

1.1.1 Mobile Robot Stability

Stability is a significant issue for mobile robot design. Loss of stability can mean the robot is unable to move and must be reorientated or retrieved, which maybe difficult or impossible in some extreme environments, such as in outer space or a nuclear fuel pool. In the worst case it can result in severe damage or destruction of the robot, and any objects it is carrying. This has become more of an issue as mobile robots have become increasingly fast and agile, often running, jumping and hopping around less controlled environments.

In many ways the consequences of stability loss in mobile robots are analogous to other situations. A human that falls over has to pick themselves up before continuing on, or if they are infirm they may require assistance. Likewise they could also suffer injury, or if walking along the edge of a long drop, fall to cause severe injury or death. A forklift truck or other

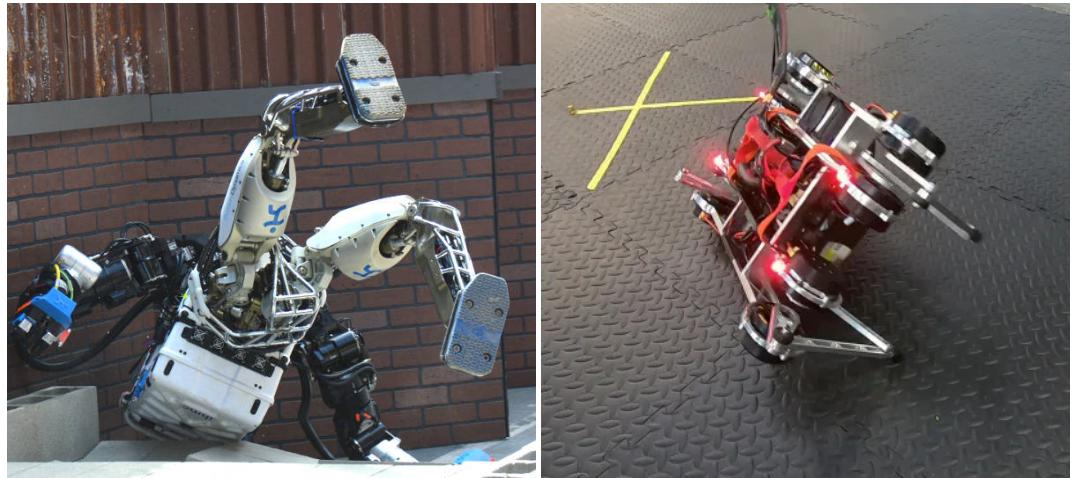


Figure 1.3: Examples of loss of stability in bipedal and quadrupedal robots. *Left Image: Public Domain, Right Image: © Google Robotics*

piece of heavy plant can topple, injuring the driver and causing the damage or destruction of vehicles and materials.

In general, stability from a biomechanical perspective can be divided into two different types, *static* and *dynamic*. While there are numerous ways to define the difference between the two, such as the maximum lyupanov exponent for dynamic stability [2], the following definitions will be used:

- **Static stability** only considers the uniform force of gravity and assumes no other forces are acting on the object.
- **Dynamic stability** considers other forces and torques on the object, both internal and external, as well as gravity.

A stationary object that has no external forces or torques being applied needs to be statically stable, a moving object, or an object that is having a force or torque applied to it other than gravity, needs to be dynamically stable.

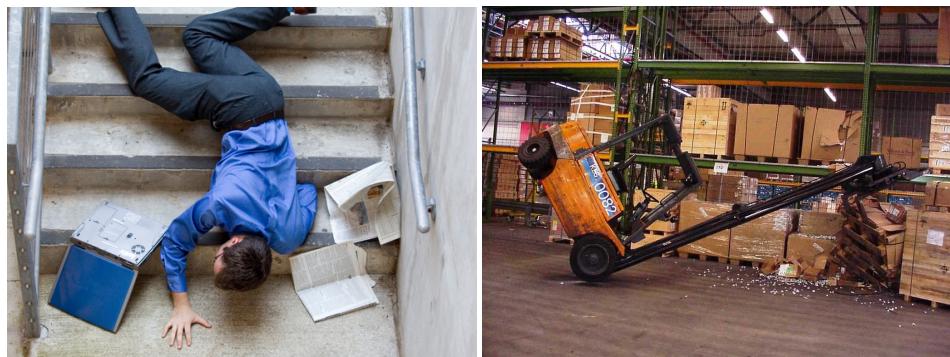


Figure 1.4: Examples of loss of stability (static or dynamic) in a human and forklift truck.

1.1.1.1 Static Stability

To determine if the robot is statically stable, the gravity axis projection of the COM, commonly referred to as the Center of Pressure (COP), needs to fall within a defined “support

“polygon” on the plane perpendicular to the gravity axis plane, as shown in figure 1.5¹. The COM can be calculated for using equation 1.1 for n bodies of masses $m_{1\dots n}$ and COM positions $\mathbf{p}_{1\dots n}$. If the gravity vector is parallel to any of the basis vectors, then the perpendicular components of the COM can be used to determine the static stability.

$$text{COM} = \frac{\sum_{i=1}^n m_i \mathbf{p}_i}{\sum_{i=1}^n m_i} \quad (1.1)$$

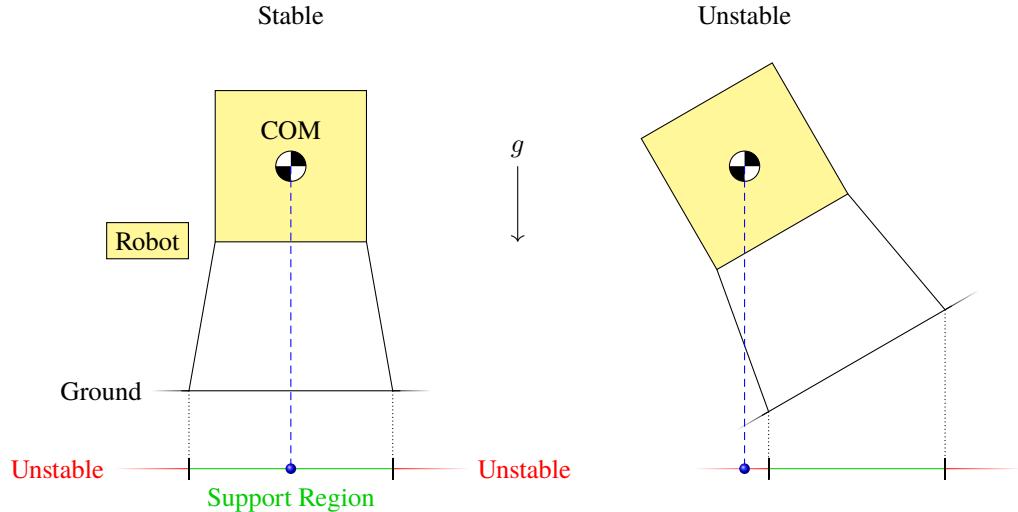


Figure 1.5: 2D representation of the static stability of a legged robot, with the support region defined by the contact points of the legs with the ground. Notice how the orientation of the robot with respect to the gravity axis can move it in and out of the static stability region.

Fundamentally, there are two different methods to maintain static stability:

- Change the region or shape of the support region so the gravity axis projection of the COM remains within the bounds of the support region.
- Move the gravity axis projection of the COM so it remains within the bounds of the support region.

The former method can be considered equivalent to using a walking pole to steady yourself on uneven ground when hiking, the leg of the pole acts as a new vertex that is used to calculate the support polygon, expanding it sufficiently, or placing your foot in front of you when walking. The latter method can be considered equivalent to leaning back to remain upright when falling forward. Leaning back moves the centre of gravity to keep it within the support polygon.

However, just because the COM falls outside of the support polygon does not mean that loss of stability is inevitable, as long as a force or torque applied to the body counteracts the forces and torques induced by the force of gravity in order to maintain stability. This is similar to applying a torque to your ankle when standing on one leg, or a strong wind keeping

¹Center of Mass and Center of Gravity are used interchangeably here. This assumes a uniform gravitational vector g , which is a suitable assumption for terrestrial robots due to the overwhelming dominance of the earth’s gravitational field. If there is a non-uniform gravitational field, then the Center of Mass and Center of Gravity will not be equivalent, and the terms can not be used interchangeably.

you upright when leaning forward. If this is the case, then *dynamic* stability is maintained while *static* stability is lost. If the force or torque is removed, then stability is lost. Conversely, forces and torques can also cause loss of stability even if the center of gravity does not fall outside of the support polygon. This is similar to being pushed over, or stopping too quickly and falling forward. So dynamic stability can maintain stability even if static stability is lost, but static stability cannot maintain stability if dynamic stability is lost.

1.1.1.2 Dynamic Stability

To determine if the robot is dynamically stable, it is not enough to only consider the position of the gravity axis projection of the COM, as it is for static stability. This is where the concept of the Zero Moment Point (ZMP) [3] is useful for mobile robots to check if dynamic stability will be maintained. It extends the calculation used for static stability by including inertial forces caused by accelerations of the bodies, as shown in figure 1.6. Though it has mostly been utilised for bipedal robots to ensure stability while walking, it is also applicable to quadruped robots [4]–[6], and has even been investigated for the development of a stability warning system in road vehicles [7]. The ZMP is formally defined as the point at which the point where the total of horizontal inertia and gravity forces equals zero. It can be thought of as a *dynamically augmented* version of the gravity axis projection of the COM. Equation 1.2 defines the position of the ZMP for a robot or vehicle with n bodies of masses $m_{1\dots n}$, COM positions $\mathbf{p}_{1\dots n}$ and COM accelerations $\ddot{\mathbf{p}}_{1\dots n}$, in contact with a planar surface of normal vector \mathbf{n} (ZMP cannot be calculated for non-planar surfaces). $\tau_{i\dots n}$ defines the torque acting on each COM, which can be calculated from .

$$\begin{aligned}\tau_i &= \mathbf{R}_i \left(\mathbf{I}_i \ddot{\theta}_i - (\mathbf{I}_i \dot{\theta}_i) \times \dot{\theta}_i \right) \\ \text{ZMP} &= \frac{\mathbf{n} \times \sum_{i=1}^n (\mathbf{p}_i \times m_i \mathbf{g} - \mathbf{p}_i \times m_i \ddot{\mathbf{p}}_i - \tau_i)}{\mathbf{n} \cdot ((\sum_{i=1}^n m_i \mathbf{g}) - (\sum_{i=1}^n m_i \ddot{\mathbf{p}}_i))}\end{aligned}\quad (1.2)$$

1.1.1.3 Addition of a Payload

When a payload is added to the robot, it is equivalent to instantaneously adding an extra body to the robot of mass m_p and position p_p thus changing the COM. Initially the payload will also create an extra contact point with the ground, changing the support polygon so the robot remains statically stable. However, as soon as contact between the payload and ground is severed, the robot can become statically unstable, as shown in figure 1.7. This does not mean the robot will immediately lose stability, the dynamic forces created picking up the payload may keep the robot dynamically stable, but once the robot is stationary without other forces acting upon it, it may lose stability. The best way to compensate for this is to change the position of the other bodies so the COM remains within the support polygon.

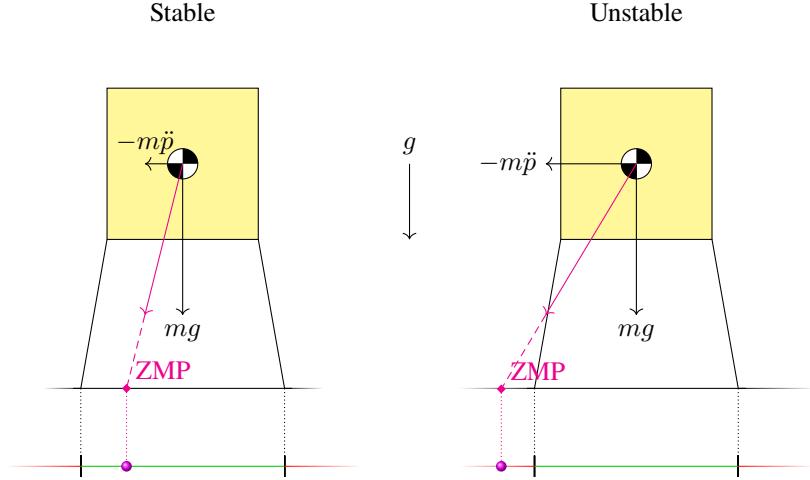


Figure 1.6: 2D representation of the ZMP of a legged robot under horizontal acceleration, with the support region defined by the contact points of the legs with the ground. Notice how increasing the horizontal acceleration of the robot can make it dynamically unstable.

This is akin to leaning back when carrying something heavy, but leaning generally has limited range, so can only compensate for lighter payloads.

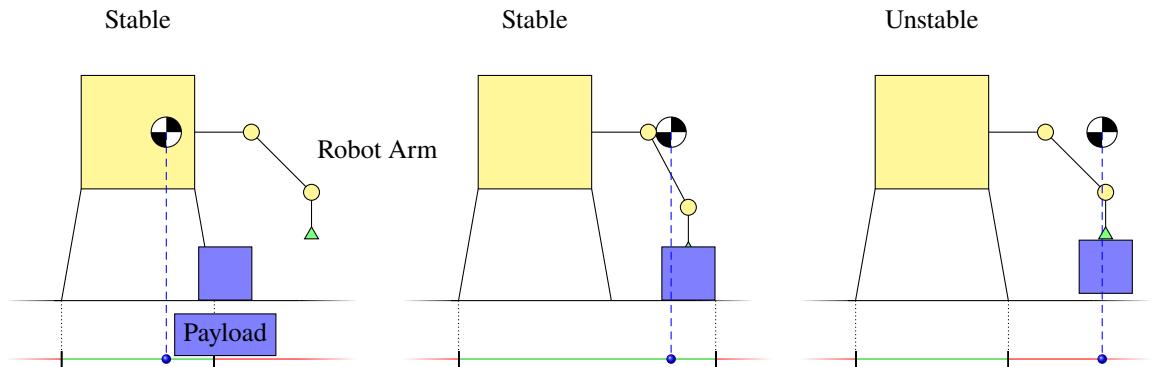


Figure 1.7: 2D representation of the static stability of a legged robot picking up a payload, with the support region defined by the contact points of the arm and legs with the ground. Notice how the payload acts as a contact point until it is lifted off the ground, preventing loss of stability even as the COM is translated due to the mass of the payload.

1.1.1.4 Use of a Tail for Maintaining Robot Stability with Addition of a Payload

One option to change the position of other bodies is to attach a “tail” to the robot at the opposite end of the robot arm. When the robot is not carrying a payload, the tail COM remains close to the body COM, and therefore has a smaller contribution to the overall COM. When the robot picks up the payload and the COM shifts forward, the tail can swing out as shown in figure 1.8, moving the COM further from the body which pulls the overall COM back towards the center of the body, maintaining the stability of the robot.

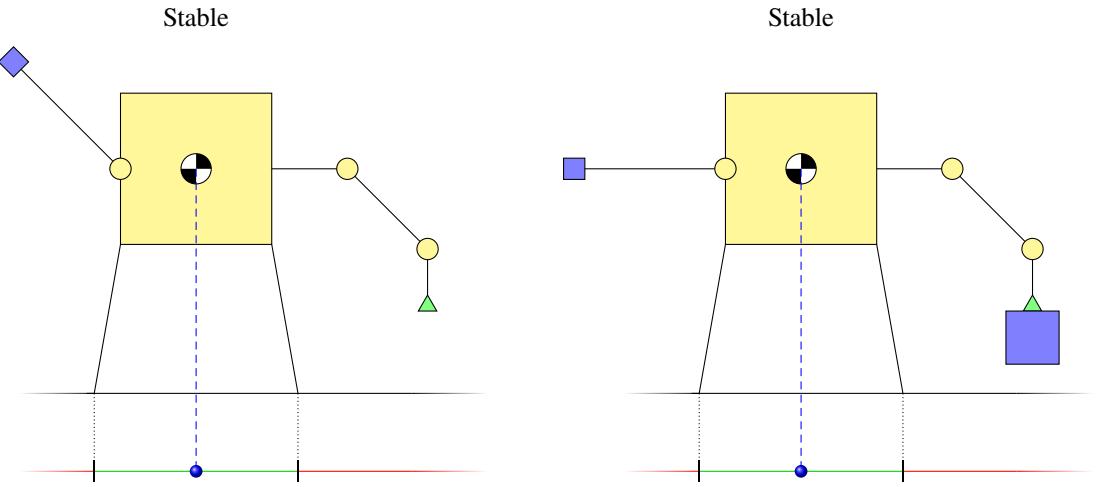


Figure 1.8: 2D representation of the static stability of a legged robot with a weighted tail picking up a payload. By changing the angle of the tail, the position of its COM can be adjusted in order to compensate for the payload.

1.1.2 Studies of Tails in Animals

Tails are a common sight in vertebrate animals, a natural extension of the spinal column. While some tails are used purely for grasping, communication or decoration, many have significant function in maintaining stability during locomotion. The following research studies demonstrate the importance of the tail for stability in two animals, by measuring their performance before and after surgical operations to disable the function or remove their tail.

1.1.2.1 Balancing Ability of a Cat Before and After Partial Tail Paralysis

In [8], experiments were conducted domestic cat uses its tail for balance when walking along a narrow beam, which was shifted laterally at a certain velocity by 2.5 cm or 5 cm while they are traversing it. Four cats were trained to walk across the beam, before and after a surgical procedure that severed the nerves in the spinal cord just above the tail, severely affecting its function by causing paralysis of the tail muscles. As can be seen from figure 1.9, this procedure caused the cats to fall from the beam far more often than before surgery.

1.1.2.2 Aerial Stability of a Jumping Lizard Before and After Tail Removal

In [9], the aerial stability of the arboreal lizard is examined, with an intact tail and with their tail removed. Lizards with the tail removed are unable to maintain their body orientation and do not land cleanly, as can bee seen in figure 1.10.

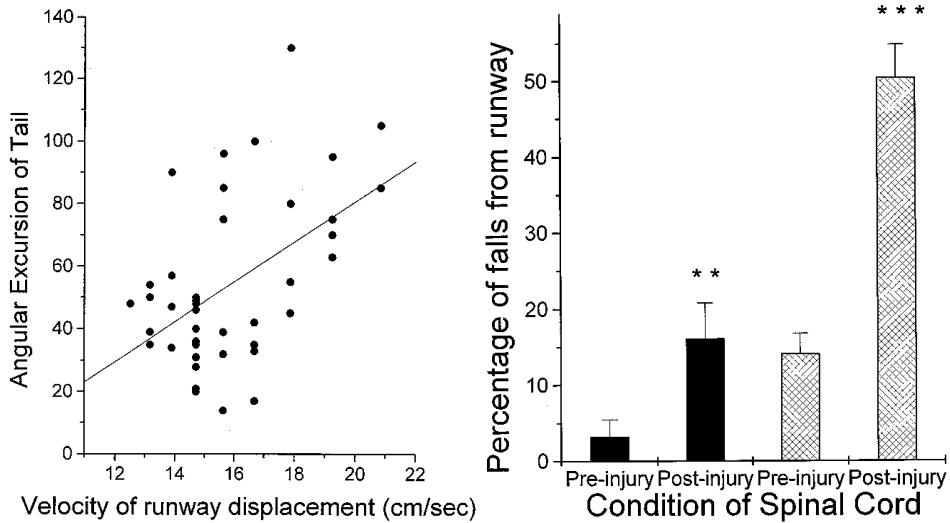


Figure 1.9: Charts from [8] showing how the cat's tail is used to maintain balance on the beam when it is shifted, and how impairing it causes a major loss of stability. Dark bars are a 2.5 cm displacement, cross-hatched bars are a 5 cm displacement.

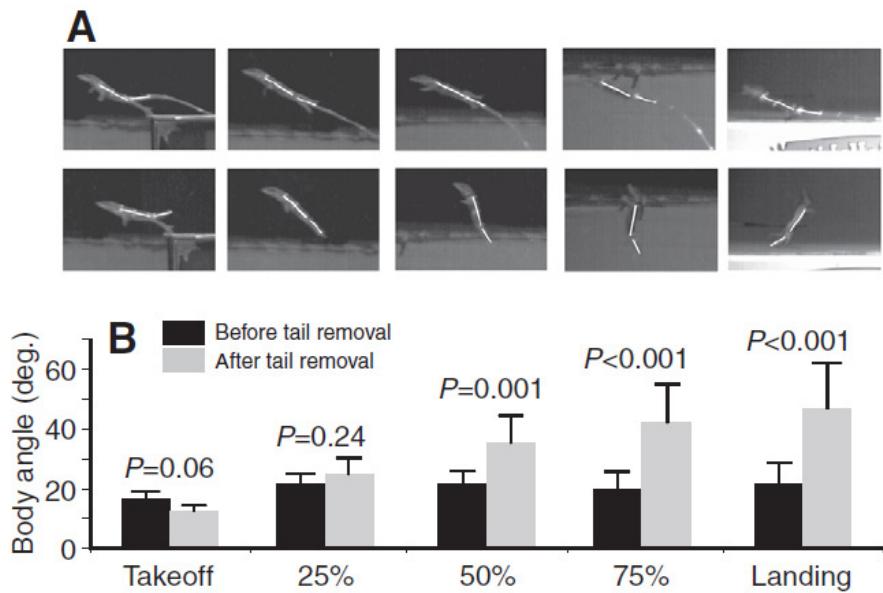


Figure 1.10: Image from [9], showing the body angle of a lizard during a jump, before and after tail removal. Section A shows still frames of the lizard at various stages when jumping, with the spinal column highlighted. Section B shows the resulting body angle at these snapshots.

1.2 Aims and Objectives

This research investigates the use of a robotic tail in maintaining the static stability of a mobile robot when carrying a payload. This is done by measuring the stability of a robotic system when carrying a payload using a robotic arm, with and without the addition of a robotic tail designed to adjust the COM to maintain stability. In order to accomplish this, a number of different strands of research need to be brought together. The following key objectives are required:

- Investigate the current uses of robotic tails in mobile robots in order to gain insight into further research.

- Investigate potential actuation methods for the robotic tail.
- Design a configurable payload that can simulate a range of mass and COM.
- Investigate if there is any advantage to using a multi-segmented tail.
- Design a system that can measure stability in a way that could be easily used on a mobile robot platform.
- Compare the stability performance of a system that is able to pick up, carry and set down the payload with and without an active tail that uses the measured stability as a closed loop control signal.

1.3 Thesis Organisation

The rest of this thesis is organised into the following chapters:

Chapter 2: A Literature Review of Terrestrial Robots with Robotic Tails and Their Functions This chapter conducts a “state of the art” review into the existing literature for robotic tails on mobile robots, with an eye towards robot stability.

Chapter ???: ?? This chapter proposes and experimentally validates a potential design for the tail actuator, based on the Twisted String Actuator.

Chapter 5: Creating a Configurable Payload for Instability Experiments This chapter describes the design and examines the performance of a dummy payload that can have its mass and COM easily adjusted.

Chapter 4: Optimisation Study for Multi-Segment Tails for COM Control This chapter simulates single and multiple segment tails of the same mass controlling their COM (which is what contributes to static stability, as mentioned in section), which answers secondary question 4 for the specific application in the primary research question.

Chapter 6: Investigating the use of a 2DOF Pendulum Tail for Compensating for Instability when Carrying a Payload This chapter describes the design of a static rig that uses a measures stability using load cells, using a design that could be easily adapted to the majority of mobile robots, answering secondary question 1. Then it proceeds to test the efficacy of the chosen tail design by using a robot arm to carry various configurations of the payload designed in chapter 5 answering the primary research question.

Chapter ???: ?? This chapter examines the findings from the previous chapters and considers the impact of the research. It also examines some of the limitations and proposes future work not covered in other chapters.

Chapter 2

A Literature Review of Terrestrial Robots with Robotic Tails and Their Functions

In this chapter, a literature review methodology is described that produces a collection of relevant publications in the field of terrestrial mobile robots with tails. A categorisation system is developed based on the function of the tail and how it contributes to the robot's dynamics. In each category, selected publications are then summarised with accompanying figures from the source material. In the discussion, broader findings about the qualities of the robot tails found in the literature are explained, which include what performance comparisons were made between different tail designs, motions, or control methods within the same publication for the same tail function, any bio-inspiration the publication took from the animal kingdom, be it through direct performance comparison, or loose "inspiration", and finally whether multiple segments confer any performance advantage in specific tail applications.

2.1 Introduction

The field of terrestrial robots with robotic tails is incredibly diverse, reflecting the many functions of tails in the animal kingdom. Even discounting tails used for fluid dynamics, e.g. swimming, water walking and flying robots, and focusing only on robots that use their tail during “terrestrial” locomotion, broadly defined as when a robot is moving along a contiguous surface, or jumping from one surface to another surface, there are many applications of robotic tails. In order to make sense of the state of the field, an abstract categorisation system is considered based on the environment the robot is in when the tail is active, the specific action the robot is taking to move itself in space, and what the specific function of the tail is with respect to the robot dynamics. Using this categorisation system, various examples are explored from a set of research articles selected using specific keywords. From this, conclusions can be derived about the general design and operation of robotic tails, which can be used to influence and guide the research covered in chapters ??.

2.2 Literature Review Methodology

Using three online publication repositories, *IEEE Xplore*, *Scopus* and *Web of Science*, a search query was conducted to find relevant publications. The query was tailored to include all publications with **tail*** or **appendage** in the title along with **robot*** (* indicates a wild-card suffix), but to exclude publications that concerned swimming, water walking, or flying robots, as using a tail as a rudder to influence fluid dynamics was outside of the scope of the research. Further exclusions were added upon experimentation with the query in order to remove false positives in areas such as chemistry (as molecules are often described as having “tails”) or medicine (as it did not pertain to mobile robots and usually concerned biological structures such as proteins and cells). The date range was set from January 1980 to June 2021, when the search was conducted, to exclude outdated publications:

(Tail* **OR** Appendage) **is contained in Document Title** & Robot* **NOT** (Fish **OR** Swim) **NOT** (Surgery **OR** Medic* **OR** Tumour) **NOT** (Helicopter **OR** Unmanned Aerial Vehicle **OR** UAV) **NOT** Underwater **NOT** (Chemical **OR** Chemistry) **NOT** Tailor* **is contained in Document Title**

As a result **498** publications were discovered. After duplicates were removed and after screening abstracts and full texts, a total of **84** unique publications were selected for inclusion. A flowchart of this process can be found in figure 2.1.

The final selection included **55** *experimental* publications, which included experimental results from a physical prototype, and **29** *simulation* publications which only included results from dynamics simulations and/or analytic functions. This review will focus on experimental publications, as they have stronger evidence for the efficacy of their research.

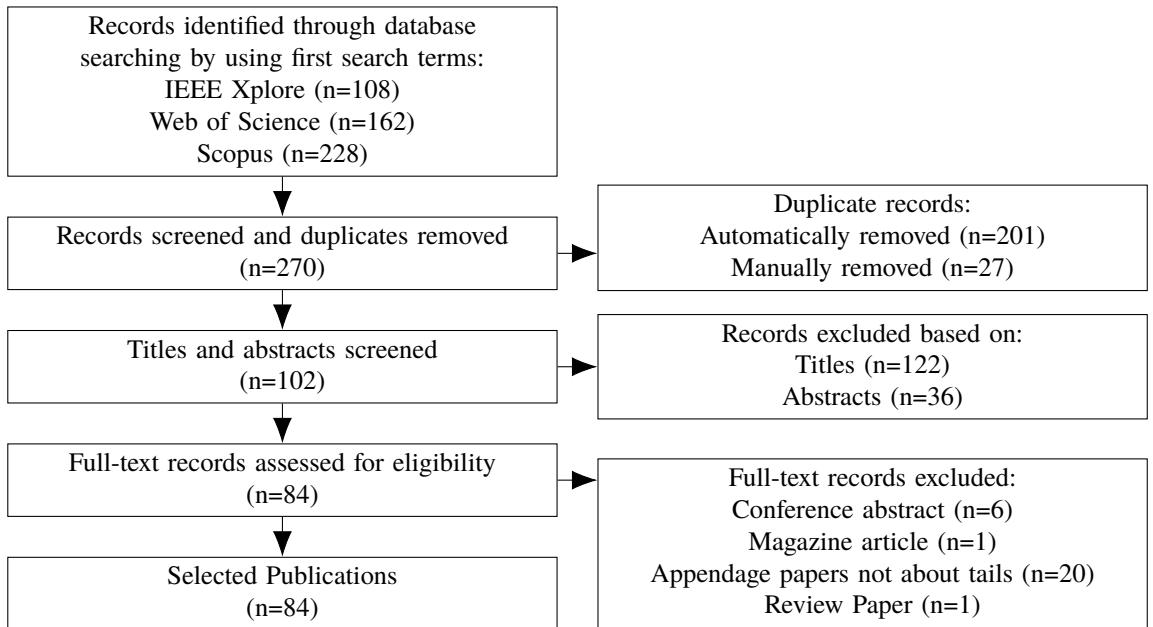


Figure 2.1: PRISMA Flowchart of the publication selection process.

2.3 Tail Functions of Terrestrial Robots

2.3.1 Categorisation System

The resulting publications represented a wide array of different robot designs and multiple forms of locomotion. Therefore, a simple categorisation system was required in order to better understand the majority of the functions of these tails. After careful analysis of the literature, two questions could be asked that provided common answers:

1. Is the robot on the ground, in the air or transitioning between those states when the tail is active?
2. What precisely is the robot doing when the tail is active?
3. What does the tail do to the dynamics of the robot (i.e. what would happen if there *wasn't* a tail present)?

Each question can then assign a tiered category to a publication based on the answer, the three tiers named *Environment*, *Action* and *Function*.

1. **Environment:** The general domain the robotic system is operating in when the tail is active. From the reviewed literature, three categories have been created:
 - *Terrestrial:* A robot with an active tail when the robot is touching a surface, such as a robot driving along the ground.
 - *Aerial:* A robot with an active tail when the robot is in free space, such as a robot which *has jumped* into the air, or is falling off a ledge.

- *Transition*: A robot with an active tail when the robot is just about to transition from between the two previous environments, such as a robot *just about* to jump into the air.

2. **Action:** The specific action the robot is performing when the tail is active. Most actions are unique to each environment, except for *hopping*.

- *Straight*: A robot travelling across a surface maintaining its direction of travel.
- *Accelerating*: A robot changing its velocity in the direction of travel across a surface. In the literature, this was a robot coming to complete stop, and starting from stationary.
- *Turning*: A robot changing its direction of travel across a surface, such as a robot turning a corner.
- *Balancing*: A robot undergoing external disturbances while travelling across a surface, typically due to adverse terrain, such as a robot navigating a rough and uneven surface without falling over.
- *Hopping*: A robot executing a sequence of periodic jumps in order to travel across a surface, similar to the method of locomotion of a Kangaroo.
- *Jumping/Falling*: A robot executing isolated non-periodic jumps, falling off a ledge or launching off a ramp, typically to transition from one surface to another at a different altitude and/or orientation.

3. **Function:** The purpose of the tail when the robot is performing the action. These categories usually apply to multiple actions.

- *Stability*: The tail is used to *maintain* some aspect of the robot's position and/or orientation from the start of the action.
- *Initiation*: The tail is used to *change* some aspect of the robot's position and/or orientation from the start of the action.
- *Amplification*: The tail is used to *amplify* the effects of an action by other parts of the robot (such as the legs) which changes the position and/or orientation.

For example, a robot with a tail that helps it increase its apex when hopping, is in the *Transition* environment as it is about to transition from being on the ground to in the air when the tail is active. The robot itself locomotes by hopping, so it is performing a *Hopping* action, and since without the tail the robot would still be capable of hopping, but would not have such a tall apex, the tail can be considered to be performing *Amplification* of the robot's existing capabilities. Overall, this results in the categorisation *Transition* → *Hopping* → *Amplification*.

Another example is a robot with a tail that prevents it from falling over on rough terrain. The robot is in the *Terrestrial* environment as it is on the ground when the tail is active.

The robot itself is performing a *Balancing* action as it is trying to remain upright during locomotion, and since without the tail the robot would fall over, the tail can be considered to be maintaining the *Stability* of the robot. Overall, this results in the categorisation *Terrrestrial* → *Balancing* → *Stability*.

This categorisation system does not include robots that have a tail but are not mobile, or where the tail is used for non-locomotion tasks, such as self righting [10] or dragging objects [11]. It also does not include robots where the tail is in constant contact with the ground as essentially an extra leg [12].

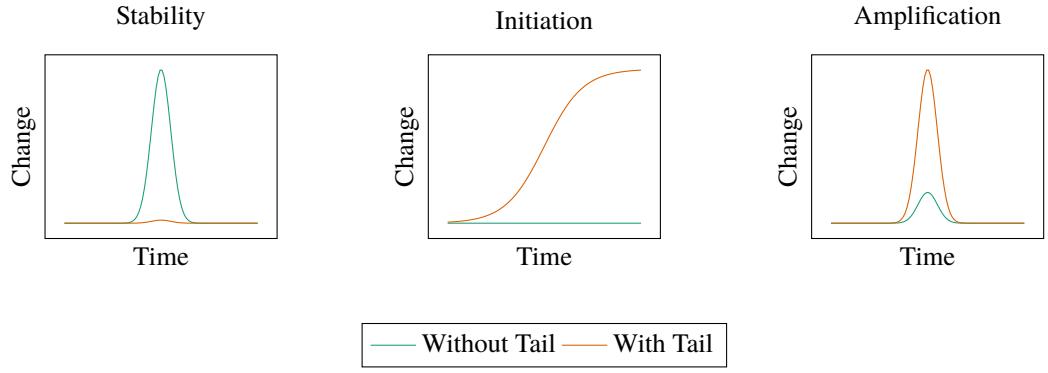


Figure 2.2: Abstract graphs of the different functions of the tail, with the magnitude representing some kind of change in the robot's position and/or orientation.

