

Adaptive Locomotion: The Cylindabot Robot

**How hybrid locomotion can improve a robot's ability
to move**

Robert Woolley

PhD

University of York

Physics, Engineering and Technology

March 2023

Abstract

Adaptive locomotion is an emerging field of robotics due to the complex interaction between the robot and its environment. Hybrid locomotion is where a robot has more than one mode of locomotion and potentially delivers the benefits of both, however, these advantages are often not quantified or applied to new scenarios. The classic approach is to design robots with a high number of degrees of freedom and a complex control system, whereas an intelligent morphology can simplify the problem and maintain capabilities. Cylindabot is designed to be a minimally actuated hybrid robot with strong terrain crossing capabilities. By limiting the number of motors, this reduces the robot's weight and means less reinforcement is needed for the physical frame or drive system. Cylindabot uses different drive directions to transform between using wheels or legs. Cylindabot is able to climb a slope of 32 degrees and a step ratio of 1.43 while only being driven by two motors. A physical prototype and simulation models show that adaptation is optimal for a range of terrain (slopes, steps, ridges and gaps). Cylindabot successfully adapts to a map environment where there are several routes to the target location. These results show that a hybrid robot can increase its terrain capabilities when changing how it moves and that this adaptation can be applied to wider environments. This is an important step to have hybrid robots being deployed to real situations.

Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References.

Contents

Abstract	2
Contents	4
Chapter 1: Introduction	8
1.1 Locomotion	9
1.2 Hypothesis	10
1.3 Chapter outline	10
Chapter 2: Literature review	11
Chapter 3: Preliminary Results	11
Chapter 4: Cylindabot Design	11
Chapter 5: Slope and Step Obstacles	11
Chapter 6: Grid Terrain	12
Chapter 7: Ridges and Bridges Map	12
Chapter 8: Conclusion	12
1.4 Contributions	12
Chapter 2: Literature Review	14
Existing robotic designs and theory of locomotion.	14
2.1 Introduction	14
2.2 Morphology	14
2.2.1 Classification by action ^[OBJ]	15
2.2.2 Robotic Platforms	16
2.2.3 Rolling Robots	17
2.2.4 Walking Robots	19
2.2.5 Jumping	23
2.2.6 Hybrid	24
2.2.7 Comparison	28
2.3 Control	30
2.3.1 Unstable Walking and Jumping	31
2.3.2 Stable Walking	32
2.3.3 Rolling Control	33
2.4 Environment and robotic testing	33
2.4.1 Applications of Locomotion	34
2.4.2 Modelling grip - Terramechanics	35
2.4.3 Navigation	36
2.4.4 Benchmarks of robots' ability	37
2.4.5 Cost of Transport	38
2.4.6 Froude Number	40

2.4.7 Contact area	41
2.4.8 Test obstacles	42
2.5 Summary	43
Chapter 3: Preliminary Results	45
Early versions of Cylindabot tested in simulation against slopes and steps.	45
3.1 Introduction	45
3.2 Simulation Environment	45
3.3 Initial testing	47
3.4 Shrinking step	49
3.5 Fixed position	50
3.6 Mathematical Analysis - Just wheels	52
3.5.1 Wheel with weight offset	53
3.5.2 Wheels and Curved Tail	56
3.5.3 Mathematics Results	59
3.5.4 Comparison with simulation	61
3.7 Changing Leg Angles	64
3.6.1 How does the robot fall on a step?	68
3.8 Size Analysis	71
3.7.1 Width Comparison	71
3.7.2 Tail Comparison	72
3.7.3 Optimal Size and Comparison	74
3.7.4 Remapped Data - Tail length comparisons	75
3.9 Summary	77
Chapter 4: Physical Cylindabot Design	78
Design process, CAD and final prototype of Cylindabot	78
4.1 Introduction	78
4.3 Background and Inspiration	79
4.3.1 Novelty	80
4.4 Initial designs	81
4.4.1 Version 1	81
4.4.2 Version 2	82
4.5 First Whег assembly	84
4.6 Overall design	86
4.6.1 How do the Whегs deploy/retract	88
4.7 Main CAD designs (3D printed parts)	89
4.7.1 Legs	89
4.7.2 Tube clamps	90
4.7.3 Drivetrain	91
4.7.4 Main body and Tail	94

4.8 Electronic components	97
4.9 Complete Robot	98
4.10 Wear and tear report	100
4.11 Summary	100
Chapter 5: Steps and Slopes	102
Cylindabot tested on slopes and steps in hardware and simulation	102
5.1 Introduction	102
5.2 Background	103
5.3 Linear Constraints for a Step	106
5.4 Simulation setup	107
5.2. Simulation Results—Single Steps	110
5.3. Simulation Results—Flat Slopes	112
5.5 Physical tests	113
5.5.1 Single Step Results	114
5.5.2 Simple Slopes Results	115
5.6 Discussion	116
5.7 Summary	118
Chapter 6: Grid Terrain	119
Cylindabot simulated traversing obstacle tiles defined by a grid of heights.	119
6.1 Introduction	119
6.1.1 Sinking grids	120
6.1.2 Attempt to smooth grids	121
6.2 Comparison - Random 5x5 Grid	121
6.3 Ten Standard Tiles	124
6.4 Checkerboard Map	126
6.5 Grid Density	128
6.6 Summary	131
Chapter 7: Ridges and Bridges Map	133
Six different obstacles show how Cylindabot can adapt to a map.	133
7.1 Introduction	133
7.2 Ridges & Bridge	134
7.3 Full grid analysis	136
8.3.1 Extra Obstacles - Gap and Long Slope	139
7.4 Map Setup	141
7.5 Map Results	142
7.6 Summary	145
Chapter 8: Conclusion	146
8.1 Thesis Summary	146
8.2 Future work	147

Chapter 4: Design	147
Chapter 5: Slopes and Steps	148
Chapter 6: Grids	149
Chapter 7: Maps	149
8.3 Final Discussion	150
Appendix	152
A.1 Double Parameter Terrain	152
A.1.1 Multi-step	152
A.1.2 Curved Step	154
A.1.3 Sloped Step	155
A.2 Frictional Coefficient	157
Cylindabot tested on slopes/steps with different frictions.	157
A.2.1 Introduction	157
A.2.2 Slope maths	158
A.2.3 Friction on Single Step	159
A.2.4 Friction on Heightfield Slopes	160
A.2.5 Friction on a cuboid slope	161
Video Links	165
Reference	166
List of Figures	172
List of Tables	181
Acknowledgments	183

Chapter 1: Introduction

It is in human nature to explore new places and for animals to move around their environment. Robotics has the promise to go where humans cannot or reduce the cost / risk of getting there.

There are a multitude of applications for robotic locomotion including planetary exploration [1], search & rescue [2], nuclear industry [3] and dangerous urban environments [4]. For all of these having a robust locomotion is essential to achieve the task and it is the premise of this thesis that adapting to the environment is key to accomplish this robustness. This involves evaluating the capabilities of the robot in different states and choosing which one to use.

Animals and humans are able to adapt to the obstacles they are trying to traverse. In the field of robotic locomotion there are two approaches to developing adaptability. Firstly, giving the robot several Degrees of Freedom (DoF) so a control system can change how each part of the robot pushes itself forward. In [5] low gear ratio motors are used that gives the robot more freedom to control its movement and acted as forerunner for the popular robot Boston Dynamics Spot [6]. These robots take inspiration from mammal quadrupeds, but other robots have taken inspiration from insects, such as [7] which is based on how a cockroach moves.

An alternative approach is to have intelligent morphology that adapts for example [8] which is again motivated by a cockroach. However, this robot has fewer active DoF and uses Whegs that allows it to achieve terrain traversal. A Whieg is a fusion between a wheel and legs and usually actuated in a continuous rotation. They can either be in fixed positions [8] or be able to be deployed [9]. The current state of the art hybrid robotics field focuses on the novelty of design. Whereas the legged or wheeled field has developed further into navigation of unknown environments, for hybrid robots more investigation into navigation decisions and fusing this with sensing data is the next major advancement.

In this thesis an adaptive robot called Cylindabot is presented. It has two wheels that can transform into three legs (Whegs). Its ability to traverse a range of terrain obstacles is tested both in simulation and hardware (a physical prototype). Knowing its capabilities is an important aspect for knowing potential applications and path planning in an environment. The thesis starts by looking at current theory and robotic platforms, before going on to the Cylindabot design and simulation methodology, moving on to show with results that adapting its mode of locomotion improves performance.

1.1 Locomotion

Put simply, locomotion is the ability to move from one place to another. The complexity comes from the interaction between the control strategy, the robot design and the environment the robot is in.

Robust and reliable locomotion is key for any form of autonomous robotics platform. Industrial robot arms are in a controlled scenario which is predictable as their base is fixed. On the other hand, in applications where the whole robot system moves the outcome can be quite unpredictable. So, if a robot attempts a terrain that it cannot traverse, it could fall or become unable to move. This is why a robot's locomotion capabilities need to be maximised.

Control of a robot covers two main aspects, movement while going over the terrain and deciding the route or approach to crossing the terrain. The complexity of the first of these is determined by the number of DoF. For legs this can be where to place its next foot as in [10]. Decisions are the other aspect; they might include creating a map of the environment, picking a route through the area, deciding whether to cross certain obstacles and finally what strategy to use for a region within the environment. For robots that only have one way to move, the path still needs to be optimised [11] but for a hybrid robot the strategy of movement needs to be considered [12].

A hybrid robot means that it has more than one style of movement that are significantly different from each other. In most cases this is defined by the parts of the robot design; wheels, legs or tracks [13]. This might either be by using the parts in conjunction with each other or separately. Categorising robots like this is useful for identifying novelty. An alternative is to define how the robot is interacting with the environment; roll, walk, jump, or fly. The advantage of this categorisation is that if new forms of propulsion for robots are invented then they are easier to fit into this framework. For clarity in this work these are referred to as modes of locomotion and Cyindabot is a hybrid robot as it can roll or walk.

Whether hybrid or not, the design of a robot has an effect on the terrain it can traverse. The interaction between the robot and the environment is defined in [14] as "robophysics". They concluded that physical prototypes need to be constructed to investigate this interaction. For hybrid robots especially, there is limited work understanding this complex relationship.

The important point is that if any of the controller, morphology or environment change then so can the outcome or success of a robot performing a task. In this thesis the primary focus is on the interaction between the robot morphology and various obstacles.

Another way of putting this is how the robot should be designed or adapt to traverse obstructions successfully.

1.2 Hypothesis

The hypothesis of this thesis set out a premise for the rest of the work:

“Adapting a ground robot's mode of locomotion shows measurable improvement for traversing a range of obstacles.”

To start with, “adapting” means that the robot is able to change for a given environment, in this case Cylindabot changes its morphology. A “ground robot” refers to an autonomous machine that moves by pushing against the surface it is on and excludes robots that fly or swim. The “mode of locomotion” means whether the robot rolls, walks or jumps. The capability of the robot being “measurably improved” not only means that there is a statistical significance but also that using the actual difference can be practically applied.

To support this hypothesis the Cylindabot’s performance is tested on a wide range of obstacles and while varying several qualities of the robot. This is an in depth investigation of the morphology/environment interaction. Work towards this hypothesis can be broken into three steps:

1. Measure: evaluate Cylindabot’s capabilities on a terrain in a given state
2. Compare: judge if certain configurations are optimal
3. Generalise: apply the optimal configuration

1.3 Chapter Outline

The entirety of the PhD could be summarised into three phases. The first phase is the Cylindabot design being iterated, initially in simulation, then on a physical robot. In the second phase, experiments are made in simulation and hardware. This allows the design to be verified and the reality gap between simulation and the actual robot to be measured. In the final phase, the design is further tested on a variety of obstacles and environments in simulation.

A brief outline of each chapter is explained and the main question each of these chapters seeks to answer is stated. The chapters of the thesis are summarised below and each conclude with a research question:

Chapter 2: Literature Review

This chapter investigates the current theories involved in locomotion and existing robotic platforms. It is split into morphology, control and environment. The chapter starts by describing the existing robotic fields. The robot morphologies are categorised by how they interact with the environment; whether they roll, walk or jump. The chapter continues with a comparison between these different platforms, looking at a range of possible capabilities. The chapter then summarises control and modelling techniques, including SLIP (Spring Loaded Inverted Pendulum), SLAM (Simultaneous Localisation and Mapping) and terramechanics as these are standard ways that the robot-environment interaction is understood. Then the chapter looks at how a robotic performance is measured by its environment with the desire to find a set of unified benchmarks that could be used to impartially compare robots' terrain capabilities. What can be learnt from existing hybrid and non-hybrid robots that could inform new design?

Chapter 3: Preliminary Results

The initial simulation work that laid the groundwork for the later chapters is covered here. Firstly the chapter covers the simulation setup and how it changed through experimentation, including the use of V-Rep simulator (later CoppeliaSim) both through its GUI interface and Python API. These early results are compared to a mathematical model which is derived for Cylindabot. Finally, the dimensions of Cylindabot are analysed through simulation to inform the physical design. How useful can simulation be before building a new robot morphology?

Chapter 4: Cylindabot Design

This chapter proposes the Cylindabot which is a novel Wheg robot that can transform between legs and wheels. This chapter explains the inspiration behind the design and then provides detail on the various iterations of design of the robot. The chapter continues to examine the overall design of the robot and each of the individual components, both 3D printed and electronic. Finally the chapter talks about the overall design, a report on wear to the robot due to experiment (predicting the longevity of the design when deployed) and future plans for improving the platform. How do physical constraints and theoretical ideals interact to give a prototype design?

Chapter 5: Slope and Step Obstacles

The core results from both simulation and hardware are presented in this chapter. The final design of Cylindabot was tested against slopes and steps to measure its capabilities. The magnitude of leg deployment and tail length were varied to analyse the obstacle

interaction. Measuring the robot's capabilities in different configurations, is there an optimal state for Cylindabot on these two obstacles?

Chapter 6: Grid Terrain

This chapter concerns Cylindabot's ability on terrains that are randomly generated. The chapter introduces terrain obstacles that are defined by a matrix of heights. This grid can then be applied to generate either steps, slope or plinth tile obstacles. Cylindabot is then tested against these tiles where each height is randomly generated for a five by five grid. The chapter concludes with investigating the effect of varying the density of the grid that defines the terrain tile. Can optimal configuration be applied to grid obstacles?

Chapter 7: Ridges and Bridges Map

In this chapter six new obstacles are used to give Cylindabot a varied challenge. Two different algorithms are used to test these obstacles. The second algorithm results in success rates that can then be used for path and adaptation planning. Cylindabot is tasked with crossing a map of these obstacles. Which is more successful for adapting Cylindabot's configuration or the route it takes?

Chapter 8: Conclusion

This chapter summarises the thesis as a whole and draws the ideas together. Future work on the Cylindabot platform and simulation is proposed. Finally a discussion about research trends is included.

1.4 Contributions

When it comes to hybrid robots, often the novelty of the morphology is the unique selling point. This could be the mechanism within the robot or the combination of the actuator parts. Hence, the design is a proof of concept and testing of the robots capabilities is often limited. With this work a more in depth analysis is undertaken.

1. Mathematical and simulation-based analysis of wheel-leg capabilities.

Kinematic and geometric analyses are used to predict how well Cylindabot will do on obstacles. A simulation model of Cylindabot and environments were generated to test the robot in a repeatable manner.

2. A mechatronic design with features and new capabilities lacking in other similar platforms.

Cylindabot is able to lock its tail and legs in different positions to allow reliable experimentation. The design uses 3D printing and electronic components that can be

reproduced. Other novel features include tube Whег axles, herringbone gearing and sprung legs.

3. Real-world verification of the robot functionality.

Hardware testing of the robot allows the design to substantiate future applications. Without these results, simulation data would be only theoretical.

4. Thorough testing of obstacle traversal capabilities in simulation and real world.

A wide variety of robot configurations and obstacles were investigated. This included comparing hardware and simulation results to investigate the reality gap.

5. Establishing the optimal modes of locomotion for different obstacle types.

Walking and rolling are the two modes applied by Cylindabot, using fixed leg or wide radius wheels respectively. Using experimental data, optimal configurations of the robot are found that vary due to obstacle type. This validates the demand for a hybrid robot.

6. Experiments on adaptation to different terrain and obstacle types.

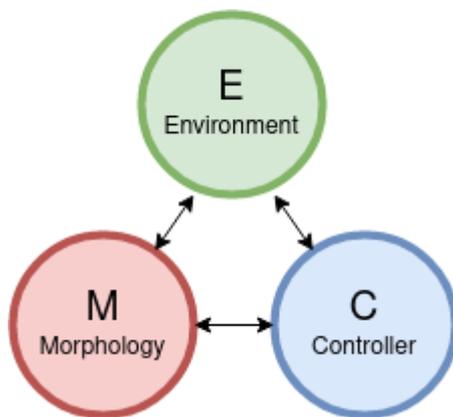
Cylindabot is able to change its configuration as it traverses a map of different obstacles. Changing its configuration is shown to be superior to choosing different routes. These obstacles were specifically designed to challenge the robot in range ways.

Chapter 2: Literature Review

Existing robotic designs and theory of locomotion.

2.1 Introduction

In this chapter we review theories that underpin locomotion of robotic systems. A robot's ability to move, specifically its capability to cross obstacles, determine/limit where it can



be deployed. Both in locomotion and general there are three spheres to be discussed: Morphology, Controller and Environment. The morphology means the physical design and components of a robot. The controller program or intelligence that determines what the morphology does. Finally the environment is what the robot interacts with and can be real or a simulation. This thesis is predominantly concerned with how the morphology and environment interacts, however all three will be discussed in this chapter.

All of these factors interact with each other, from type/size of wheels or legs to the system used to navigate. They combine together in a complex manner and generate a binary outcome: does the robot successfully traverse an environment. Although this thesis is primarily concerned with hybrid locomotion, singular locomotion robots are referred to here as well as whether it is possible to create benchmarks for robotic locomotion.

2.2 Morphology

To begin this chapter, a discussion on how different robots are classified by their morphology. This means that each type of robot can be discussed and analysed in turn. There are several different ways that a locomotion system could be categorised. For robots with a single form of locomotion they are often characterised by the deployment environment whereas hybrid robots are often defined by the components of the robot. A hybrid locomotion robot is one that takes advantage of more than one approach to moving. In [13] each single form of locomotion is defined by whether the robot has wheels, track or legs and a robot is a hybrid of these if it has more than one of them. Although this paper fails to mention Whegs as a compromise between legs and wheels, it does give a full comparison between these hybrid and non-hybrid designs.

In [15] robots for nuclear environments are categorised by both components and which power station they were deployed in. In [16] the types of components are taken further by

defining how these parts are connected together; legs with wheels at the end, robots with separate wheels/legs, legged-tracks and leg-wheels.

The robots reviewed in [14], were grouped by the type of terrain that they were meant to traverse, as the paper focused on the robot and physics interaction and the material it moves on is key to this. The paper concluded that the robo-physics needs to be investigated in the real world so that new interactions could be discovered. Robo-physics is a term introduced by the paper and describes the new physics generated by a robot.

In [17] the authors define five types of locomotion components; wheels, tracks, legs, body articulation and non-contact examples are given but precise definitions are not. The paper goes on to focus only on the first two stating, "legged motion involves considerably more complex control algorithms ". It finished with selling the advantages of the Elastic Loop Mobility System (ELMS) as being superior to rocker-bogie wheel rovers and traditional tracked systems.

2.2.1 Classification by Action

The issue with this classification by component is that it could never be complete for all possible future designs. If a new part was invented then a new class would need to be generated and hence no longer be complete.

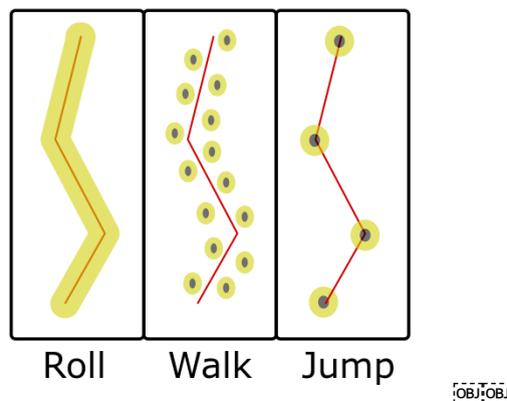


Figure 1. Diagram showing the different forms of locomotion

An improved system is to use a complete categorisation into three types of locomotion; rolling, walking and jumping. This only applies to robots moving across land where the contact with the ground provides both thrust and support of weight. It relies on how continuous the contact with the floor is and the footprint path. For rolling robots there is constant contact with the floor and the wheels have to map out an unbroken line. Walking robots are still touching the ground at all times but the path of the contact footprint has gaps between contact points. Whereas with jumping there are discrete intervals in floor contact both in time of contact and the path of the foot print. The

difference between the last two is that there are times when no part is touching the ground for jumping. A design could be hybrid either transforming between different types of locomotion or acting like more than one of them at once. An example of this would be crawling robots where the actuation is done by limbs (walk) but a part of the robot is dragged along (roll), whereas a snake sliding would be under rolling as no part of it leaves contact with the floor.



The same categorisation was used in [18]- jumping, walking and rotating (wheeled)- which is used to explain the versatility of its design. Jumping was not implemented with the design as “there are no actuators” [18] that meet the requirements and the difficulty making the design tough enough for jumping. They give one gait possibility for walking and four different configurations for rolling (one of which was implemented). In walking they kept the weight within the area of the foot so it did not need to be dynamically balanced. [19] defines “a rolling robot as one that rolls on its entire outer surface” and does not apply torque to the ground which is different to how it



is referred to in this thesis. This style of rolling is achieved by rotating/moving internal weight within a round outer surface of the robot. It goes on to divide natural rolling as either passive or active and robotic movement into seven categories. Spherical and cylinder designs are promoted by this paper to justify the author's later design, Jollibot [20], with the premise that these designs can function after landing in any orientation.

Table 1. Categorisation by action

Mode	Floor Contact (is it touching the ground)	Path Footprint (of each leg)
Roll	Continuous	Continuous
Walk	Continuous	Discontinuous
Jump	Discontinuous	Discontinuous

2.2.2 Robotic Platforms

The shape and component of a physical robot has a radical effect on its capabilities. In this section we look at the hardware designs of certain robots. This could never be an up to date complete list of all robot morphologies but is more of a snapshot of the current designs with a focus on hybrid designs.

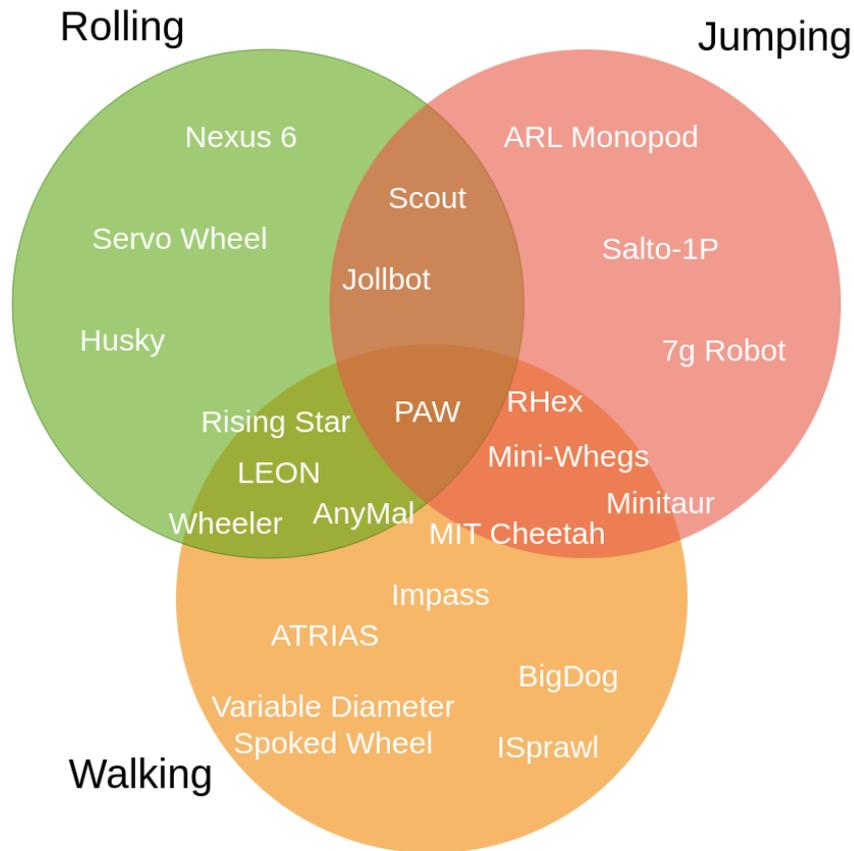


Figure 2. Venn diagram of the morphologies discussed here and what type of locomotion they use

2.2.3 Rolling Robots

Re-inventing the wheel is what will be discussed here, that being said wheels are still the most dominant mechanical transport mechanism used both in robotics and general transport.

A key feature of rolling robots is whether they will be able to grip a given terrain to traverse it. This can be divided into two key aspects; deforming to move over obstacles with a form of suspension and the actual interaction between the robot's wheel/track and the surface. If no deformation is performed, the robot will experience more impedance to its movement caused by every bump in terrain. It seems that the development of grip models are linked to the space program and the robots involved in it.

If we talk about where reliability is paramount, the rover missions to Mars become an obvious choice, all of which used rolling wheels as their form of locomotion. They have another thing in common; for the Sojourner, Spirit, Opportunity and Curiosity missions, the rovers all used a Rocker-bogie suspension system. The Nexus 6 is an example of this platform in [21]. This system allows smooth contact with the ground over a variety of feature shapes.

The Husky robot [22] is currently synonymous with field robots and has four wheels that do not turn making it a slip steering quad. It has been used for a range of applications including debris clearing [23], off-sea oil rigs [24] , agriculture [25] and the DARPA subterranean challenge [26]. The Husky is a four wheeled design that has a high payload capacity and can be fitted with a UR5 robot arm [27] or LIDAR systems (laser imaging, detection, and ranging).



Figure 3. Husky robot from Clear Path [22]

Rolling robots can still adapt to obstacles. In [28] a wheel is composed of servos in a loop. This mechanism could change shape to conform to the terrain it was crossing. Changing the shape gait, the robot was able to traverse flat ground, slopes, steps and round ridges. A mathematical model was used to develop these gaits and it was shown that the adaptation helped it to traverse more challenging obstacles.

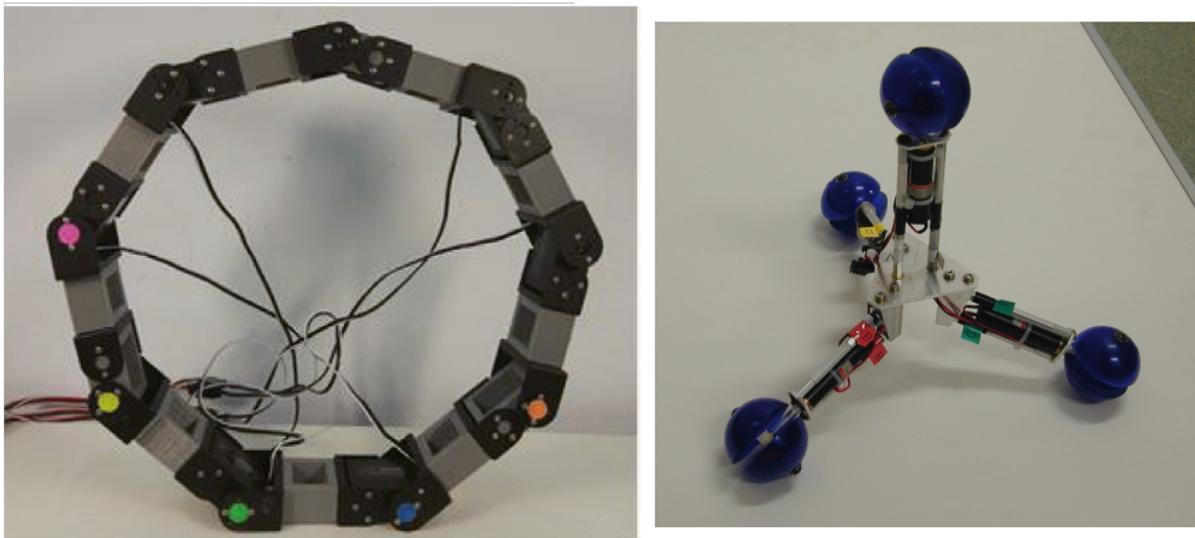


Figure 4. Servo wheel [28] and Tetrahedron robot [29] with “Omni-balls”

An omnidirectional robot uses wheels that are able to slide in the direction of the axis of rotation. This means the wheel can apply without limit to the other direction of

movement. Omnidirectional robots are usually applied to flat ground application, however an “Omni-ball” is developed in [30]. The wheel design was also used into a tetrahedron shaped robot [29] and later able to transform between a tetrahedron and a flat configuration [31]. A downside to the “Omni-ball” is that it can only roll in the orthogonal direction when at certain angles which limits its omnidirectional movement.