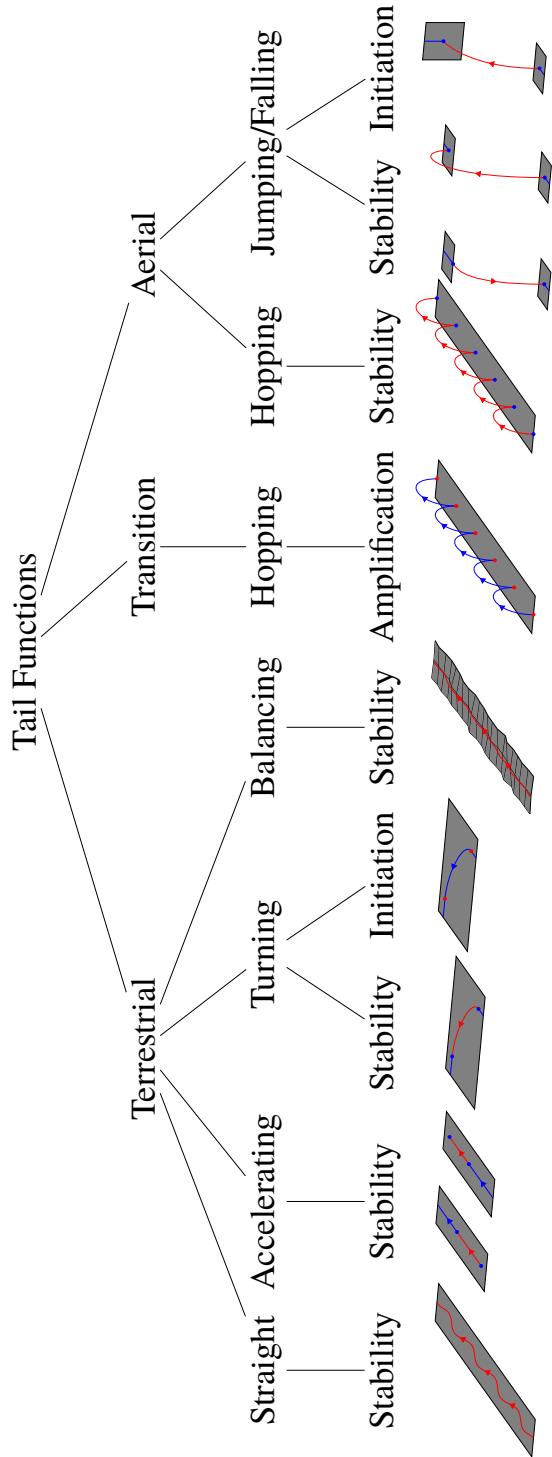


Figure 2.3: Tree diagram of all the categorisations found in the publications with accompanying visual diagrams.

2.3.2 Examples from each Category

2.3.2.1 *Terrestrial → Straight → Stability*

[13] uses a quadrupedal robot with a gait controlled by a Central Pattern Generator (CPG).

Upon initial experiments with a trotting locomotion with no active tail, the robot would not maintain a set heading, it would instead slowly begin to drift in a circle. Visual observations noted that the robot would topple onto its front left leg that was in “swing” phase (lifted off the ground), and it would drag on the ground until in “stance” phase. This resulted in a difference between the Ground Reaction Force (GRF) of the left and right feet which caused the drift in locomotion path.

By implementing a swinging tail that imparted an opposing torque to the direction of the topple, the differences between the left and right GRF were reduced, and the robot could maintain its heading, as can be seen in figure 2.5.

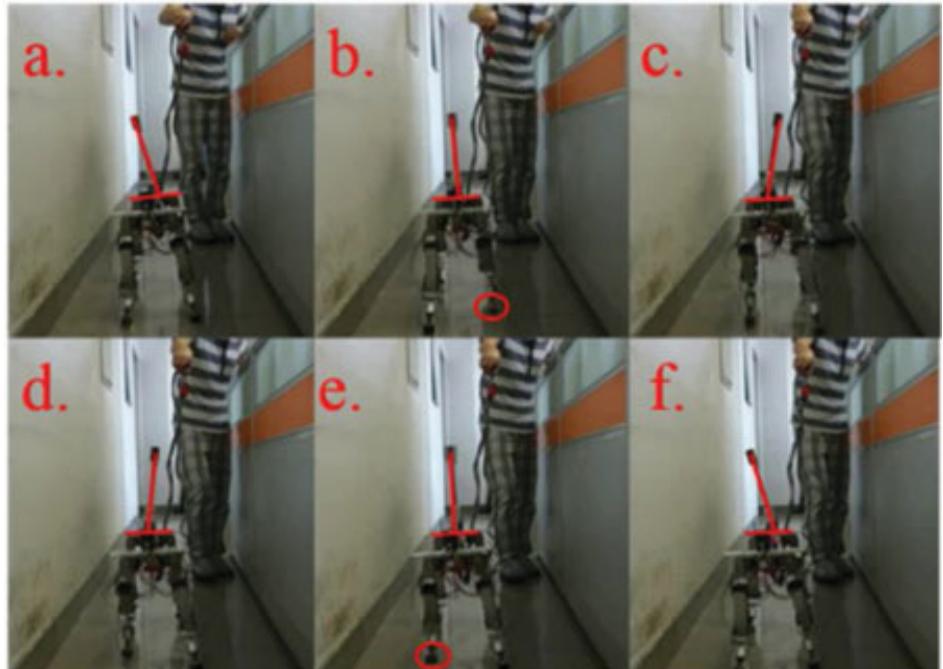


Figure 2.4: Image from [13] showing how the tail moves during the gait in order to correct for heading drift.

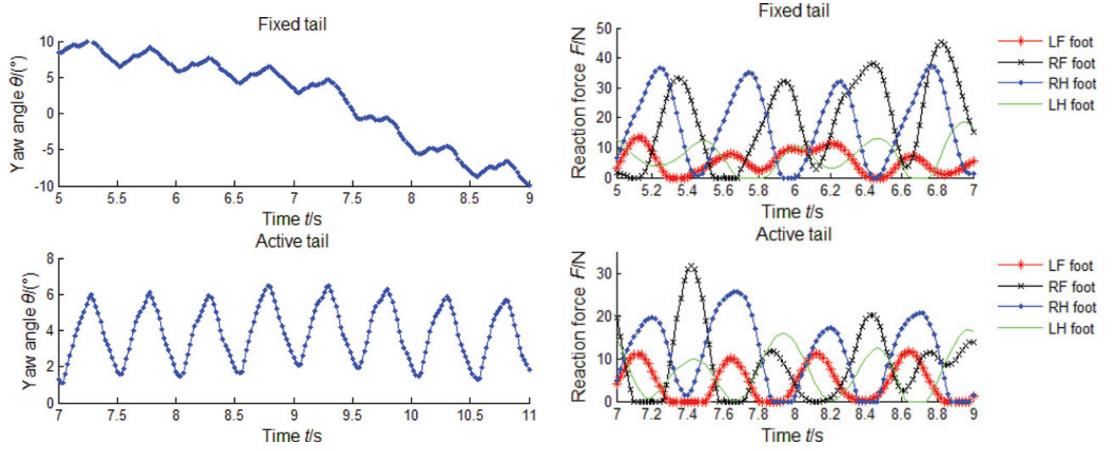


Figure 2.5: Data from [13] showing the effects of a static and dynamic tail on maintaining robot heading in a trotting gait.

2.3.2.2 *Terrestrial → Accelerating → Stability*

[14] used inspiration from the Cheetah to improve the acceleration and braking capabilities of a wheeled robot. The research is based on findings from [15], which shows that quadruped acceleration and deceleration in the animal kingdom is limited by their ability to constrain body pitch to prevent toppling over. It can be considered analogous to a motorcycle: accelerate too fast and the vehicle will “pop a wheelie” and potentially flip backwards, decelerate too fast and the opposite may occur.

Using a combined state feedback and Proportional Integral (PI) controller based on the angular position of the tail, and the angular velocity of the tail and body, the researchers were able to increase the acceleration and braking capabilities of the robot by using the tail to generate an opposing torque to the direction of body pitch, as can be seen in figure 2.8. This was verified by running a series of experiments, increasing the acceleration/braking magnitude until the robot failed to complete the test by toppling over.

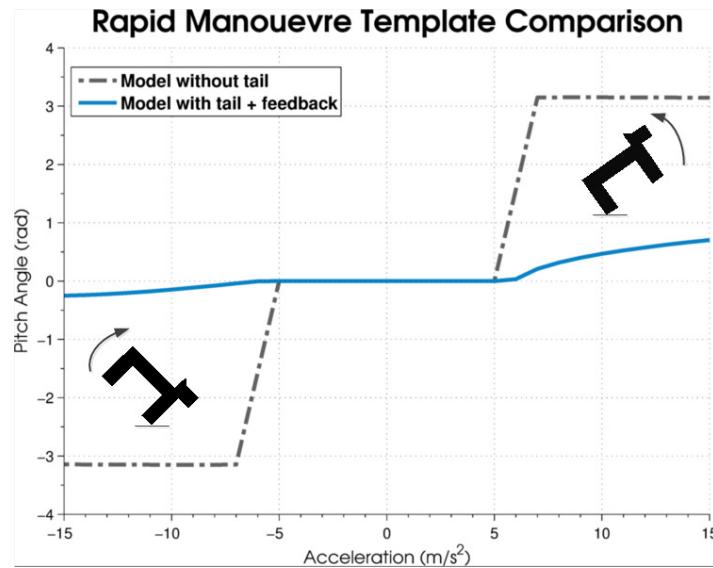


Figure 2.6: Simulation Data from [14] showing the how the body pitch would be reduced when accelerating or braking with an active tail.



Figure 2.7: Images from [14] showing the robot performing a rapid acceleration test with the tail.

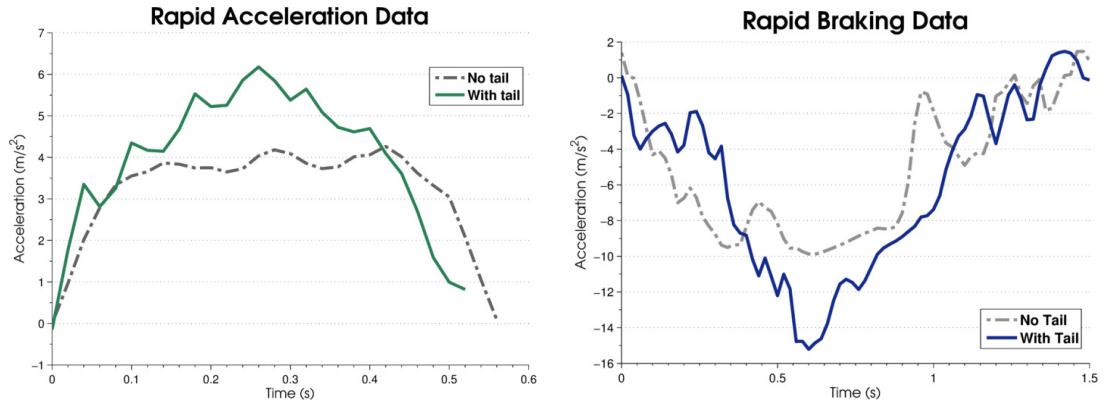


Figure 2.8: Experimental Data from [14] showing the maximum acceleration achieved with and without an active tail.

2.3.2.3 *Terrestrial → Turning → Stability*

[16], [17] took similar inspiration from the Cheetah to allow for tighter turns by allowing greater lateral acceleration. [16] swings the tail out in a single motion in the direction of the turn, producing an opposing torque to the centrifugal force that would otherwise topple the robot during the turn. In contrast, [17] moves the tail constantly in a conical motion, the direction of rotation in the direction of the turn. This also produces an opposing torque in the same fashion, but was not limited in duration, as in the first strategy the tail would eventually contact the ground. This allowed for turns of longer duration to be stabilised.

The control system and experimental procedures were similar to those in [14].

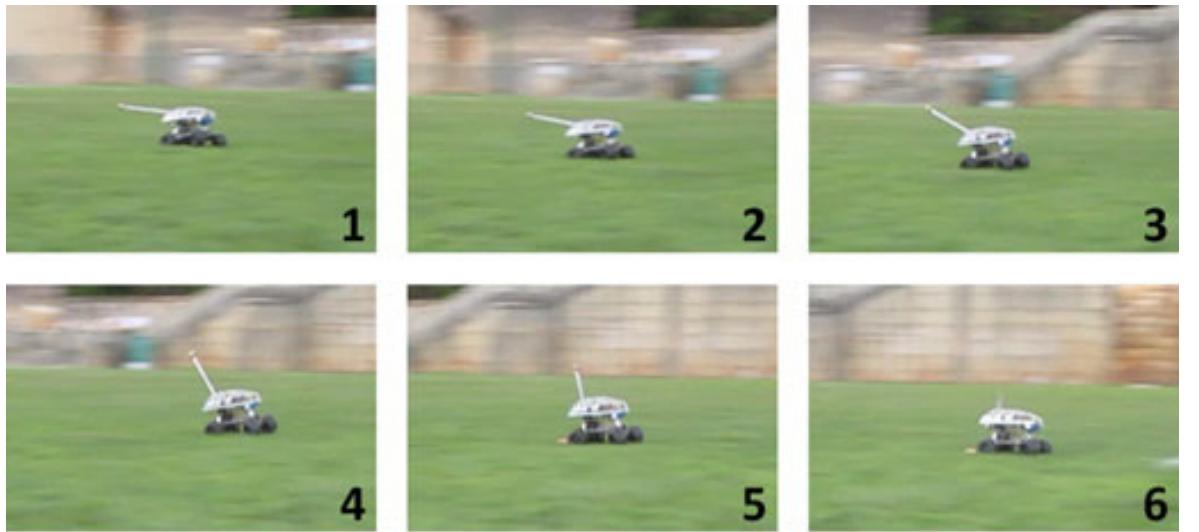


Figure 2.9: Images from [17] showing the robot performing a turn with the tail.

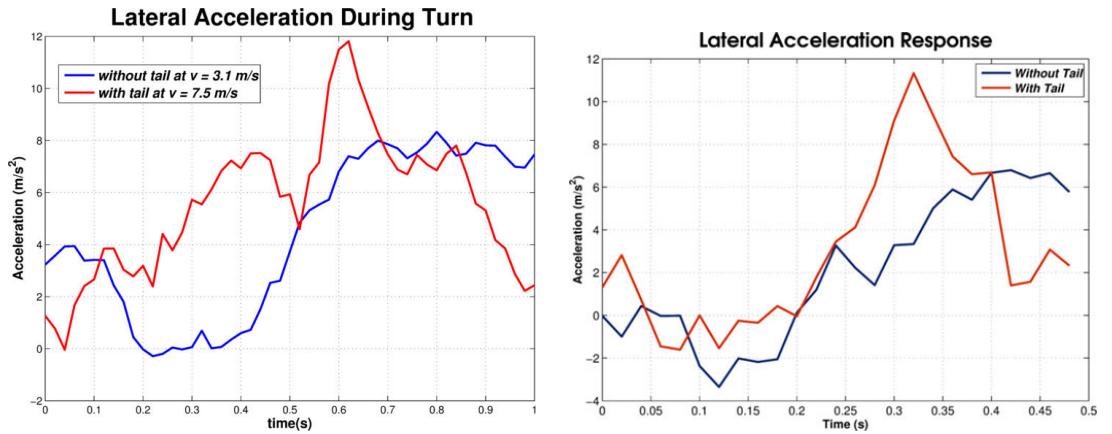


Figure 2.10: Experimental Data from [16] (left) and [17] (right) showing the maximum lateral acceleration achieved with and without an active tail. Note the right graph manages similar results to the left graph.

2.3.2.4 *Terrestrial → Turning → Initiation*

[18], [19] both use similar robot designs, insect like robots that locomote using 6-8 pairs of small legs (in this case [18] used a robot with eight legs, and [19] used six legs) as shown in figure 2.11. Both robots are designed to be very light (52 g in [18] and 46 g in [19]) so the legs have a low friction force with the ground. A suitably weighted tail, when swung out in a horizontal motion, can overcome this friction force and impart enough torque to rotate the body of the robot to a new heading.

[19] compared an open and closed loop response of the tail, while [18] only compared open loop tail responses at different frequencies and amplitudes, as can be seen in figure 2.13.

Both experiments were able to greatly increase the turning rate of the robot over a turn using a differential drive, at 360° s^{-1} for [19] and 400° s^{-1} for [18].

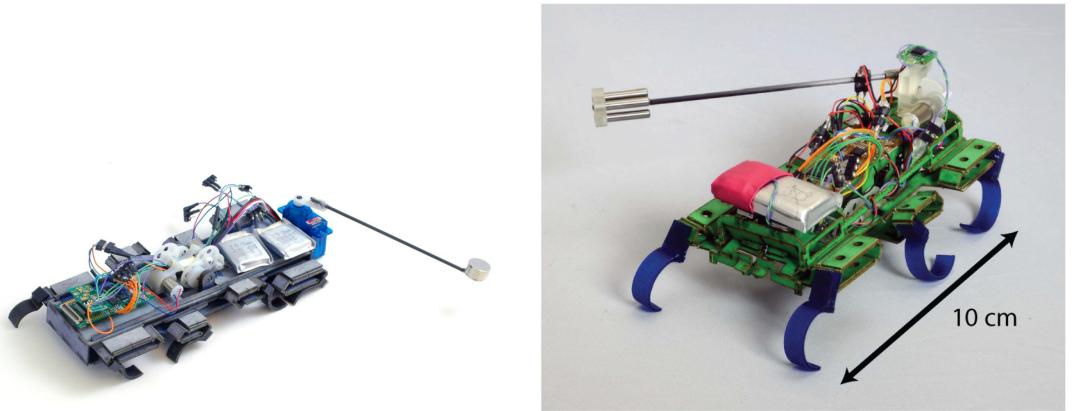


Figure 2.11: Images of the hexapodal and octopodal insect like robots in [18] (left) and [19] (right).

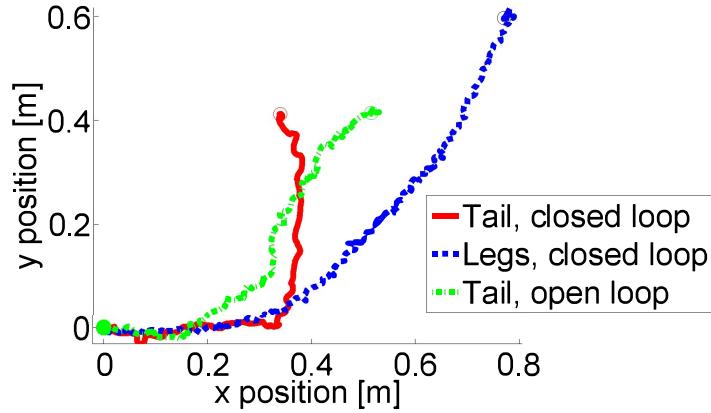


Figure 2.12: Data from [19] comparing the robot path on an XY plane for a 90° differential drive turn, open loop tail turn, and closed loop tail turn. Note the vastly increased sharpness of the turn when a tail is used.

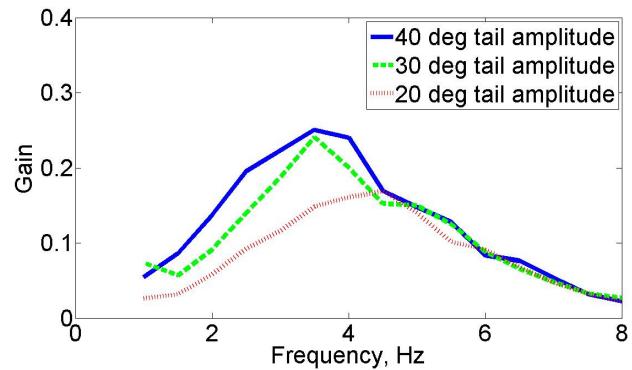


Figure 2.13: Data from [18] showing the gain in yaw rate for various open loop tail trajectories and different frequencies and amplitudes.

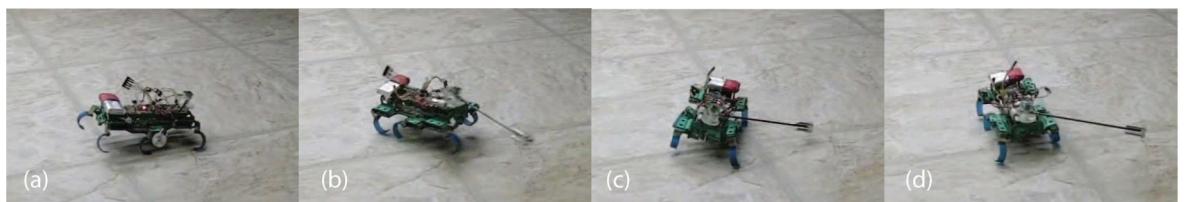


Figure 2.14: Images from [19] showing the robot using its tail to make a turn.

2.3.2.5 Terrestrial → Balancing → Stability

The forces that may cause the robot to topple over when balancing can result from both *internal* forces: robot design or joint inertia when moving at high speed, and *external* forces: uneven terrain or impact. [20] is concerned with internal forces, whereas [21] is concerned with external forces.

[20] used a dinosaur like bipedal robot with a long neck and tail. The neck and tail then swung from side to side during the gait, maintaining the stability of the robot. Two experiments were conducted using different strategies for maintaining stability. The “static” method swung the neck and tail in a trapezoidal motion in order to keep the COM within the area of the current foot on the ground. The “dynamic” method calculated the ZMP of the robot and instead constrained that to keep it within the are of the foot. This resulted in a smaller motion of the neck and tail which enabled a faster gait, from 19.5 mm s^{-1} up to a theoretical 208 mm s^{-1} , though in practice the velocity was limited by the motor performance.

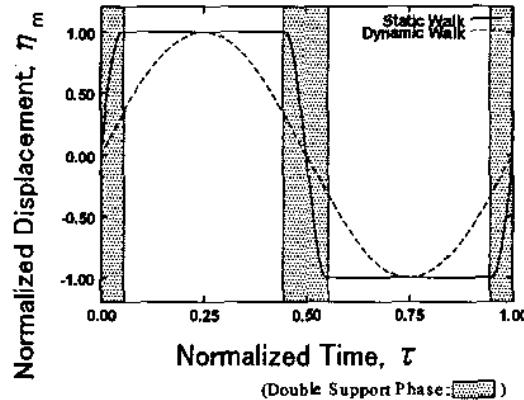


Figure 2.15: Data from [20] showing the normalised change in COM for a static and dynamic walk.

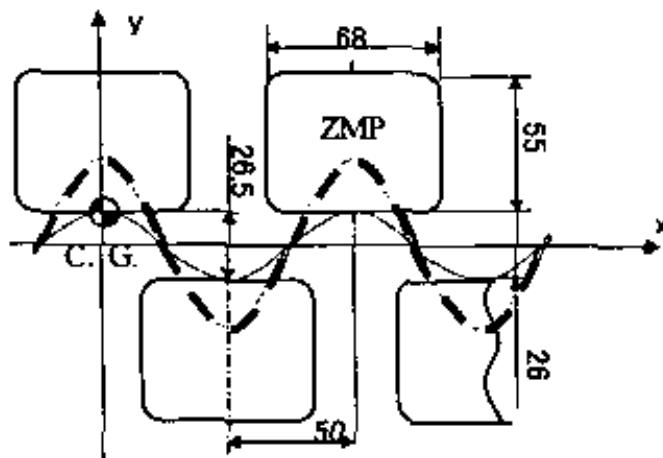


Figure 2.16: Data from [20] showing the trajectory of the COM and ZMP for a static and dynamic walk.

[21] uses a quadrupedal robot closely modelled on a Cheetah which is hit in the torso by a “wrecking ball” to simulate a disturbance. In the control experiment the weighted tail remains static, in the active tail experiment the tail responds in an open loop trajectory when

triggered by an accelerometer that sensed the impact. The active tail experiment was able to significantly reduce the hip displacement after impact, as can be seen in figure 2.18.

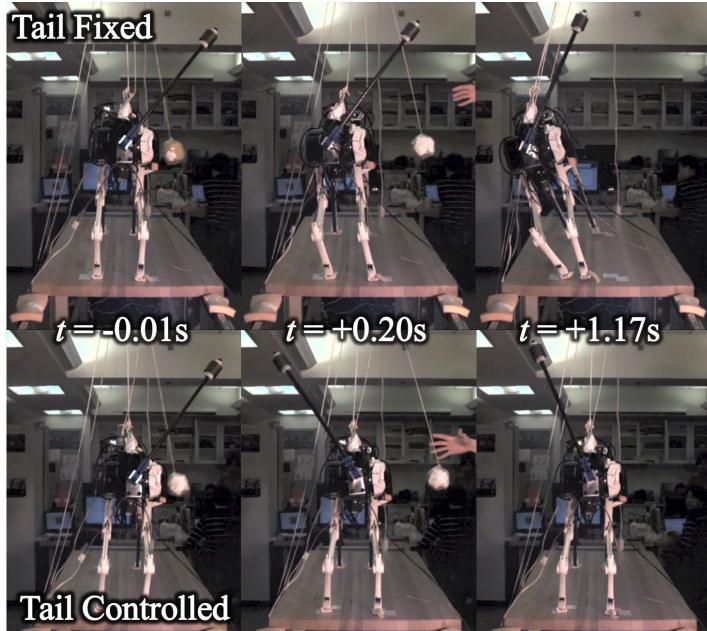


Figure 2.17: Images from [21] showing how the tail or body deflect when hit by the wrecking ball, depending on if the tail is active.

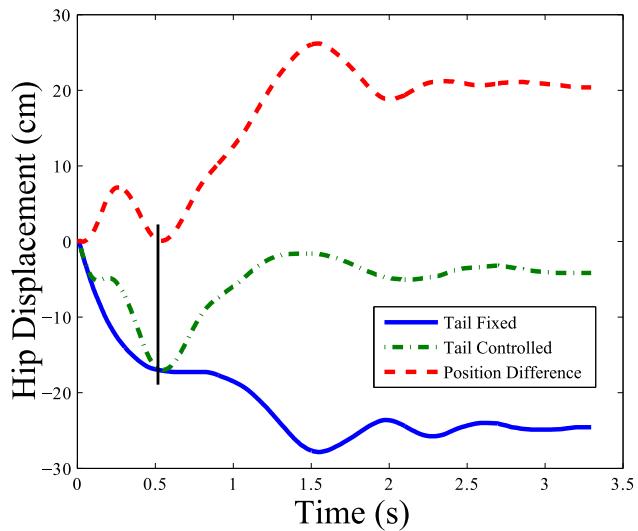


Figure 2.18: Data from [21] showing the difference in hip displacement between a fixed and controlled tail.

2.3.2.6 *Transition → Hopping → Amplification*

[22], [23] both use a tail to increase the magnitude of a hopping gait. [22] increases the *height* of the hop, while [23] increases the *length*.

[22] uses a bipedal robot with a long, flexible tail. The tail consists of six segments, connected together by passive spring revolute joints. The only active joint connects the tail assembly to the robot body. This creates a “whip-like” motion in the tail when the joint is actuated. By experimenting with different spring constants, an optimum value that maximises

jumping height can be found, as can be seen in figure 2.21. Results showed a significant increase in jump height, 256 mm over 240 mm for a model with a rigid tail, and 210 mm for a mode with an inactive tail, demonstrating the superiority of a flexible tail in this application.

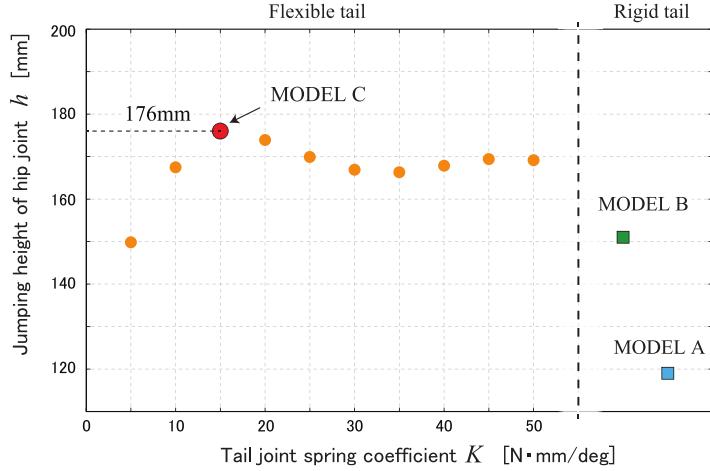


Figure 2.19: Data from [22] showing the optimal spring constant for maximising jumping height.

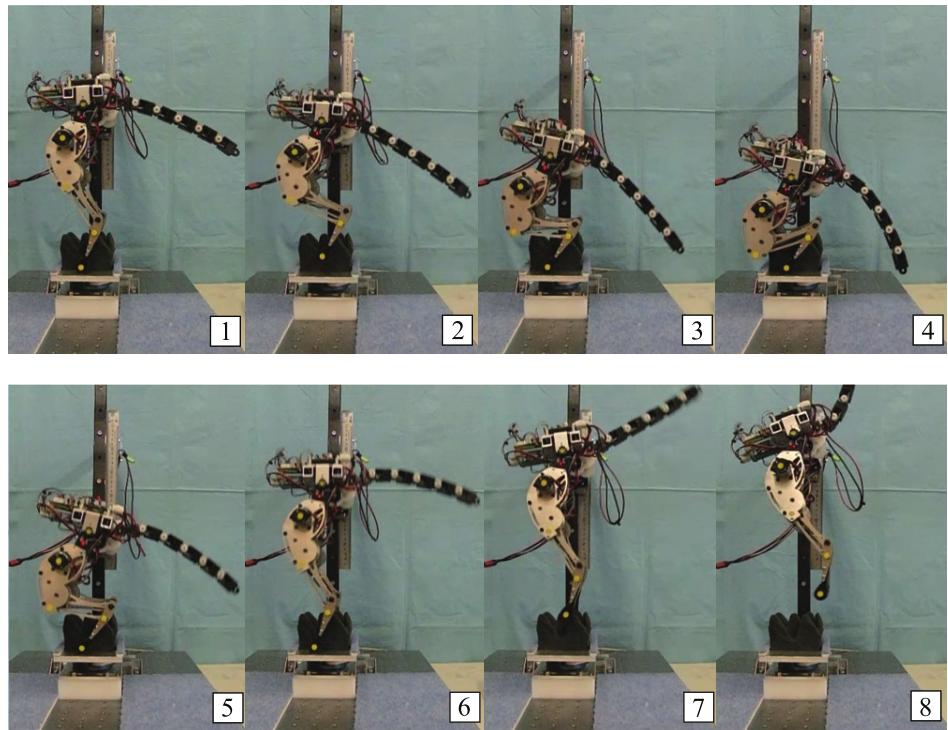


Figure 2.20: Images from [22] showing the motion of the tail during a jump.

[23] uses a quadrupedal robot that locomotes in a galloping motion. By using a weighted tail in an open loop trajectory, the robot is able to increase its forward velocity while also reducing body pitch. Two different tail lengths and masses were used for experiments, a long, light tail (31 g/168 mm), and a short heavy tail (53 g/128 mm). Different open loop amplitudes, 35° and 65° , were also used for each experiment. Results showed an increase in forward velocity and reduction in body pitch, with the best forward velocity of 0.558 m s^{-1} from the short, heavy tail at 35° , and the best reduction in body pitch per stride of 4.6° from the long, light tail at 65° . These results compared to a forward velocity of 0.479 m s^{-1} and

body pitch per stride of 8.5° for a fixed “passive” tail.

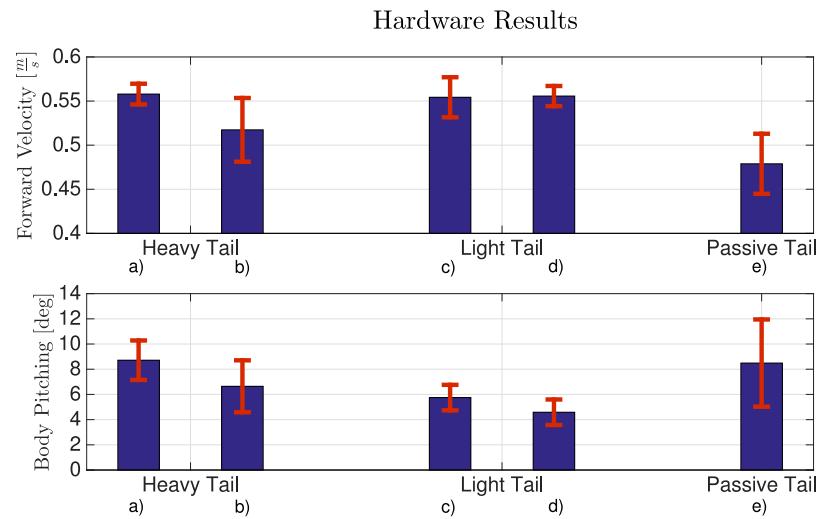


Figure 2.21: Data from [23] showing the increase in forward velocity and/or reduction in body pitch for the different tail lengths and masses. Note that a) and c) correspond to a 35° tail amplitude, while b) and d) are a 65° amplitude.