2.2.4 Walking Robots

One of the key factors which determine a robots complexity is its number of Degrees of Freedom (DoF). A small number of DoF means the robot's movement is limited. However as more DoF are introduced the weight, number of points and complexity of the controller all increase. In [32] more legs are credited with a faster design. This was based around the travel time of the legs to move forward for the next step and mentions that the control of all these legs would be complex. One thing it fails to consider is the added weight of all these actuators (and power) would make the robot considerably more heavy. Possibly to the point it would no longer be able to lift its own weight, which is why the paper states that most legged robots have 6 or less legs (with 3 DoF per leg).

An interesting insight within this paper is the distinction made between insect and mammal style leg configuration. An insect style has the knee joint perpendicular to the direction of travel and produces thrust by a turn at the hip joint. A mammal style leg has the knee in the direction of travel and propels itself by either having its weight forward to fall or pushing when the leg is behind the hip. The efficiency of legged designs is then discussed with the main loss of power being put on negative work. This is when the actuators in a leg are wasting energy by creating torque in opposite directions. The authors suggest gravitational decoupling with either orthogonal mechanisms [33] or some sort of pantograph [10]. A pantograph is a parallel bar system that allows motion to be translated to a different place.

Using continuous rotation to create a walking gait in the R-Hex [34] allows the designer to take advantage of how motors were designed to be used. The compliance on the legs are important to how the RHex handles terrain and, even with a simple controller, it is able to handle an array of different terrain. Several different models of the RHex have been released and over a dozen papers published.



Figure 5. a) Original RHex [35], b) RHex 0.9

A unique design is the Mochibot [36] which uses 32 linearly extending legs that allows it to roll like a sphere on legs. An extension longer than a normal linear actuator was required, this was achieved by a slide rail mechanism that was able to achieve an extension of 252% of an original extension length. This new style of actuators are quite slow and were only tested on a flat surface.

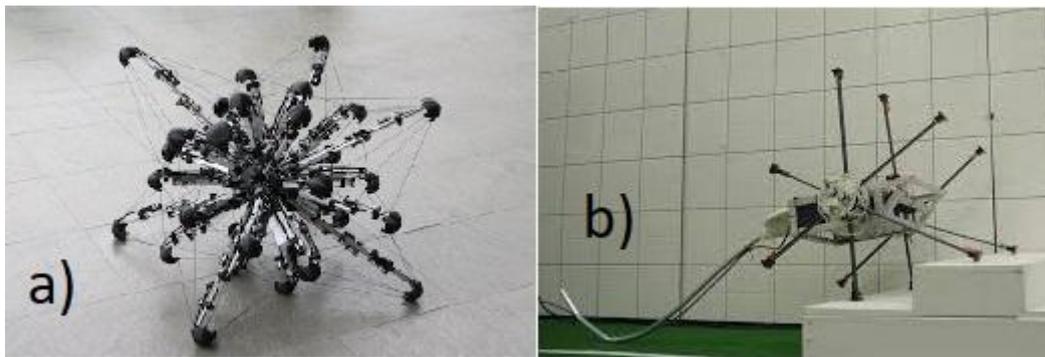


Figure 6. a) Mochibot [37] b) Impass [36]

The IMPASS [37] robot combines the features of linearly extending rods in the Mochibot and the turn of a central hub from the Mini-Whegs. It is able to perform even longer extensions than the Mochibot by using carbon fibre rods that contain a chain that can be driven. The robot is able to traverse obstacles taller than itself without jumping. This again comes down to a novel actuator mechanism.

The variable diameter robot [2] had a spoked wheel design which was able to vary the length of its legs and tail length. This variability was extensively tested against a single step and a geometric theory was developed for its capability. The six legs deployed using a pantograph mechanism and were described as wheels without spokes. Another example of a rimless wheel is the road runner [38] which had eight legs that were sprung to absorb impacts.

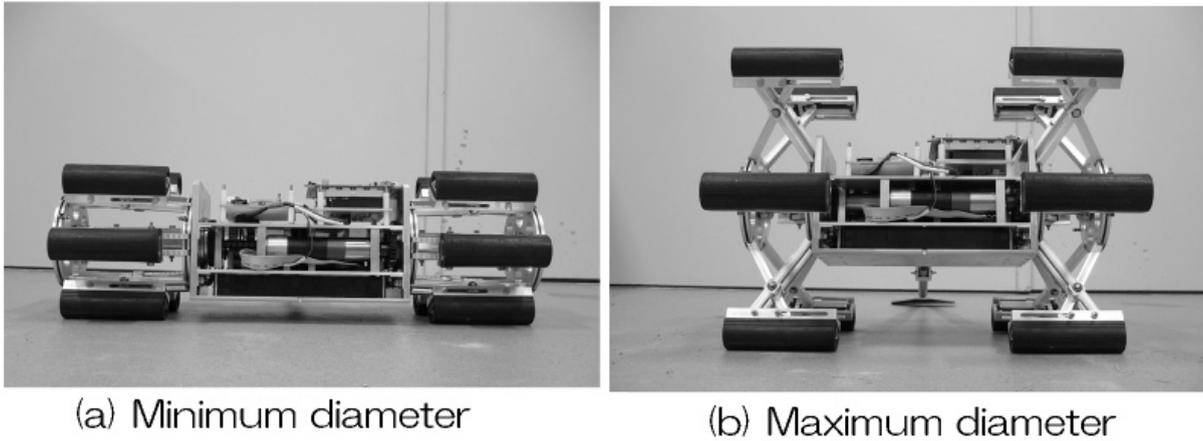


Figure 7. Variable diameter spoked wheel robot [2]

The adaptation of leg length is not limited to Wheg style robots. The quadruped in [39], [40] is able to adjust its leg length. The issue with the adaptation of this leg length was that a new controller was required and evolution was used to do this. It was shown to be feasible ([41]), however the improvement in obstacle traversal was not shown.

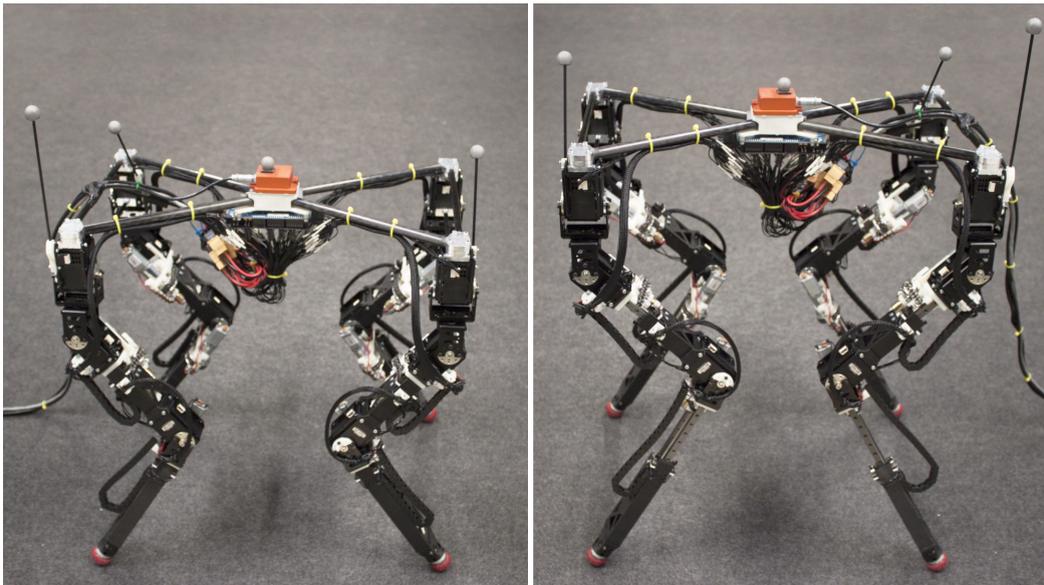


Figure 8. Leg adapting quadruped [39]. Left: leg retracted, right: legs extended.

A new feature of legged locomotion is having deformation within the actuation of the limb. This is taken to one end of the spectrum with the Direct Drive (DD) robot family described in [42]. These DD motors have several advantages layed out in the paper but also mean the motors are working far from optimal conditions. The case for no or low gear ratio is made well and the limitations presented.

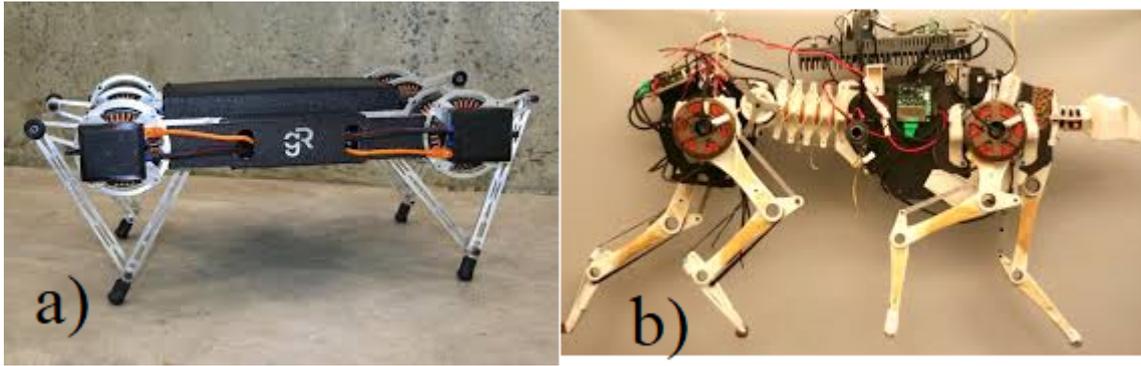


Figure 9. a) Minitaur [42] b) MIT Cheetah [5]

The MIT Cheetah [5] was a walking robot specifically designed to reduce the energy required to move. One part of achieving this was creating large radius motors with a low gear ratio (planetary gear). These motors were mounted on the hip of the robot and are able to regenerate energy when back turned. It is an innovative design that achieves its objectives well. Further to it making the motor more efficient, it also allowed greater control of the leg it actuates. This style back drivable motor can be seen in the Anymal [43] and Boston Dynamics Spot [6]. These type quadruped robots are reviewed in [44] and which gait are applicable to different terrains, the problem being that the gait comparison is qualitative rather than based on empirical quantitative data.

One of the heavy weights of the walking robots is the BigDog [45], which is one of the few of Boston Dynamics designs to be published as academic work. At the time, it was the culmination of Raibert work on dynamic locomotion that will be discussed later. It is the largest robot being analysed here and has around 50 sensors. Even with this number of sensors, it is not clear to what extent it can measure the world around it. It is also given navigation instructions from a human operator.

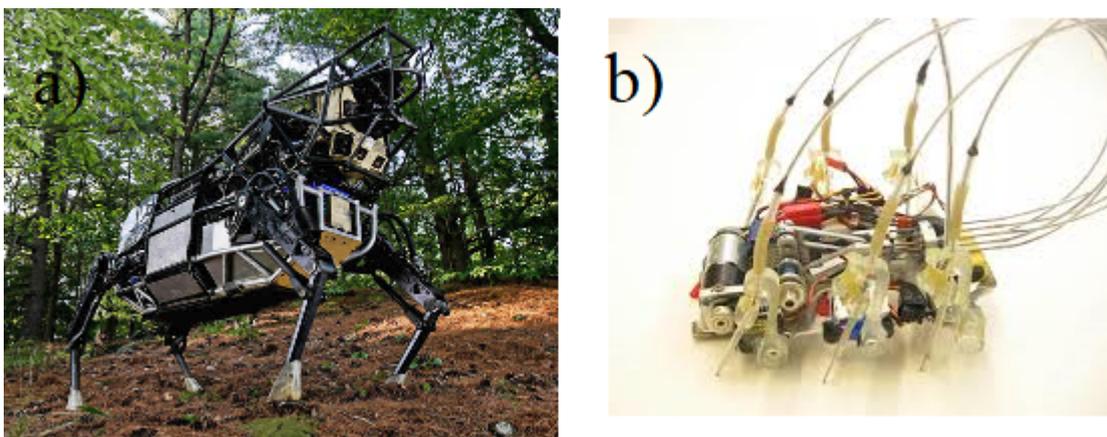


Figure 10. a) BigDog [45] b) ISprawl [46]

At the opposite end of the scale in terms of complexity and size is the ISprawl robot [46] which takes the pneumatic design of the original Sprawl robot and makes it electric. It has a single motor that powers all six linear moving legs with a crank system. Each leg has

flexible compliance, can deform at the hip and is moved by a push-pull cable. There is a good analysis of its capabilities on a flat surface and a comparison to the older Sprawl designs.

2.2.5 Jumping

Jumping a robot requires it to be able to have a high power output and be able to survive the impact of landing. The advantages are that entire sections of terrain can be bypassed which might be impossible to get over normally. There are two main design features that are used to help robots to jump; energy storage that allows more power at the point of take off and compliance in the physical leg or the actuators, which mean some of the energy from landing is sprung into the next jump. The end result of this is a power to weight ratio that determines how high a robot can jump.