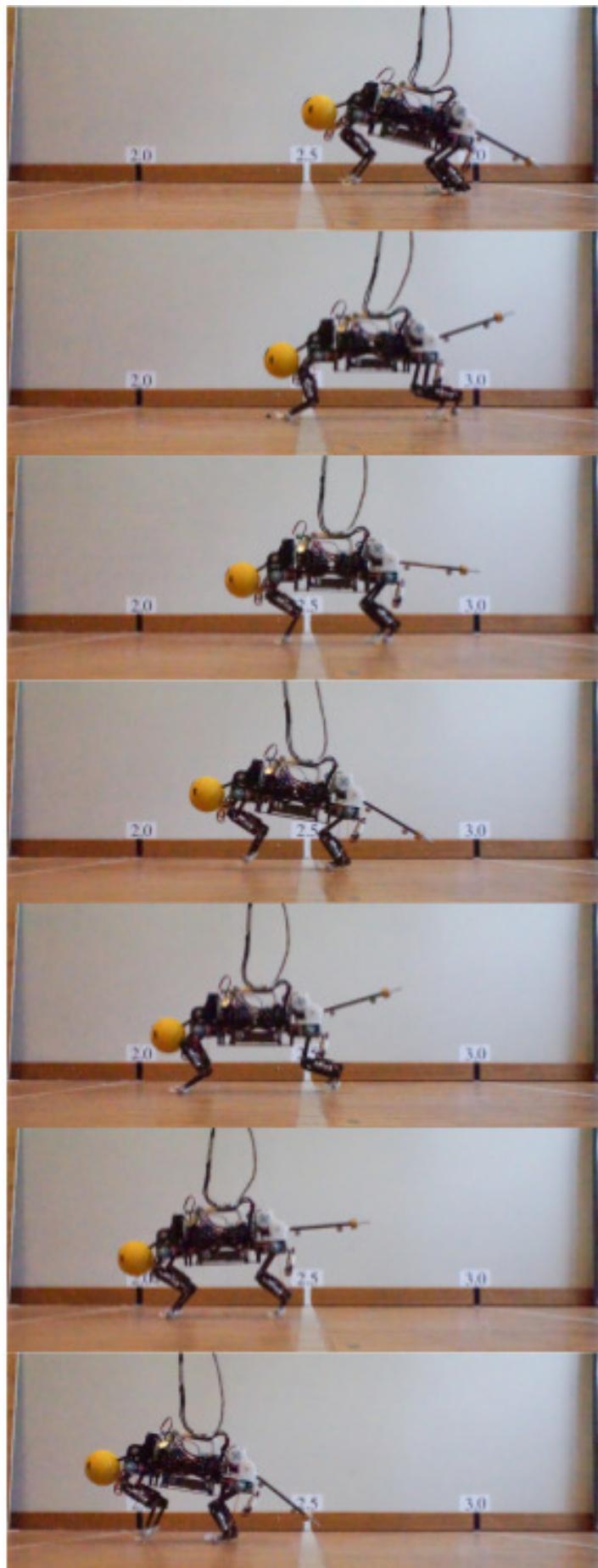


Figure 2.22: Images from [23] showing the motion of the body and tail over a single stride cycle.

2.3.2.7 Aerial → Hopping → Stability

[23] also used the tail to reduce body pitch, which prevents the robot from falling over forward when hopping. [24] used a kangaroo like robot with a tail to also decrease body pitch, in both an open and closed loop tail trajectory. The open loop tail was able to reduce the body pitch range to 5.17° and Root Mean Square (RMS) error to 1.17° , and the closed loop tail to 4.49° with an RMS of 0.96° , as compared to 7.18° with an RMS of 2.24° for a stationary tail (in this case, *range* refers to the difference between the maximum and minimum pitch angle during the hop).

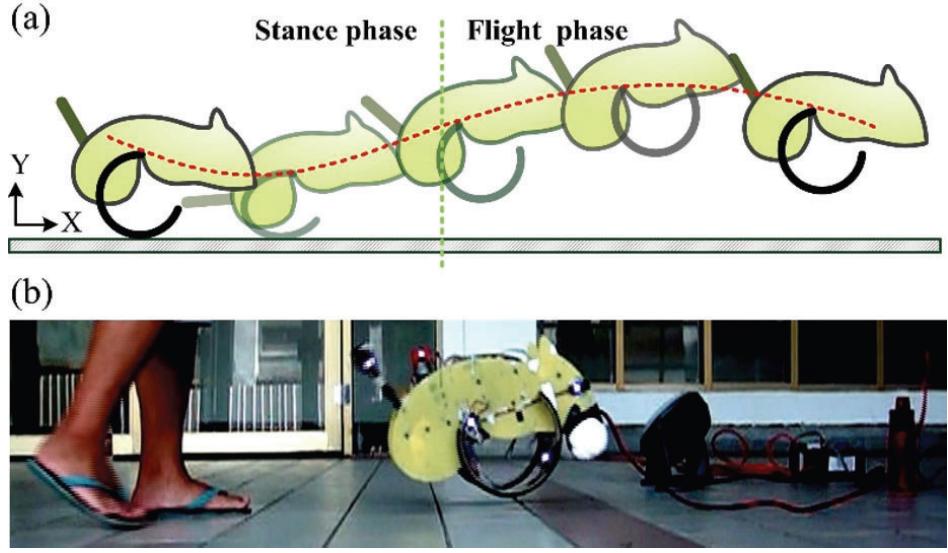


Figure 2.23: Images from [24] showing the a single stride for the robot, and an image of the robot just before landing.

[25] also developed a bipedal robot with a 2 Degrees of Freedom (DOF) tail that was used to maintain a stable hopping motion.

2.3.2.8 Aerial → Jumping/Falling → Initiation

[26] uses a wheeled robot with a weighted tail to reorient the robot in mid-air. Two experiments are carried out to this end, where the robot is released from a vertical position on a wall and attempts to land horizontally, and where the robot drives off a ledge and attempts to land on a 45° sloped surface. The tail used a Proportional Derivative (PD) feedback controller with body orientation as input. In both experiments, the robot was able to successfully reorient itself.

[27] used a robot that was able to wheel itself to a specific position, then rotate itself using the tail as an appendage into a different orientation which allowed it to launch itself into the air. Then once in the air it would use the tail itself to orient itself back so it landed on its wheels.

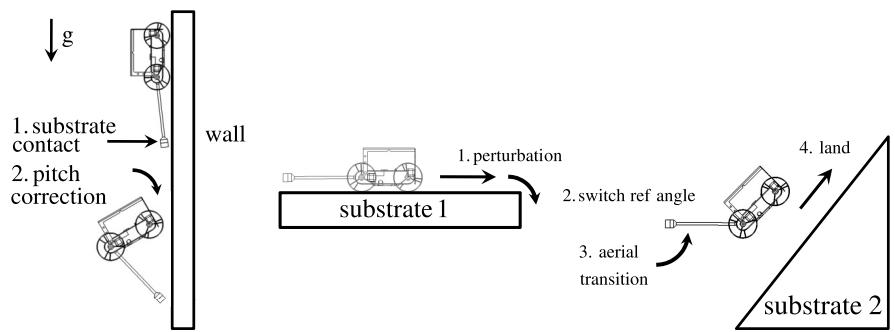


Figure 2.24: Diagrams from [26] outlining the two reorientation experiments.

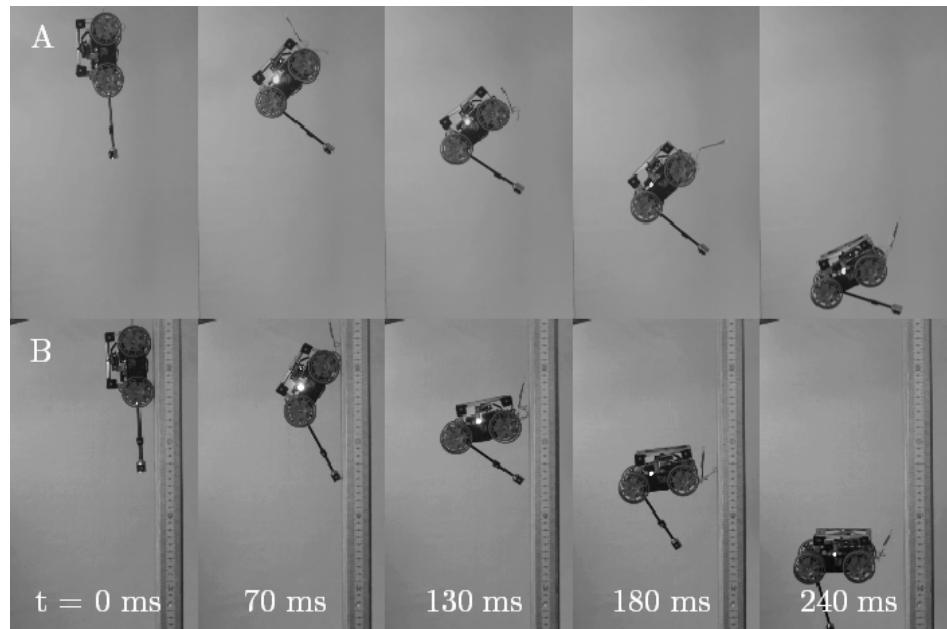


Figure 2.25: Images from [26] showing the wall reorientation experiment.



Figure 2.26: Images from [26] showing the slope reorientation experiment.

2.3.2.9 Aerial → Jumping/Falling → Stability

[26] also conducted an experiment where the robot would hit a small obstacle when driving along a surface, causing it to go airborne. Experiments were conducted without a tail, with a static tail, and with a closed loop controlled tail.

[28] had a hexapod robot that was able to remain upright when running off a ledge.

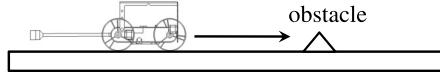


Figure 2.27: Diagram from [26] showing the obstacle impact test.

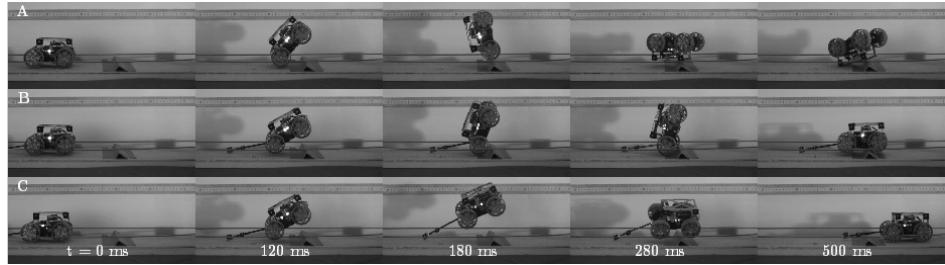


Figure 2.28: Images from [26] showing the slope obstacle impact test without a tail, with a static tail, and with a dynamic tail with closed loop feedback.

2.4 Discussion

Many tails were designed to be multi functional, as evidenced by experiments of different functions within the same publication, such as [23], and/or multiple publications that included the same robot, such as [14], [16], [17].

2.4.1 Performance Comparisons in the Selected Publications

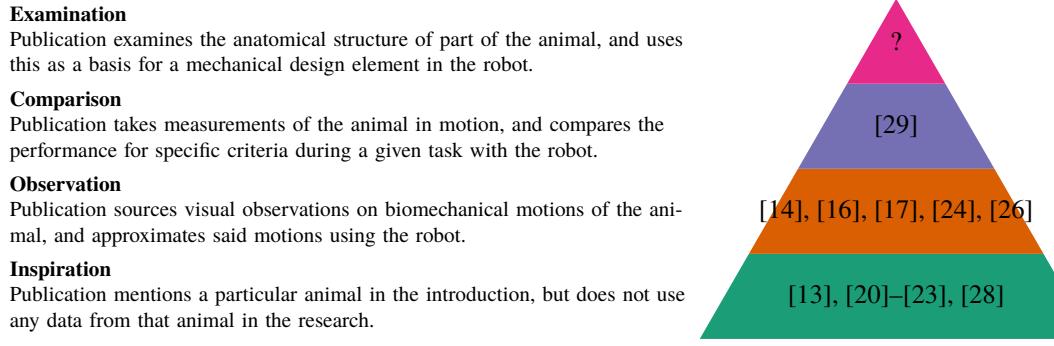
A number of experimental publications varied their tail design [22], [23] or control system [24], [27], and were able to record improvements in performance for various metrics. Figure 2.29 shows selected publications performance comparisons, normalised to the best performing variation for each metric, for a unitless comparison. For control systems, the more complex and advanced the better the performance, understandable since a publication is unlikely to test a more complex control system unless there is high confidence in a performance improvement. For tail design, the picture is more complex. While [22] demonstrates a clear improvement in performance for a flexible tail, a more complex design, [23] shows how tail design can result in trade-offs between different metrics, with different tail designs being superior. However, it is possible to see the *Long Light Tail* (65°) is the optimal choice, since reduction in *Forward Velocity* is slight, but the reduction in *Body Pitch* (a negative metric, where a larger value is worse) is significant, assuming both metrics are equally weighted. Figure 2.29 shows these performance improvements normalised to the best performing configuration at 100%, e.g. a measured result of [5, 10, 30, 50] would be represented as [10%, 20%, 60%, 100%].



Figure 2.29: Graphs of various publications that varied tail design or control, normalised to the best performing variation (= 100%) for each metric, for a unitless comparison.

2.4.2 Bio-Inspiration

The selected publications contained varying degrees of bio-inspiration, which can be fitted into a hierarchy as shown in figure 2.30. Publications with the least influence mentioned an animal in their research only in a vague fashion, mainly as an introduction to their research, possibly to provide loose justification for their approach. Other publications did take this one step further, and made direct observations on the motion of the animal tail in order to closely approximate them in a robot. For example, [14], [16], [17] all observe the tail motion of a Cheetah when performing certain actions (accelerating, decelerating and turning), and while they do not measure the exact trajectory of those observations, they use a similar motion for their robot tail when the robot is performing the same action. A more intensive approach does take data from an animal performing an action, and compares the data to a



Note: [21] does use *observation* for the simulation, but does not for the experimental work.

Figure 2.30: Hierarchy of bio-inspiration with examples from the selected literature, from least to most influential on the research.

robotic equivalent. For example, [29] makes a direct comparison between the body rotation sensitivity from a perturbation between an Agama Lizard, a robot, and a simulated Velociraptor. What has not been observed in the selected literature is an attempt to replicate the anatomical structure of an animal tail using mechanical components. This is likely due to a number of reasons, complexity of design and control, redundant actuation, and technological limitations among others. While this approach has been conducted before [30], it has not been seen in the selected literature. This kind of intensive bio-inspiration could be useful in the field of robotic tails in order to explore the potential for performance improvements, and could also help answer questions about why naturally evolved tails are generally more complex and redundant than their robotic counterparts.

2.4.3 Use of Multiple Segment Tails

Only a couple of publications [22], [31] experimented with mobile robots using tails with multiple segments. Other publications [32]–[37] only experimented with multi-segment tails in isolation, some with speculation on future inclusion in a mobile robot. As mentioned in subsections 2.4.1 and 2.3.2.6, [22] managed to achieve a 25 mm improvement in jump height using a sprung flexible tail instead of a rigid one. The relative paucity of experimental multi-segment tails could be explained due to complexities in dynamics and control, as even in [22] only the first joint was actuated and the rest passively spring-loaded, resulting in a simple design and control system. If the future research of some isolated multi-segment tail experiments is realised, then more data will be available.

2.5 Conclusion

This chapter has reviewed a curated selection of the literature on terrestrial mobile robots with robotic tails, focused around experimental papers that generated results from physical prototypes. It has discovered a commonly occurring set of tail functions that assist in the dynamics of the robot, in the air, on the ground, and transitioning between the two en-

vironments. Some publication used bio-inspiration, such as the kangaroo and cheetah for hopping and turning in [17], [24] and some did not, such as the addition of a tail to insect like hexapod robots in [18], [19] for inducing steering, which does not occur in nature. It has discovered by far the most common design for a tail is a rigid single segment with 1-2 DOF, unlike in the animal kingdom where multi-segment tails are the norm. However, a publication that did use multi-segment tails did find an improvement in performance [22], even when only using passive spring joints for all segments except the first.

There are still several gaps in the research, which has very limited experimental studies into multiple segment tails on mobile robots, and no quantitative research in the use of a tail for controlling robot dynamics when carrying a payload. [38] does use a hexapod robot with a tail to carry a payload, but does not indicate how effectively, if at all, the tail contributed to the robot stability.

While simulation studies [35], [39]–[42] and studies with tails in isolation have been conducted [32]–[37], there is still a relative paucity of experimental research, the only notable result still using mostly passive joints. This is most likely down to technical limitations, the additional mass and bulk of additional actuators is more of an issue on a mobile robot than a static tail or simulation study. There is also the increase in control complexity, which will make stable control of the tail more of a challenge in most of the functions discussed. The development of a compact, light 2DOF actuator could assist in making multi segment tails on mobile robots more feasible. A proposed actuator based on the Twisted String Actuator is investigated in chapter ???. A simulation study is also conducted in chapter 4, in order to see if any improvement can be gained using a multi-segment tail for the specific application in this thesis.

The dynamics of picking up, carrying, and setting down a payload pose a unique challenge for a robotic tail. However the literature has proven that using a robotic tail for maintaining stability, by keeping the robot body at a certain angle in the air, or preventing the robot toppling over on the ground, is viable. A closed loop control system that is able to measure the stability of the robot and control the tail with said measurement during the transport of a payload as feedback is structurally similar to a system that uses body angle or acceleration, both of which have been described in the literature. In chapter 6, a measurement system that does this based on four load cells is devised, and a closed loop controller is connected to a robotic tail.

Chapter 3

A Novel Triad Twisted String Actuator for Controlling a Two Degrees of Freedom Joint: Design and Experimental Validation

Actuated universal joints, or equivalent joint systems, are found in a number of robotic applications, in particular mobile snake robots, continuum robots and robotic tails. These joints have two degrees of freedom on two axes, each perpendicular to a third axis and to themselves. Such joints use a variety of actuation methods, including direct drive motors, linear screw drives, cable based systems, and hydraulics/pneumatics. In this paper the authors design and validate a mechanism that uses the Twisted String Actuator (TSA) in an antagonistic triad to actuate the universal joint, using orientation sensors and load cells to create a robust cascading closed loop control system. This results in a light, compact, high-performance actuation system that avoids the extra mass and hardware complexity that alternative actuation methods present, with the additional challenge of nonlinearity.

3.1 Introduction

AUJ mechanisms are found in a wide range of robotic applications, such as confined space inspection using continuum robots [43], highly manoeuvrable mobile snake robots [44], and biomimetic robot tails for stability [45]. Mobile snake robots must usually incorporate electric actuators inline with their joints. This results in an AUJ having to shift the mass of the follower segments and all the actuators inside the follower segments, which results in high torque requirements. Continuum robots and robotic tails can reduce the mass and size of the AUJ by moving their actuators away from the AUJs and use cables to transfer the force to the joints, or use hydraulic or pneumatic actuators which tend to be lighter than equivalent electric motors. This comes at the expense of increased mass and bulk at the base of the arm or tail.

First developed by Würtz *et al.* [46] in 2010, the TSA uses two or more strings between two fixtures as a linear actuator. When one fixture is rotated (typically by an electric motor), the looped string twists into a helix, decreasing the distance between them. TSA actuators have been used for a hand orthosis [47], elbow joint [48] and foldable robot arm [49] among other functions.

The primary advantage of TSA over similar linear actuators such as a leadscrew is the reduction (lower velocity, higher torque) the TSA provides is not proportional to the mass of the actuator, in fact it is slightly inversely proportional. Generally, to increase the reduction in an actuator requires the addition of a gearbox which increases mass, but in the case of the TSA, by decreasing the string cross-section radius, the reduction increases given a constant unwound length and motor angle, resulting in a greater reduction with no increase, or even a slight decrease, in actuator mass.

While the reduction in a leadscrew can be increased by decreasing the lead on the thread, which also has no increase in mass, this has a limited range and can quickly run up against manufacturing tolerances or material strength requirements. In order to achieve greater or more robust reductions, the screw radius has to be increased, or the driving motor has to have a larger reduction before driving the screw, both of which usually result in more material (typically steel) and therefore more mass.

However, TSA does have some disadvantages, the most significant of which is a nonlinear reduction equation, which is also dependent on the motor angle (and therefore actuator position). The reduction decreases in a nonlinear fashion as the angle increases, with the derivative decreasing as the angle increases. There is also the compliance of the strings to consider, depending on the thickness and material chosen, which becomes a significant factor under high forces. Both of these issues can be addressed with accurate modelling [50] and/or a robust control strategy, as demonstrated in [46]. What is more of an issue is the unidirectional force of the TSA, which can only impart force in tension. This means that for an AUJ, which is a 2 DOF joint, a minimum of three TSA are required, unless spring return mechanisms are used, which would impart additional force on the TSA and therefore

reduce performance. However, the potential high force to mass ratio of the TSA due to the non-proportional reduction may adequately compensate for the additional actuator requirement.

The focus of this research is to investigate if the TSA is a suitable candidate for control of an AUJ considering both the benefits and drawbacks. To this end, the objective is to simulate a model and then construct a physical experimental prototype to validate the proposed control system.

Table 3.1: Model coefficients.

Coefficient	Value	Coefficient	Value
l_1	41.8 mm	J	$1 \times 10^{-6} \text{ kg m}^{-2}$
l_2	0 mm	K_L	1000 N m^{-1}
r	13 mm	f_{\min}	3 N
l_u	41.8 mm	ω_s	441.9 rad s^{-1}
r_s	$200 \mu\text{m}$	I_s	0.19 A
m	72.619 13 g	K_t	$0.0263 \text{ N m A}^{-1}$
C	0.1315 N mm	τ_s	4.5 mN m
α_s	$1 \times 10^5 \text{ rad s}^{-2}$		
Coefficient	Value		
I	$\begin{bmatrix} 3 \times 10^{-5} & 0 & 0 \\ 0 & 3.2 \times 10^{-5} & 0 \\ 0 & 0 & 1.4 \times 10^{-5} \end{bmatrix} \text{ kg m}^{-2}$		

3.1.1 Twisted String Actuator

Given the unwound length l_u and the cross-section radius of the string r_s , the actuator length is given by

$$l_s(\theta_s) = \sqrt{l_u^2 - \theta_s^2 r_s^2} \quad (3.1)$$

where θ_s is the motor angle, as shown in figure 3.1. This equation assumes an infinite string stiffness, so is only reasonably accurate under low tension. Although theoretically the stroke of the TSA can be the entire domain of $[0, l_u]$, in reality the thickness of the string prevents a geometric helix from forming once the helix pitch $q < 4r_s$ (or $q < 2nr_s$ for n strings) as mentioned in [46]. This limits the lower bound of the stroke as follows,

$$l_{\min} = \frac{l_u}{\sqrt{\frac{\pi^2}{2} + 1}} \approx 0.46 l_u \quad (3.2)$$

or approximately 46% of l_u for a two string TSA.

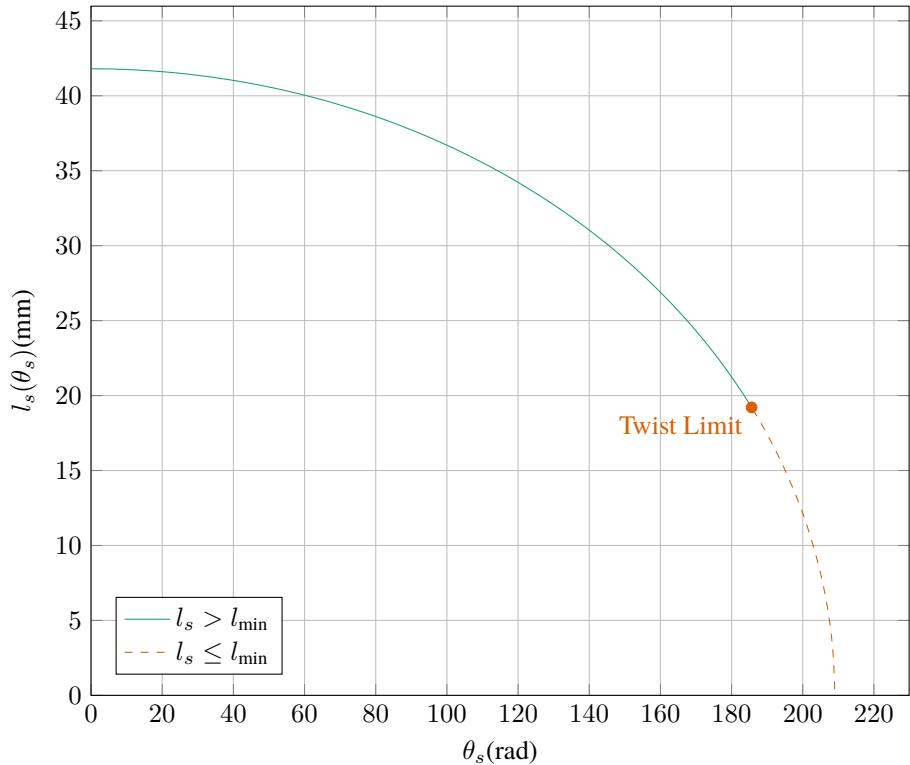


Figure 3.1: TSA string length against motor angle with coefficients from table 3.1.

3.1.2 Antagonistic Triad

As mentioned in the introduction, because the TSA provides only tensile force, a minimum of three actuators are required for a 2 DOF actuation system. These can be arranged in a triangular configuration to create an “antagonistic triad”, akin to the antagonistic pairs of muscles found in animals. In a pair arrangement, one actuator contracts while the other relaxes, but in a triad, up to two actuators can share the same action, and one actuator can be inactive if the other two are performing different actions. The geometric structure of the system as shown in figure 3.2 can be described with two equilateral triangles of inradius r on two planes separated in the z axis. The centroids are then connected via a universal joint from each plane normal to an intersecting point, let the vector $\theta = [\theta_1 \ \theta_2]$ denote the rotation of the second plane relative to the first, in the y and x axes around the intersecting point, and let l_1 and l_2 denote the normal distance from the intersection to the first and second plane centroids respectively. When $\theta = [0 \ 0]$ the triangles are parallel to each other. The distance between the vertex pairs of each triangle is then denoted as $[\lambda_1 \ \lambda_2 \ \lambda_3]$ for the “top”, “left” and “right” vertices of the triangles. When θ is changed, this will change λ_1 , λ_2 and λ_3 respectively.

To calculate the lengths of the strings for a given θ of the universal joint, we define a vector function $\Lambda(\theta) = [\lambda_1(\theta) \ \lambda_2(\theta) \ \lambda_3(\theta)]$ as follows.

$$\begin{aligned}
\lambda_1(\boldsymbol{\theta}) &= \sqrt{(l_1 + l_2 \cos \theta_1 \cos \theta_2 + r \cos \theta_1 \sin \theta_2)^2 \\
&\quad + (r - r \cos \theta_2 + l_2 \sin \theta_2)^2 \\
&\quad + (l_2 \cos \theta_2 \sin \theta_1 + r \sin \theta_1 \sin \theta_2)^2} \\
\lambda_2(\boldsymbol{\theta}) &= \sqrt{(a - b + c)^2 + (l_1 - d)^2 + e^2} \\
\lambda_3(\boldsymbol{\theta}) &= \sqrt{(a + b - c)^2 + (l_1 + d)^2 + e^2}
\end{aligned} \tag{3.3}$$

where:

$$\begin{aligned}
a &= -\frac{\sqrt{3}r(\cos \theta_1 - 1)}{2} \\
b &= l_2 \cos \theta_2 \sin \theta_1 \\
c &= \frac{r \sin \theta_1 \sin \theta_2}{2} \\
d &= \frac{\sqrt{3}r \sin \theta_1}{2} + l_2 \cos \theta_1 \cos \theta_2 - \frac{r \cos \theta_1 \sin \theta_2}{2} \\
e &= \frac{r \cos \theta_2}{2} - \frac{r}{2} + l_2 \sin \theta_2
\end{aligned}$$

Surface plots of the functions in (3.3) are shown in figure 3.3 for a domain of $[-\frac{\pi}{2}, \frac{\pi}{2}]$.

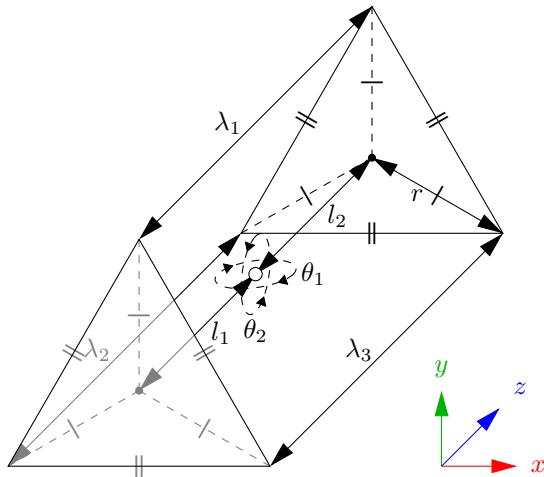


Figure 3.2: Kinematic diagram of the antagonistic triad, where the universal joint rotation is defined by $\theta_{1,2}$ on the y and x axes respectively, and the actuator lengths are defined by $\lambda_{1,2,3}$ for the “top”, “left” and “right” actuators. r and $l_{1,2}$ define the anchor points of the actuators.

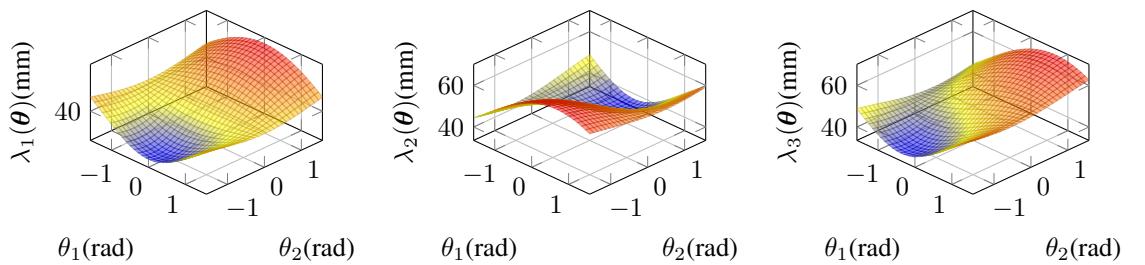


Figure 3.3: Surface plots of each element of the vector function $\Lambda(\boldsymbol{\theta})$, assuming coefficient values from table 3.1. Note that λ_2 and λ_3 are symmetric.

3.2 Control System

The control system is a four layer cascade design, joining an inverse dynamic control system [51], to the triad force controller in [52], to a proportional controller for each TSA. It uses feedback signals of the joint position from the accelerometers and TSA force from the load cells. A second order setpoint trajectory q is used as the input, which can either be pre-defined or generated dynamically from user input. Feedback is provided by the AUJ angular position θ as shown in figure 3.2, angular velocity $\dot{\theta}$, and TSA tension force \hat{f} . Figure 3.4 shows a complete block diagram of the control system.

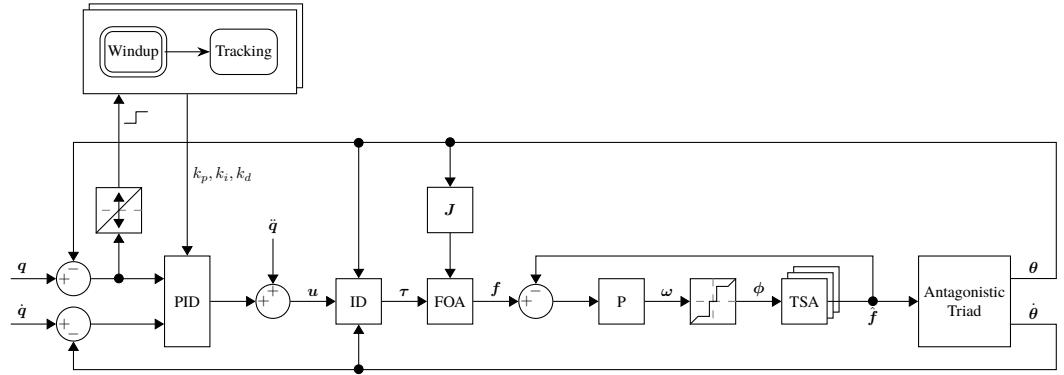


Figure 3.4: Block diagram of the complete experimental control system, excluding the hardware velocity controllers for the motors.