The Jollbot [20] is a jumping robot based around a sphere to mimic the movement of tumbleweeds. The metal bars that make up its shell are compressed by motors in the centre then released to allow it to jump. Once it has jumped, its shape allows it to continue to roll passively under its own momentum. The other actuators control weight ballast so that the angle of jump may be controlled. It was designed from a set of biological inspirations that were then converted to requirements which is a unique way of prototyping a robot. The design is robust to the collision from jumping, though the exact precision of jumping would be difficult to control.

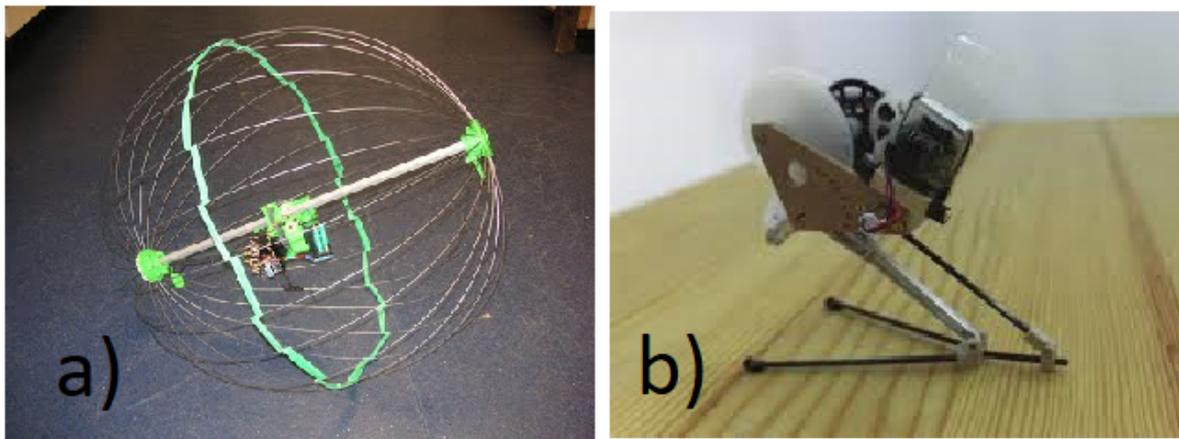


Figure 11. a) Jollbot [20] b) 7g Jumping robot [47]

The 7g Jumping robot [47] was designed to be as light as possible to maximise its jump height, hence its mass is given as part of its name. This does come with the limitation that it does not have much on it apart from the jumping mechanism. A good comparison with other jumping robots of the time is made but this design does have limited autonomy compared to other designs listed.

The highest jumping robot was set using a hybrid spring mechanism [48] which combines carbon fibre rod under compression and latex rubber under tension. This allowed the force to the total compression graph to be quite flat and the total energy stored much greater. With its lightweight design, it was able to jump to a height of 32 metres.

2.2.6 Hybrid

A hybrid robot has more than one way in which it can move and there are a plethora of designs each with different features. The Scout robot [49] is a miniature two wheeled robot that is designed for surveillance tasks. It has two wheels with a cog profile to achieve grip and uses a sprung sheet of metal to be able to jump. One end of it is wound in using a cord around the body of the robot and then is released to jump. This jump is not controlled as it will not matter which way the robot will land as its cylindrical design means it does not have a correct orientation. The Scout robot is designed to survive these impacts as well as being fired from a larger deployment robot. (Not to be confused with the Scout II which is a walking robot mentioned later in this work). Another robot that used jumping to climb steps the wheels were unable to was [50] which had four wheels and moved internal weight to provide the jump. There were two later developments to the scout platform. The first added various locomotion improvement and sensing capabilities in [51] to the original design, including a set of spokes to allow it to walk. Obstacle capabilities were never tested and the tail was not extended; it is suspected that the short tail would severely limit the use of these spoke legs. The second was a full design into a MegaScout, [52] which increased the size and various capabilities of the platform. Although the MegaScout removed the robot's ability to jump there was a version that had legs that encased the body. The obstacle capabilities of all scout designs are not detailed which is often a problem with hybrid robot documentation.

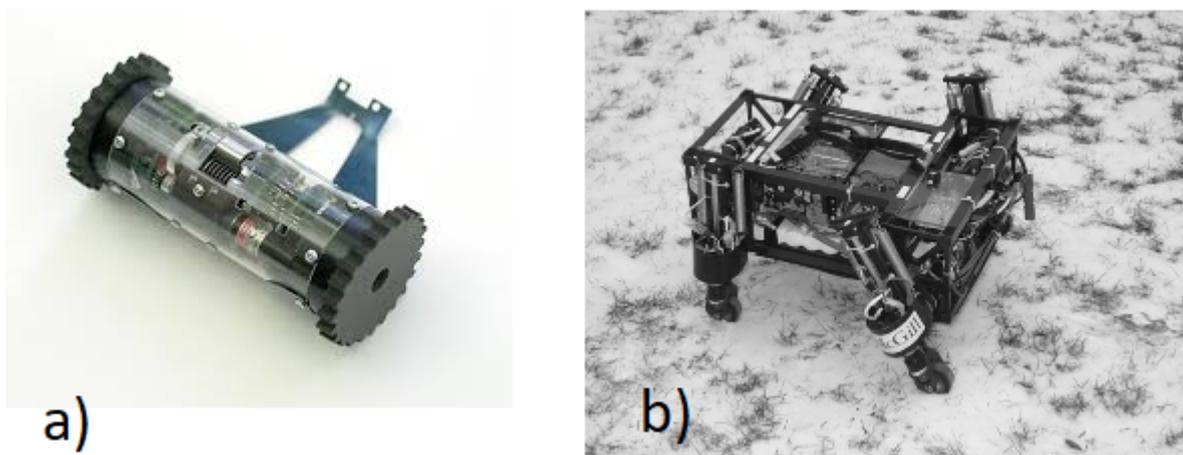


Figure 12. a) Scout robot [49], b) PAW robot [53]

The PAW robot [53] mounts wheels on the end of legs to provide the adaptability of walking with the efficiency of rolling. It has a similar frame to the Scout II and is said to be designed to handle simple, complex and highly complex environments. However, in this paper, the leg action is used to supplement braking and turning on a flat surface. In a later paper the robot negotiates a single step [54]. Although not documented within papers the PAW robot is the only platform that is able to walk, roll and jump in videos at least.

The SAPPHYR design [55] is a pneumatically powered hybrid robot where legs at the front pull forward. It was designed as a test platform for developing leg control, as the wheels allowed a stable platform for experimentation. A similar design was the Wheeleg robot [33] which was created to be like a cart. The key difference between these two designs is the orientation of the joints used to actuate them. The SAPPHYR has a mammal style configuration and the Wheeleg robot has an insect style arrangement. The Wheeleg robot was tested on a diverse set of terrain but concludes that this design takes the drawbacks of both wheels and legs. It is important with a hybrid system that considerations on whether the separate systems hamper each other's function or work well together.

Closer to the design of a wheel is a Wheg [56] which is a wheel with deliberate gaps put in to allow it to act as a given number of legs, similar to the Isle of Man flag. It supposedly has the advantages of both legs and wheels. An example hybrid design using them is the Mini-Whegs robot; it is a small manoeuvrable robot that is also able to jump, which was based on bigger earlier designs. The Whegs move in a fixed pattern and the robot steers like a car. It is meant to emulate the movement of a cockroach.



Figure 13. a) Mini-Whegs [57], b) Rising Star [58]

The Rising Star [58] robot takes the Whegs design in a new direction by adding a mechanism for manipulating where a line of Whegs are deployed. The Rising Star has two extra DoF, the first that is able to change the roll angle of Whegs, which is referred to as sprawl angle. The second is able to move the centre of the robot forward and

backwards using a parallel bar mechanism. This extra movement allows more versatile movement; the robot is able climb a vertical chimney, handle a step and not get stuck in soft ground. It also means that a new type of locomotion is possible where the Whegs do not rotate. Instead the robot moves like a turtle with the entire line of Whegs lifted over the body and then the body is pulled over the line of Whegs. It is also able to be fitted with normal wheels to reduce its cost of transport. The detailed force analysis of the robots joints are given and it is tested on a variety of surfaces. However more detail is needed about these tests; without this data it is hard to compare Rising Star to other platforms.

There are two similar robots that are able to transform from a quad wheeled mode to using legs. They both split the wheels in two, however how they are then used is different. The TurboQuad [59] moves the semicircles of the split wheels perpendicular to the axis of rotation to form a two legged Wheg.

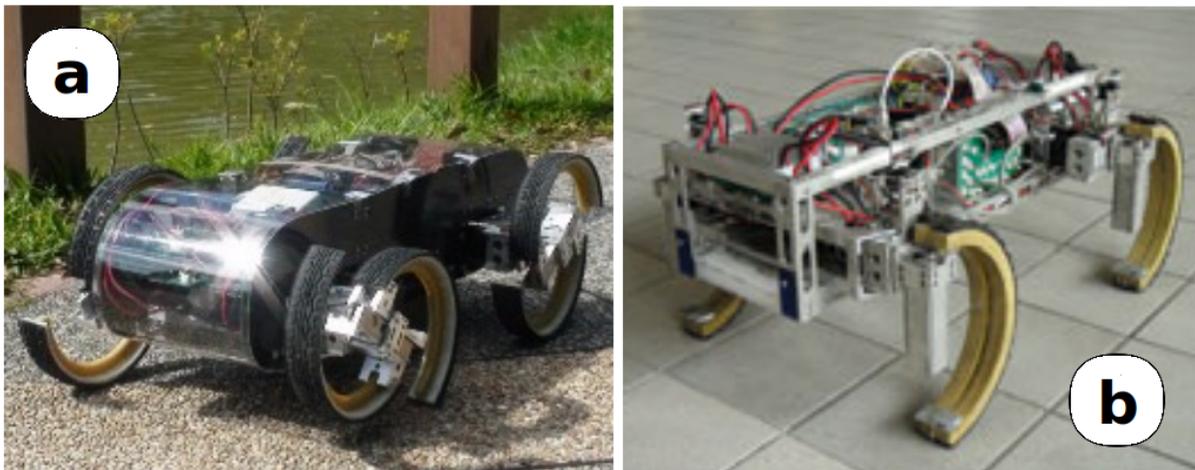


Figure 14. a) TurboQuad [59], b) Quattroped [60]

On the other hand the Quattroped [60] transforms by having the wheels hinge over in half to make a single leg similar to those from X-RHex [61]. The Quattroped can move by turning these legs continuously like Wheg or using the three degrees of freedom each leg has to walk like a quadruped animal would. Both TurboQuad and Quattroped capabilities are well analysed to step climbing and movement efficiency however their ability on a slope is not mentioned.

Large radius wheels are advocated in [62] where the robot is inspired by armadillo. It used a slip ring to transfer data power to the leg servos. [62] says that large radius wheels that transform into legs should be optimal for traversing terrain however does not present any experimentation to validate this, mainly focusing on how the wheel/legs can manipulate objects. Further variations in wheel/leg transformation are proposed in [63] to take this design further. Again, the proposed advantages of hybrid locomotion are stated but not proven to be true.

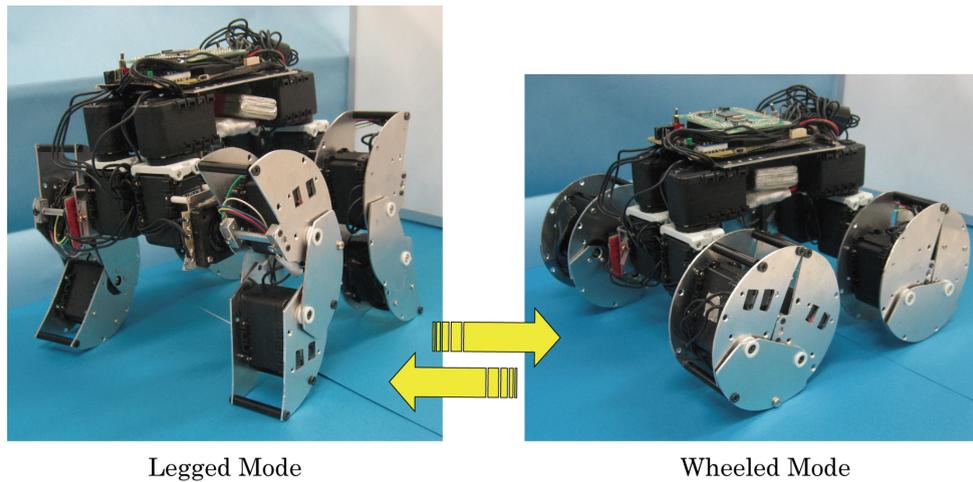


Figure 15. Armadillo inspired robot [62]

There are a few hybrid robots similar to the Scout [49] that have only two wheels but transform in different ways. This reduces the number of actuators the robot needs. These robots reshape their round wheels into a Whег that is used to improve its obstacle traversal capabilities. The first T-shaped Whег robot [64] has three legs that rotate from a position parallel to the axis of rotation. The T at the end of each leg is designed to catch on obstacles. A full grid approach is used to test the robot’s ability on a step in simulation similar to the results in Chapter 7 and is one of the few examples of hybrid robots being simulated.