3.2.1 Actuated Universal Joint Position PID Controller with Acceleration Feedforward

Firstly, a PID controller is used to generate a control signal u with the input q as the set-point, and the AUJ angular position θ and velocity $\dot{\theta}$ as feedback, plus the addition of a feedforward term for the input acceleration \ddot{q} , i.e.

$$u = k_p (q - \theta) + k_i \left(\int_0^t (q - \theta) dt \right) + k_d (\dot{q} - \dot{\theta}) + \ddot{q}. \quad (3.4)$$

3.2.2 Inverse Dynamics

The control signal u from the PID controller is then converted to the desired AUJ torque τ as follows

$$\tau = D(\theta) u + C(\theta, \dot{\theta}) \dot{\theta} + G(\theta). \quad (3.5)$$

3.2.3 Twisted String Actuator Force Optimisation Algorithm

This uses a modified algorithm from [52], which proposes an inverse force transformation algorithm to control an antagonistic triad using force controlled linear actuators, to select

an optimal force vector from the desired joint torque. A force matrix \mathbf{F} is created from the torque input $\boldsymbol{\tau}$, jacobian J_Λ from the vector function Λ as defined in (3.3), and minimum force constant f_{\min} . The diagonal components $f_{i,i}$ are equal to f_{\min} , while the other elements in the column are based on a calculation using $J_{\Lambda_{-i,*}}$ where $-i$ is a row removed from the matrix.

$$\begin{aligned} J_\Lambda &= \begin{bmatrix} \frac{\partial \lambda_1}{\partial \theta_1} & \frac{\partial \lambda_2}{\partial \theta_1} & \frac{\partial \lambda_3}{\partial \theta_1} \\ \frac{\partial \lambda_1}{\partial \theta_2} & \frac{\partial \lambda_2}{\partial \theta_2} & \frac{\partial \lambda_3}{\partial \theta_2} \end{bmatrix} \\ \boldsymbol{\gamma}(i) &= -J_{\Lambda_{-i,*}}^{-\top} \left(J_{\Lambda_{i,*}}^\top f_{\min} + \boldsymbol{\tau} \right) \\ \mathbf{F}(\boldsymbol{\tau}, \boldsymbol{\theta}) &= \begin{bmatrix} f_{\min} & \boldsymbol{\gamma}(2)_1 & \boldsymbol{\gamma}(3)_1 \\ \boldsymbol{\gamma}(1)_1 & f_{\min} & \boldsymbol{\gamma}(3)_2 \\ \boldsymbol{\gamma}(1)_2 & \boldsymbol{\gamma}(2)_2 & f_{\min} \end{bmatrix} \end{aligned} \quad (3.6)$$

The following algorithm then selects one column of \mathbf{F} to be the output force vector \mathbf{f} , where \top and \perp are boolean *true* and *false* respectively

```

1:  $s \leftarrow [\top \ \top \ \top]$ 
2: if  $f_{23} > f_{\min}$  then  $s_2 \leftarrow \perp$  else  $s_3 \leftarrow \perp$  end if
3: if  $f_{31} > f_{\min}$  then  $s_3 \leftarrow \perp$  else  $s_1 \leftarrow \perp$  end if
4: if  $f_{12} \geq f_{\min}$  then  $s_1 \leftarrow \perp$  else  $s_2 \leftarrow \perp$  end if
5: for  $i = 1$  to  $3$  do
6:   if  $s_i \rightarrow \top$  then  $\mathbf{f} \leftarrow \mathbf{f}_{*,i}$  end if
7: end for
```

3.2.4 Twisted String Actuator Force Proportional Controller

The selected forces are then used as an input to a P controller with gain k_{ps} using the measured load cell forces $\hat{\mathbf{f}}$ as feedback. The output from this can then be used to control the top, left and right TSA motors, corresponding to the actuators in figure 3.2.

3.2.4.1 Simulation Current Control

In the simulation, each TSA was modelled as a state-space system which takes motor current u as an input and outputs y as the TSA tension force. [46] defines it as such, where J is the motor inertia, C is the motor coulomb friction (modified from viscous friction as the motor only has dry friction), K_t is the motor torque constant, and K_L is the load stiffness. As the original definition is for a fixed load l_u distance from the motor a modified model is required which takes into account the varying length between the motor and load defined by $\Lambda(\boldsymbol{\theta})$. A saturation function, with the compact notation $\text{sat}_x^y z = \max(x, \min(y, z))$ is used to prevent incorrect compression forces when the string is slack. All of the motor coefficients were taken from the Faulhaber 1724TSR datasheet [53] as this is the motor used in

the experimental model. An estimated value is used for the load stiffness, this was chosen to be a high number as the model is expected to be very stiff.

$$\begin{aligned}
h(\theta_s) &= \frac{\theta r_s^2}{\sqrt{l_u^2 - \theta_s^2 r_s^2}} \\
k(\theta_s, \boldsymbol{\theta}) &= \lambda_n(\boldsymbol{\theta}) - \sqrt{l_u^2 - \theta_s^2 r_s^2} \\
\dot{\boldsymbol{x}} &= \begin{bmatrix} x_2 \\ -\frac{K_L}{J} h(x_1) k(x_1, \boldsymbol{\theta}) - \frac{C}{J} \operatorname{sgn}(x_2) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_t}{J} \end{bmatrix} u \\
y &= K_L \operatorname{sat}_0^\infty k(x_1, \boldsymbol{\theta})
\end{aligned} \tag{3.7}$$

The state space model was then adapted to include constraints on motor velocity and acceleration set by the motor controller in order to keep the motor within design limits, by replacing $\dot{\boldsymbol{x}}$ with $\dot{\boldsymbol{x}}'$ which contains saturation functions for maximum motor velocity v_s and acceleration α_s .

$$\dot{\boldsymbol{x}}' = \begin{bmatrix} \operatorname{sat}_{-\omega_s}^{\omega_s} \dot{x}_1 \\ \operatorname{sat}_{-\alpha_s}^{\alpha_s} \dot{x}_2 \end{bmatrix} \tag{3.8}$$

3.2.4.2 Experimental Velocity Control with Deadband Compensation

Due to a controller deadband within $\pm 10 \text{ min}^{-1}$, an adjustable deadband compensator is used,

$$\phi_i(\omega_i) = \begin{cases} 10 & 10 \leq \omega_i < h \\ -10 & -h < \omega_i \leq -10 \\ 0 & h \leq \omega_i \leq -h \\ \omega_i & \text{otherwise} \end{cases} \tag{3.9}$$

where ϕ_i is the compensator for the controller i . An adjustment value $h \in [0, 10]$ changes the threshold at which the compensator switches on and off, allowing for a small deadband to remain.

The result from the TSA is then a compressive force acting between each of the three TSA and its corresponding endpoint on the Antagonistic Triad, imparting a torque on the axes of the universal joint.

Table 3.2: PID gains in the simulation and experiment.

Gain	Value	
	Simulation	Experiment*
k_p	800	3×10^4
k_i	3000	350
k_d	50	50
k_{p_s}	19	100

* Tracking mode, see section 3.3.2.

3.3 Simulation & Experimental Results

3.3.1 Experimental Setup

For the experimental validation, a physical prototype of the mechanism was constructed with coefficients from table 3.1 as design parameters. This was mounted vertically, in order for the Inertial Measurement Unit (IMU) to measure the orientation of the universal joint. The TSA mechanisms consist of a compact high torque motor attached to the base segment, with a string clamp attached to the motor shaft. On the follower segment, a load cell is mounted on top of a universal joint to ensure a purely axial load, with a capstan bolt attached to the load cell. The string itself is attached to the clamp at both ends using two grub screws for extra security and easy adjustment, and looped through the hole in the capstan bolt. The total mass of the prototype, excluding the mount, is ≈ 176 g. Figure 3.8 details the construction of the experiment with all the constituent parts.

3.3.2 Windup & Tracking States

The tracking controller is activated after a “wind up” stage when the operating conditions are met. The controller parameters for the Windup and Tracking states are $k_p = 800$, $k_i = 3000$, and $k_d = 3 \times 10^4$, $k_{p_s} = 350$ respectively. This transition trigger is defined on a per-axis basis, as the first zero crossing of the angle error (as $q = 0$ this is effectively θ). A graph showing the difference this state change makes to the AUJ orientation is shown in figure 3.5.

3.3.3 Results

Figure 3.6 plots the tracking response of both the simulation and experiment. Three trajectories were created to test the capabilities of the mechanism. Two were only in one axis of the Universal Joint, and the third was in both axes. The deflection angle range was limited to $\pm 11^\circ$ on a single axis, and $\pm 6^\circ$ on both axes. This was partly due to the low value of l_u , but can easily be extended by increasing this value, a low l_u was chosen as it resulted in easier installation.

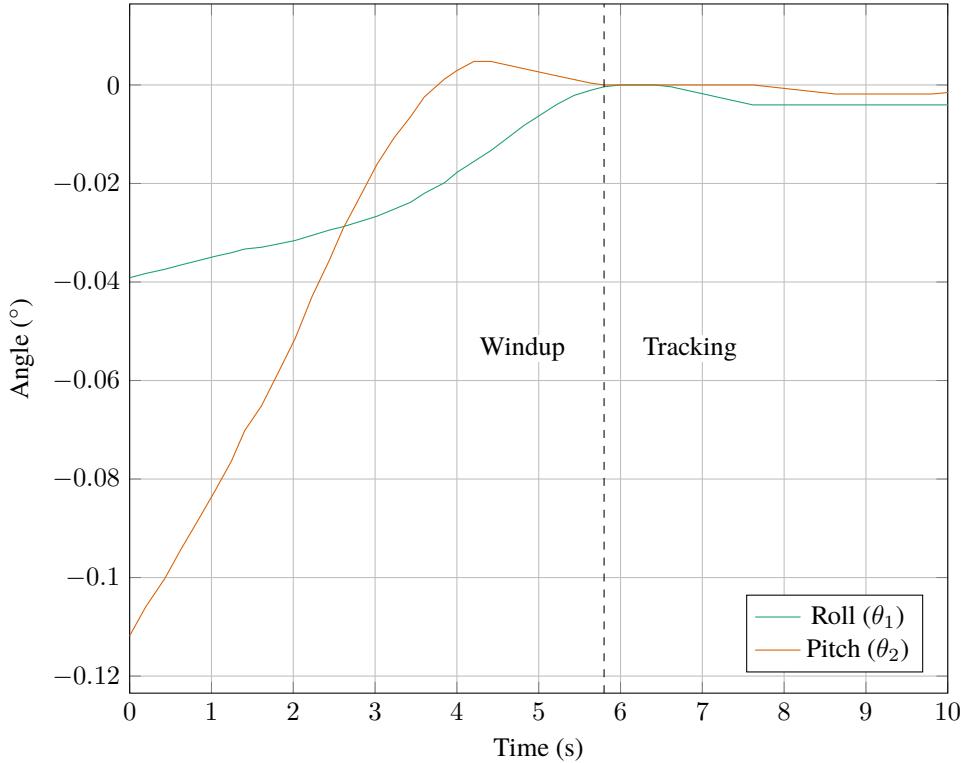


Figure 3.5: The initial startup of the mechanism showing the transition between “windup” and “tracking” states.

3.4 Performance Comparison

To compare the performance of a TSA AUJ against alternatives, we can measure two metrics, the maximum tension force f_{\max} and maximum stroke velocity \dot{p}_{\max} . This corresponds to the equivalent of maximum torque and maximum velocity in a rotary motor, a larger f_{\max} would be able to actuate a larger follower mass, and a larger \dot{p}_{\max} would be able to rotate the AUJ more quickly. The alternatives chosen for comparison are leadscrews of various rod diameters d_m and pitches λ , and a “direct drive” where the motor is rotating the universal joint directly without any reduction or motion transformation.

3.4.1 Twisted String Actuator

For the TSA metrics, the equations from [46], in particular $h(\theta)$ and $k(\theta)$ as used for the state space, which can be used to determine f_{\max} and \dot{p}_{\max} . By extracting coefficient r_s as an input to make $f(p, r_s)$ and $\dot{p}(\dot{\theta}, p, r_s)$ the performance of different string thicknesses can be compared for a given unwound length l_u and $\tau_{\max}, \dot{\theta}_{\max}$ over the range of the contraction length p .

$$\begin{aligned}
k^{-1}(p) &= \pm \frac{\sqrt{p(2l_u - p)}}{r_s} \\
h^{-1}(\theta) &= \frac{\sqrt{l_u^2 - r_s^2 \theta^2}}{r_s^2 \theta} \\
f(p) &= h^{-1}(k^{-1}(p)) = \pm \frac{\sqrt{(l_u - p)^2}}{r_s \sqrt{p(2l_u - p)}} \\
f_{\max} &= f(p)\tau_{\max}
\end{aligned} \tag{3.10}$$

$$\begin{aligned}
\dot{k}(\dot{\theta}, \theta) &= \frac{\dot{\theta} r_s^2 \theta}{\sqrt{l_u^2 - r_s^2 \theta^2}} \\
\dot{p}(\dot{\theta}, p) &= \dot{k}(\dot{\theta}, k^{-1}(p)) = \pm \frac{\dot{\theta} r_s \sqrt{p(2l_u - p)}}{\sqrt{(l_u - p)^2}} \\
\dot{p}_{\max} &= \dot{p}(\dot{\theta}_{\max}, p)
\end{aligned} \tag{3.11}$$

3.4.2 Leadscrew

For the leadscrew metrics, the raising torque calculation [54] can be used as the absolute value of f_{\max} , since the TSA only operates in tension, which can be used to determine the same metrics. The performance of different screw diameters d_m and leads λ can then be compared for a given τ_{\max} and coefficient of friction μ . \dot{p}_{\max} is then calculated by multiplying λ with $\dot{\theta}_{\max}$. The performance of different λ can then be compared for a given $\dot{\theta}_{\max}$.

$$\begin{aligned}
|\tau(f)| &= \frac{d_m f (\lambda + \pi d_m \mu)}{2(\pi d_m - \lambda \mu)} \\
|f(\tau)| &= \frac{2\tau(\pi d_m - \lambda \mu)}{d_m (\lambda + \pi d_m \mu)} \\
f_{\max} &= |f(\tau_{\max})|
\end{aligned} \tag{3.12}$$

$$\begin{aligned}
\dot{p}(\dot{\theta}) &= \lambda \frac{\dot{\theta}}{2\pi} \\
\dot{p}_{\max} &= \dot{p}(\dot{\theta}_{\max})
\end{aligned} \tag{3.13}$$

3.4.3 Direct Drive

The direct drive metrics are trivially calculated using the lever force and tangential velocity that would be generated at the endpoint of a linear actuator able to impart equivalent angular velocity and torque on the universal joint.

$$f_{\max} = \frac{\tau_{\max}}{\sqrt{l_2^2 + r^2}} \tag{3.14}$$

$$\dot{p}_{\max} = \dot{\theta}_{\max} \sqrt{l_2^2 + r^2} \quad (3.15)$$

3.4.4 Comparison

As the values for τ_{\max} and $\dot{\theta}_{\max}$ for the TSA depend on p , but remain constant for the lead-screw, the performance of the TSA is going to be better or worse than a given leadscrew depending on the value p . Figure 3.7 compares the TSA configuration using the coefficients from table 3.1 against a number of common leadscrew configurations that are practical for the dimensions of the AUJ. The TSA outperforms or underperforms different leadscrew configurations depending on p . In simpler terms, the performance of the TSA is dependent on the contraction length. The maximum linear velocity increases with the contraction length, and the maximum tension force decreases with contraction length, both in a non-linear fashion.

3.5 Conclusion

This research has demonstrated the robust control of the orientation of a universal joint using TSA in an antagonistic triad configuration, with a low tracking error ($\pm 1.8^\circ$) at low speed (max. 0.6° s^{-1}) in 2 DOF control with a vertical base orientation. It has also compared the performance of the system to alternative actuation methods. Future developments would include improvements to the orientation sensors, the data from the IMU proved to be unreliable and of poor resolution, so either a superior IMU will be used or an alternative method for sensing the universal joint orientation will be investigated, such as Linear Variable Differential Transformers (LVDTs), hall effect sensors or potentiometers. The system was tested in a vertical base orientation in order for the IMU to be able to measure the joint orientation. With suitable modifications to the inverse dynamics function and a known base orientation, it will be possible to test the controller at non-vertical base orientations. We will also increase the joint velocity, joint range and follower mass (m) in order to test the performance of the system under more strenuous conditions. The controller performance will also be quantified, examining the open and closed loop bandwidth. Eventually, we wish to develop a system comprised of multiple segments, to demonstrate its suitability for applications such as mobile snake robots or continuum robots.

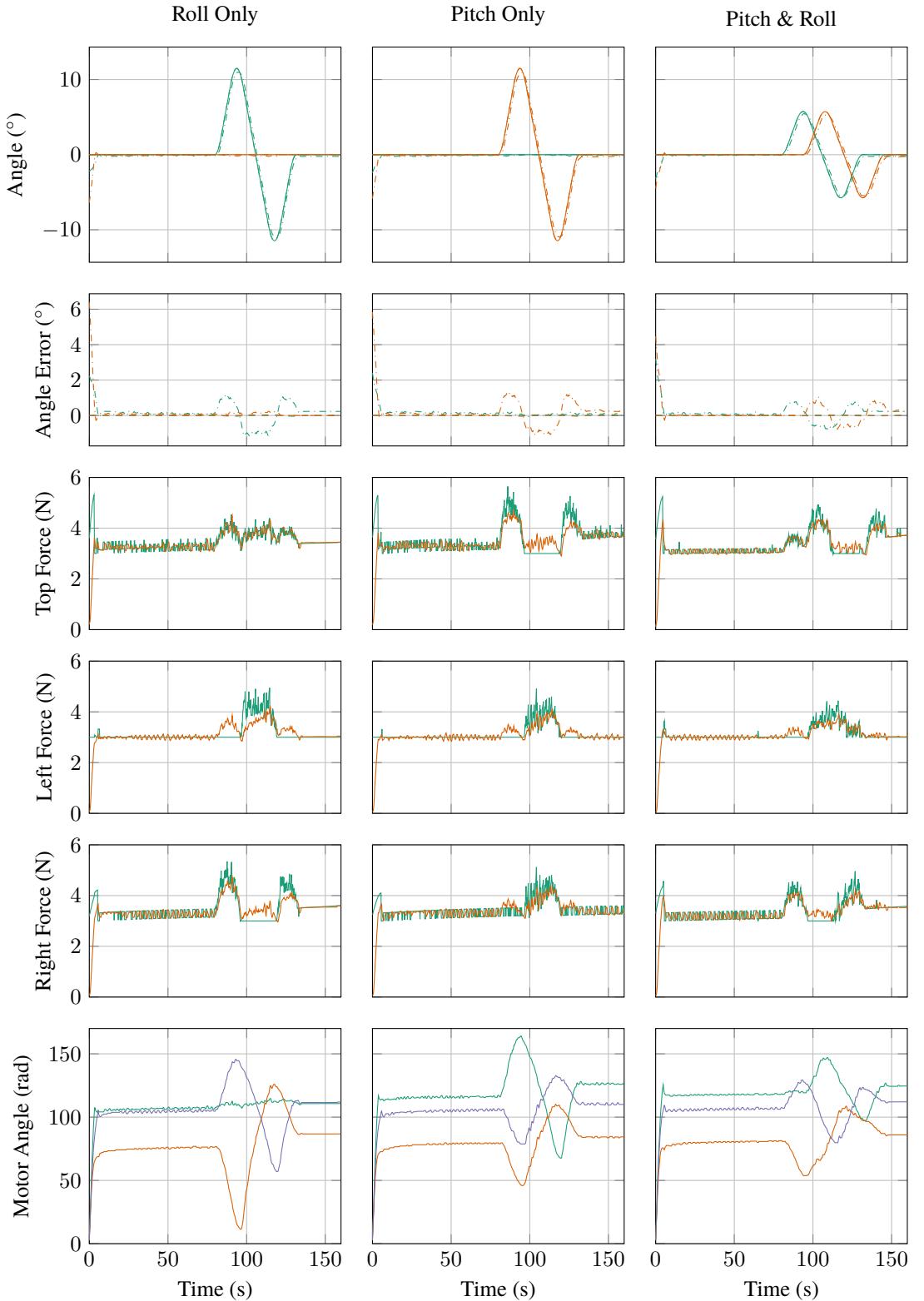


Figure 3.6: Plots of the setpoint and response for three different trajectories, one on only the roll axis (column 1), one on only the pitch axis (column 2), and one on both axes (column 3). Plots include AUJ orientation, forces at the top, left and right TSA, and the motor positions.

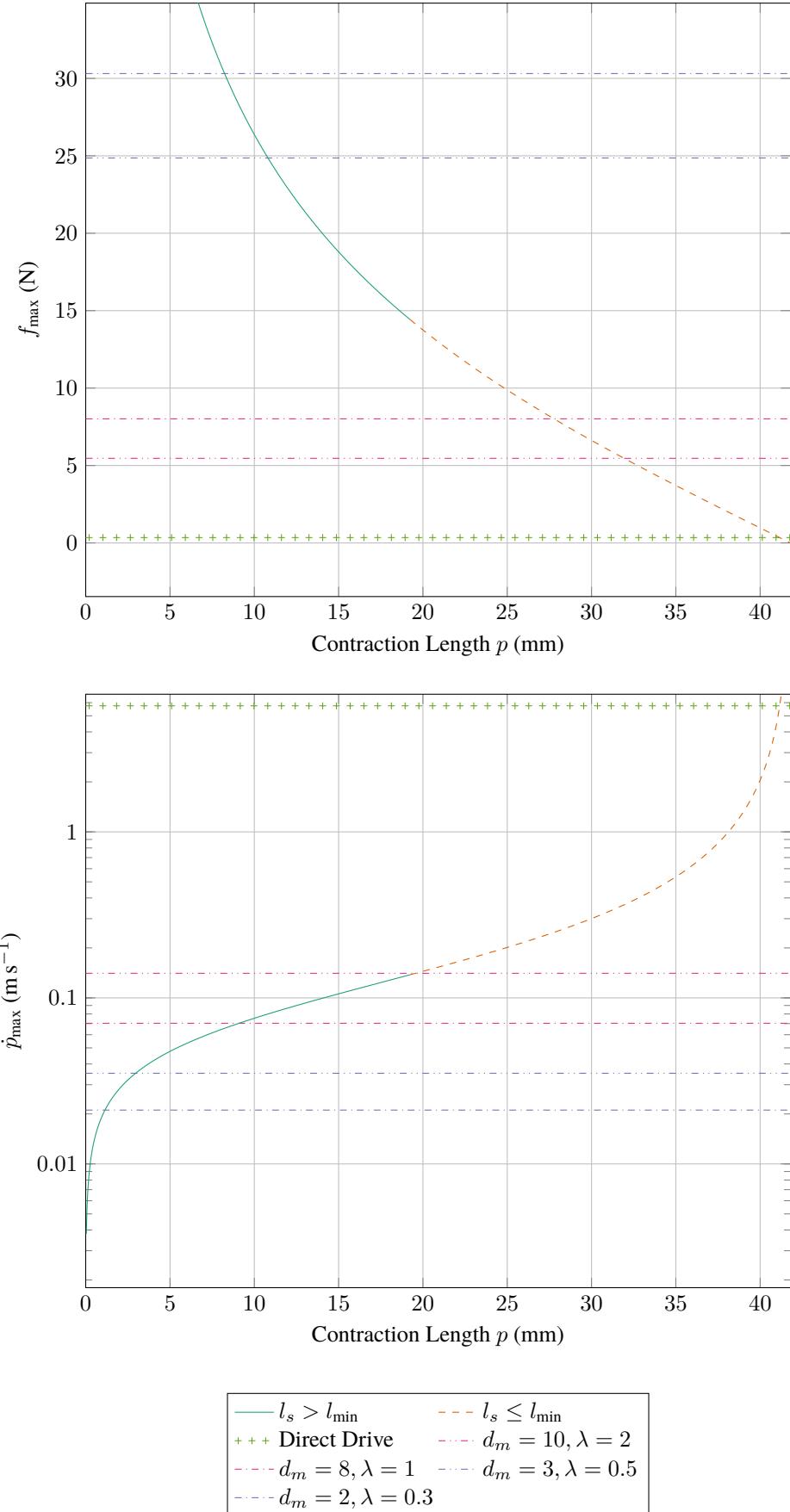


Figure 3.7: Performance comparison of the TSA configuration using coefficients from table 3.1 to various leadscrew configurations with different d_m and λ , and the direct drive, where $\mu = 0.1$ for the leadscrews.

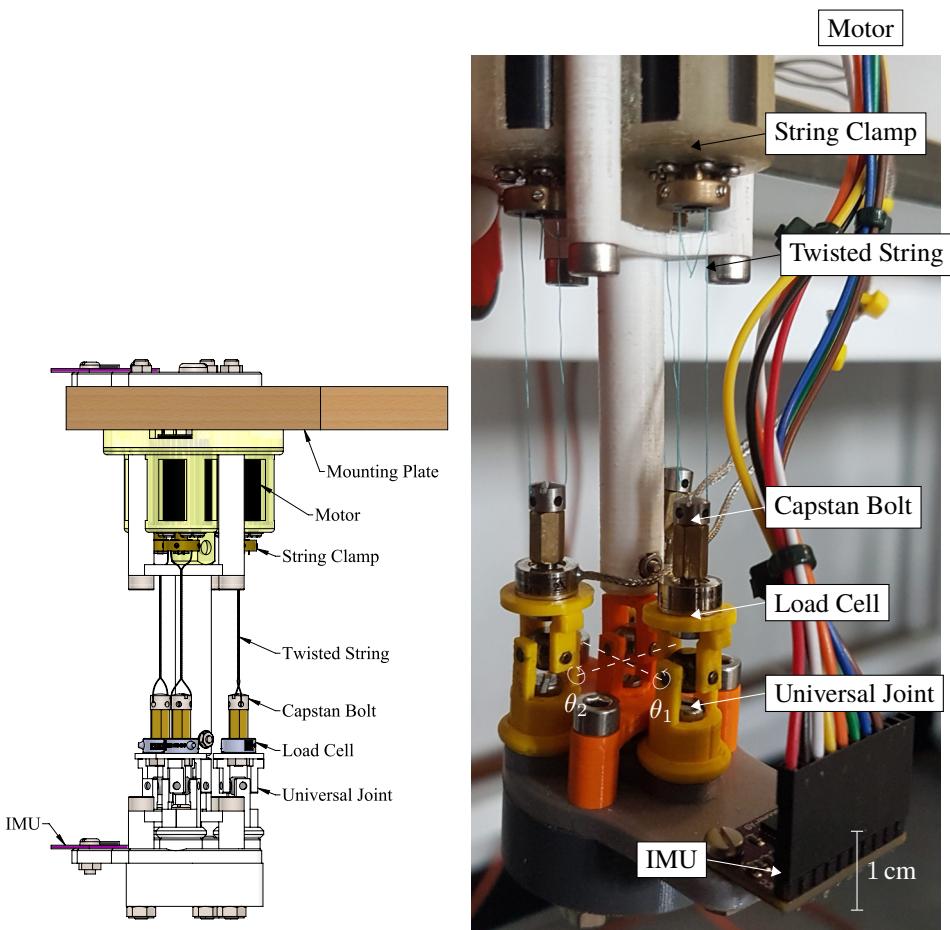


Figure 3.8: Annotated photograph of the experimental model, with the roll θ_1 and pitch θ_2 axes marked.

Chapter 4

Optimisation Study for Multi-Segment Tails for COM Control

4.1 Introduction

One of the most noticeable differences between robotic and animal tails, as discovered in chapter 2, is the far greater number of segments in most animal tails when compared to robot tails. The aim of this chapter was to discover if there is any merit to using multiple segments for tracking a COM trajectory using closed loop control, since this would be similar to the control system the tail will use in the experiments in chapter 6. To this end, a simulation study was constructed in MATLAB where a tail of the same mass and length is divided into 1-6 segments, and two different “inverse kinetic” algorithms are used, which are designed to minimise joint velocity, joint torque or both. Results from this study determined whether it was worth pursuing a multi-segment approach for the experimental work in chapter ??.

4.1.1 Existing Study for Other Tail Function

[55] conducted a simulation study using a tail with 1-6 DOF, with 1 DOF joints that moved the tail horizontally. The aim was to see if a tail with more segments could increase the yaw magnitude of a quadrupedal robot, in the manner of the *Terrestrial → Turning → Initiation* robots found in section 2.3.2.4. It was discovered that in theory, a tail with more segments and the same mass, length and maximum joint velocity would increase the yaw magnitude, as shown in figure 4.1.

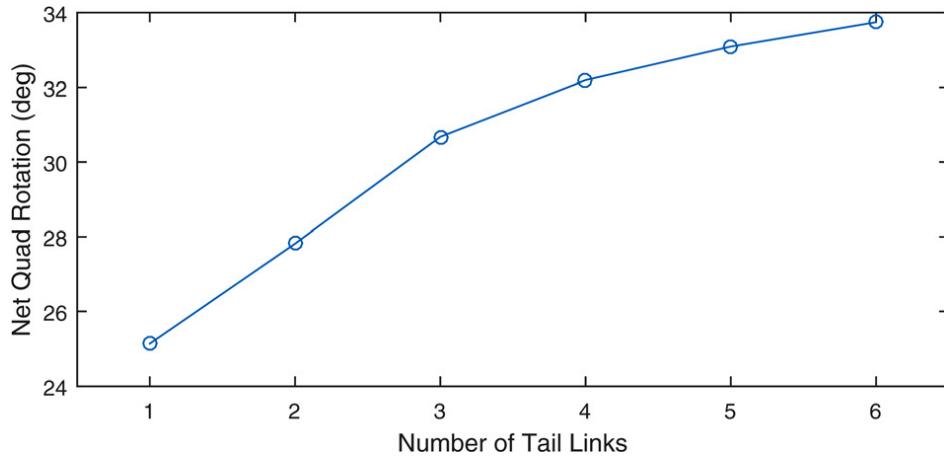


Figure 4.1: Chart from [55] showing how the net quadruped rotation increases with the number of segments.

While this study did suggest multiple segments could improve the performance of a tail for a specific function, there are important differences between the tail function of the study and the simulation conducted in this chapter. It used an open loop controller which constrained the maximum deflection $\Delta\theta$ of each joint to $\frac{360}{n+1}$ where n is the number of segments, and the duration of the trajectory ΔT to $\frac{2\Delta\theta}{|\dot{\theta}_{\max}|}$ where $\dot{\theta}_{\max}$ is the maximum tail velocity. In contrast, this simulation uses a closed loop controller that is tracking a specific COM trajectory each time, it is not trying to increase the magnitude of an output. In fact, it can be seen as having the opposite goal of this study, keeping the output constant and trying to optimise the inputs, tail velocity and tail torque.

4.2 Mathematical Definition of the Model

Consider a tail with an arbitrary number of segments as a chain of bodies connected by revolute joints, where l are the lengths of the bodies, m are the masses of the bodies, l_c are the offsets of the COM from the origin of each body along the axis of the chain (assuming the COM remains on that axis for the sake of simplicity), and θ are the angles of each revolute joint, as shown in figure 4.2.

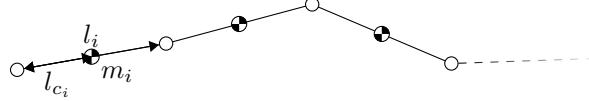


Figure 4.2: Diagram of a 2D tail, with all parameters annotated.

The transformation matrices for a given body's origin and COM can then be computed as follows:

$$T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & l_i \\ \sin \theta_i & \cos \theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

$$T_{c_i} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & l_{c_i} \\ \sin \theta_i & \cos \theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The forward kinematics for the tail are then computed. This will give the position of the endpoint of the tail for a given set of joint angles θ :

$$T(\theta) = \prod_{i=1}^n T_i \quad (4.2)$$

A similar equation can be used for the forward *kinetics*, with the addition of the body masses and COM offsets as parameters. This will give the position of the COM for the entire tail:

$$R(\theta) = \frac{\sum_{i=1}^n m_i \prod_{j=1}^{i-1} (T_j) T_{c_i}}{\sum_{i=1}^n m_i} \quad (4.3)$$

4.3 Inverse Kinetics

As the forward kinetics can be computed in a very similar fashion to the inverse kinematics, it follows that the inverse kinetics can be computed in a similar fashion [56]. Firstly, define the Jacobian:

$$\mathbf{J}_R = \begin{bmatrix} \frac{\partial r_{13}}{\partial \theta_1} & \frac{\partial r_{13}}{\partial \theta_1} & \dots & \frac{\partial r_{13}}{\partial \theta_n} \\ \frac{\partial r_{23}}{\partial \theta_1} & \frac{\partial r_{23}}{\partial \theta_1} & \dots & \frac{\partial r_{23}}{\partial \theta_n} \end{bmatrix} \quad (4.4)$$

Then, assuming a COM position defined by $\begin{bmatrix} q_x & q_y \end{bmatrix}^\top$ the COM velocity can be calculated using the Jacobian:

$$\begin{bmatrix} \dot{q}_x \\ \dot{q}_y \end{bmatrix} = \mathbf{J}_R \cdot \dot{\theta} \quad (4.5)$$

To get the joint velocities (or torques) for a given COM velocity, any inverse kinematic algorithm can be used, treating the COM in a similar fashion to a 3 DOF end effector. The COM velocity $\dot{\mathbf{q}}$ can be calculated from a Proportional Integral Derivative (PID) controller which has the COM error as an input.

4.4 Inverse Kinematic Algorithms

Two different inverse kinematic algorithms were chosen for the experiment that would be adapted into inverse kinetic algorithms, a simple Damped Least Squares (DLS) algorithm able to minimise joint velocity, and a more complex algorithm that uses a weighting constant to minimise both joint velocity and joint torque.

4.4.1 Damped Least Squares

DLS, also known as Levenberg-Marquardt [57], is the simplest algorithm to implement. In this implementation, it is a single-step analytic function, requiring no iteration as the error term is calculated using a PID controller based on the current state of the system. It is an improvement on the pseudoinverse or transpose method, which have issues with singularities:

$$\dot{\theta} = \mathbf{J}^\top (\mathbf{J} \mathbf{J}^\top \lambda^2 \mathbf{I}) \begin{bmatrix} \dot{q}_x \\ \dot{q}_y \end{bmatrix} \quad (4.6)$$

Where \mathbf{I} is the identity matrix and $\lambda \in \mathbb{R}^+$ is a suitable damping constant.