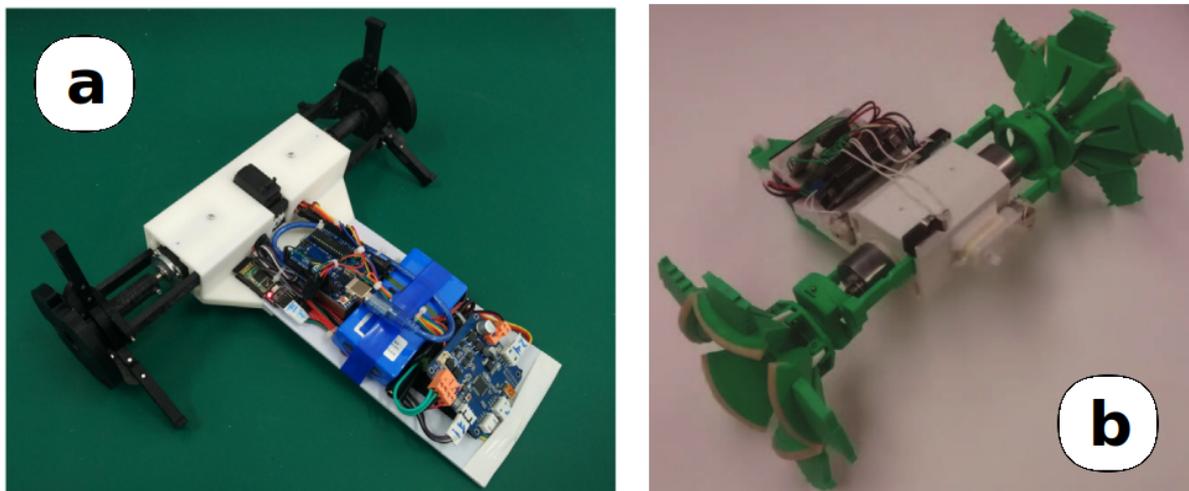


Figure 16. a) T-shaped Whег robot [64], b) Passive Wheel robot [65]

The passive wheel robot [65] has five jagged plastic fins to act as legs. Rubber has been added to the outside of the wheel to provide improved grip. The robot was tested against wooden obstacles, sand and a floor of leaves. In each case, the robot was able to move when using the legs and was unable to move with wheels. There are only images of these terrains, rather than any details, so replication of these tests would be difficult. Both the T-shaped and passive wheel robots use a separate motor that deploy/retract the legs and only have one configuration once deployed. They both use rack and pinion systems to have a motor’s rotation translated to a linear slide.

It is possible to reduce the number of actuators even further by having the Whег deployment system be passive. The Wheel Transformer robot [66] uses one of the feet of its leg as a trigger to release the Whег deployment. The situation used was that the robot is stuck with the wheel rotating against a step. The trigger foot would catch the corner of the step and deploy the legs. The paper shows that wheeled mode is more effective for speed on the flat and that legs are able to climb over a taller step. This means that both configurations have been justified.

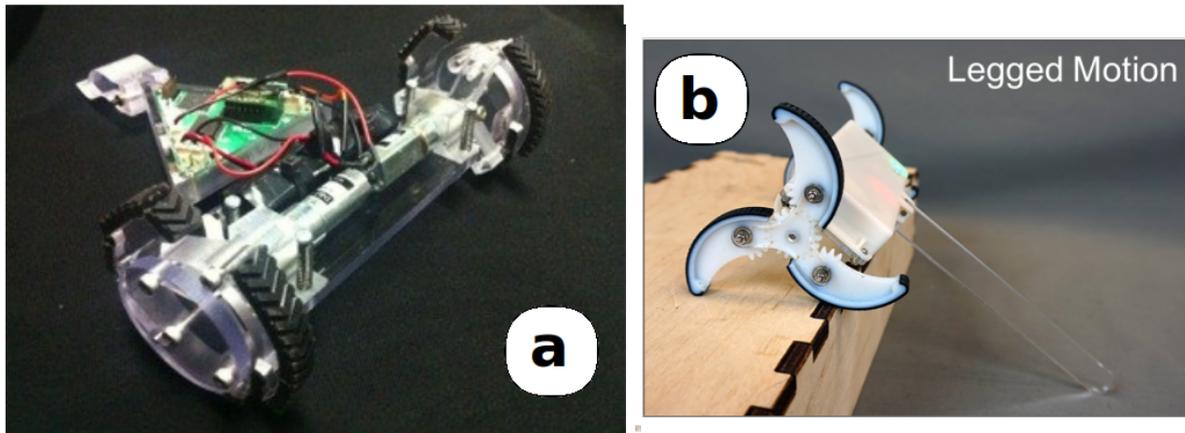


Figure 17. a) Wheel Transformer [66], b) Wheeler robot [9]

The wheeler robot [9] is a small minimally actuated design that can transform from legs to wheels. Its unique selling point is the leg deployment which uses the drive from a single motor to both drive the Whег and deploy/retract the legs of the Whег. This is achieved by a cog in the centre of the Whег which is able to rotate the entire Whег assembly or move the legs. The direction of drive of the Whегs also determines whether the legged or wheeled modes are being used. The robot flips upside down when using different modes due to its symmetric design. The robot is tested on a range of surfaces including steps, slopes, asphalt, carpet and grass. In each case specific details were included making it first repeatable and one of the few hybrid platforms tested on both slopes and steps.

The majority of development in Whег design had been in its deployment method. Unpowered deployment reduces the number of actuators. In [67] powered deployment is used to allow the wheel to be omni-directional. This allows for controlled movement in any direction as well as good step climbing ability which was previously achieved by legged locomotion. The advantage of these “omni-Whегs” is the energy efficiency on flat ground and their effectiveness on a slope were not tested.

2.2.7 Comparison

The problem with this large variety of morphologies is that to know which is the best for a specific application is difficult. This section aims to compare these robots and suggest

some new ways in which they could be analysed. Many of the papers for their robots, however, do not fully document their capabilities.

Table 2. Morphology Comparison

Name	Mode	DoF	Kg	B	S	C	N	F	T	Notes	Year
7g Jumping	J	1	0.007	✓				✓	2	Very light jumping mechanism	2008
Salto-IP	J	5	0.103	✓	✓			✓	3	Highly accurate jumps using rotors	2018
Mini-Whegs	W, J	3	0.146	✓				✓	2	Fixed gait, based on cockroach	2003
Scout	R, J	3	0.2	✓	✓			✓	3	Symmetric design to survive falls	2000
ISprawl	W	1	0.3	✓				✓	2	Flexible piston hexapod with high gait frequency	2006
Rising Star	R, W	4	0.308	✓		✓		✓	3	Whegs with control of sprawl angle	2018
Jollbot	R, J	2	0.465	✓				✓	2	Jumping tumbleweed	2007
Rhex	W, J	8	6	✓	✓	✓	✓	✓	5	Hexapod with continually rotating flexible legs	2001
Minitaur	W, J	8	5	✓		✓		✓	3	Directly driven wide radius motors	2016
Nexus 6	R	10	5.8	✓	✓	✓	✓	✓	5	Rocker-Bogie suspension	2002
Twin-Frame	R, W	6	10					✓	1	Jointed arm connecting two wheels/feet	2003
Mochibot	W	16	10.5	✓			✓	✓	3	Sphere of spikes that linearly extend	2018
ARL Mono	J	2	15						0	First electrically actuated jumper	1997
PAW	R, W, J	8	15.7	✓		✓		✓	3	Wheels on the end of legs	2006
Scout II	W	24	4	✓		✓		✓	3	Sprung leg bounding robot	2005
Wheleg	R/W	8	25			✓		✓	2	Insect orientated legs and powered wheels	2003
SAPPHYR	R, W	4	30*					✓	1	Mammal orientated legs pulling passive wheels	2002
MIT Cheetah	W, J	12	33		✓	✓		✓	2	Planetary geared wide radius motors	2015

Name	Mode	DoF	Kg	B	S	C	N	F	T	Notes	Year
IMPASS	W	9	50					✓	1	Actuated spoke design using chains	2018
ATRIAS	W	6	63						0	Pantograph biped so motors at hip	2018
Big Dog	W	16	109	✓	✓	✓		✓	4	Hydraulic mini-tank, heavy and lots of sensors	2008

In Table 2 a range of different platforms are listed in order of their weight. The weight of some of these robots had to be estimated by the general size which is indicated by the star. An important qualifier of the complexity of the locomotion systems is the number of Degree of Freedom (DoF), in this case only at active DoF are counted. The level of autonomy is compared by whether the robot has certain hardware or is able to perform important tasks.

- B = Battery; Does the robot contain the ability to power itself
- S = Sensing; The robot has sensors to observe the world around it
- C = Computing; Is the robot able to compute what it needs to do
- N = Navigation; The robot actually makes the decisions on where it tries to go
- F = Free Standing; Does the robot require any exterior balancing
- T = Total of autonomy points awarded

If these are present they are given a value of one then summed together to give an autonomy score that is given out of five. High scores are preferable and mean a robot has many of the requirements to act autonomously. This gives a general sense of whether the robot would be able to function on its own. The rarest of these checks is the navigation which is often in a moving system one of the hardest to implement and has many other checks as prerequisites.

2.3 Control

In this section the various control systems and their locomotion that they accomplish will be discussed. For legs a key distinction made here is between statically stable systems and dynamically balanced ones. The first being if the actuators locked up or no action was performed the robot would not fall over. Whereas, the second must constantly be correcting its balance angle to stay upright or has airborne points in its movement. Finally wheeled control systems are discussed.

One of the main factors that make a control system complex is the number of Degrees of Freedom (DoF). Most robots have at least 2 DoF however many have much more

(Table 2). Each of these DoF needs to be programmed/tuned to react in the right way. This is true for all modes of locomotion and for Cylindabot the number of DoF was minimised. The reasons for reducing DoF for Cylindabot included decreased weight, impact resistance and to make the control system simpler.

2.3.1 Unstable Walking and Jumping

One of the founders of modern jumping and walking robots was Raibert [68] who developed the Spring-Loaded Inverted Pendulum (SLIP) model which is still used as a yardstick for comparing control designs. SLIP is based around modelling a leg mathematically as a simple spring and a point mass. This simplifies the dynamics for a controller to be designed. Multiple legs could be controlled using this single controller design like a Gatling gun with each leg being set to jump at different points in time. He went on to show how the control of a single leg can be applied to multiple limbs, using virtual legs and three different quadruped gaits [69]. The idea behind virtual legs is that if pairs of legs perform the same action at the same time the robot acts like it has less legs.

One of the main problems with Raibert's physical designs was that they were pneumatic and hydraulic which, at the time, limited its autonomy. These actuators needed to be externally powered by a pump and therefore can only function via tether. This tether would mean that it would be unable to function outside on its own. The ARL Monopod [70] took the SLIP model and used springs plus electrical actuators which removed the need for a pump. The biggest jump towards realising Raibert's control scheme was large radius motors used in [5]. This breakthrough allowed robots like Boston Dynamics Spot [6] and the ANYmal [43] to come into existence. The new actuator was able to have controllable spring and damping. Adding these mechanical dimensions to the actuation means the robot can land softly and control its movement.

One of the motivations behind jumping robots is to be able to choose the placement of landing. In [71], the robot Salto was able to achieve this and it was a single legged hopping robot that could jump 0.90 m high. This jumping robot used a combination of drone rotors and a balancing tail to be able to precisely plan its jumps. It was able to land on different surfaces and perform planned routes, jumping from the floor to a chair then onto a desk. In a more recent video it has been shown jumping outside of the test arena without the tracking system used there but instead under remote control. The main issue with Salto is that it uses drone propellers for balance and therefore would struggle to be scalable to larger robots or in certain space applications where there is limited atmosphere.

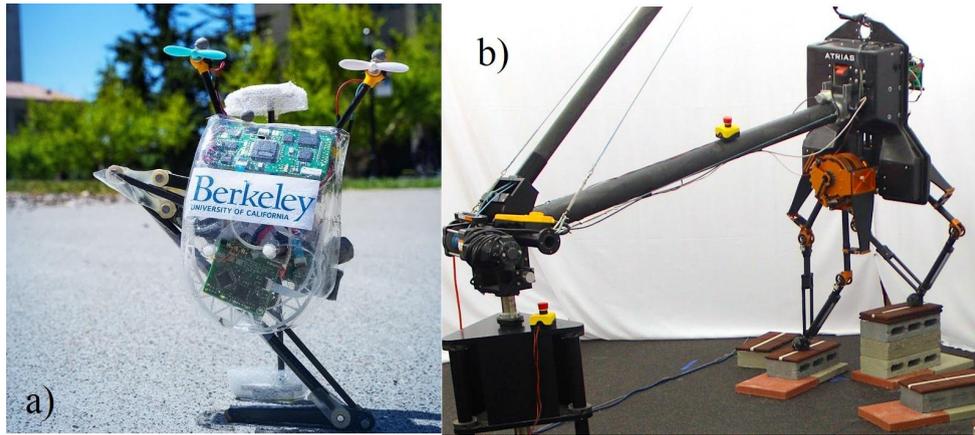


Figure 18. a) Salto-1P [71], b) ATRIAS [10]

Using a completely different method in [10] foot placement was also achieved for a two legged walking robot. It was implemented on the ATRIAS robot platform which uses a pantograph system to keep the actuator at the hip of the robot. Although the robot was mounted on a boom which means it was not balancing in three dimensions. The design used a two-step gait optimisation to be able to place its feet on discrete stepping stones. The gaits were pre-programmed and then an appropriate one was used at any given time taking into account the next two steps.

2.3.2 Stable Walking

An interesting example of controlled stable movement was Mochibot [36] which managed to use its various legs to create a continuous path in any direction. The author said that other similar polyhedron robots fall between different polygon faces in discrete falls. What makes their design unique is that it is able to deform and extend this polygon of balance in any direction by using an imaginary surface. In such a way it can move along any path that could be devised. It does this by balancing on three legs then retracts/extends the legs so that it falls over placing a new leg on the floor.

Zero Moment Point (ZMP) describes a way a legged robot tries to control its movement so there is no rotational moment that would topple the robot. ZMP is common in walking systems with six legs as they are able to perform an alternating tripod gait [34]. This is where three legs stay on the ground while the others are moved forward. The advantages of this is that the robot is stable and that the controller is easy to design. The robot is stable as long as its centre of mass stays above the triangle made by the three feet. The movement path is defined by where the second triangle of three legs is placed, can often be omnidirectional and is not time dependent like unstable walking control.

Stable walking allows controlled movement and has been applied to a range of terrains. The limitations are that it tends to be slow and obstacle traversal is limited to the reach of

the legs. Whereas jumping (including running), although more risky in an unknown environment, is faster and able to traverse obstacles taller than the robot.

2.3.3 Rolling Control

Rolling robots often have less degrees of freedom (DoF) than other modes of locomotion and often move in a statically stable fashion. There are still a range of ways that robots can roll. In [72] four different designs are modelled and analysed (differential, 2x skid steer and omni-directional). Differential drive was chosen for Cylindabot for its simplicity and turn radius. Below is a brief description of four drive styles:

- Differential drive; when the speeds of the two wheels are varied to turn
- Skid steer; similar to differential but multiple wheel or track are used so they have a skid as they turn
- Car style; separate DoF that angles wheels to allow turn
- Omni-directional; Swedish wheels used to so the robot can move in any direction

One of the main advantages of rolling platforms is the stable frame for sensors or manipulators to operate from. SLAM was achieved on wheeled systems much earlier than walking ones and SLAM has yet to be done on a purely jumping platform. Having mapping and localisation is important for controlling a robots movement as, without it, planning a movement or getting back to a start position needs exterior assistance.