While it is easy to implement, DLS by itself can only minimise a single metric at once, in this case the joint velocity 2-norm $\|\dot{\theta}\|_2$.

4.4.2 Weighted Quadratic Programming

In [58] an algorithm was designed based on the pseudoinverse method [57], that allowed for a weighted output that balances both the velocity 2-norm and torque 2-norm $\|\tau\|_2$ by using a weighting constant $\alpha \in [0, 1]$:

$$\begin{aligned}
\ddot{\mathbf{q}}_0 &= \begin{bmatrix} \ddot{q}_x \\ \ddot{q}_y \end{bmatrix} - \mathbf{J}\dot{\boldsymbol{\theta}} \\
\mathbf{J}_\tau &= \mathbf{J}D(\boldsymbol{\theta})^{-1} \\
\boldsymbol{\tau} &= \alpha \left(D(\boldsymbol{\theta}) \mathbf{J}^+ \ddot{\mathbf{q}}_0 + D(\boldsymbol{\theta}) (\mathbf{I} - \mathbf{J}^+ \mathbf{J}) \mathbf{J}^+ \begin{bmatrix} \dot{q}_x \\ \dot{q}_y \end{bmatrix} + (C(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + G(\boldsymbol{\theta})) \right) \\
&\quad + (1 - \alpha) \left(\mathbf{J}_\tau^+ \ddot{\mathbf{q}}_0 + \mathbf{J}_\tau^+ \mathbf{J}_\tau (C(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + G(\boldsymbol{\theta})) \right)
\end{aligned} \tag{4.7}$$

Where \square^+ is the moore-penrose pseudoinverse, and $D(\boldsymbol{\theta})$, $C(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$, $G(\boldsymbol{\theta})$ are the lagrangian inertia matrix, coriolis/centrifugal vector, and gravity vector of the system respectively.

Then [59] took this idea and formulated an alternative to the pseudoinverse using quadratic programming, allowing the inclusion of position, velocity and acceleration limits, at the expense of an iterative algorithm:

$$\begin{aligned}
\xi_i^+ &= \min \left\{ \kappa_p (\mu \theta_i^+ - \theta_i), \kappa_v (\dot{\theta}_i^+ - \dot{\theta}_i), \ddot{\theta}_i^+ \right\} \\
\xi_i^- &= \max \left\{ \kappa_p (\mu \theta_i^- - \theta_i), \kappa_v (\dot{\theta}_i^- - \dot{\theta}_i), \ddot{\theta}_i^- \right\} \\
\mathbf{Q} &= \alpha \mathbf{I} + (1 - \alpha) D(\boldsymbol{\theta})^2 \\
\mathbf{p} &= \alpha \lambda \dot{\boldsymbol{\theta}} + (1 - \alpha) D(\boldsymbol{\theta})^\top (C(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + G(\boldsymbol{\theta})) \\
\ddot{\boldsymbol{\theta}} &= \text{minimise } \frac{1}{2} \ddot{\boldsymbol{\theta}}^\top \mathbf{Q} \ddot{\boldsymbol{\theta}} + \mathbf{p}^\top \ddot{\boldsymbol{\theta}} \\
&\text{subject to } \mathbf{J} \ddot{\boldsymbol{\theta}} = \ddot{\mathbf{q}}_0, \\
&\quad \xi^+ \leq \ddot{\boldsymbol{\theta}} \leq \xi^-
\end{aligned} \tag{4.8}$$

where λ is a design constant similar to what is used for the DLS method, and ξ^\pm define the position θ^\pm , velocity $\dot{\xi}^\pm$ and acceleration $\ddot{\xi}^\pm$ limits. $\kappa_{p,v} \in \mathbb{R}^+$ define the declaration gain to the position and velocity limits, and $\mu \in (0, 1)$ defines the “critical area” in terms of joint position where the joints will be decelerated when they enter. The COM acceleration $\ddot{\mathbf{q}}$ can be calculated from a PID controller with an acceleration feedforward term, which has the COM error as an input.

4.5 Simulation Setup

The simulation used the `rigidBodyTree` object found in MATLAB, which allows for dynamics simulations to be performed based on a physical description of bodies and joints. Each simulation was performed using the following steps:

1. A `rigidBodyTree` was generated in MATLAB with 1-4 bodies connected by revolute joints, with the appropriate mass and length.

2. The lagrangian dynamic functions ($D(\theta), C(\theta, \dot{\theta}), G(\theta)$) were calculated based on the properties of the `rigidBodyTree`.
3. An `ode45` solver ran with both the DLS and **QP!** (**QP!**) inverse kinetic controllers, recording the joint velocities and torques.

4.5.1 Model Parameters

Each model had a total mass of 100 kg and a total length of 10 m. These would be divided by the number of segments, so the tail would always have the same mass and length. These numbers were chosen entirely arbitrarily, and had no bearing on the values to be used in the payload stability experiments in chapter 6. For both PID controllers, simple proportional controllers were found to be sufficient, with $k_p = 50$. Standard gravity was defined to be in the $-y$ direction, as shown in figure 4.3.

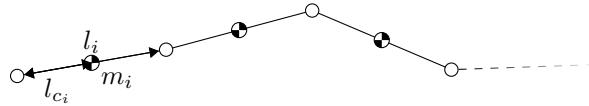


Figure 4.3: Diagram of the 1,2,3 and 4 segment tails, showing the gravity direction.

4.5.2 COM Trajectory

The COM x component was defined as the gaussian membership function $g(t) = \exp\left(\frac{-(t-c)^2}{2\sigma^2}\right)$ where $c = 5, \sigma = 1$, as shown in figure 4.4. This would allow for a smooth trajectory that is easily defined. The COM y component was left uncontrolled, the inverse kinetics functions were allowed to choose the most appropriate value to minimise their goals. This reflected the Three Dimensional (3D) equivalent of the payload stability experiments in chapter 6, where the z component is also ignored.

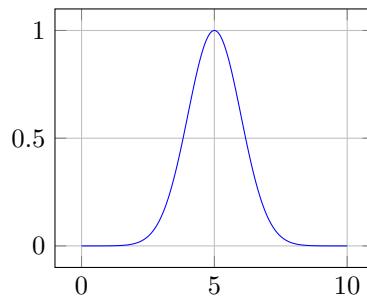


Figure 4.4: Gaussian membership function with $c = 5, \sigma = 1$.

4.6 Results

When attempting to run the **QP!** algorithm with $\alpha < 1$, except with a single segment where there is no optimisation possible, it was not possible to achieve a stable result. Therefore,

From the results in figure 4.5 and figure 4.6, the following trends can be noted:

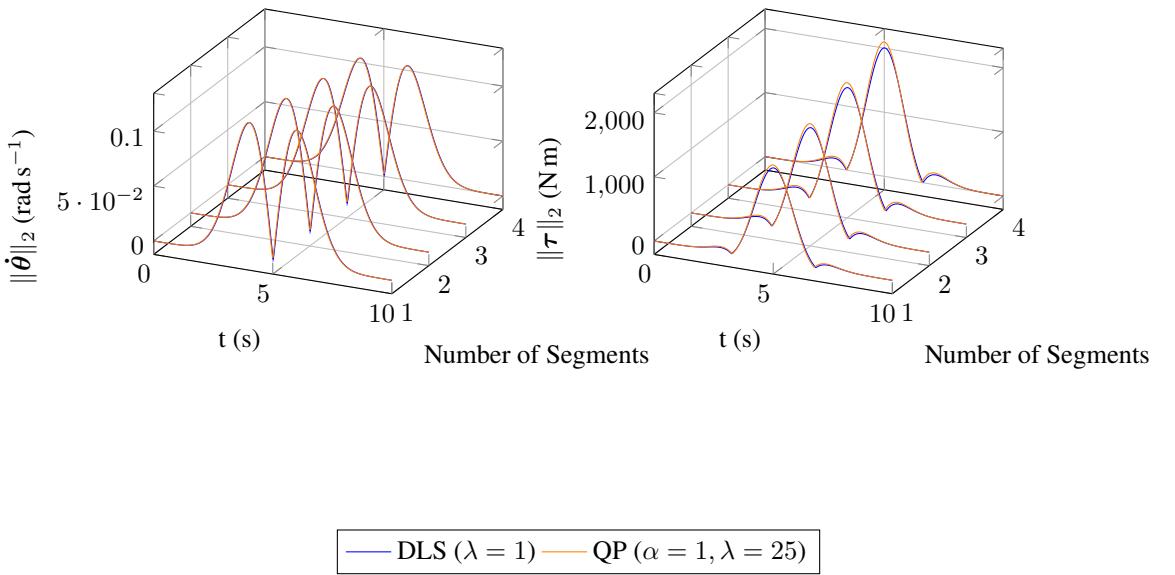


Figure 4.5

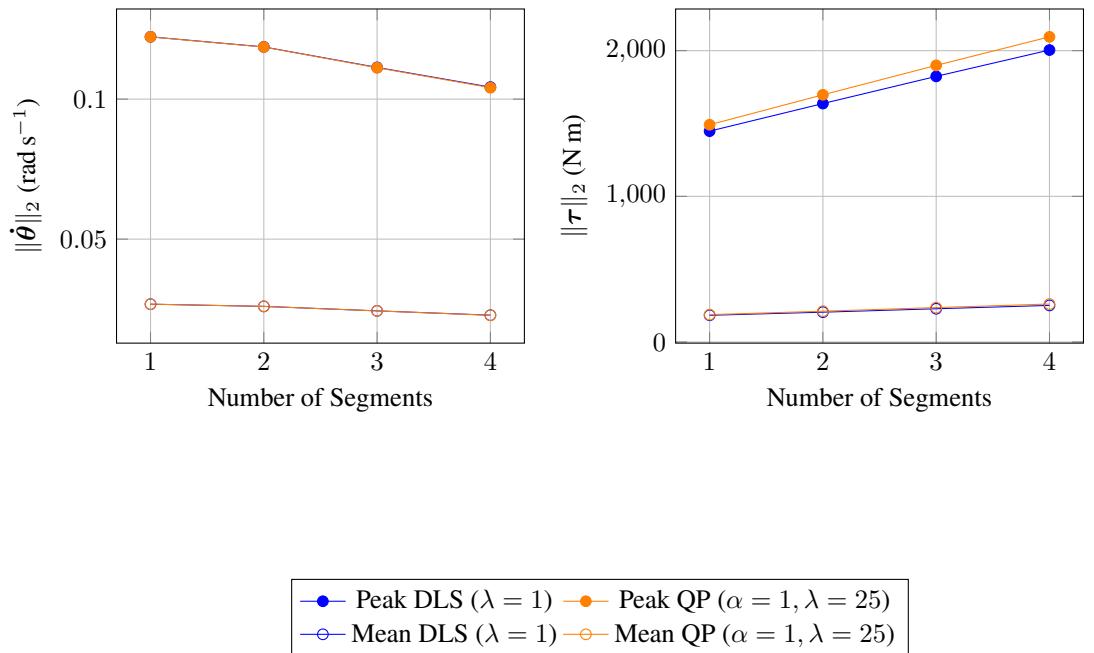


Figure 4.6

- The joint velocity 2-norm $\|\dot{\theta}\|_2$ decreases as the number of segments increase, under both DLS and QP! algorithms.
- The joint torque 2-norm $\|\tau\|_2$ increases as the number of segments increase, under both DLS and QP! algorithms.

4.7 Conclusion and Discussion

In this chapter, a simulation study has been conducted that explores if there are performance improvements in using multi segment tails for the payload stability experiments in chapter 6. A simulation model has been constructed, and two inverse kinetics algorithms, DLS and QP!, have been used to track a gaussian COM trajectory on a single axis.

Based on the research conducted in this chapter, it appears the advantages to using multi segment tails for COM tracking are marginal at best, and counterproductive at worst, with only an XX% reduction in peak velocity 2-norm for 4 segments in stead of a single segment, and coupled with a XX% increase in peak torque 2-norm, which is typically a more limiting factor for actuator selection. However, as noted, there were issues with the **QP!** algorithm that prevented effective use of the torque reduction???

Chapter 5

Creating a Configurable Payload for Instability Experiments

In order to generate a diverse set of test data for the payload stability experiments in chapter 6, where a robot arm will carry a payload with and without a robot tail to control stability, a configurable payload was conceived, an object that could be configured to have different masses and COM relative to the geometric centre of the object. A series of test points can then be generated which have a specific mass and COM, and a matching algorithm can be used to find the configuration that mostly closely matches these parameters. The experiments can then be run with each of these test points to generate the test data. In this chapter, an abstract mathematical model of a configurable payload is conceived, which the mass and COM of a specific permutation of the payload can be derived from. Then three separate sets of “test points” are considered to cover a wide range of the available configuration space within the chosen robot arms performance limits, using a combined mass and COM vector as a target which finds the closest permutation using a search function, or extrema which maps onto a specific permutation. For target test points, two search methods are examined, one brute force method for small permutation sets and one for larger sets where a brute force method would be computationally intractable. Then the chosen implementation is described, including the number of block materials, the number of blocks, the size and design of the container and the chosen search method. The results of this implementation for all test point sets are then tabulated and graphed. In the discussion, the limitations of the mass and COM range are considered, and potential designs to expand the range and scope of the experiments in chapter 6 are conceived, specifically improving the range of the COM and simulating dynamic loads where the COM changes over time.

5.1 Introduction

The payload consists of a matrix of cubes of various materials packed tightly into a 3D printed container. The cubes are designed to be changed after each experimental run to alter the mass and COM of the payload. A lid on the container prevents the cubes from falling out during the experiment, and the exterior design of the box may accommodate additional features to improving the handling of the payload by the robot arm. An abstract model of this is shown in figure 5.1.

The test points can then be generated by considering the *Configuration Space* of the payload design, i.e. how many permutations can be generated given a $n \times n \times n$ matrix of cubes, where each cube can be a number of different materials. The mass and COM can then be calculated for each permutation, taking into account the material density and mass and COM of the container. “Extrema” test points can then be found simply by finding the permutation with the maximum or minimum mass and COM, or combination thereof. Depending on the number of permutations, a search method can then be used that accepts an arbitrary mass and COM as a target, and finds the nearest permutation to that target for other test point sets.

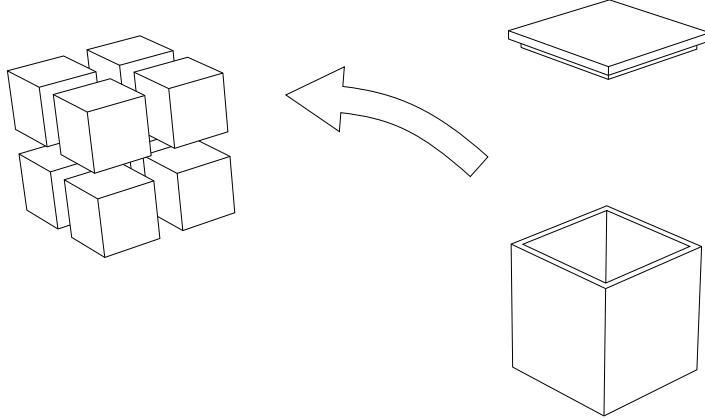


Figure 5.1: Concept drawing of the configurable payload, with swappable cubes that allow the COM to be varied relative to the geometric center.

5.2 Mathematical Design of the Configurable Payload

5.2.1 Mass and COM Configuration Space

Firstly, consider a positive real set of material densities $\mathcal{P} \in \mathbb{R}^+$, each element the density (in kg m^{-3}) of a material to be used:

$$\mathcal{P} = \{\rho_1, \rho_2 \dots \rho_n \mid \rho_i > 0\} \quad (5.1)$$

Each permutation can then be defined as an $n \times n \times n$ matrix C , such that each element is an element of \mathcal{P} , where n^3 is the number of cubes in the matrix:

$$\mathbf{C} = (c_{ijk}) \in \mathbb{R}^{n^n} \mid (c_{ijk}) \in \mathcal{P} \quad (5.2)$$

\mathcal{X} can then be defined as the set of all permutations of \mathbf{C} .

To calculate the mass of the permutation, take the sum of all the cube densities multiplied by their volume a^3 , where a is the cube edge length, plus the container mass m_c :

$$M(\mathbf{C}) = \left(\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n c_{ijk} a^3 \right) + m_c \quad (5.3)$$

To calculate the COM, take the sum of each cube mass multiplied by its position relative to the centroid of the center cube (c_{222}), which can be calculated from the cube indexes ijk , plus the container COM \mathbf{r}_c if non-zero:

$$R(\mathbf{C}) = \frac{\left(\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n c_{ijk} a^4 \left([i \ j \ k] - n + 1 \right) \right) + \mathbf{r}_c}{M(\mathbf{C})} \quad (5.4)$$

The *configuration space* \mathcal{Y} can then be considered as a set of \mathbb{R}^4 vectors containing the target mass and COM concatenated as $[m_i \ \mathbf{r}_i]$ of each element. As such, \mathcal{Y} is a codomain of \mathcal{X} , such that $H : \mathcal{X} \mapsto \mathcal{Y}$ where the map function H is $[M(\mathbf{C}) \ R(\mathbf{C})]$.

5.2.2 Mass Limitation

As \mathcal{X} may have mass values that are greater than what the robot arm can reliably lift, a subset \mathcal{X}_{lim} can be defined as such:

$$\mathcal{X}_{\text{lim}} = \{ \mathbf{x} \subset \mathcal{X} \mid M(\mathbf{x}) \leq m_{\text{max}} \} \quad (5.5)$$

Where m_{max} is the maximum mass the robot arm can reliably lift. This can be mapped onto $H : \mathcal{X}_{\text{lim}} \mapsto \mathcal{Y}_{\text{lim}}$ in a similar fashion.

5.2.3 Test Point Sets

Test points can either be derived from subsets of \mathcal{X}_{lim} defined by logical expressions, or the nearest neighbours of \mathcal{Y}_{lim} from a target mass and COM concatenated into a vector as in H , found by a *search method*.

5.2.3.1 Extrema Set (\mathcal{E})

The extrema set is designed to test the extremas of the space of \mathcal{X}_{lim} for both $M(\mathcal{X}_{\text{lim}})$ and $R(\mathcal{X}_{\text{lim}})$. This allows for test points to be generated at the limits of the capabilities of the payload or robot arm, i.e. the lightest and heaviest possible payloads, and the limits of the payload COM on the xy plane, both on the individual axes $\pm x$, $\pm y$, but also on combined axes (diagonals) $\pm x = \pm y$. The extrema set is defined from a set of logical constraints. The first two constraints of the set find the maximum and minimum values of the payload mass using $M(\mathcal{X}_{\text{lim}})$, and the next four constraints use the payload COM using $M(\mathcal{C})$ to get the maximum and minimum values of the x and y component of the COM. Finally, the last four constraints define the diagonal maximum and minimum values where the COM components match $x = y$ or $x = -y$.

$$\mathcal{E} = \left\{ \boldsymbol{x} \subset \mathcal{X}_{\text{lim}} \mid \begin{array}{l} M(\boldsymbol{x}) = \max \{M(\mathcal{X}_{\text{lim}})\} \\ M(\boldsymbol{x}) = \min \{M(\mathcal{X}_{\text{lim}})\} \\ R(\boldsymbol{x})_x = \max \{R(\mathcal{X}_{\text{lim}})_x\} \\ R(\boldsymbol{x})_x = \min \{R(\mathcal{X}_{\text{lim}})_x\} \\ R(\boldsymbol{x})_y = \max \{R(\mathcal{X}_{\text{lim}})_y\} \\ R(\boldsymbol{x})_y = \min \{R(\mathcal{X}_{\text{lim}})_y\} \\ R(\boldsymbol{x})_x = \max \{R(\mathcal{X}_{\text{lim}})_x\} \wedge R(\boldsymbol{x})_x = R(\boldsymbol{x})_y \\ R(\boldsymbol{x})_x = \min \{R(\mathcal{X}_{\text{lim}})_x\} \wedge R(\boldsymbol{x})_x = R(\boldsymbol{x})_y \\ R(\boldsymbol{x})_x = \max \{R(\mathcal{X}_{\text{lim}})_x\} \wedge R(\boldsymbol{x})_x = -R(\boldsymbol{x})_y \\ R(\boldsymbol{x})_x = \min \{R(\mathcal{X}_{\text{lim}})_x\} \wedge R(\boldsymbol{x})_x = -R(\boldsymbol{x})_y \end{array} \right\} \quad (5.6)$$

\mathcal{E}_1 Target Because it was possible for $\max \{M(\mathcal{X}_{\text{lim}})\}$ to have multiple solutions due to the mass constraint, \mathcal{E}_1 was changed to a target $S \left(\begin{bmatrix} m_{\text{max}} & 0 & 0 & * \end{bmatrix} \right)$ where S is the search method function with target structure $\begin{bmatrix} m & \mathbf{r} \end{bmatrix}$ that returns an element in \mathcal{X}_{lim} , and $*$ is a “don’t care” notation, to indicate to the search method to not include that component as part of the target. One of the search methods described in section ?? can then be used for S to find the nearest point in configuration space.

5.2.3.2 Cube Set (\mathcal{C})

The cube set is defined by the vertices of a cube of size b centred around the COM origin $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$. This provides a suitable method to generate arbitrary test points within the configuration space, and also tests if the z component of the COM has any effect on the payload stability experiment in chapter 6.

$$\mathcal{C} = \left\{ \boldsymbol{x} \subset \mathcal{X}_{\text{lim}} \mid \boldsymbol{x} = S \left(\begin{bmatrix} * & \pm \frac{b}{2} & \pm \frac{b}{2} & \pm \frac{b}{2} \end{bmatrix} \right) \right\} \quad (5.7)$$

5.2.3.3 Balanced Set (\mathcal{B})

The balanced set is defined by q points in \mathcal{X}_{lim} subject to the constraint $R(\mathbf{x})_x = 0 \wedge R(\mathbf{x})_y = 0$. This can be defined as a ‘‘balanced’’ set as the COM x and y components are both zero. The points are evenly spaced between the maximum and minimum mass as defined in section ???. This provides a range of ‘‘balanced’’ test points with zero COM but varying mass, to isolate the mass influence of the payload in the payload stability experiment in chapter 6.

$$\begin{aligned} m_r &= \frac{\max\{M(\mathcal{X}_{\text{lim}})\} - \min\{M(\mathcal{X}_{\text{lim}})\}}{q+1} \\ \mathbf{z} &= \begin{bmatrix} m_r & 2m_r & \cdots & qm_r \end{bmatrix} \\ \mathcal{B} &= \left\{ \mathbf{x} \subset \mathcal{X}_{\text{lim}} \mid \mathbf{x} = S \left(\begin{bmatrix} z_i & 0 & 0 & * \end{bmatrix} \right) \right\} \end{aligned} \quad (5.8)$$

5.2.4 Search Methods for Finding the Nearest Configuration to a Target

When considering a viable search method given a target, the cardinality of \mathcal{X} is important to consider. It is defined as $|\mathcal{X}| = |\mathcal{P}|^{n^3}$ which increases super exponentially with n . For example, when $|\mathcal{P}| = 4$, $n = 2$ results in 65536 permutations and $n = 3$ results in approximately 1.8×10^{16} permutations. It’s very clear that when $n > 2$ for non-trivial cardinalities of \mathcal{P} , any kind of brute-force method is not computationally tractable. Therefore, a brute-force nearest neighbour method (see section 5.2.4.2) would be suitable for when $n = 2$, and a heuristic search method such as simulated annealing (see section 5.2.4.1) method would be suitable for when $n > 2$. Simulated annealing was chosen over alternatives such as a simple gradient descent as it is less likely to get stuck in local minima.

5.2.4.1 Simulated Annealing Search Method

Simulated annealing [60] is a modification to gradient descent optimisation [61] that allows the algorithm the chance to ‘‘jump out’’ of local minima early on (even though the approximation becomes temporarily worse). However, as the number of remaining steps decreases, that probability becomes smaller, becoming more and more like gradient descent. First, like any gradient descent algorithm, two things need to be generated, the initial configuration \mathbf{C}_0 , which can be random or manually selected, and the function $\mathcal{N}(\mathbf{C})$ which creates a set of all the ‘‘neighbours’’ of \mathbf{C} . In this case, this can be defined as the subset of \mathcal{X}_{lim} where the difference between \mathbf{C} and an element of $\mathcal{N}(\mathbf{C})$ is one and only one $c_{ijk} \neq c_{ijk}$:

$$\mathcal{N}(\mathbf{C}) = \{x \subset \mathcal{X}_{\text{lim}} \mid \exists! (x_{ijk} \neq c_{ijk})\} \quad (5.9)$$

Then the simulated annealing function can be described as follows:

1. Set \mathbf{C} to the initial permutation \mathbf{C}_0 .

2. For each of the optimisation steps:

- (a) Set the temperature value t with function $T\left(\frac{s_{\max}}{s}\right)$ which takes into account the number of remaining steps s , where s_{\max} is the total number of steps.
- (b) Set \mathbf{C}_{new} as a random element from the set of all neighbours of \mathbf{C} as defined by $\mathcal{N}(\mathbf{C})$.
- (c) Use acceptance probability function $P(E(\mathbf{C}), E(\mathbf{C}_{new}), t)$ where $E(\mathbf{C})$ is the energy function.
- (d) Compare that value with a random uniformly distributed real number between 0 and 1. If greater than or equal to, then replace \mathbf{C} with \mathbf{C}_{new} . Otherwise, keep it the same.
- (e) Repeat with \mathbf{C} until there are no remaining steps.

3. Return the approximated permutation \mathbf{C} .

```

 $C = \mathbf{C}_0$ 
for  $s \leftarrow 1, s_{\max}$  do
     $t = T\left(\frac{s_{\max}}{s}\right)$ 
     $\mathbf{C}_{new} = \mathcal{N}(\mathbf{C}) \xleftarrow{R} x$ 
    if  $P(E(\mathbf{C}), E(\mathbf{C}_{new}), t) \geq x \sim U([0, 1])$  then
         $\mathbf{C} = \mathbf{C}_{new}$ 
    end if
end for
return  $\mathbf{C}$ 
```

Simulated Annealing Energy Function Simulated annealing can also be adapted for multi-objective optimisation [62], so it is possible to generate test points that approximate a desired mass and COM simultaneously.

Simulated Annealing Cooling Function The function which controls the probability of exiting local minima (known as the *temperature*) is known as the *cooling function*. This function can be any function which monotonically decreases (except in adaptive simulated annealing where it is dependant on the accuracy of the current approximation). Different functions will result in a different cooling profile, generally decreasing quickly in the first few steps, and then slowing down after that. A graph of different cooling functions over 100 steps is shown in figure 5.2.

5.2.4.2 Nearest Neighbour Search Method

If \mathcal{X}_{\lim} is suitably small, then a brute-force method can be used which is guaranteed to find the nearest element to the target within a finite time. This can be done by calculating the L2

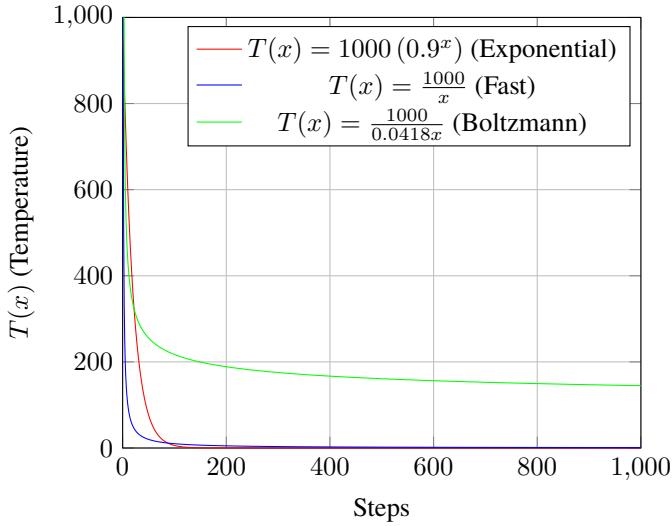


Figure 5.2: Various temperature cooling profiles for simulated annealing, assuming 1000 steps.

norms between the target vector t and all the elements of \mathcal{Y}_{lim} and finding the minimum. If there are several elements in the domain of \mathcal{X}_{lim} , then one is chosen at random from this set.

$$\text{NN}(t, \mathcal{X}_{\text{lim}}) = \min \{ \|t - x\|_2 \mid \forall x \in \mathcal{X}_{\text{lim}} \} \quad (5.10)$$

5.3 Implementation of the Configurable Payload

5.3.1 Container Design

The internal width of the container was chosen to be 76 mm to fit comfortably between the robot arm grippers, which combined with a wall width of 5 mm gives an overall width of 81 mm. Therefore, each cube would be $\frac{75}{n}$ mm in size, to allow for fitting clearance. For additional resiliency, the base would be an additional 4 mm, giving the container an overall height of 90 mm with the lid (not including gripper guides). A dimensioned schematic is shown in figure 5.3.

Two sides of the container were made of a textured pattern designed to enhance grip when picked up by the robot arm. The other two sides had small viewports centred around each cube, for permutation verification during the experiments in chapter 6. The lid of the container had a small notch to allow for it to be levered off in case of a tight fit, and guides to ensure the robot arm grippers would pick up the container in the same location each time, ensuring experiment repeatability. A labelled photograph is shown in figure 5.4.

The cube container was 3D printed in Polylactic Acid (PLA) with an 80% fill density. This gave it a total mass of 0.221 kg which provides the value for m_c . By design, the COM of the container would be 0 mm in both the x and y direction, to within a reasonable manufacturing tolerance. Due to the minimal contribution of the container to the total mass, this was a suitable assumption.

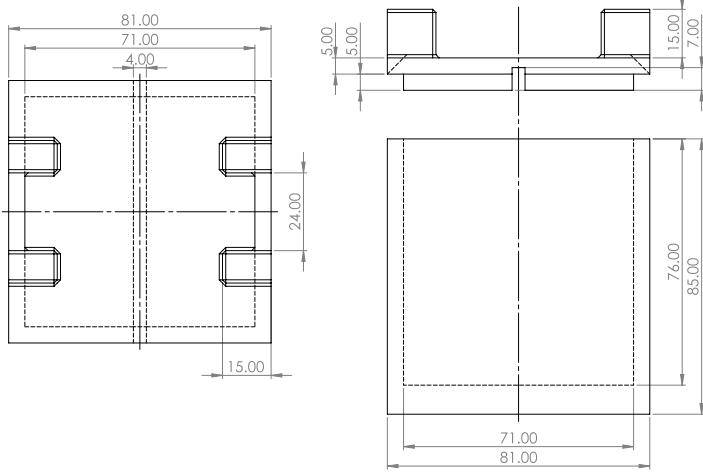


Figure 5.3: Schematic of the container, including body and lid.

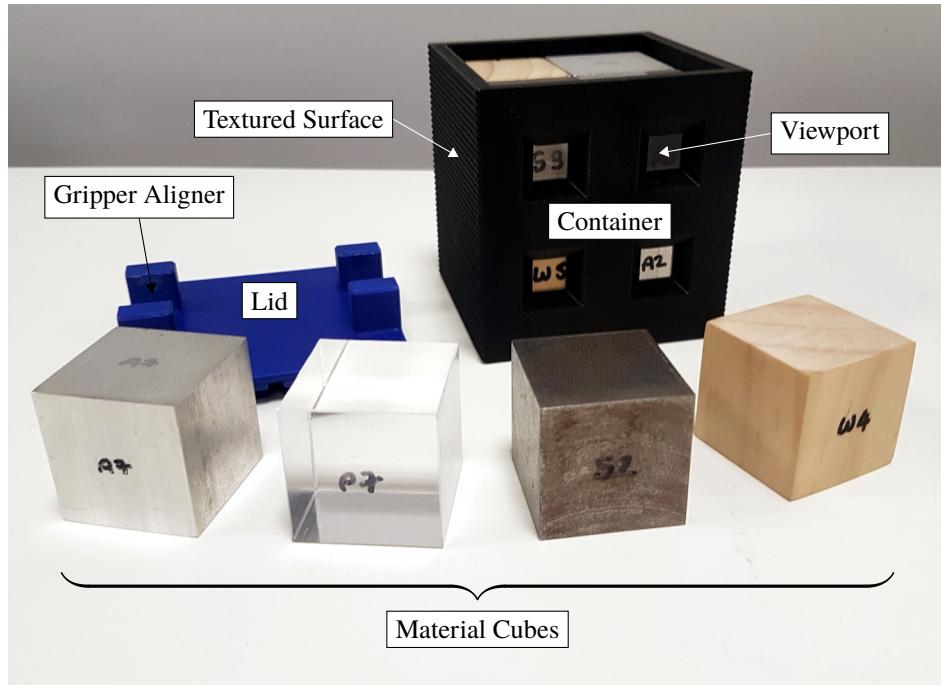


Figure 5.4: Labelled photograph of the final container design, along with the material cubes.

5.3.2 Selected Search Method on the Configuration Space for the Cube Set and Balanced Set

Initially an $n = 3$ configuration was used, using the simulated annealing search method with the acceptance probability function *Rule M* from [62] to allow for a multiobjective search. This is a weighted blend of two other algorithms defined in the paper, *Rule P* and *Rule W* with a weighting coefficient $\alpha \in (0, 1) \subset \mathbb{R}$. There is also a weighting vector for each element of the test point $\mathbf{w} \in \mathbb{R}^4 \mid w_i \in (0, 1)$.

$$P(\mathbf{x}, \mathbf{y}, \mathbf{w}, t) = \underbrace{\alpha \prod_{i=1}^m \min \left\{ 1, e^{\frac{w_i(x_i - y_i)}{t}} \right\}}_{\text{Rule P}} + (1 - \alpha) \underbrace{\min \left\{ 1, \max_{i=1,\dots,m} \left\{ 1, e^{\frac{w_i(x_i - y_i)}{t}} \right\} \right\}}_{\text{Rule W}} \quad (5.11)$$

Unfortunately it was difficult to find a stable and consistent result even after 4.3×10^4 steps, as shown in figure 5.5. Therefore as an alternative to using simulated annealing, the $n = 2$ configuration was used, with larger cubes to compensate. As this only had 65536 permutations, a brute force search would be tractable using

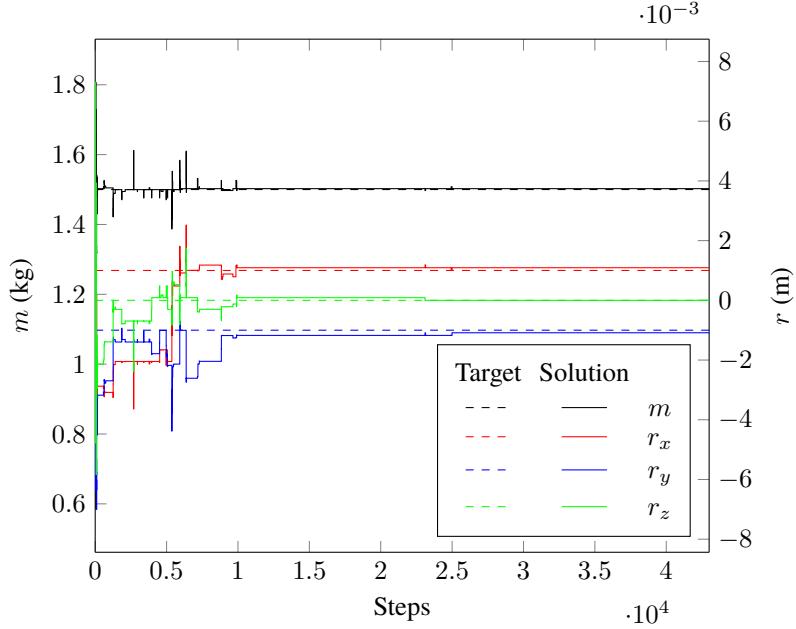


Figure 5.5: Simulated annealing output for the target $[1.5 \ 0.001 \ 0.001 \ 0.001]$ with $\alpha = 0.997$ and even weighting $w_m, w_r = 0.25$ for 4.3×10^4 steps.

5.3.3 Material Selection

In order to produce a reasonably wide and dense configuration space, four materials: *wood*, *plastic*, *aluminium* and *steel*, were chosen. More dense materials, such as nickel and lead, were rejected due to difficulty sourcing stock of the correct size, or potential issues with machining. Initially estimated densities were used in order to test the search methods, but after the cubes were manufactured, it was possible to get an average density based on the measured mass of each cube as seen in figure 5.6, given a cube size of 35 mm. Table 5.1 lists the exact kind of material used for each of the four materials, and its average density based on the measured mass. These differ from many stated values available from other sources such as online material databases, likely due to small discrepancies in cube size due to manufacturing tolerances and variability in material composition (particularly for wood since the blocks required sanding in order to fit in the container, and due to the less precise properties of natural materials).

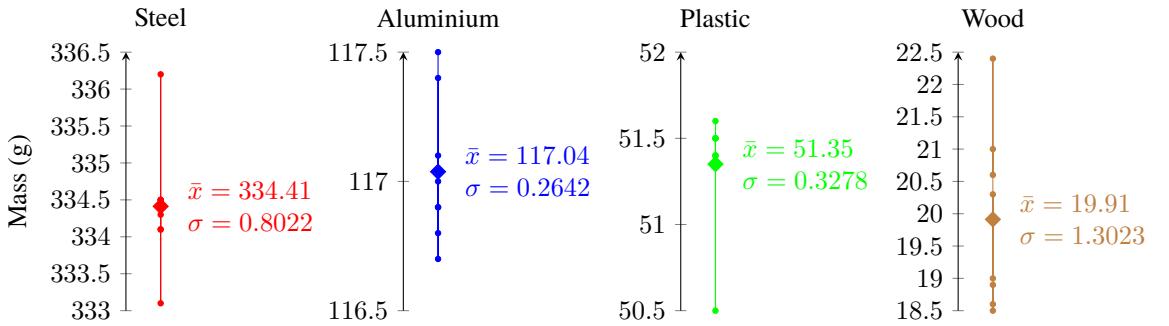


Figure 5.6: Masses for each set of eight 35 mm cubes of the configurable payload for each of the four materials, with the mean mass \bar{x} and standard deviation σ of each set.

Material	Variant	Density (kg m^{-3})
Wood	Pine	464.37
Plastic	Acrylic	1201.9
Aluminium	6082	2740
Steel	EN3B	7800

Table 5.1: The materials chosen for the cubes and their calculated density.

5.4 Resulting Mass and COM Configurations of the Physical Payload

5.4.1 Configuration Space

Figure ?? shows a scatter graph of all the permutations of mass and COM of the constructed payload.

Given a $2 \times 2 \times 2$ matrix of cubes with the materials in table 5.1, there were a total of 65536 permutations, with a total mass range of $[0.38, 2.90]\text{kg}$, and a total COM range of $[-13.44, 13.44]\text{mm}$ on all axes. As some permutations mapped to the same point in configuration space, there were 62969 unique mass and COM vectors.

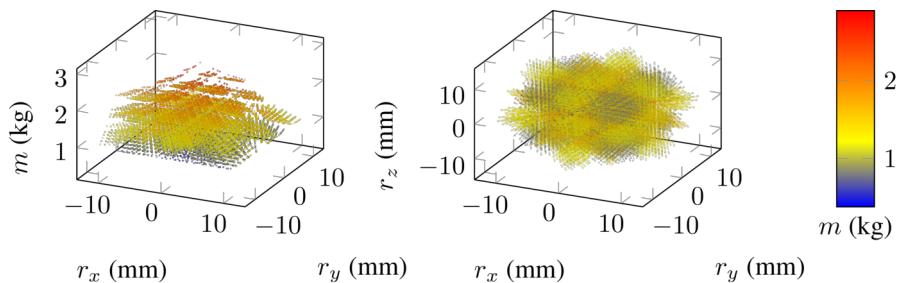


Figure 5.7: Scatter graph of all 62969 unique mass and COM vectors in the configuration space, each point represents a unique permutation.

5.4.2 Test Point Results

Figure 5.8 shows scatter plots of the mass and COM for each test point.

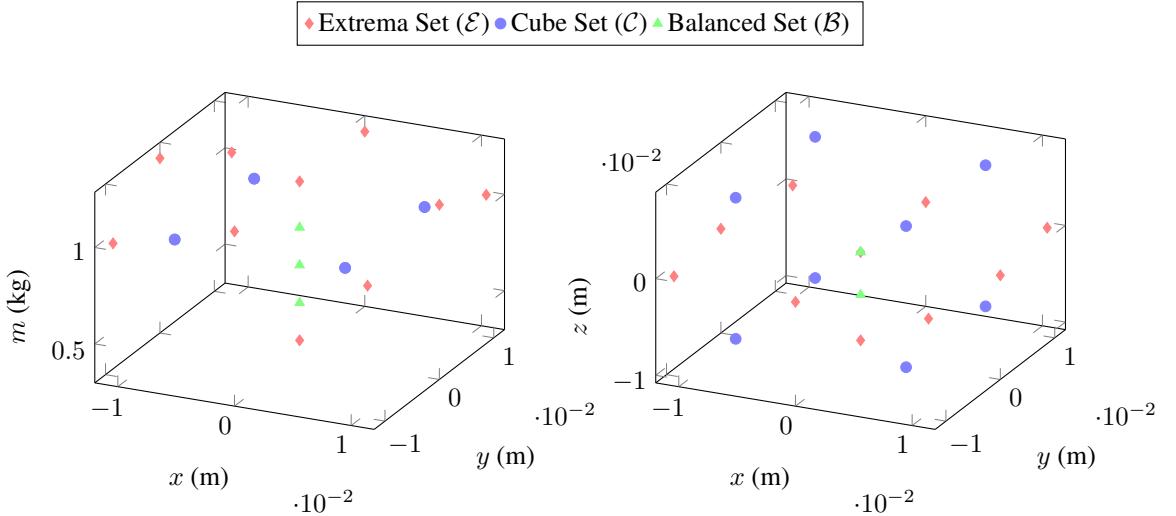


Figure 5.8: Mass and COM coordinates for each test point set.

5.4.2.1 Test Point Mass and COM

Tables 5.2 and 5.3 show the target positions defined by the notations in section . Table 5.3 also shows the permutation found in the configuration space nearest neighbour search method. The mass limit m_{\max} for the extrema set was set at 1.25 kg, and the cube set length parameter b was set at 7 mm.

m	r
Extrema Set (\mathcal{E})	
1.204	$[0.012 \quad 0.000 \quad -0.000]$
1.204	$[0.000 \quad 0.012 \quad -0.000]$
1.204	$[-0.012 \quad 0.000 \quad -0.000]$
1.204	$[0.000 \quad -0.012 \quad -0.000]$
0.380	$[0.000 \quad 0.000 \quad 0.000]$
1.009	$[0.011 \quad 0.011 \quad -0.000]$
1.009	$[-0.011 \quad -0.011 \quad 0.000]$
1.009	$[0.011 \quad -0.011 \quad 0.000]$
1.009	$[-0.011 \quad 0.011 \quad 0.000]$

Table 5.2: Table of the mass and COM vectors of \mathcal{E} , excluding \mathcal{E}_1 .

5.4.2.2 Test Point Material Permutations

Tables 5.4, 5.5 and 5.6 show the material permutations, i.e. the material of each cube in the container, for each point in the test point sets.

Target				Nearest		L2 Norm Error	
m	\mathbf{r}	m	\mathbf{r}				
Extrema Set (\mathcal{E})							
1.250	$[0.000 \ 0.000 \ *]$	1.204	$[0.000 \ 0.000 \ -0.009]$			4.556×10^{-2}	
Cube Set (\mathcal{C})							
*	$[-0.007 \ -0.007 \ -0.007]$	0.987	$[-0.007 \ -0.007 \ -0.007]$			5.244×10^{-4}	
*	$[-0.007 \ -0.007 \ 0.007]$	0.987	$[-0.007 \ 0.007 \ -0.007]$			2.023×10^{-2}	
*	$[-0.007 \ 0.007 \ -0.007]$	0.987	$[0.007 \ -0.007 \ -0.007]$			2.023×10^{-2}	
*	$[-0.007 \ 0.007 \ 0.007]$	0.987	$[0.007 \ 0.007 \ -0.007]$			2.023×10^{-2}	
*	$[0.007 \ -0.007 \ -0.007]$	0.987	$[-0.007 \ -0.007 \ 0.007]$			2.023×10^{-2}	
*	$[0.007 \ -0.007 \ 0.007]$	0.987	$[-0.007 \ 0.007 \ 0.007]$			2.023×10^{-2}	
*	$[0.007 \ 0.007 \ -0.007]$	0.987	$[0.007 \ -0.007 \ 0.007]$			2.023×10^{-2}	
*	$[0.007 \ 0.007 \ 0.007]$	0.987	$[0.007 \ 0.007 \ 0.007]$			5.244×10^{-4}	
Balanced Set (\mathcal{B})							
0.598	$[0.000 \ 0.000 \ *]$	0.575	$[0.000 \ 0.000 \ 0.000]$			2.229×10^{-2}	
0.815	$[0.000 \ 0.000 \ *]$	0.771	$[0.000 \ 0.000 \ -0.004]$			4.459×10^{-2}	
1.033	$[0.000 \ 0.000 \ *]$	0.966	$[0.000 \ 0.000 \ 0.000]$			6.688×10^{-2}	

Table 5.3: Table of the target and actual mass and COM vectors for \mathcal{C} , \mathcal{B} and \mathcal{E}_1 with the L2 norm error. * notation indicates “don’t care” and is excluded from the search algorithm.

m	r	Material Matrix		3D Preview
		Extrema Set (\mathcal{E})		
1.204	[0.000 0.000 -0.009]	$\begin{bmatrix} S & A \\ W & S \end{bmatrix} \begin{bmatrix} W & W \\ A & W \end{bmatrix}$	$\begin{bmatrix} W & W \\ A & W \end{bmatrix}$	
1.204	[0.012 0.000 -0.000]	$\begin{bmatrix} W & W \\ S & A \end{bmatrix} \begin{bmatrix} W & W \\ A & S \end{bmatrix}$	$\begin{bmatrix} W & W \\ A & S \end{bmatrix}$	
1.204	[0.000 0.012 -0.000]	$\begin{bmatrix} W & S \\ W & A \end{bmatrix} \begin{bmatrix} W & A \\ W & S \end{bmatrix}$	$\begin{bmatrix} W & A \\ W & S \end{bmatrix}$	
1.204	[-0.012 0.000 -0.000]	$\begin{bmatrix} A & S \\ W & W \end{bmatrix} \begin{bmatrix} S & A \\ W & W \end{bmatrix}$	$\begin{bmatrix} S & A \\ W & W \end{bmatrix}$	
1.204	[0.000 -0.012 -0.000]	$\begin{bmatrix} A & W \\ S & W \end{bmatrix} \begin{bmatrix} S & W \\ A & W \end{bmatrix}$	$\begin{bmatrix} S & W \\ A & W \end{bmatrix}$	
0.380	[0.000 0.000 0.000]	$\begin{bmatrix} W & W \\ W & W \end{bmatrix} \begin{bmatrix} W & W \\ W & W \end{bmatrix}$	$\begin{bmatrix} W & W \\ W & W \end{bmatrix}$	
1.009	[0.011 0.011 -0.000]	$\begin{bmatrix} W & W \\ W & S \end{bmatrix} \begin{bmatrix} W & W \\ W & S \end{bmatrix}$	$\begin{bmatrix} W & W \\ W & S \end{bmatrix}$	
1.009	[-0.011 -0.011 0.000]	$\begin{bmatrix} S & W \\ W & W \end{bmatrix} \begin{bmatrix} S & W \\ W & W \end{bmatrix}$	$\begin{bmatrix} S & W \\ W & W \end{bmatrix}$	
1.009	[0.011 -0.011 0.000]	$\begin{bmatrix} W & W \\ S & W \end{bmatrix} \begin{bmatrix} W & W \\ S & W \end{bmatrix}$	$\begin{bmatrix} W & W \\ S & W \end{bmatrix}$	
1.009	[-0.011 0.011 0.000]	$\begin{bmatrix} W & S \\ W & W \end{bmatrix} \begin{bmatrix} W & S \\ W & W \end{bmatrix}$	$\begin{bmatrix} W & S \\ W & W \end{bmatrix}$	

Table 5.4: Table of all test point configurations of the *Extrema Set*, showing the mass and COM and material permutation in matrix and 3D form.

m	r	Material Matrix	3D Preview
Cube Set (\mathcal{C})			
0.987	$[-0.007 \quad -0.007 \quad -0.007]$	$\begin{bmatrix} [S \quad A] \\ [A \quad W] \end{bmatrix} \begin{bmatrix} A \quad W \\ W \quad W \end{bmatrix}$	
0.987	$[-0.007 \quad 0.007 \quad -0.007]$	$\begin{bmatrix} [A \quad S] \\ [W \quad A] \end{bmatrix} \begin{bmatrix} W \quad A \\ W \quad W \end{bmatrix}$	
0.987	$[0.007 \quad -0.007 \quad -0.007]$	$\begin{bmatrix} [A \quad W] \\ [S \quad A] \end{bmatrix} \begin{bmatrix} W \quad W \\ A \quad W \end{bmatrix}$	
0.987	$[0.007 \quad 0.007 \quad -0.007]$	$\begin{bmatrix} [W \quad A] \\ [A \quad S] \end{bmatrix} \begin{bmatrix} W \quad W \\ W \quad A \end{bmatrix}$	
0.987	$[-0.007 \quad -0.007 \quad 0.007]$	$\begin{bmatrix} [A \quad W] \\ [W \quad W] \end{bmatrix} \begin{bmatrix} S \quad A \\ A \quad W \end{bmatrix}$	
0.987	$[-0.007 \quad 0.007 \quad 0.007]$	$\begin{bmatrix} [W \quad A] \\ [W \quad W] \end{bmatrix} \begin{bmatrix} A \quad S \\ W \quad A \end{bmatrix}$	
0.987	$[0.007 \quad -0.007 \quad 0.007]$	$\begin{bmatrix} [W \quad W] \\ [A \quad W] \end{bmatrix} \begin{bmatrix} A \quad W \\ S \quad A \end{bmatrix}$	
0.987	$[0.007 \quad 0.007 \quad 0.007]$	$\begin{bmatrix} [W \quad W] \\ [W \quad A] \end{bmatrix} \begin{bmatrix} W \quad A \\ A \quad S \end{bmatrix}$	

Table 5.5: Table of all test point configurations of the *Cube Set*, showing the mass and COM and material permutation in matrix and 3D form.

m	r	Material Matrix	3D Preview
Balanced Set (\mathcal{B})			
0.575	[0.000 0.000 0.000]	$\begin{bmatrix} \begin{bmatrix} A & W \\ W & W \end{bmatrix} & \begin{bmatrix} W & W \\ W & A \end{bmatrix} \end{bmatrix}$	
0.771	[0.000 0.000 -0.004]	$\begin{bmatrix} \begin{bmatrix} A & W \\ A & A \end{bmatrix} & \begin{bmatrix} W & A \\ W & W \end{bmatrix} \end{bmatrix}$	
0.966	[0.000 0.000 0.000]	$\begin{bmatrix} \begin{bmatrix} A & A \\ W & A \end{bmatrix} & \begin{bmatrix} A & W \\ A & A \end{bmatrix} \end{bmatrix}$	

Table 5.6: Table of all test point configurations of the *Balanced Set*, showing the mass and COM and material permutation in matrix and 3D form.

5.5 Discussion and Conclusion

Due to the discrete nature of a composition of cubes of various densities, exact mass and COM configurations were often not possible, so a search method algorithm was required to find the permutation closest to the target. With a more granular ability to vary mass and COM, it would be possible to dispense with the need for a search method, as long as a suitably limited level of precision is considered. But this may result in more time required for reconfiguration for each test point, which would make the payload stabilisation experiments in chapter 6 more difficult.

The results show it is possible to generate a very large number of permutations (65536) with only 8 cubes of 4 different densities, and increasing to 12 cubes generates such an enormous number of permutations (1.8×10^{16}) it is not tractable to use a brute force search method given a target mass and COM. However, the COM extrema is small, only $[-13.44, 13.44]\text{mm}$ on all axes, due to the container and material cubes needing to fit inside the robot arm grippers. A different design that allowed the payload to be larger than the gripper width would allow for a significantly greater COM range. A potential design that achieves this is discussed in section 5.5.1.1.

5.5.1 Future Work

5.5.1.1 Payload with Improved COM Range

As mentioned in section 5.4.1, the COM range of the configurable payload is only $[-13.44, 13.44]\text{mm}$ on all axes. Similarly to the mass range, the COM range could be increased by the use of denser and lighter materials. However, with the current design, it is not possible to increase the COM range by a significant amount, even the inclusion of densest and lightest elements, osmium at $22\,590\,\text{kg m}^{-3}$ [63], and hydrogen at $0.089\,88\,\text{kg m}^{-3}$ [64], which would be completely impractical or impossible to use, the COM range only increased by 2.7 mm. This is due to the small offset distance of the COM of each material cube, only 17.5 mm from the geometric centroid of the payload. A design that could increase the maximum offset distance would therefore increase the COM range significantly. One proposal would be to use a rod with a mass on one end and a handle for the grippers on the other. The mass material, thickness, and length of the rod could then be varied to adjust the mass and COM of the payload. Since the gripper can rotate 360° with the chosen robot arm, any COM can be simulated within a annulus shaped configuration space around $z = 0$ for a given mass, but could be extruded to an annular cylinder if some mechanism was used to vary the z element of the COM (asymmetric masses, rod sliders on the handle, etc.). If two rods either side of the handle are used, the annuli become disks and cylinders respectively, allowing for “balanced” payloads, as can be seen in figure 5.9. This is an advantage over the original design, and eliminates the need for any kind of search algorithm for the COM, since it could be found analytically for a given mass using polar coordinates, where

the *radius* is the rod length and the *azimuth* is the gripper rotation angle, though it must be noted the mass and maximum rod length will determine the available range of values for the COM. A visualization of a potential design is shown in figure 5.10.

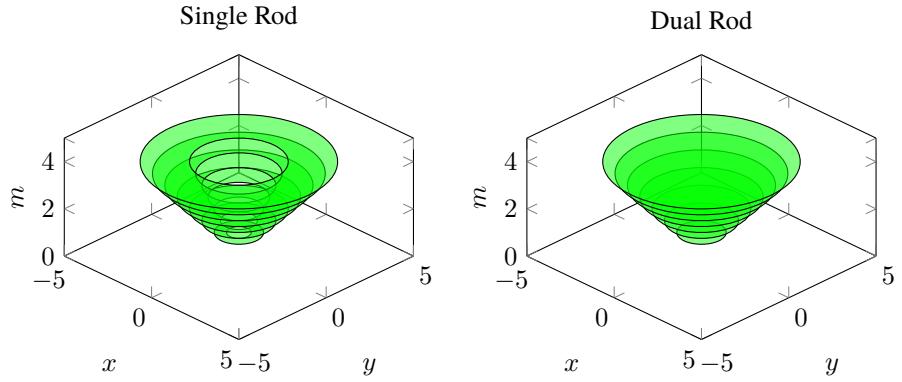


Figure 5.9: Example configuration space for the new configurable payload design, where $z = 0$.

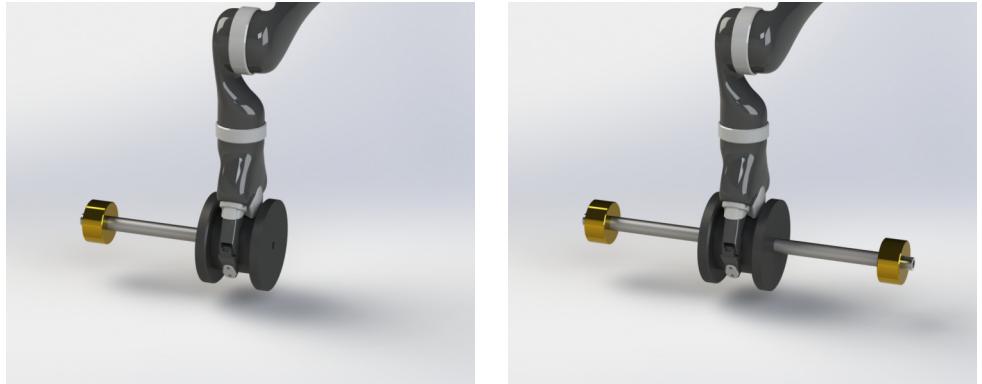


Figure 5.10: Renders of a concept design for the new configurable payload, with a single rod and dual rod configuration.

5.5.1.2 Dynamic COM Implementation: Fluid Filled Container

Not all payloads that a robot may pick up are solid objects, some may have COM that shift during transit due to external disturbances. Generally any container partially filled with either loose material or fluid, such as a bottle or sack, could have a so-called “dynamic” COM. Therefore, a container partially filled with fluid, with a specific viscosity and mass would be a suitable simulacrum for these kinds of payloads, as shown in figure 5.11. The robot arm could then shake the container from side to side briefly to simulate a disturbance, creating a standing wave inside the container that would have an effect on the COM.

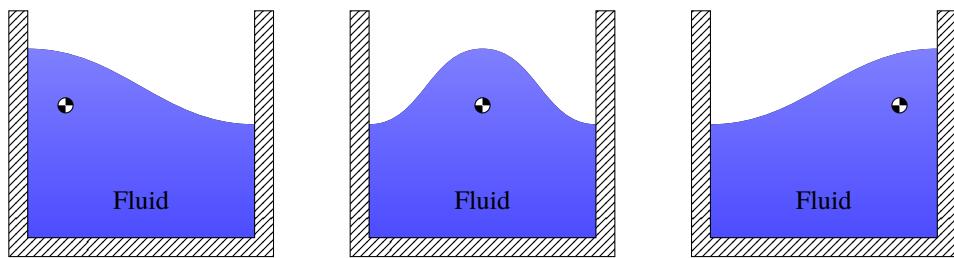


Figure 5.11: Diagram on the COM dynamics of a fluid filled container. If the container is disturbed, a standing wave forms inside the container which causes the COM to oscillate.

5.5.2 Conclusion

This work has outlined the design and implementation of a configurable payload that is suitable for the payload stability experiments in chapter 6. Using this payload, a range of objects with different mass and COM can be emulated in order to generate a significant range of experimental data for different simulated payloads.

Chapter 6

Investigating the use of a 2DOF Pendulum Tail for Compensating for Instability when Carrying a Payload

In this chapter, the design of the static rig is outlined, made up of a base, a robotic arm, the configurable payload from chapter 5, a 2 DOF “inverted pendulum” tail driven by two brushless motors using a bevel gear arrangement, and four load cells used to calculate the COP. N cost functions for stability are then derived from the COP data when the payload is being lifted by the arm. An experimental trajectory is described which replicates the motions of a mobile robot moving and object from one location to another. A simple proportional controller is then developed and simulated, and its efficacy is compared with experimental results from the constructed rig for each test point set in chapter ???. The discussion then examines the efficacy of the chosen design and actuation method for the robotic tail, and its limitations and potential improvements, as well as examining the ability of the configurable payload to generate a suitably wide range of test points, given the variation in the motion of the tail for each point.

6.1 Introduction

As mentioned in chapter 1, the primary aim of this research is to quantify the efficacy of a robotic tail for maintaining the stability of a mobile robot when carrying a payload. While the ultimate verification for this question would be to attach a tail to a mobile robot and engage in field trials, carrying objects in a close simulacrum to a real operating environment, a simpler and more flexible approach is to use a static rig which can measure the static stability using the COP. By comparing the measurement to a robot's known stability region and initial COM, it is possible to determine if a specific mobile robot would remain statically stable without having to topple a mobile robot, and risk equipment damage.

However, demonstrating an improvement in a discrete set of pass/fail stability tests based on specific mobile robots is not necessary to answer the research question. Instead, a continuous approach can be used, which considers stability as a cost function rather than a binary result. This has the advantage of being able to quantify *how* stable the experiment is, and the magnitude of dynamic forces a given robot could withstand. The smaller the cost and the larger the support polygon, the greater the magnitude, depending on the distance to the polygon boundary in a specific direction.

6.2 Measuring the COP using Four Load Cells

The standard method of measuring the COP (see definition in chapter 1) has been to use a *force plate*, which uses 4 3-axis load cells in a square or rectangular arrangement as shown in figure 6.1.

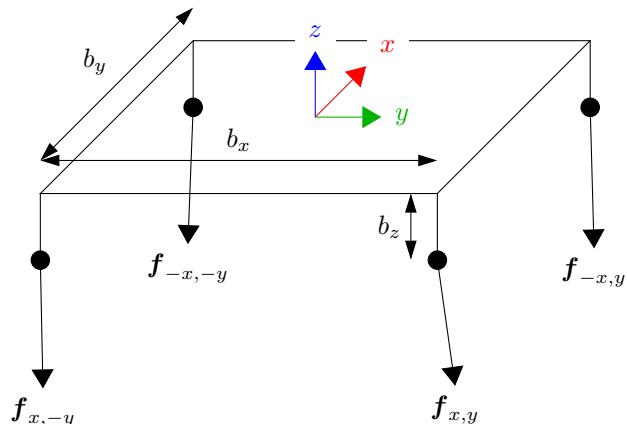


Figure 6.1: Diagram of a typical force plate setup.

To calculate the location of the COP, the following equation can be used:

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_{x,y} & \mathbf{f}_{-x,y} \\ \mathbf{f}_{x,-y} & \mathbf{f}_{-x,-y} \end{bmatrix}$$

$$CP(\mathbf{F}) = \begin{bmatrix} \frac{b_x((f_{11z} + f_{21z}) - (f_{12z} + f_{22z})) + b_z \sum_{i=1}^2 \sum_{j=1}^2 f_{ijz}}{\sum_{i=1}^2 \sum_{j=1}^2 f_{ijz}} \\ \frac{b_y((f_{11z} + f_{12z}) - (f_{21z} + f_{22z})) + b_z \sum_{i=1}^2 \sum_{j=1}^2 f_{ijy}}{\sum_{i=1}^2 \sum_{j=1}^2 f_{ijz}} \end{bmatrix} \quad (6.1)$$

This could be easily adapted to a mobile robot with an arbitrary number of legs or wheels. The following equation can be used for n ground contact points with associated load cells:

$$CP(\mathbf{F}) = \frac{\sum_{i=1}^n [b_{x_i} \ b_{y_i}] f_{iz} + b_z [f_{ix} \ f_{iy}]}{\sum_{i=1}^n f_{iz}} \quad (6.2)$$

Where $[b_{x_i} \ b_{y_i}]$ is the x and y position of the load cell.

It is possible to further simplify the design if only static stability is measured and the load cells Z axis is always in parallel with the gravity vector. Because $\sum_{i=1}^2 \sum_{j=1}^2 f_{ij} = [0 \ 0 \ mg]$ and therefore has no x or y component, 3-axis load cells can be substituted for single axis load cells, which are considerably cheaper and easier to use. By letting $b_z = 0$, the equation now only relies on the scalar z component of the load cells:

$$CP(\mathbf{F}) = \begin{bmatrix} \frac{b_x((f_{11z} + f_{21z}) - (f_{12z} + f_{22z}))}{\sum_{i=1}^2 \sum_{j=1}^2 f_{ijz}} \\ \frac{b_y((f_{11z} + f_{12z}) - (f_{21z} + f_{22z}))}{\sum_{i=1}^2 \sum_{j=1}^2 f_{ijz}} \end{bmatrix} \quad (6.3)$$

This is similar to the design of the Wii Fit™ Balance Board, which is able to measure the COP fairly accurately, as demonstrated in [65], [66] after calibration. The experimental setup used significantly higher quality load cells, resulting in additional measurement accuracy over these studies.

Based on this simplified force plate concept, a “static rig” can be designed that can measure the COP of a robot arm carrying a payload and a robot tail. This can act as a suitable simulacrum for a generic mobile robot, with any number of legs or wheels, that is picking up, carrying, and setting down a payload. The robot arm and tail are positioned along the origin X axis at a_y and $-t_y$ on the Y axis, and a_z and t_z on the Z axis. The tail is a rod attached to a 2 DOF joint, with the orientation controlled by input angles θ . The robot arm is a 4 DOF design with a pair of grippers to pick up a payload, with the input angles q . The following masses and COM then need to be considered, the base m_b, \mathbf{r}_b , the arm $m_a, \mathbf{r}_a(q)$, the payload $m_p, \mathbf{r}_p(q)$ the tail drive $m_d, \mathbf{r}_d(q)$ ¹ and the tail, which is calculated from m_t, l_t and θ . A free body diagram is shown in figure 6.2.

¹separated from the base so it can be removed from the equation when the tail is removed.