Balance control is not limited to walking or jumping robots though. Although Cylindabot has a tail to help balance itself, other robots need to balance themselves. Balancing two wheeled robots are part of a subclass of control problems known as inverted pendulums . This journal, [73], reviews the range of control algorithms used on these two wheeled balancing robots. There has been little investigation into how these balancing robots deal with different terrain. In [74] a two-wheeled robot manages to make a controlled ascent of a step. This break in smoothness is handled by three different controllers that are designed using Linear Quadratic Regulator (LQR). In [75] a two wheeled robot was simulated going over terrain but did not manage to do physical experiments.

2.4 Environment and Robotic Testing

The environment means the situation that robots are applied to or tested within. It is where the morphology and controller come together to react to what is around the robot. In this section, different applications and tests for robots are discussed. Modelling this robot environment interaction means the outcome of a situation can be predicted and planned for. This means whether the robot will fall over, get stuck or traverse the environment successfully. The only way to answer this question is to do limit testing; finding out what a robot can/cannot traverse.

2.4.1 Applications of Locomotion

There are a range of applications where the locomotion of a robotic platform is paramount. The four that will be discussed here are search & rescue, planetary exploration, nuclear industry and dangerous urban environment.

Search and rescue involves going into an area after a natural or man-made disaster has struck. According to [2] it involves being able to fit under and over obstacles; it achieves this by varying the diameter of the six legged Whegs. In [64], it states the key criteria is high ground adaptability for construction sites or disaster areas. Ability to survive falls and impacts is given as the most important feature in this type of environment in [76].

Planetary Exploration has mostly been done on Mars with rovers [77]. The first planetary rover was Lunokhod (1970) [78] which was remotely controlled on the moon. The majority of Mars rovers (Curiosity, Sojourner, Opportunity) use six wheels with a Rocker-Bogie suspension. In [1] the key to planetary robotics is allowing them to be more autonomous.

Nuclear Industry means man made environments which often have high amounts of radiation which are harmful for humans. In [15] robots are deemed useful in an unknown environment, whereas [3] focused on different forms of locomotion strategy to solve problems in nuclear environments.

Dangerous Urban Environments are situations where the human factor provides the dangerous element. Examples of these are bomb threats, hostage situations and police raids. Robots like the throwbot in [4] that are thrown into the environment need to be able to survive the impacts of this sort of deployment, whereas the Scout robot [49] is able to jump independently as well. In this work it states that for these sort of covert situations the small size of the robot is key.

The common thread for these situations is that the robot needs to be able to act independently. This means not getting stuck and acting autonomously or with limited human control. All the environments had unknowns which cannot be explored beforehand. This means that a robot needs to be capable on a range of terrains and be able to adapt to changes in the environment. The problem is that there is often little information about the specific sizes or shapes of terrain that might be found. There needs to be a standardised way of measuring and categorising an environment so the limit (however wide) could be given to what should be expected for a given application. In [9] the specific details of the environment are given so that the experiments could be completed by different platforms.

2.4.2 Modelling grip - Terramechanics

Terramechanics is the study of wheel and soft ground interaction [79], coined by Mieczysław G. Bekker. It is the study of how wheels sink into soil or sand and how this causes the wheel to slip. Over the years a range of theories have been developed and are reviewed in [80]. Its main conclusion is that a standardisation of how parameters are collected is needed. This is corroborated by [81] where specific shortcomings are discussed. Finally in [82] it was shown that adding ridges (grousers) to a wheel help it prevent sinkage.

These theories were first applied to the Lunar Roving Vehicle (LRV). Most grip design models for a wheel can be traced back to Bekker's work [79] on how ground deforms. He coined the phrase and discipline of terramechanics which is how soil properties interact with wheeled or tracked vehicles. It is based around two formulae one for the sinkage of a wheel (Equation 1.1) and the second Mohr-Colomb criterion for when a surface will fail. The combination of these equations gives the maximum grip of a wheeled system (Equation 1.2); the problem is getting the correct parameters about the surface. The collection of these parameters is covered in [81] which refer to Bekker's work as "pioneering" and [83] which use both these formulas.

Equation 1. Terramechanics equations

$$\sigma = \left(\frac{k_c}{b} + k_\phi \right) z^n \quad [1]$$

$$\tau_{max} = \sigma \tan(\phi) + c \quad [2]$$

Where in the first equation, σ is the maximum normal stress, k_c and k_ϕ are pressure-sinkage moduli, b is the wheel width and n is the sinkage exponent. In the second τ_{max} is the maximum shear strength, ϕ is the internal friction angle and c is the terrain cohesion.

Since then a variety of formulas have been developed and have been reviewed in [80]. It concluded that more standardisation is required, especially in the collecting of parameters. These models do not change the underlying friction that a robot has but instead allow for the possibility of sinkage. Once sinkage occurs, a robot needs more propulsion to make progress up a slope or even on the flat. These factors are often only considered for wheeled rovers used for planetary exploration.

2.4.3 Navigation

Navigation involves finding a way of moving around an environment. According to [84] there are four key concepts to navigation: sensing, mapping, localization and action. It is

possible to explore an environment without generating a map of the area or localising where the robot is. The actions in this case are directly determined by what the robot senses around it. Obstacle avoidance or wall following are examples of moving without a map. This approach is limited as without saving what the robot has experienced the robot will only be able to react to what is around it and will not be able to plan ahead.

Navigation involves generating or using a map of the surrounding terrain and then deciding a route through it. This map is a model of what is around the robot and should contain information on whether the robot can cross that area. The difficulty is that to make a map the location of the robot is needed and to calculate the location of the robot a map of the surrounding area is needed. A solution is to generate both at the same time through a process called Simultaneous Localisation And Mapping (SLAM) [85], [86]. Once a map is generated the path that a robot takes through the environment still needs to be determined. SLAM requires as much information and computer processing power as possible to be able to function effectively. In [87] the LIDAR and camera data generated an elevation map that was then updated with sensory data from the footsteps of the robot. This fusion of data allowed the AnyMal to navigate across a wide range of terrain. It is possible for localisation or map to be provided to the robot externally. For instance GPS or another tracking system could determine the robots location.

In [11] different routes are compared using distance, terrain traversability, and energy consumption. These are then combined to best overall route with Dijkstra's algorithm for the six wheeled robot used. Alternatively [88] uses B-spline numerical method to calculate whether sections of the map are traversable. The focus here is making sure the model of the environment is accurate. In [89] terramechanics is applied to a rover to plan the best path for a planetary rover. This involved two models; the dynamics of the robot itself and wheel-soil contact model.

The lack of research into path planning to hybrid robots is pointed out in [12]. A hybrid robot called LEON is tested against slope and rough terrain (rocks). From this an action planner decides whether the robot should move in wheeled or legged mode. The main issue with the LEON design is the 20 DoF it requires. This number of DoF means that design is quite heavy and its locomotion is inefficient. One of the main issues with hybrid robot locomotion is that they often have no or limited sensing capabilities. The lack of sensing means the robots are unable to know enough about their environment to make good decisions about their movement.

2.4.4 Benchmarks of robot's ability

A benchmark means that different robots can be compared to each other using a standard criteria. The question of which is the best suited robot from the earlier section is difficult

to answer.

The National Institute of Standards and Technology (NIST) has tried to create a benchmark by creating a robotic test facility [90]. It involves obstacles - slopes, staircases, hurdles, gaps and step fields - as well as different surfaces such as sand, gravel and mud. These were designed to test emergency response robots. Although quite comprehensive, it requires a large amount of setup and is infeasible for all robotic labs to invest as much. This means that not all robots will be tested against these requirements and therefore using it as a universal comparison is impossible.

A comparison was made using three types of obstacles: slopes, steps and gaps in [3]. The robots used were spherical, wheeled, tracked and Whegged robots. It found that tracks were best overall with spherical best at crossing a gap and Whegs best at climbing the step. The problem is that there are no details about the size or design of the robots so it is unknown if these played a factor in the outcome. Also, a comparison to other platforms is not possible without this information. These obstacles are a good choice but crucial information was missing.

The fairest metrics are ones that are relative to size of the robots themselves as otherwise building bigger robots would improve their score. Some benchmarks are already fair by their nature. The angles of slopes for instance are already fair for all robot sizes as the size of the robot is irrelevant (assuming the slope has the same amount of grip). Other metrics can be corrected to incorporate size as a factor. In [13] a range of metrics are used; slopes, shaped obstacles, speed and energy efficiency. In this paper, a robot's ability on steps, round ridge and uneven terrain is divided by the height of the robot. Speed is divided by the robot's length. Energy efficiency is calculated by dividing potential energy gained by climbing a slope by the energy used by actuators. These divisions turn the measurement into ratio and as such provide a much better benchmark for comparing robots. In [91] similar metrics are used for speed, obstacle, slope, soil sinkage and efficiency, however in this case they are measured against a benchmark vehicle rather than the size of each robot.

The question still arises whether these metrics of simple obstacles can be used to predict the outcome on complex terrain. It is proposed here that a robot's ability on slopes and steps can be used to anticipate what it can do on smooth and jagged terrain respectively. The advantage of using steps and slopes is that they can be defined by a single parameter and are easy to construct for physical testing.

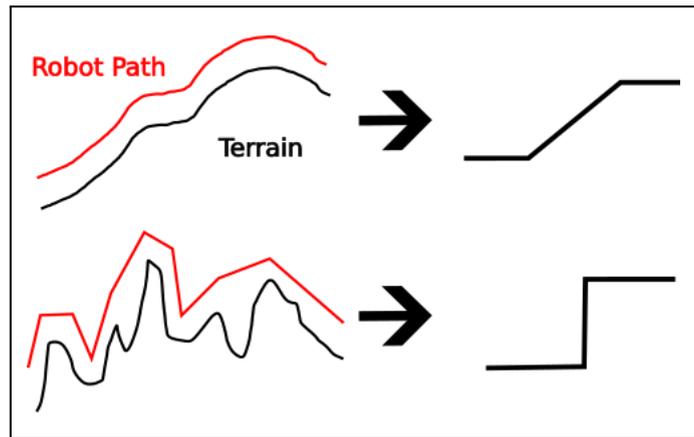


Figure 19. Smooth terrain measured by slopes and jagged terrain measured by steps.

The need for a standardised way of comparing robots that try to handle terrain was proposed in [35] which was a later paper featuring the R-Hex robot. In it the robot was tested on several different surfaces, steps and an obstacle course. Details of what these entailed were given which will allow these sorts of tests to be repeated, but whether they could be used for a standardised test platform is yet to be determined.

2.4.5 Cost of Transport

A decent way to measure a robot's capability is to measure its efficiency of movement. This is done by Cost of Transport (CoT) [92] and in [93] it is defined as the power used by the robot divided by the weight multiplied by velocity of the robot. It goes on to compare different systems in a scatter diagram of mass vs CoT. A similar graph can be found in [5]. Cost of Transport is one of the few comparison metrics which is morphology and application independent as well as widely applied.

One of the key factors in comparing these robots is Cost of Transport (Specific Resistance, COT). It can be defined using the power needed to maintain a velocity or the energy required to move a distance. This measure allows a robot's efficiency to be compared and with information about the battery, the distance it would be able to travel. With this R-Hex paper it was measured across a range of surfaces.

Equation 2. Cost of transport

$$COT = \frac{E}{mgd} = \frac{P}{mgv}$$

Where m is the mass of the robot, g is the gravitational constant, E is the energy to cover the distance d and P is the power to travel at velocity v .

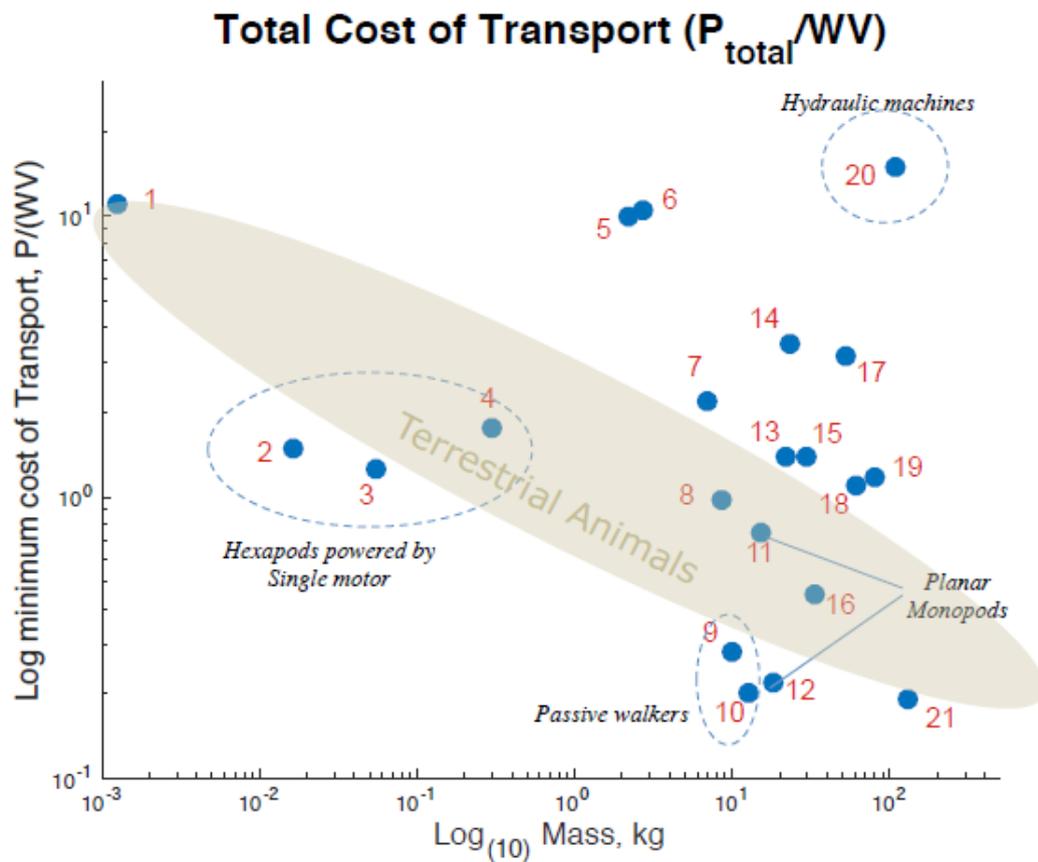
Table 3. Cost of Transport

Robot	Cost of Transport
PAW	0.18
ARL Monopod	0.22
MIY Cheetah	0.51
R-Hex	0.6/2*
Scout II	1.4
ATRIAS	1.46
ISprawl	1.75
Minitaur	2.3
7g Jumper	4.5

*=depending on version

It is interesting that the three top robots within this Table 3 display the three different styles of locomotion defined earlier. However, this list is not complete and the only rolling robot here is the PAW robot as many papers do not include this sort of information. This is the most common quantity used for comparing robot locomotion and is still not always present.

A common approach when dealing with cost of transport is to compare it to the mass of the robot as within the natural kingdom there is an inverse relationship with mass and cost of transport. A graph like this features in [5] but a simpler one is displayed here, figure 20 from [93]. It groups certain types of robots and has a shaded area so robots can be compared to wild animals. In [70] the cost of transport is compared with speed in a graph.



1 HAMR-VP (Baisch et al., 2014)	12 ARL Monopod II (Ahmadi et al., 2006)
2 X2-VelociROACH (Haldane et al., 2013)	13 Scout II (Buehler, 2002)
3 DASH (Birkmeyer et al., 2009)	14 StarLETH (Hutter et al., 2013)
4 iSprawl (Kim et al., 2006)	15 Fastrunner (Cotton et al., 2012)
5 Cheetah Cub (Sprowitz, 2013)	16 MIT Cheetah (Seok et al., 2013)
6 MIT Learning Biped (Collins et al., 2005)	17 ATRIAS 2.1 (Renjewski et al., 2013)
7 Rhex hexapod (Salanli 2001)	18 Asimo (Collins et al., 2005)
8 RHex-biped (Neville et al., 2003)	19 KOLT (Estremera et al., 2008)
9 Cornell Ranger (Bhounsule, 2012)	20 Big Dog (Raibert et al., 2008)
10 Cornell Biped (Collins et al., 2005)	21 ETH Cargo (Günther et al., 2015)
11 ARL Monopod I (Buehler, 2002)	

Figure 20. Graph of different robots comparing their Cost of Transport with their mass. A log graph is used to compress data so comparison can be made. Graph used here from [93] and contains robots not analysed here.

2.4.6 Froude Number

Originally from fluid dynamics, the Froude number relates speed and length. It is applied to robots as a measure of the excitement of a gait for a robot. A high Froude number would relate to a running pattern ($Fr > 1$) whereas a low one would mean a walking or climbing movement ($Fr < 1$). In [94] a comparison of the Froude number is made with compliant legged robots. The highest being awarded to the DASH robot with RHex coming in third but is the highest of the robots compared here.

Equation 3. Froude number

$$Fr = \frac{v}{\sqrt{gl}}$$

Where v is the robot velocity, g is gravity constant and l is the leg length from the hip.

Table 4. Froude number

Robot	Froude Number
PAW	0.69
Scout II	0.7
BigDog	0.98
Cheetah Cub	1.3
ISprawl	1.9
RHex	2.1

The advantage of the Froude number is that it is dimensionless and does not depend on the morphology of the robot. It gives an idea of how energetic a robot's gait is without specific or measured bias. The problem is how this sort of measure could be applied to wheeled robots. One possibility is to use the wheel or track radius as the leg length.

2.4.7 Contact area

The final proposed comparison variable, which is about the design of the robot, is contact area. This means the actuated area which is in contact with the ground. This is an important factor when determining sinkage which links back to Bekker theory. The sinkage will not only determine if a robot is able to traverse a terrain but also how much energy will be required to do so.

As far as the author is aware, this sort of surface sinkage has not been investigated for walking or jumping robots. The focus of most terramechanics is on wheeled or tracked vehicles. The sinkage comparison in [17] was between Rocker-bogie, tracks and the Elastic Loop Mobility System (ELMS), which the paper proposes. In this, the Track and ELMS drew with the Rocker-bogie design being an order of magnitude worse.

This would be important for soft terrain to know if a robot would become stuck and whether it would cause damage to the environment it moved through. The grip of a robot could also be estimated assuming the material of the contact area was consistent. To actually calculate this on real robots, a consistent amount of deformation of the ground would be needed. Then the area in contact after that would be calculated. An example of this would be a perfectly round wheel technically has an area contact of zero.

When you allow it to sink a small amount then a rectangle is formed and the area could then be calculated.

2.4.8 Test obstacles

Next there is a need for a standardised set of test obstacles. Proposed here are two obstacles which are a variable slope and a single step. The first would be a slope that is angled until the robot is no longer able to climb up it. It would test how well the robot could grip a surface, has the power to pull itself along and doesn't topple over. The Salto robot [71] is shown bouncing off an angled surface as an obstacle course. Although a slope would seem a good test of a robot's grip and drive power it does not appear to be a comparison benchmark in the literature.

One potential reason for this could be in constructing a variable slope that could handle the weight of the robots. Another could be the material the surface could be made out of, could mean standardisation would be difficult. It is also the case that robot design could be customised specifically for the slope task, like the climbing Mini-Whegs [57] or the more recent T-RHex [95]. Both of which are able to maintain grip beyond vertical, which could then make the next test of a step height redundant. This could be mediated by saying that the vertical surface of a step must be smooth so a robot cannot just climb up that.

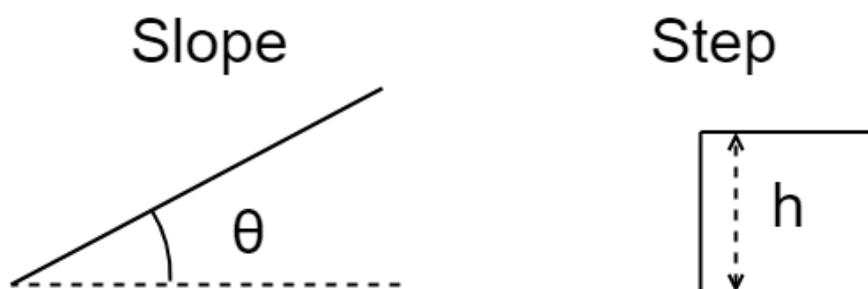


Figure 21. Slopes and steps diagram

The second would be the highest step a robot could make it over which then could be compared with the robot's height. Jumping robots could have an advantage with these sorts of obstacles. The Impass platform [37] is the most effective walking robot in this category being able to lift itself over a step $\sqrt{3}$ of its nominal height. Walking robots like this have an advantage to wheeled ones as even with a large amount of grip a circular wheel could only do a step less than half its diameter. In [74] it was only 5cm and the wheel was 19.6cm in radius. When a wheel has an active suspension system or instead uses track this can increase though. The comparison of Rocker-bogie, tracks and the

ELMS in [17] found they could scale obstacles 1.5, 2 and 3 times their wheel/track diameter respectively.

The reigning champions of step ascending would have to be jumping robots which the jump height will now be compared. In Table 5 they are ordered by jump height over the height of the robot. The majority of the robots in the table use mechanical springs with the exception of the Sand Flea [96]. It is a Boston Dynamics design and does not have any academically published information. The Sand Flea also uses a CO2 piston to power these large jumps and it is the heaviest robot jumping to these sorts of heights.

Table 5. Jump comparison table

Robot	Weight (Kg)	Jump GPE (mJ)	Jump Height (cm)	Robot Height (cm)	Jump Ratio (jump h ÷ robot h)
Jollbot [20]	0.465	101.4	21.8	15.6	1.4
Mini-Whegs [57]	0.146	46.9	22	10	2.2
Scout [49]	0.2	79.4	35	10	3.5
Salto-1P [71]	0.103	100.9	90	17	5.3
7g Jumping [47]	0.007	0.966	138	5	27.6
Sand Flea [96]	5	40000	800	15.24	51.9
Hybrid Spring [48]	0.0225	8	3200	30	106.7

2.5 Summary

There are several factors that determine a robot's capability in terrain traversal. First its morphology determines the components, size and potential agility. The classification system used here splits robots into how they interact with the ground, whether they roll, walk or jump. The complexity of the control system needed to move is determined by the number of degrees of freedom and whether the robot is stable. Finally, the environment the robot is tested against needs to be suitable for its design and gait. If the wrong metric

is used then a judgement may be unfair to certain robots. Although locomotion benchmarks exist for robotic platforms, they are often not applied to hybrid robots.

Fully autonomous navigation on unstructured terrain is often the last feature to be achieved by a mode of locomotion. It was first done with rolling robots then more recently by walking robots and has yet to be done with jumping ones. When it comes to hybrid robot navigation, more investigation is needed as the field is reasonably unexplored. A hybrid robot needs to know which mode of locomotion is suited to each terrain type and decide which route might be best. Switching between modes of locomotion needs to be further investigated with an emphasis on determining which mode is optimal.

In the following chapter of this thesis, Cylindabot is simulated crossing a wider range of obstacles and tested in hardware. The design decisions and inspiration from literature of Cylindabot is discussed in Chapter 4. In Chapter 5 its ability to adapt to slopes and steps is shown to be advantageous. Before that, in the next chapter, initial designs are tested in simulation to inform the later design and experimental setup.

Chapter 3: Preliminary Results

Early versions of Cylindabot tested in simulation against slopes and steps.

3.1 Introduction

Before designing a physical robot it was decided that simulation experiments could be used to inform the final design. From the literature review, it was clear a hybrid locomotion robot could adapt to traverse its environment. To achieve this, a simulation environment was created and some initial tests were carried out.

First the simulation program was chosen as V-Rep (later CoppeliaSim). V-Rep had a useful graphic interface that was used to control basic designs of Cylindabot manually. Single steps and flat slopes were to be used as obstacles to test ideas. The process of testing then needed to be automated. It was attempted to use an obstacle that slowly shrunk until it became traversable, but too often the robot failed before crossing it. The next option was to have the object at a fixed height during a simulation run. Cylindabot was tested repeatedly at multiple heights until it consistently failed. This method was computationally expensive but effective and is similar to the algorithm used in [64].

Mathematical models were generated to predict these simulated results for a wheeled version of Cylindabot [97]. The models proved to be effective at predicting performance for slopes but not so well steps. Next, the deployment of the legs was investigated and compared with other dimensions of Cylindabot. Specifically, the width between the Whlegs and the length of the tail were varied. This work is important because it laid the foundation for all future simulation work in this thesis and allowed the design of Cylindabot to be tested before the physical designs began.

3.2 Simulation Environment

A simulation environment allows for repeated and precise testing of a robot in a range of environments. There are a variety of potential platforms that can be used to simulate the robot and the environment. The main area of study here is the interaction between the robot and its environment, which means the quality and reliability of the physics engine is one of the most important requirements. For ease of use and to check the validity of the simulation, the program should have a high quality Graphical User Interface (GUI). This means that the simulation can be edited and monitored through this GUI. Finally, the design of Cylindabot and environment terrains might need to be altered during experimentation, so they need to be easy to edit and be encoded separately.

Following various tests and reading of the literature CoppeliaSim, previously known as V-rep, was used to simulate the Cylindabot and obstacles that it attempts to traverse. The two main competitors to CoppeliaSim are Gazebo and Unity. The pros and cons of these three platforms are compared here. All three programs are cross platform so can run on more than one operating system.

Gazebo is a robot simulation program from Open Robotics which is run from the terminal. It has a strong integration with the Robot Operating System (ROS). ROS allows communication between different sub-programs and access to a wide range of robotic libraries. The main reason Gazebo was not chosen for this project is because of graphical user interface limitations.



GAZEBO



Unity is primarily used for game development and can be adapted for use as a robotic simulator. The majority of the physics engines used for simulation are for the computer game industry. Unity's main strengths are the stability of its code base as it is used professionally and the high level graphics make Computer Vision much more realistic. Although considered it was not used as a more specialised simulation program was more appropriate.

CoppeliaSim from Coppelia Robotics was previously known as V-Rep and shares most of its characteristics. The main advantages of CoppeliaSim are the seven programming languages that can be used to control it and the range of physics engines that it incorporates. Every aspect within the program is modular so if one part needs alteration it can be done in isolation. It has an easy-to-use user interface which allows rapid



CoppeliaSim

testing of platforms and user observation of experiments. For these reasons CoppeliaSim/V-Rep was chosen to be the simulation engine for Cylindabot. When first simulating Cylindabot, V-Rep was used and later in this research it was upgraded to CoppeliaSim. They function almost identically so this change did not affect how the simulation functioned or slowed down progress.

There are three main components that were used to simulate the robot and its environment within CoppeliaSim: scenes, models and Lua scripts. The scene is a file that contains everything that the simulator uses. This includes robots, environment and scripts as well as various settings for that simulation.

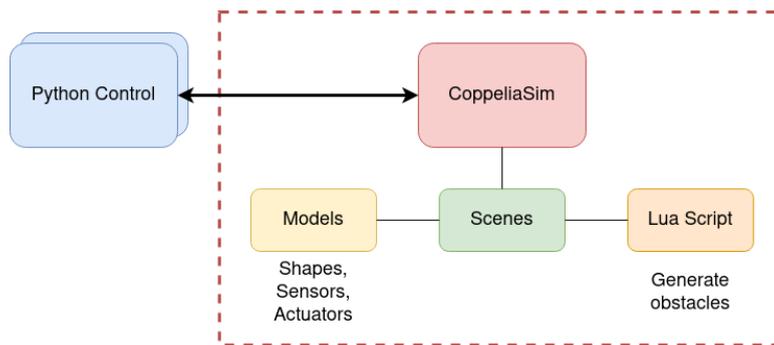


Figure 22. Simulation architecture for CoppeliaSim

The model files in CoppeliaSim hold a simulated robot, or any other object, made up of various parts. This allows items to be saved separately from a scene as a combined object. For Cylindabot three main parts were used: primitives, actuators and sensors. Primitives are basic shapes that can be grouped together to make more complex shapes. Specifically they are cuboids, cylinders or spheres. These were used for generating the shape and creating the weight imbalance for the wheeled version of the Cylindabot design. Actuators control the movements within a robot. For Cylindabot, they control the Wheg rotation, leg deployment and tail position. The Wheg rotation is how the robot moves whereas the other two are the configuration of Cylindabot.