6.2.1 Theoretical COM Derivation

If the mass and COM of every component of the system is known, it is possible to calculate the total mass and overall COM theoretically. Using the terms from the free body diagram in figure 6.2, the COM can be found as follows:

$$M = m_b + m_d + m_t + m_a + m_p$$

$$R(\theta, \mathbf{q}) = \frac{\mathbf{r}_b + \left(\begin{bmatrix} 0 & a_y & a_z \end{bmatrix} + \mathbf{r}_a(\mathbf{q}) + \mathbf{r}_p(\mathbf{q}) \right) + \left(\begin{bmatrix} 0 & -t_y & t_z \end{bmatrix} + (\mathbf{r}_d + l_t \cos \theta) \right)}{M} \quad (6.4)$$

If the tail is removed, then the calculations for the tail COM and tail drive COM are removed from the equation:

$$\bar{M} = m_b + m_a + m_p$$

$$\bar{R}(\mathbf{q}) = \frac{\mathbf{r}_b + \left(\begin{bmatrix} 0 & a_y & a_z \end{bmatrix} + \mathbf{r}_a(\mathbf{q}) + \mathbf{r}_p(\mathbf{q}) \right)}{\bar{M}} \quad (6.5)$$

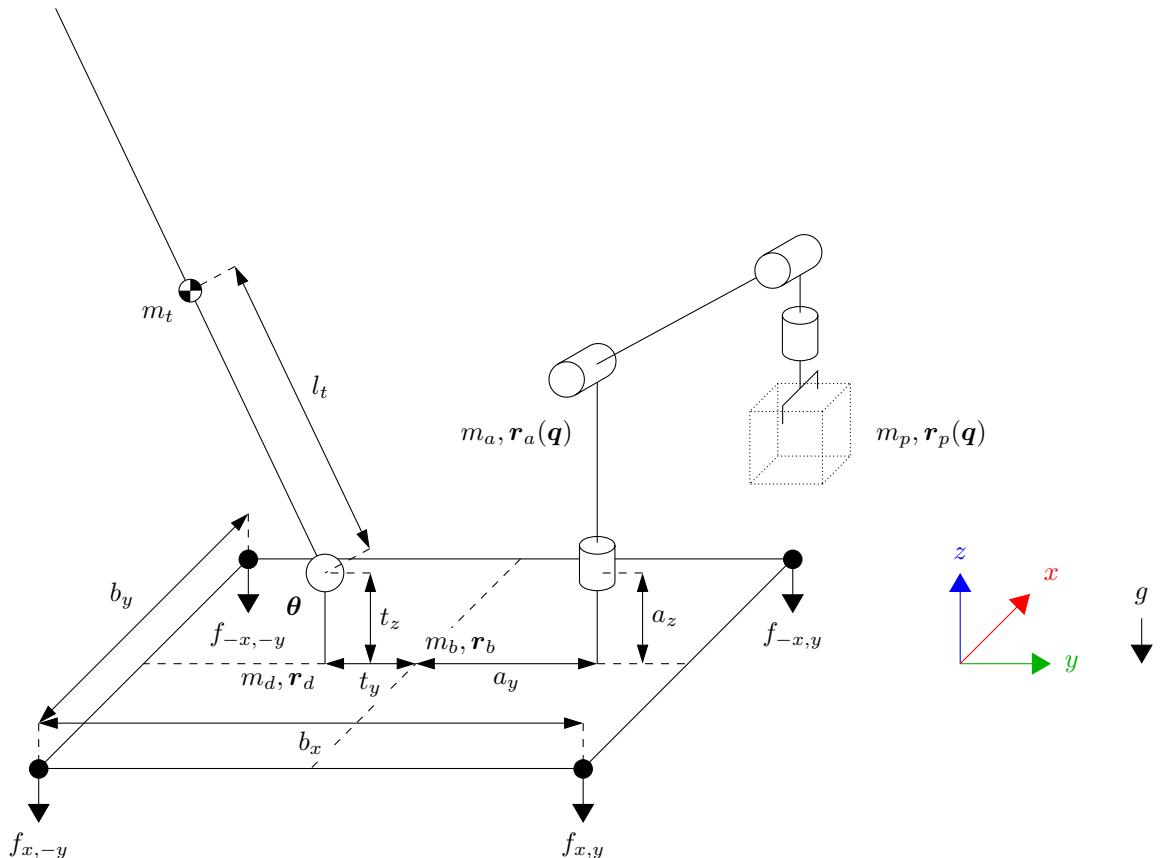


Figure 6.2: Free body diagram of the static test setup.

6.3 Test Sequence

The experiment test sequence is designed to replicate the actions of a mobile robot that is moving an object from one place to another. The robot arm grasps the payload and lifts it off the ground plane, then holds it for a predetermined period of time, before placing it back down on the ground plane and releasing it. This sequence is illustrated in figure 6.3.

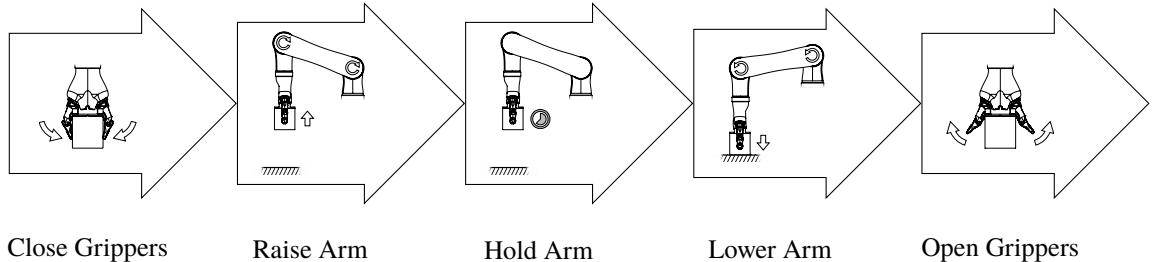


Figure 6.3: Trajectory sequence of the robot arm and payload.

This sequence is repeated twice for each test point in the sets designed in chapter 5, once with no tail attached to the rig as a control, and once with the active tail attached to the rig. The area of interest for the research is the “carry” interval of the experiment as shown in figure 6.4, between the *Raise Arm* and *Lower Arm* stages, specifically when the payload is not in contact with the ground plane.

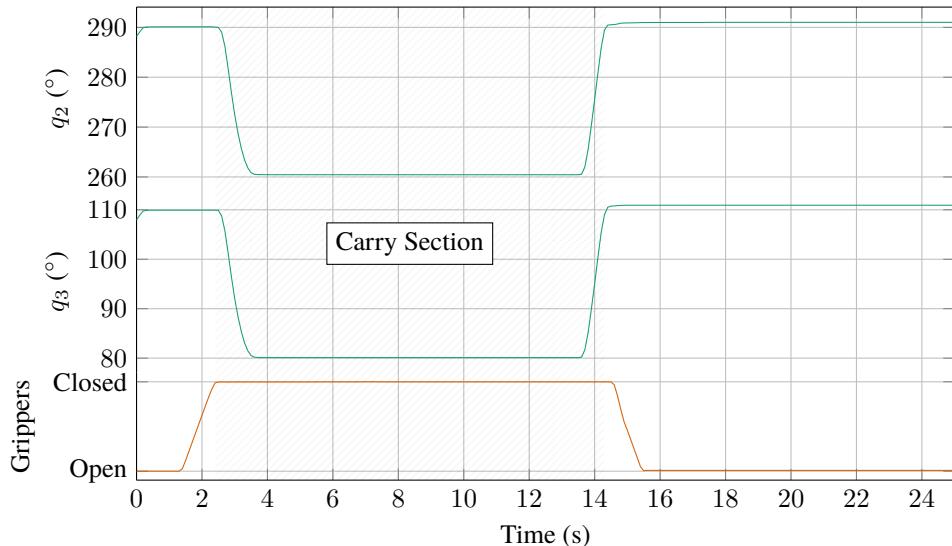


Figure 6.4: Graph of joint trajectories q_2 and q_3 during the test sequence (all other joints remain static), and the position of the grippers relative to the fully open position and the maximum closed position to grip the payload. Hatched area indicates the “carry” interval, where the tail will be compensating for the payload mass.

6.4 Simscape Multibody™ Simulation

Based on the diagram in figure 6.2 a Simscape Multibody™ simulation was built. Simscape Multibody™ is a Simulink library that allows the construction of dynamic models from solid bodies, rigid transform and joint blocks. These models can then be simulated, and inputs and output from the joints can be used to control the model.

Coefficient	Value
b_x, b_y	400 mm
a_y, t_y	170 mm
a_z	132 mm
t_z	116 mm
l_t	439 mm ^a
m_t	1.66 kg ^a
m_a	5 kg ^b
m_p	Variable

^a Composite value calculated from the rod and tip masses and lengths and the position of the tip mass at the end of the rod.

^b Robot arm mass sourced from [67] plus the mass of the robot mount.

Table 6.1: Table of the values of the coefficients from figure 6.2 used in the simulation and experimental apparatus.

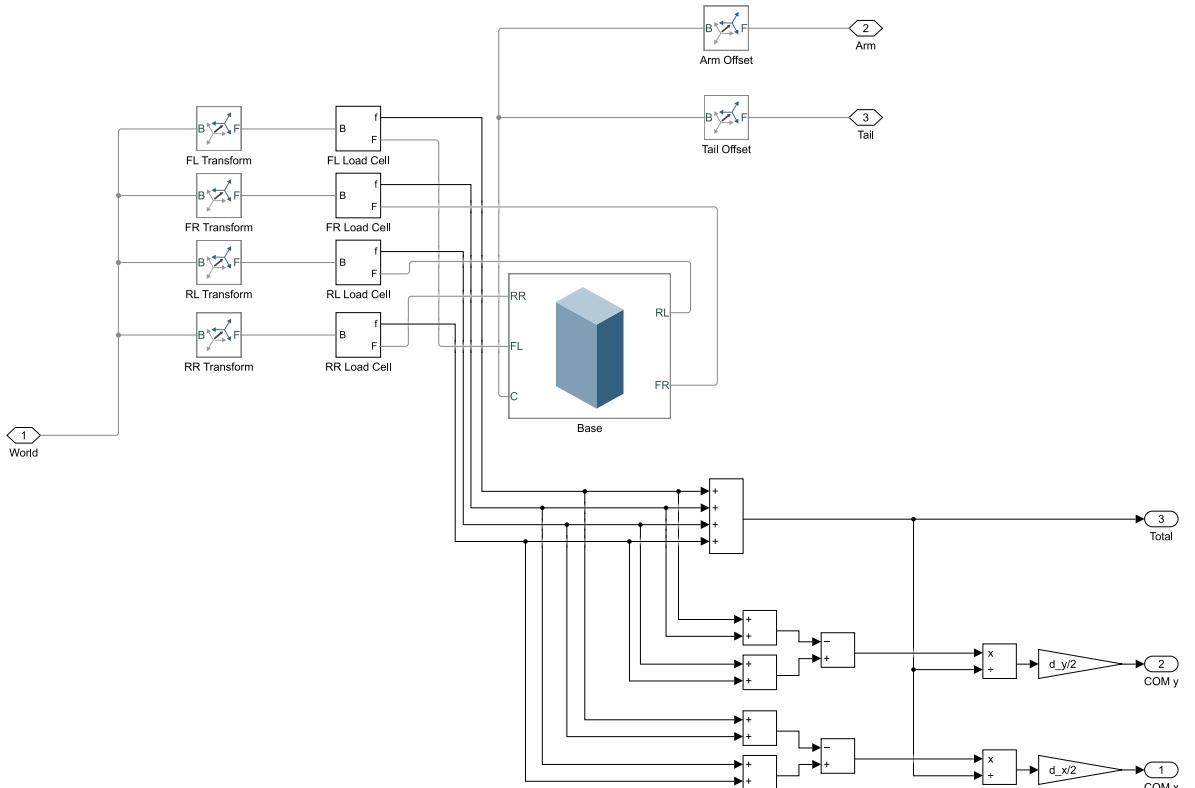


Figure 6.5: Simulink model of the simplified force plate using the Simscape Multibody™ library.

6.4.1 Simulation Construction

6.4.1.1 Simplified Force Plate

For the force plate, a rectangular solid body of dimensions $[b_x \quad b_y \quad 1 \times 10^{-5}]^T$ ² is connected to the ground (or “world frame” as used in the software) via four weld joints on the lower corners. These joints have no DOF, but act as sensors for the force imparted on them, which is measured and used to calculate the COP using the method described in section 6.2. The diagram of the simplified force plate model is shown in figure 6.5.

²The non-zero z value is required due to simulation limitations. As explained in section 6.2 this will have no effect on the results.

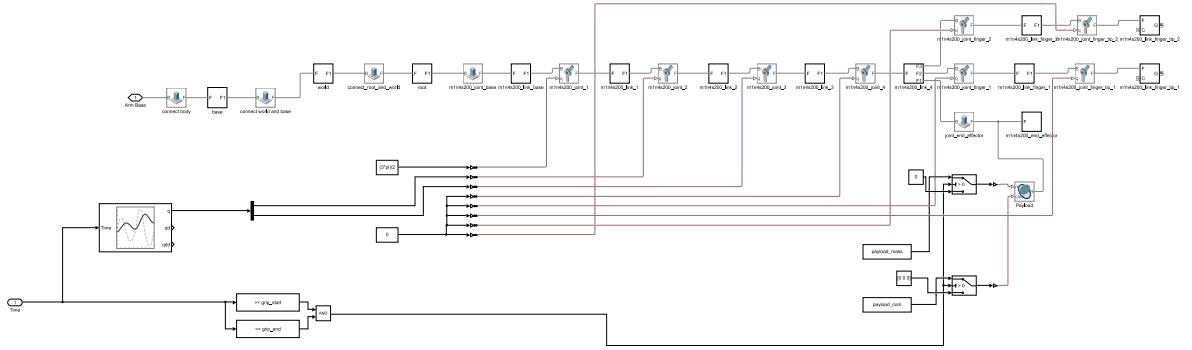


Figure 6.6: Simulink block diagram of the robot arm using the Simscape Multibody™ library.

6.4.1.2 Robot Arm

For the robot arm, a Unified Robotics Description Format (URDF) model of the arm to be used in the experiment, the Kinova™ MICO² 4DOF model with a KG-2 gripper end effector, was imported and automatically converted into a chain of revolute joints and solid bodies. The gripper action was not modelled as their low mass had a negligible contribution to the dynamics of the system. To simulate picking up and setting down a payload, a variable mass and COM solid was attached to the end effector frame of the arm, initially set to the values for the test point being simulated. A boolean switch would then set the mass to 0 outside a defined interval, as in the equation below:

$$\bar{m}_p(t) = \begin{cases} m_p & \text{if } t_r \leq t \leq t_l \\ 0 & \text{otherwise} \end{cases} \quad (6.6)$$

where the interval is defined as $[t_r, t_l]^3$. The diagram of the simplified force plate model is shown in figure 6.6.

6.4.1.3 Robot Tail

For the robot tail, a simple rod and mass model was constructed using cylindrical solids and an actuated universal joint. The diagram of the simplified force plate model is shown in figure 6.10.

6.4.2 Simulation Control System

The control system is a simple proportional controller that is designed to minimise the value of the COP relative to the origin at the center of the base, where a mobile robot would typically have the most stability. Using the calculation for the COP as defined in section 6.2, the tail joint velocity $\dot{\theta}$ is calculated as follows:

³In the simulation, the COM was also controlled in a similar fashion, again due to simulation limitations.

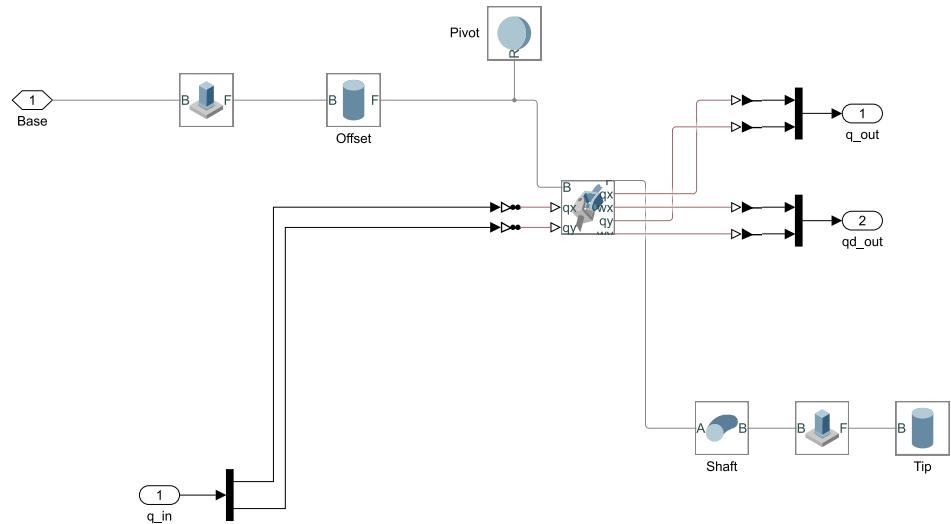


Figure 6.7: Simulink block diagram of the robot tail using the Simscape Multibody™ library.

$$\dot{\theta} = k_p (-CP(\mathbf{F})) \quad (6.7)$$

A low pass filter is also applied to the COP input to mitigate noise from the simulation solver. A block diagram of the simulation control system is shown in figure 6.8.

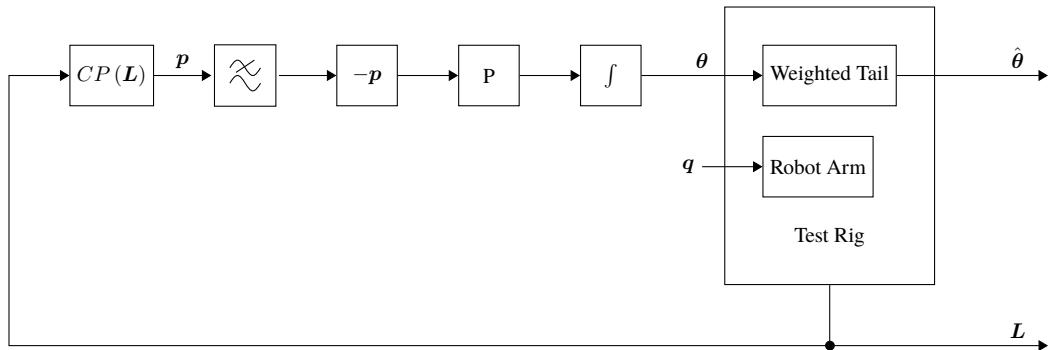


Figure 6.8: Simulation block diagram.

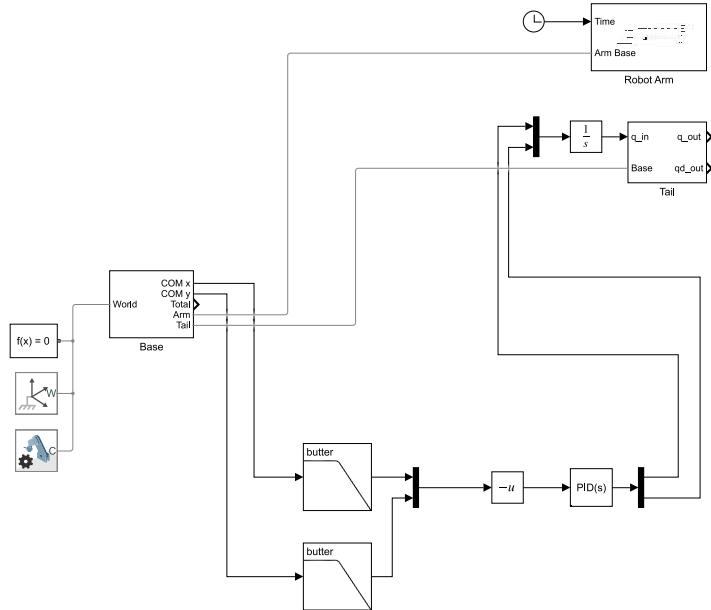


Figure 6.9: Simulink block diagram of the control system using the Simscape Multibody™ library.

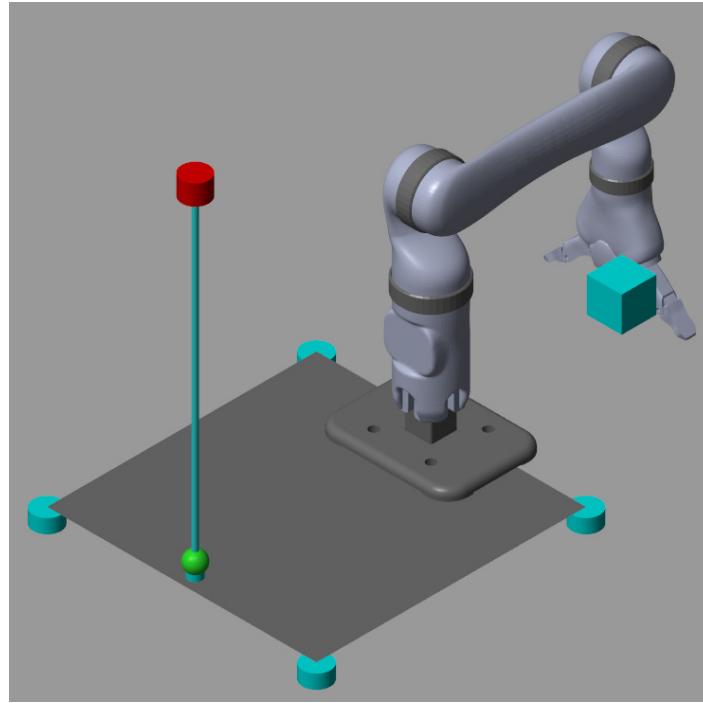


Figure 6.10: 3D preview of the dynamic model of the robot using the Simscape Multibody™ library.

6.5 Experimental Apparatus

The control hardware consists of a National Instruments (NI) myRIO® as a real-time controller connected to a Personal Computer (PC). This provides an interface for the closed loop control of the robot tail, from the load cell inputs to the motor outputs. The tail orientation is also recorded via the myRIO®. The robot arm is controlled via the PC since it has no real-time control driver. A diagram of the control system is shown in figure 6.11, and an annotated photograph of the apparatus is shown in figure 6.12.

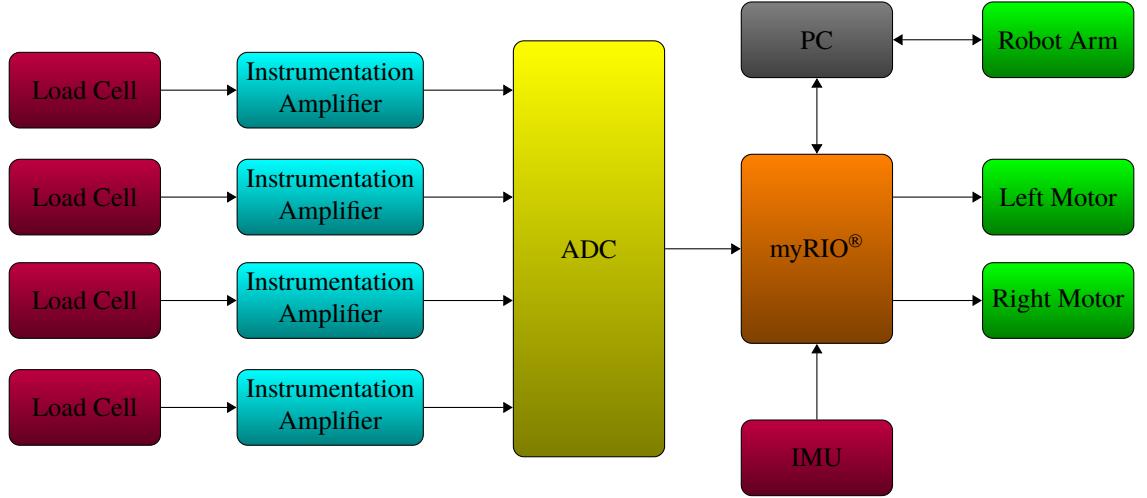


Figure 6.11: System diagram of the static rig.

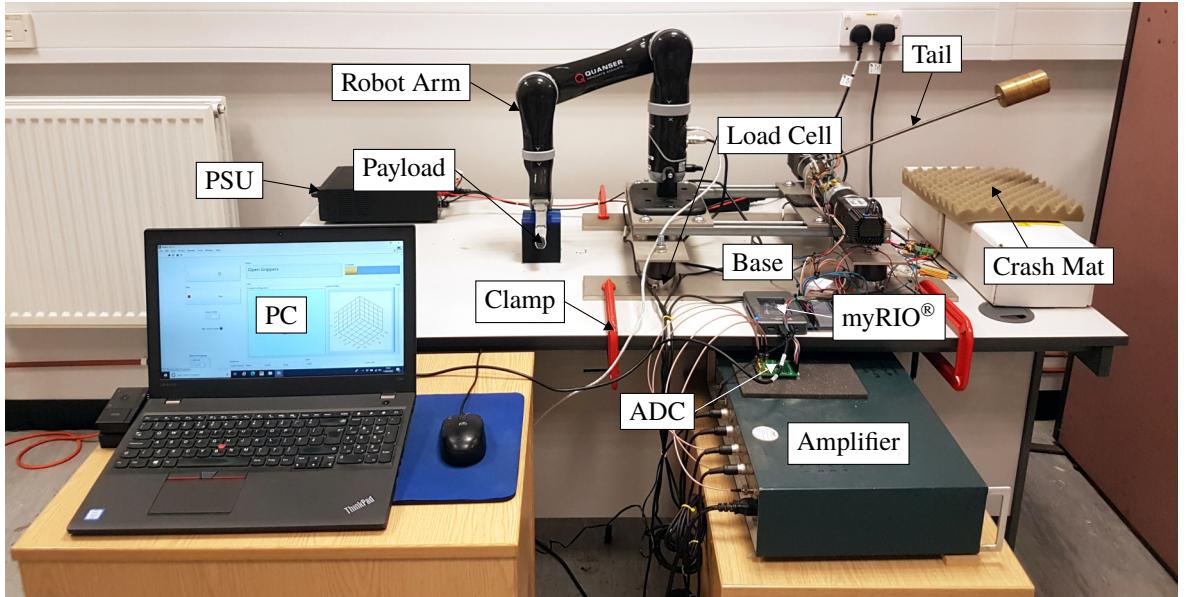


Figure 6.12: Labelled photograph of the static rig.

6.5.1 Base

The base consists of two pairs of steel plates orthogonal to each other, separated by load cells and rubber vibration dampers. The lower plates are clamped to a sturdy table at the four positions indicated in figure 6.13, and a pair of shallow Unistrut® channels are bolted between the upper plates. The plates for the robot arm and tail are then secured to the rig with nuts that slot into the channels, allowing them to be shifted along the Y axis so a_y and t_y can be adjusted.

The load cells are Omega™ LCM204 with a range of $\pm 200\text{N}$, connected to Fylde FE-359-TA instrumentation amplifiers that fed the amplified signal to an Analog Devices AD7606 Analog to Digital Converter (ADC), which was connected to the NI myRIO®. A separate ADC was chosen due the limited number of differential analog inputs on the myRIO®, with only two on the MSP connector [68]. The chosen ADC also had a greater resolution of 16 bits [69] than the built-in myRIO® ADC of 12 bits [68].

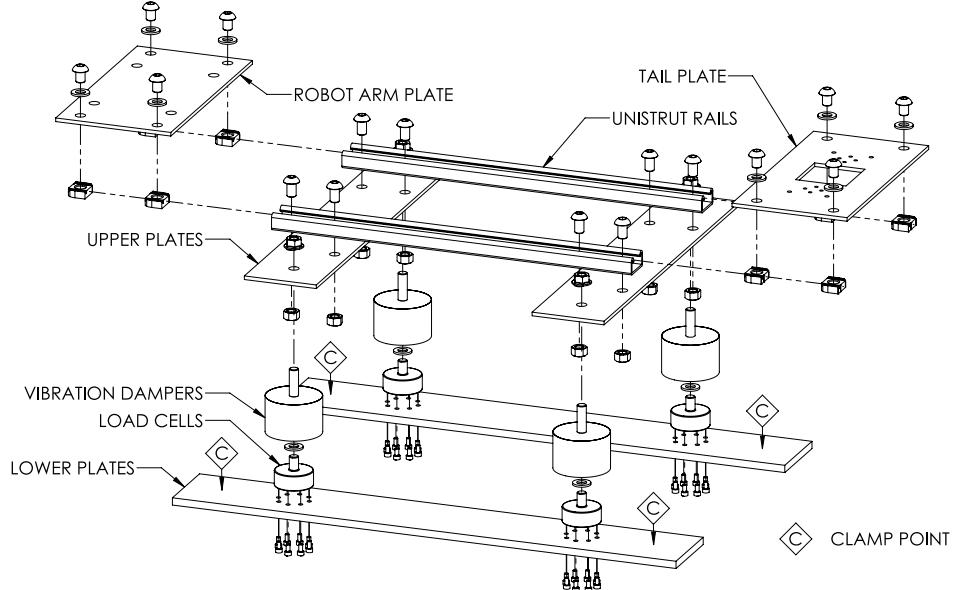


Figure 6.13: Exploded schematic of the base of the static rig, with important components labelled and clamping points marked.

6.5.2 Robot Arm

The robot arm is a Kinova™ MICO² 4DOF model with a KG-2 gripper end effector. It can carry a maximum payload of 2 kg when fully extended, though in experimental conditions for the configurable payload designed in chapter 5 the friction force between the grippers and textured surface of the container was insufficient for loads much lower than this limit. It is attached to the base using a mount which is bolted to the robot arm plate.

6.5.3 Robot Tail

While the actuator designed in chapter — proved that it could be a successful actuation system for the robot tail, it was not considered mature enough to be applied to this experiment. Instead, a simpler design was used, based on the common “inverted pendulum” designs used in many of the publications found in chapter 2, with a 2 DOF joint attached to a rod with a mass on the end.

6.5.3.1 Bevel Gear Mechanism

The mechanical design of the tail joint uses the design from [17], which uses a 3 way bevelled gear system to create a 2 DOF joint, which allows both motors to be mounted to the base, increasing the available tail torque. This design works as shown in figure 6.14, where rotating both motors in the same direction rotates the tail joint around the axis of the *driving* gears connected to the motor shafts, and rotating the motors in opposite directions rotates the tail in the axis of the *driven* gear attached to the tail.

In order to translate motor rotations to tail motion, the following equation can be used, where $\phi = [\phi_1 \ \phi_2]^\top$ are the motor angles:

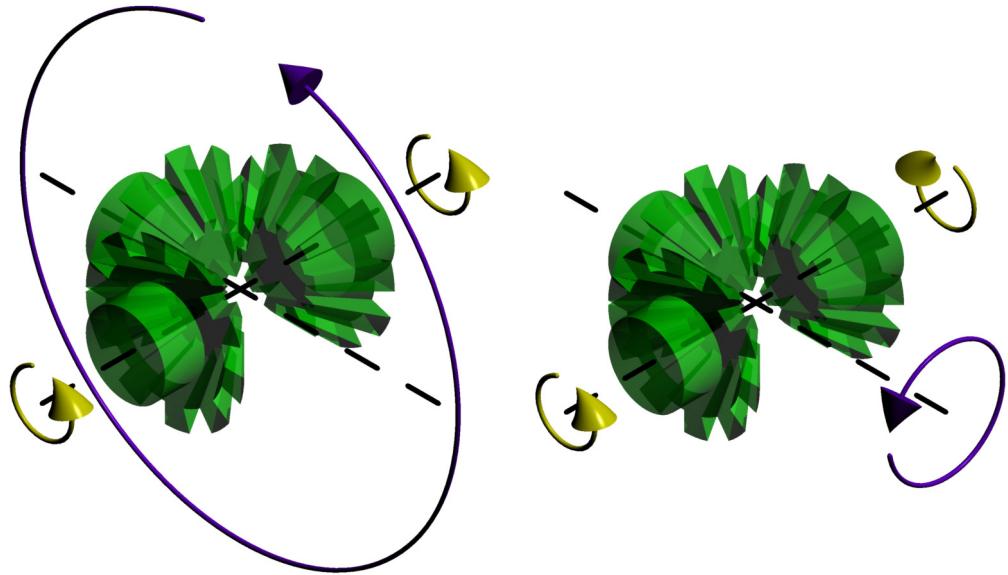


Figure 6.14: Visualisation of the bevel gear system. When both driving gears turn in the same direction, the driven gear rotates around the driving gears axis. When the driving gears turn in opposite directions, it rotates around its own axis.

$$\dot{\phi} = \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix} \dot{\theta} \quad (6.8)$$

6.5.3.2 Motors

The motors chosen for the tail mechanism are Crouzet DCmind Brushless Gearmotors, with a maximum output speed of 208 min^{-1} at 12 V(DC) and a maximum torque of 6.8 N m [70]. These consisted of a brushless motor and a 19:1 ratio gearbox. 6.8 N m is comfortably in excess of half the maximum torque required to move the tail, which can be calculated as $\tau_{\max} = l_t m_t g$ which corresponds to a horizontal tail orientation. Using the coefficients in table 6.1, this is 7.1 N m . Cheaper brushed motors were initially used, but were unable to move reliably at low velocity, making smooth control of the tail difficult.

6.5.3.3 Inertial Measurement Unit

The tail also includes a Bosch Sensortec BNO080 IMU as used in chapter —, mounted to the base of the tail to sense the angular orientation of the tail. This is used to limit the tail range and for data acquisition purposes.

6.5.4 Control System

The control system is similar to the system used in the simulation in section 6.4.2, with the addition of a saturation function to replicate the joint limits, and the function to convert the joint angles to motor angles as defined in section 6.5.3. There is also a conversion from

rad s^{-1} to min^{-1} (Rotations per Minute (RPM)) and a gain to compensate for the gear reduction. A block diagram of the system is shown in figure 6.15.

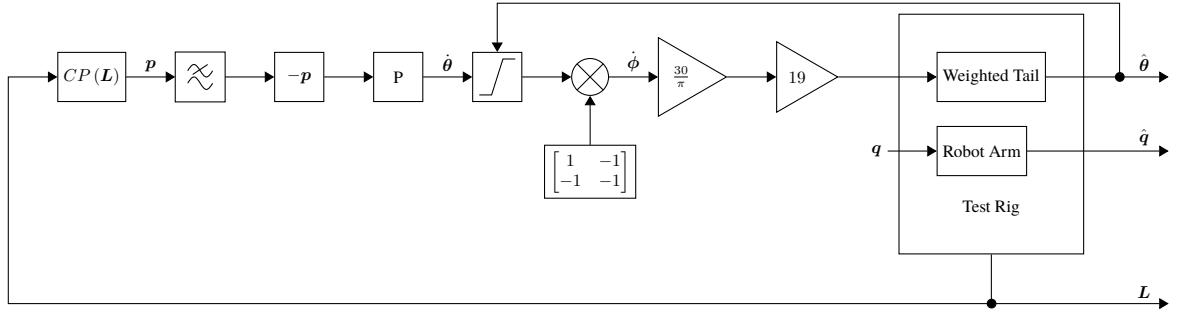


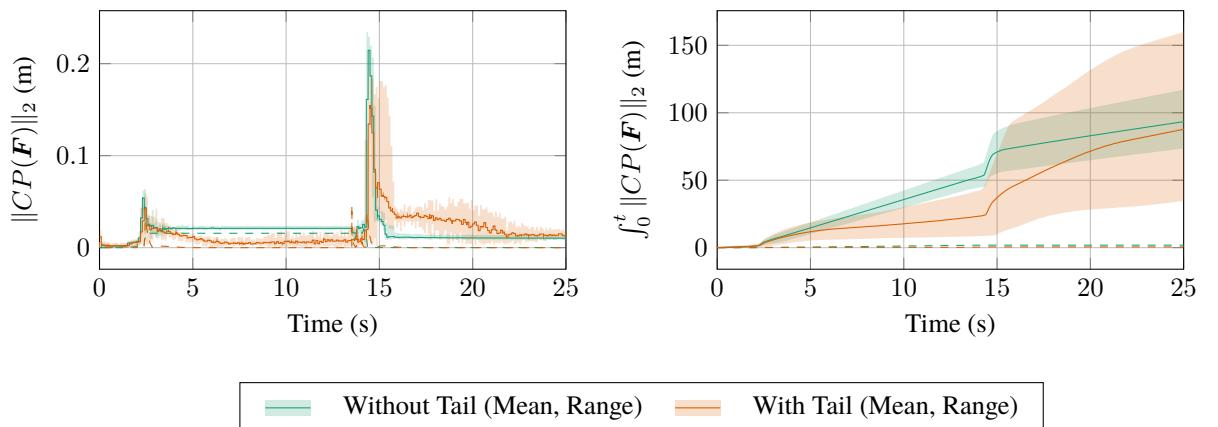
Figure 6.15: Block diagram of the experimental control system.

6.6 Results

For each test point, 20 experiments were run, 10 *control* experiments without the tail attached, and 10 *tail* experiments with the feedback controlled tail. The mean values for the 10 experiments were then taken as the measurements, with the minimum and maximum values considered as the measurement error.

In order to provide a fair comparison between the control and tail experiments, a perfectly balanced initial condition is assumed for the control experiments. This was done by offsetting the COP data by the value of the initial condition.

$$1.291 [0.000 \quad -0.000 \quad -0.008]$$



In order to compare the performance with and without the robot tail, the stability measurement given by the COP can be quantified into a “cost function” that gives a single scalar or two-vector (for both components of the COP) value for each test point. These values can then be used to score the performance of the experiment, and provide a means to compare the control and tail experiments.

Since the area of interest for the experiments is the carry

6.6.1 Minimum COP Deviation

This cost function measures the smallest (or minimum) value of the COP. This effectively measures the most stable state of the system during the experiment. This can be defined by the following equation:

$$\min \{CP(\mathbf{F}(t)) \mid t \in T, 5 < t < 13\} \quad (6.9)$$

Figure 6.16 shows the mean minimum COP deviation position for the control and tail experiments. As can be clearly seen, the y component of the COP is reduced to less than mm in every tail experiment. Though the x component does increase, this is by at least an order of magnitude less than the reduction in the y component. This is due to instabilities caused by the dynamic properties of the motor and tail as explained in section —.

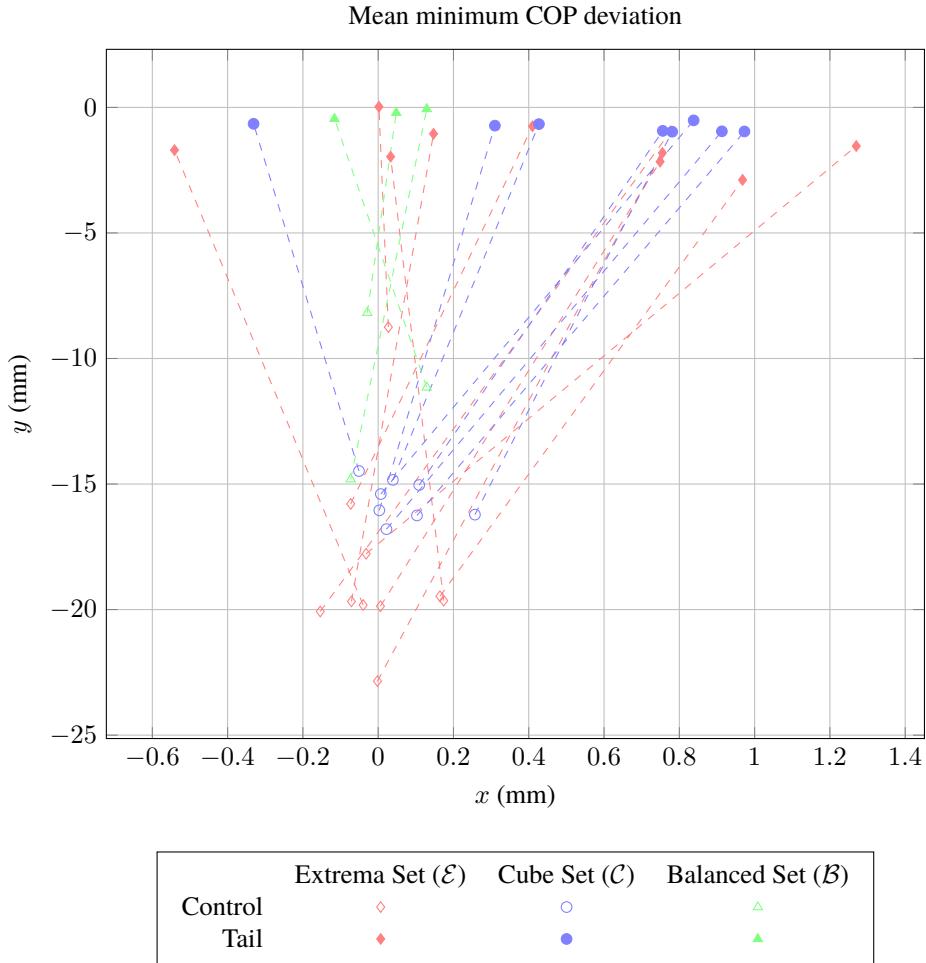


Figure 6.16: Graph of the mean COP during the *carry* interval of the experiment.

The confidence in the change in value can be measured using a paired t-test, which measures the probability of two hypothesis, that a set of value pairs *is* zero and a set of value pairs is *not* zero. It does this by comparing the mean difference between each pair of values. The result from all sets shows a statistically significant mean difference bewteen the y component of the cost function between the control and tail experiments, with $p = 1.826 \times 10^{-15}$. The mean reduction is not stastically useful, since it would be expected to very greatly

depending on the mass and COM of the payload.

6.6.2 Cumulative COP Deviation 2-Norm

This cost function measures the cumulative deviation of the euclidean norm of the COP. Whereas the previous cost function measured the most stable state the experiment achieved, this measures how well the stability was maintained over the entire interval. This can be defined by the following equation:

$$\sum_{t \in \{T, 5 < t < 13\}} \|CP(\mathbf{F}(t))\|_2 \quad (6.10)$$

Figure 6.17 shows the mean cumulative COP deviation 2-norm for the control and tail experiments. Every tail experiment achieves a significant reduction of this cost function.

Using the same paired t-test used in the previous cost function, the reduction is again shown to be statistically significant, with $p = 4.780 \times 10^{-14}$.

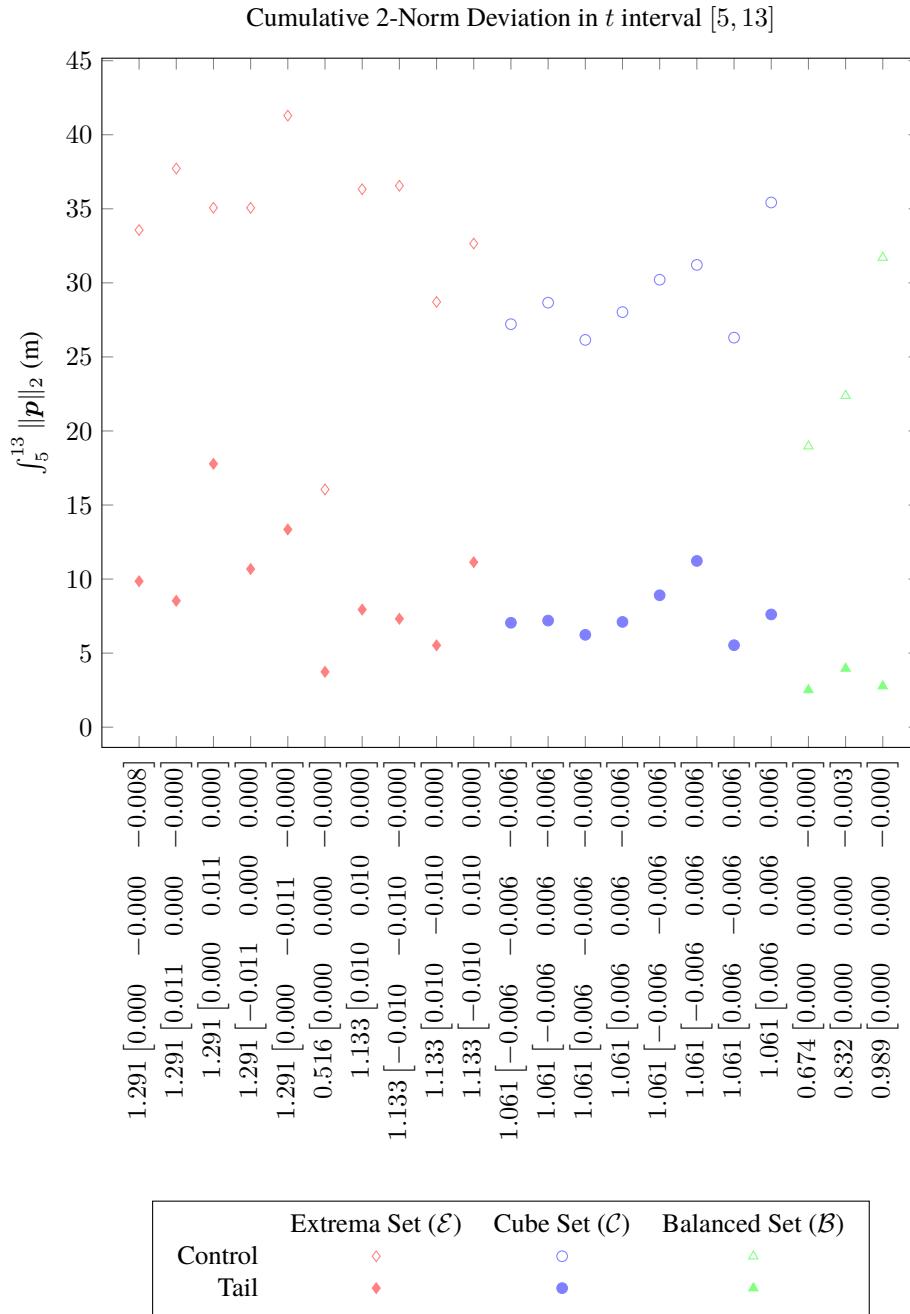


Figure 6.17

6.7 Discussion

6.7.1 Payload and Tail Angle Regression Analysis

To examine the effects of varying the payload mass and COM on the tail, a regression study was conducted comparing mass and COM *inputs* to tail angle *outputs*, specifically the peak angle output measured by the maximum magnitude function, where θ represents the vector of tail angle measurements from the experiment:

$$\wedge(\theta) = \begin{cases} \max \theta & |\max \theta| \leq |\min \theta| \\ \min \theta & |\max \theta| > |\min \theta| \end{cases} \quad (6.11)$$

Figure 6.18 shows the correlations between the maximum magnitude of the tail on both pitch and roll axes, and the mass and corresponding COM of the payload. A change in the x component of the COM would only affect the roll of the tail, and the y component would only affect the pitch, but a change in mass could effect both, though the y component to a much greater degree.

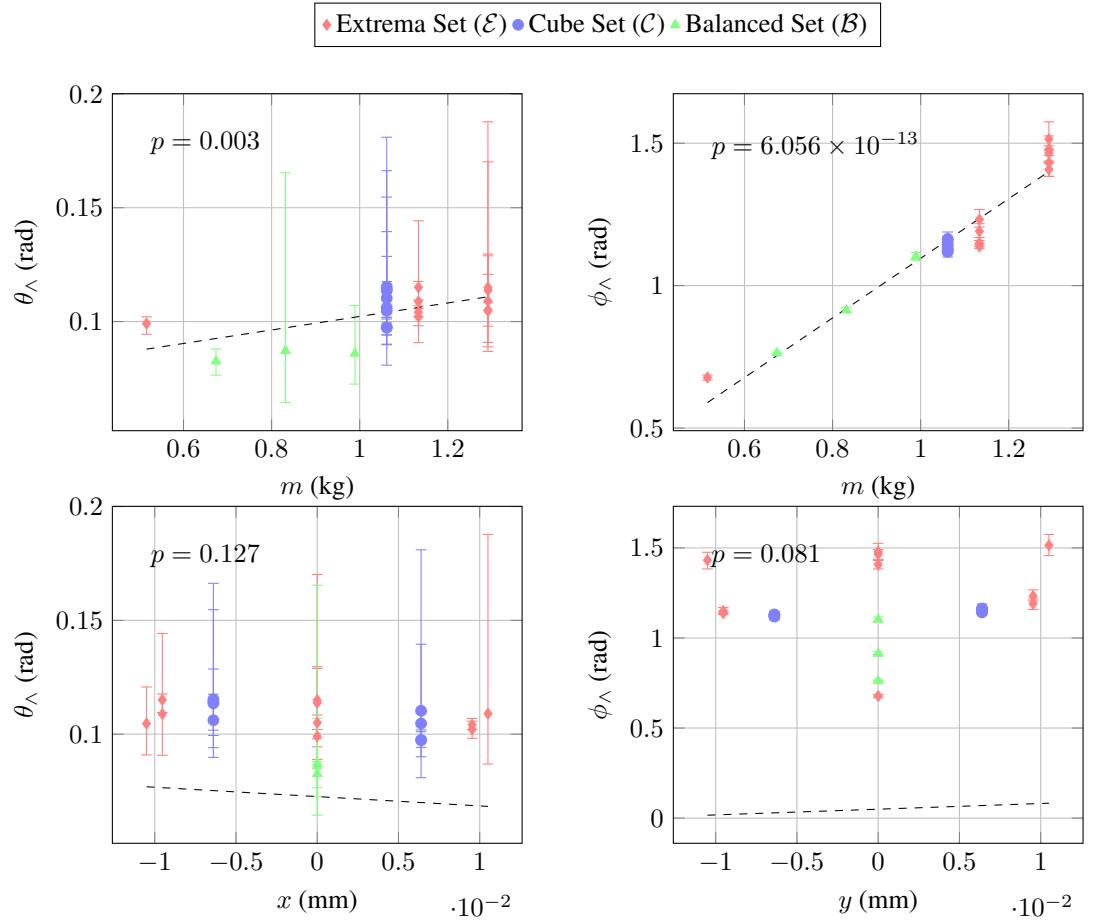


Figure 6.18

The results show that there is a strong correlation ($p = 6.056 \times 10^{-13}$) between the payload mass and the pitch maximum magnitude of the tail of $1.046 \text{ rad kg}^{-1}$.

6.7.2 Vibration Analysis

A major issue with the experiments that likely prevented the collection of more accurate data and more stable control of the tail was the vibration of the tail shaft. This was due to greater than expected elastic deformation of the rod that held the weight at the end of the tail.

In order to characterise this vibration, the tail was removed from the system and subjected to an impact test in order to find the resonant frequency. This was achieved by fixing the base of the tail and striking the tip mass with a small steel hammer. The BNO080 IMU was fixed above the mass and the accelerometer data was recorded. Figure 6.19 shows the resulting y component of the accelerometer data, and a Discrete Fourier Transform (DFT) of the data shows the resonant frequency.

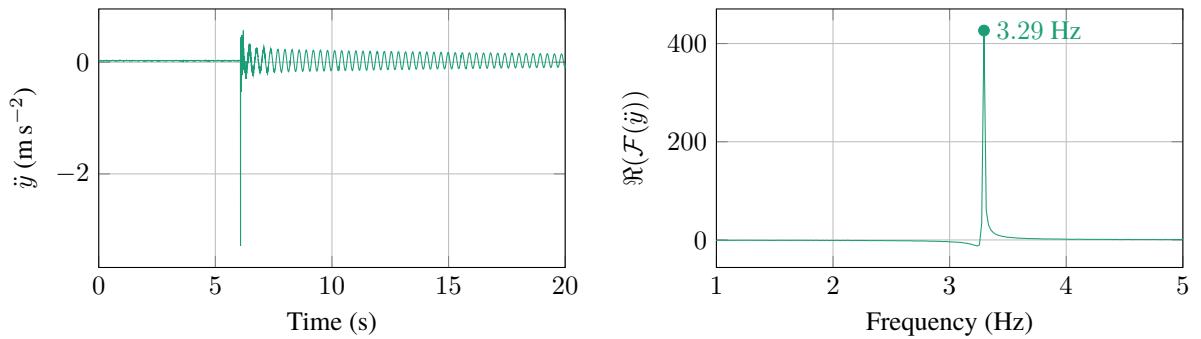


Figure 6.19: Resonant frequency of the tail calculated from accelerometer data captured during an impact with a small steel hammer. The left graph is the accelerometer data, and the right graph is the real part from a DFT on the data.

What to put here?

6.8 Conclusion

This could potentially be alleviated by using a stiffer material for the shaft, or using a shaft of a larger diameter. However, this would likely increase the mass of the tail shaft, or would require more specialised materials to be sought.

References

- [1] M. Gor, P. Pathak, A. Samantaray, J.-M. Yang, and S. Kwak, “Fault accommodation in compliant quadruped robot through a moving appendage mechanism,” *Mechanism and Machine Theory*, vol. 121, pp. 228–244, 2018.
- [2] A. Ekizos, A. Santuz, A. Schroll, and A. Arampatzis, “The maximum lyapunov exponent during walking and running: Reliability assessment of different marker-sets,” *Frontiers in physiology*, p. 1101, 2018.
- [3] M. Vukobratović and B. Borovac, “Zero-moment point—thirty five years of its life,” *International journal of humanoid robotics*, vol. 1, no. 01, pp. 157–173, 2004.
- [4] H. Montes, S. Nabulsi, and M. Armada, “Detecting zero-moment point in legged robot,” in *Climbing and Walking Robots*, Springer, 2005, pp. 229–236.
- [5] T. Akbas, S. E. Eskimez, S. Ozel, O. K. Adak, K. C. Fidan, and K. Erbatur, “Zero moment point based pace reference generation for quadruped robots via preview control,” in *2012 12th IEEE International Workshop on Advanced Motion Control (AMC)*, IEEE, 2012, pp. 1–7.
- [6] A. W. Winkler, F. Farshidian, D. Pardo, M. Neunert, and J. Buchli, “Fast trajectory optimization for legged robots using vertex-based ZMP constraints,” *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2201–2208, 2017.
- [7] S. Lapapong, A. Brown, and S. Brennan, “Experimental validation of terrain-aware rollover prediction for ground vehicles using the zero-moment point method,” p. 6, Oct. 2020.
- [8] C. Walker, C. J. Vierck Jr, and L. A. Ritz, “Balance in the cat: Role of the tail and effects of sacrocaudal transection,” *Behavioural brain research*, vol. 91, no. 1-2, pp. 41–47, 1998.
- [9] G. B. Gillis, L. A. Bonvini, and D. J. Irschick, “Losing stability: Tail loss and jumping in the arboreal lizard *anolis carolinensis*,” *Journal of Experimental Biology*, vol. 212, no. 5, pp. 604–609, 2009, ISSN: 0022-0949. doi: 10.1242/jeb.024349. eprint:

<https://jeb.biologists.org/content/212/5/604.full.pdf>. [Online].

Available: <https://jeb.biologists.org/content/212/5/604>.

- [10] C. C. Kessens and J. Dotterweich,
“Ground-based self-righting using inertial appendage methods,”
in *Unmanned Systems Technology XIX*,
International Society for Optics and Photonics, vol. 10195, 2017, p. 1 019 505.
doi: 10.1117/12.2262535.
- [11] Y.-H. Kim and D. A. Shell, “Bound to help: Cooperative manipulation of objects via compliant, unactuated tails,” *Autonomous Robots*, pp. 1–20, 2018.
- [12] B. McInroe, H. C. Astley, C. Gong, *et al.*, “Tail use improves performance on soft substrates in models of early vertebrate land locomotors,”
Science, vol. 353, no. 6295, pp. 154–158, 2016. doi: 10.1126/science.aaf0984.
- [13] Z. Xiuli, G. Jiaqing, and Y. Yanan, “Effects of head and tail as swinging appendages on the dynamic walking performance of a quadruped robot,”
Robotica, vol. 34, no. 12, pp. 2878–2891, 2016.
doi: 10.1017/S0263574716000011.
- [14] A. Patel and M. Braae,
“Rapid acceleration and braking: Inspirations from the cheetah’s tail,”
in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, IEEE, 2014, pp. 793–799. doi: 10.1109/ICRA.2014.6906945.
- [15] 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, 2009.
- [16] A. Patel and M. Braae,
“Rapid turning at high-speed: Inspirations from the cheetah’s tail,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, IEEE, 2013, pp. 5506–5511. doi: 10.1109/IROS.2013.6697154.
- [17] A. Patel and E. Boje, “On the conical motion of a two-degree-of-freedom tail inspired by the cheetah,”
IEEE Transactions on Robotics, vol. 31, no. 6, pp. 1555–1560, 2015.
doi: 10.1109/TRO.2015.2495004.
- [18] A. O. Pullin, N. J. Kohut, D. Zarrouk, and R. S. Fearing,
“Dynamic turning of 13 cm robot comparing tail and differential drive,”
in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, IEEE, 2012, pp. 5086–5093. doi: 10.1109/ICRA.2012.6225261.

- [19] N. J. Kohut, A. O. Pullin, D. W. Haldane, D. Zarrouk, and R. S. Fearing, “Precise dynamic turning of a 10 cm legged robot on a low friction surface using a tail,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, IEEE, 2013, pp. 3299–3306. doi: [10.1109/ICRA.2013.6631037](https://doi.org/10.1109/ICRA.2013.6631037).
- [20] K. Takita, T. Katayama, and S. Hirose, “Development of dinosaur-like robot titrus-motion control of the head and tail of miniature robot titrus-iii,” in *SICE 2002. Proceedings of the 41st SICE Annual Conference*, IEEE, vol. 5, 2002, pp. 2984–2989. doi: [10.1109/SICE.2002.1195580](https://doi.org/10.1109/SICE.2002.1195580).
- [21] R. Briggs, J.-W. Lee, M. Haberland, and S.-B. Kim, “Tails in biomimetic design: Analysis, simulation, and experiment,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, IEEE, 2012, pp. 1473–1480. doi: [10.1109/IROS.2012.6386240](https://doi.org/10.1109/IROS.2012.6386240).
- [22] R. Sato, S. Hashimoto, A. Ming, and M. Shimojo, “Development of a flexible tail for legged robot,” in *Mechatronics and Automation (ICMA), 2016 IEEE International Conference on*, IEEE, 2016, pp. 683–688. doi: [10.1109/ICMA.2016.7558645](https://doi.org/10.1109/ICMA.2016.7558645).
- [23] S. W. Heim, M. Ajallooeian, P. Eckert, M. Vespiagnani, and A. J. Ijspeert, “On designing an active tail for legged robots: Simplifying control via decoupling of control objectives,” *Industrial Robot: An International Journal*, vol. 43, no. 3, pp. 338–346, 2016. doi: [10.1108/IR-10-2015-0190](https://doi.org/10.1108/IR-10-2015-0190).
- [24] L. Guan-Horng, L. Hou-Yi, L. Huai-Yu, C. Shao-Tuan, and L. Pei-Chun, “A bio-inspired hopping kangaroo robot with an active tail,” *Journal of Bionic Engineering*, vol. 11, no. 4, pp. 541–555, 2014. doi: [10.1016/S1672-6529\(14\)60066-4](https://doi.org/10.1016/S1672-6529(14)60066-4).
- [25] J. An, T. Chung, C. H. D. Lo, C. Ma, X. Chu, and K. S. Au, “Development of a bipedal hopping robot with morphable inertial tail for agile locomotion,” in *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, IEEE, 2020, pp. 132–139.
- [26] E. Chang-Siu, T. Libby, M. Tomizuka, and R. J. Full, “A lizard-inspired active tail enables rapid maneuvers and dynamic stabilization in a terrestrial robot,” in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, IEEE, 2011, pp. 1887–1894. doi: [10.1109/IROS.2011.6094658](https://doi.org/10.1109/IROS.2011.6094658).

- [27] Z. Jianguo, Z. Tianyu, X. Ning, M. W. Mutka, and X. Li, “Msu tailbot: Controlling aerial maneuver of a miniature-tailed jumping robot,” *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 6, pp. 2903–2914, 2015.
doi: 10.1109/TMECH.2015.2411513.
- [28] T. Libby, A. M. Johnson, E. Chang-Siu, R. J. Full, and D. E. Koditschek, “Comparative design, scaling, and control of appendages for inertial reorientation,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1380–1398, 2016.
doi: 10.1109/TRO.2016.2597316.
- [29] T. Libby, T. Y. Moore, E. Chang-Siu, *et al.*, “Tail-assisted pitch control in lizards, robots and dinosaurs,” *Nature*, vol. 481, no. 7380, pp. 181–184, 2012.
doi: doi:10.1038/nature10710.
- [30] J. A. Nyakatura, K. Melo, T. Horvat, *et al.*, “Reverse-engineering the locomotion of a stem amniote,” *Nature*, vol. 565, no. 7739, pp. 351–355, 2019.
- [31] B. Simon, R. Sato, J.-Y. Choley, and A. Ming, “Development of a bio-inspired flexible tail system,” in *2018 12th France-Japan and 10th Europe-Asia Congress on Mechatronics*, IEEE, 2018, pp. 230–235.
- [32] W. S. Rone, W. Saab, and P. Ben-Tzvi, “Design, modeling, and integration of a flexible universal spatial robotic tail,” *Journal of Mechanisms and Robotics*, vol. 10, no. 4, p. 041001, 2018.
- [33] W. Saab, W. S. Rone, and P. Ben-Tzvi, “Discrete modular serpentine robotic tail: Design, analysis and experimentation,” *Robotica*, pp. 1–25, 2018.
- [34] W. Saab, J. Yang, and P. Ben-Tzvi, “Modeling and control of an articulated tail for maneuvering a reduced degree of freedom legged robot,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 2695–2700.
- [35] W. S. Rone and P. Ben-Tzvi, “Continuum robotic tail loading analysis for mobile robot stabilization and maneuvering,” in *ASME 2014 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2014*, American Society of Mechanical Engineers, 2014. doi: 10.1115/DETC2014-34678.
- [36] Y. Liu, J. Wang, and P. Ben-Tzvi, “A cable length invariant robotic tail using a circular shape universal joint mechanism,” *Journal of Mechanisms and Robotics*, vol. 11, no. 5, 2019.

- [37] Y. Liu and P. Ben-Tzvi, “Design, analysis, and integration of a new two-degree-of-freedom articulated multi-link robotic tail mechanism,” *Journal of Mechanisms and Robotics*, vol. 12, no. 2, 2020.
- [38] Z. Mu, X. Liu, W. Xu, and Y. Lou, “A multi-legged biomimetic robot with double-motion modes and 4-dof-balance tail,” in *2018 13th World Congress on Intelligent Control and Automation (WCICA)*, IEEE, 2018, pp. 1611–1616.
- [39] N. Iwamoto and M. Yamamoto, “Jumping motion control planning for 4-wheeled robot with a tail,” in *System Integration (SII), 2015 IEEE/SICE International Symposium on*, IEEE, 2015, pp. 871–876. doi: [10.1109/SII.2015.7405114](https://doi.org/10.1109/SII.2015.7405114).
- [40] N. Iwamoto and A. Nishikawa, “Distributed model predictive control-based approach for flexible robotic tail,” *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 31–36, 2018.
- [41] W. S. Rone and P. Ben-Tzvi, “Static modeling of a multi-segment serpentine robotic tail,” American Society of Mechanical Engineers, 2015. doi: [10.1115/DETC2015-46655](https://doi.org/10.1115/DETC2015-46655).
- [42] Y. Liu and P. Ben-Tzvi, “Dynamic modeling, analysis, and comparative study of a quadruped with bio-inspired robotic tails,” *Multibody System Dynamics*, vol. 51, no. 2, pp. 195–219, 2021.
- [43] R. Buckingham and A. Graham, “Nuclear snake-arm robots,” *Industrial Robot: An International Journal*, 2012.
- [44] M. Luo, R. Yan, Z. Wan, *et al.*, “Orisnake: Design, fabrication, and experimental analysis of a 3-d origami snake robot,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1993–1999, 2018.
- [45] W. S. Rone, W. Saab, and P. Ben-Tzvi, “Design, Modeling, and Integration of a Flexible Universal Spatial Robotic Tail,” *Journal of Mechanisms and Robotics*, vol. 10, no. 4, Apr. 2018, 041001, ISSN: 1942-4302. doi: [10.1115/1.4039500](https://doi.org/10.1115/1.4039500).
- [46] T. Würtz, C. May, B. Holz, C. Natale, G. Palli, and C. Melchiorri, “The twisted string actuation system: Modeling and control,” in *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Jul. 2010, pp. 1215–1220. doi: [10.1109/AIM.2010.5695720](https://doi.org/10.1109/AIM.2010.5695720).

- [47] P. Muehlbauer, M. Schimbera, K. Stewart, and P. P. Pott, “Twisted string actuation for an active modular hand orthosis,” in *ACTUATOR; International Conference and Exhibition on New Actuator Systems and Applications 2021*, VDE, 2021, pp. 1–4.
- [48] J. Park, J.-i. Park, H.-T. Seo, Y. Liu, K.-S. Kim, and S. Kim, “Control of tendon-driven (twisted-string actuator) robotic joint with adaptive variable-radius pulley,” in *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, IEEE, 2020, pp. 1096–1098.
- [49] B. Suthar and S. Jung, “Design and feasibility analysis of a foldable robot arm for drones using a twisted string actuator: Frad-tsa,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5769–5775, 2021.
doi: 10.1109/LRA.2021.3084890.
- [50] S. Nedelchev, I. Gaponov, and J. Ryu, “Accurate dynamic modeling of twisted string actuators accounting for string compliance and friction,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3438–3443, 2020.
doi: 10.1109/LRA.2020.2970651.
- [51] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons, 2020.
- [52] F. Dessen, “Coordinating control of a two degrees of freedom universal joint structure driven by three servos,” in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3, Apr. 1986, pp. 817–822. doi: 10.1109/ROBOT.1986.1087559.
- [53] *Dc micromotors - precious metal commutation*, Dr. Fritz Fauhaber GmbH & Co. KG, Feb. 2020.
- [54] J. Shigley, C. Mischke, and R. Budynas, *Mechanical Engineering Design* (McGraw-Hill series in mechanical engineering). McGraw-Hill, 2004, ISBN: 9780072520361.
- [55] W. S. Rone and P. Ben-Tzvi, “Dynamic modeling and simulation of a yaw-angle quadruped maneuvering with a planar robotic tail,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 138, no. 8, p. 084502, 2016.
doi: 10.1115/1.4033103.
- [56] R. Boulic, R. Mas, and D. Thalmann, “Inverse kinetics for center of mass position control and posture optimization,”

- in *Image Processing for Broadcast and Video Production*, Springer, 1995, pp. 234–249.
- [57] S. R. Buss, “Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods,” *IEEE Journal of Robotics and Automation*, vol. 17, no. 1-19, p. 16, 2004.
- [58] S. Ma, “A balancing technique to stabilize local torque optimization solution of redundant manipulators,” *Journal of Robotic Systems*, vol. 13, no. 3, pp. 177–185, 1996.
- [59] Y. Zhang, D. Guo, and S. Ma, “Different-level simultaneous minimization of joint-velocity and joint-torque for redundant robot manipulators,” *Journal of Intelligent & Robotic Systems*, vol. 72, no. 3-4, pp. 301–323, 2013.
- [60] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [61] A. des Sciences (Paris), *Comptes rendus hebdomadaires des séances de l'Académie des sciences*. Gauthier-Villars, 1857, vol. 45.
- [62] P. Serafini, “Simulated annealing for multi objective optimization problems,” in *Multiple criteria decision making*, Springer, 1994, pp. 283–292.
- [63] A. M. (M. James, *Macmillan's chemical and physical data*, eng. London: Macmillan, 1992, ISBN: 0333511670.
- [64] *Crc handbook of chemistry and physics*. eng, Boca Raton, Fla, 1978.
- [65] H. L. Bartlett, L. H. Ting, and J. T. Bingham, “Accuracy of force and center of pressure measures of the wii balance board,” *Gait & posture*, vol. 39, no. 1, pp. 224–228, 2014.
- [66] J. M. Leach, M. Mancini, R. J. Peterka, T. L. Hayes, and F. B. Horak, “Validating and calibrating the nintendo wii balance board to derive reliable center of pressure measures,” *Sensors*, vol. 14, no. 10, pp. 18 244–18 267, 2014.
- [67] *Kinova mico™ robotic arm 4dof specifications*, 1.2, Kinova, Inc., Apr. 2018.
 [Online]. Available: <https://drive.google.com/file/d/1xCmpqSDMZvKU4IypSaXBLIEX6p7pigGr/view>.
- [68] *NI myRIO-1900 user guide and specifications*, National Instruments, May 2016.
- [69] *AD7606/AD7606-6/AD7606-4 data sheet*, version F, Analog Devices, Apr. 2020.
- [70] *Dcmind brushless gearmotor data sheet*, Crouzet Automatismes, Jul. 2016.

Appendices