Two programming languages were used to control the robot and the environment: Lua and Python. The Lua scripts were used to generate terrain obstacles. The Python scripts were used to control the movement of Cylindabot, start/stop the simulations, call Lua functions and save required data. This arrangement of different scripts did evolve during the course of the experiments. For instance, the first testing did not use scripts and was controlled manually, whereas the final experiments used several different Python scripts for different aspects of the simulation.

3.3 Initial Testing

The initial tests of the robot were done by using the Graphical User Interface (GUI) of CoppeliaSim. This supplied a rapid testing of the initial designs for Cylindabot. The model of the robot was created by placing primitives that were grouped together to form more complex shapes like the legs or body. These were then connected in a hierarchy by actuators and then all components were saved as a model. Different versions of Cylindabot were saved as named models so they could be used interchangeably with environments.

Here is an example of a hierarchy of the curved tail robot in figure 23:

- Main body (transparent cylinder)
 - Right motor (rotational actuator)

- Right wheel (flat cylinder)
 - Left motor (rotational actuator)
 - Left wheel (flat cylinder)
 - Tail motor (rotational actuator)
 - Tail (4 convex shapes grouped together)

The different layers of bullet points represent the hierarchy where more indented points are a child of the previous one. The main body acts as the root of the robot model. Each of the child items were positioned into the correct location and orientation, then attached to their parent object. Once attached, if the parent is manually moved then all descendants in the hierarchy move with them. Also, once the simulation is running, forces are transferred both up and down the hierarchy.

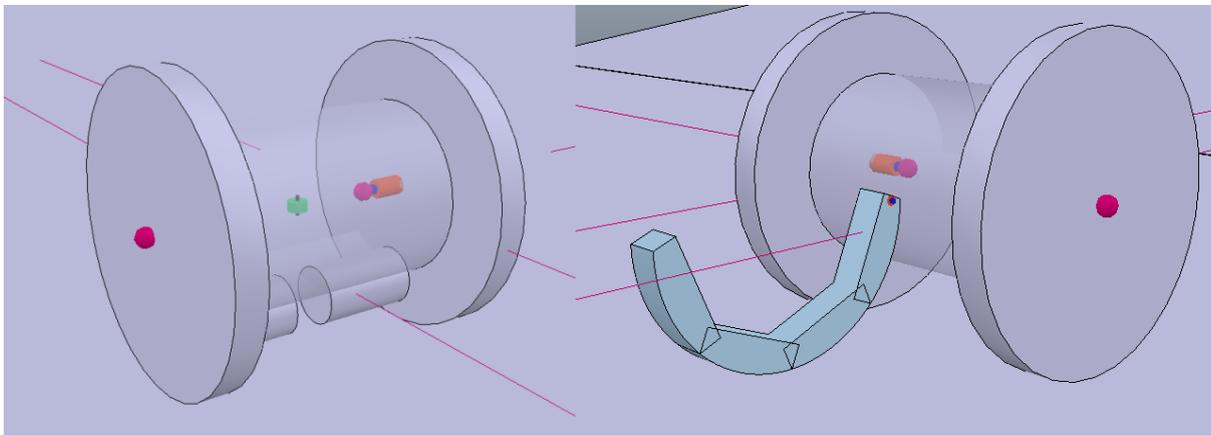


Figure 23. Wheeled simulation models. Without tail with offset mass and curved tail.

To start with, to move the robot, the speed of the Whegs were manually set in the parameter window for the actuators. This meant that the robot would move once the simulation was started and the user could observe its movement. Although these tests were not scientifically rigorous, they did give insights into potential flaws in the design and control problems. For instance, when moving with a counter balance weight and without a tail a proportional controller was needed. Without this controller the robot would oscillate as it moved quite wildly. The curved tail design (Figure 23, right) was not feasible for using legs as it would become too unstable when legs fully deployed. When going over steps with just wheels the speed of the robot had a significant impact on success as it would often bounce up the step. These steps were fixed in place with increasing height and how many the robot made it over was its success measure. These observations led to a second design with a permanent straight tail to allow uses of the legs. The wheel-only version of Cylindabot was used in section 3.5. A mathematical model was created to predict the robot's capabilities on steps and slopes. The majority of the other work in this research focused on deploying and retracting legs so a straight tail was used.

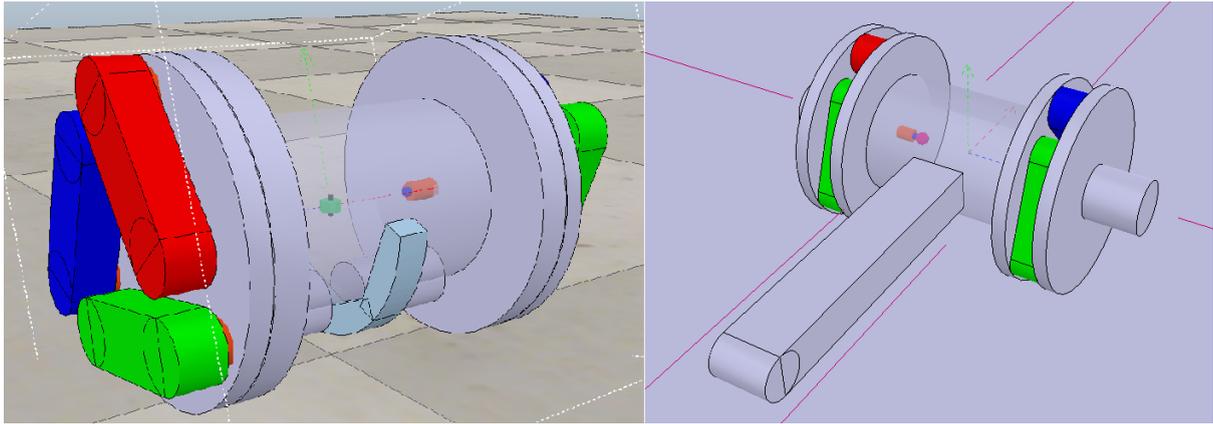


Figure 24. Curved tail with end mounted legs vs straight tail with encased legs. Early versions of Cylindabot.

The straight tailed version of Cylindabot meant that the tail was always present and had advantages. The legs were now encased from both sides to improve mechanical robustness. The longer tail meant that the legs' full potential could be realised. A small cylinder was added to the outside of the Whlegs to prevent the robot being stuck on its end. To enable changing the robot's tail length or width, a different robot model was created that could be loaded when needed. The next question was how to calculate the size of obstacles Cylindabot can traverse.

3.4 Shrinking Step

The idea here was to slowly make the step obstacle easier until the point Cylindabot was able to climb over it. The step shrunk at a constant rate, so the time at which it crossed the step could compute the size of the step. This strategy did not work. What actually happened is that the robot, more often than not, had a problem before the step was short enough. These problems caused one of three outcomes: the robot fell off the side of the environment, it became stuck on its side or a physics error occurred (its position became not a number). Only the runs where the robot did not do one of these gave useful data. The proportion of these successful runs were found to be too low to be tenable. A variety of attempts were made to amend these errors, the success rate of which is summarised in Table 6.

Table 6. Success Rate of a Shrinking Step

	Tail Length (cm)	Successes	Fall	Time out	Physics Error	Number of runs
No Controller	15	0.057	0.529	0.000	0.402	87
	25	0.046	0.759	0.000	0.195	87
	35	0.276	0.632	0.000	0.092	87

	Tail Length (cm)	Successes	Fall	Time out	Physics Error	Number of runs
Proportional controller	15	0.178	0.200	0.244	0.378	45
	25	0.148	0.403	0.103	0.345	290
	35	0.156	0.644	0.044	0.156	45
Big Increment	25	0.172	0.366	0.214	0.248	145
Small Increment	25	0.269	0.317	0.193	0.221	145

To begin with the robot had its wheels at a fixed speed and ran at the step which is labelled as no controller. Then, the robot was run with a proportional controller. This is where the angle of the robot was calculated, subtracted from the target direction and the speed changed in the Whlegs to adjust the direction of the robot. The big and small increments refer to how much the step height was reduced by. These strategies varied the success rate between 5 and 27%. The number of runs for each of these varied as they were simply seeing which setup was viable rather than trying to get reliable results. This method was meant to reduce the number of simulation runs required. It was unsuccessful, during the later work two different simulation approaches were generated: the fixed position approach and the ‘beer mat’ method.

3.5 Fixed Position

The next method used was to fix the obstacle’s position. The obstacle is set to a static height and the robot is given a short amount of time to cross it. This led to all heights being attempted. The number of errors became insignificant and hence the data more reliable. In Table 7 the the success rate is shown for the legs, set to 30 degrees for the straight tailed version of Cylindabot. Each height was run 30 times and with 20 different heights meaning 600 runs for this leg angle.

Table 7. Success rate for a fixed step with a leg angle of 30 degrees

Step Height (mm)	Success Rate %	Time Outs %	Falls %	Mean Time of Successes (sec)
0	100	0	0	3.57
5	100	0	0	3.56
10	100	0	0	3.57
15	100	0	0	3.63
20	100	0	0	3.64

Step Height (mm)	Success Rate %	Time Outs %	Falls %	Mean Time of Successes (sec)
25	100	0	0	3.68
30	100	0	0	4.07
35	100	0	0	4.09
40	100	0	0	4.19
45	100	0	0	4.35
50	100	0	0	4.45
55	100	0	0	4.48
60	100	0	0	4.42
65	100	0	0	4.48
70	100	0	0	4.43
75	100	0	0	4.43
80	100	0	0	4.45
85	100	0	0	4.51
90	100	0	0	4.53
95	100	0	0	4.57
100	100	0	0	4.56
105	100	0	0	4.67
110	100	0	0	5.29
115	100	0	0	5.08
120	100	0	0	5.52
125	100	0	0	5.32
130	100	0	0	6.98
135	63.3	36.7	0	7.92
140	16.7	83.3	0	7.83
145	3.3	93.3	3.3	14.15
150	0	90	10	-
155	0	100	0	-
160	0	90	10	-
165	0	86.7	13.3	-
170	0	96.7	3.3	-
175	0	96.7	3.3	-
180	0	96.7	3.3	-
185	0	93.3	6.7	-
190	0	90	10	-
195	0	86.7	13.3	-

The most important characteristic of this data is the sharp drop-off after 130mm. The drop-off is where the robot shifts from succeeding to failing in the space of 20mm. The fall in success rate can be seen in Figure 25. This is the main area of interest when looking at terrain traversal. When does it start and what is its distribution? It can be used to estimate the maximum step the robot can surmount.

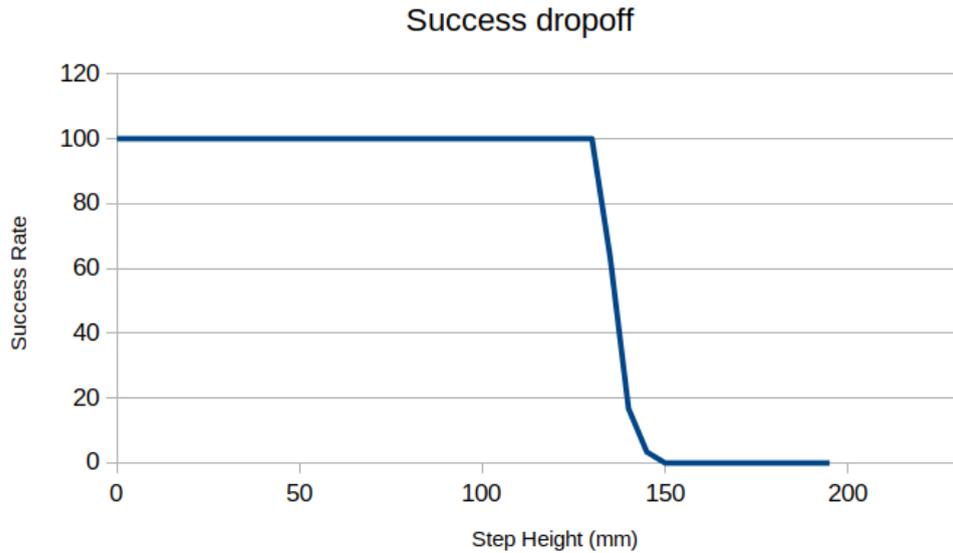


Figure 25. Success Drop-off for a step with a leg angle of 30 degrees. Simulation results every 5 mm with 30 runs at each height.

The problem here is how to estimate the maximum obstacle height that the robot can traverse as the outcome of each simulation run is binary: success or fail. There are two potential methods that can be used to estimate this failure region; one based on the failure drop being a sharp linear drop and the other on mean average. The first procedure estimates the maximum height by stacking all the successes that happen in order and linearly interpolating the height. This finds the centre of the failure region by inferring the drop is linear. The formula for this estimated maximum height is shown in Equation 4. The second, the mean approach, is simply doubling the mean height of successful runs. The mean of the success should calculate the middle of the success region but is susceptible to erroneous data. As these fixed positions start from 0 difficulty which should always be successful if the mean is doubled it should estimate a point in the failure region.

The first approach was chosen as it is more robust to outliers and skewed results curves. The difference between the estimated maximum and the doubled mean was used to calculate the error bars of the data. This is due to the skew of the data being analogous to the spread of the data.

Equation 4. Equation for Maximum Estimate

$$\text{Estimated Max Height} = \frac{\text{Successes} * \text{Height Increment}}{\text{Runs per Increment}} - \text{Height increment}$$

3.6 Mathematical Analysis - Just Wheels

Now that a method was determined to estimate the maximum height of an obstacle that a robot could cross, the first round of data collection was done. This was for the first version of Cylindabot without legs. The mathematical kinematics were derived for a comparison with simulated results.

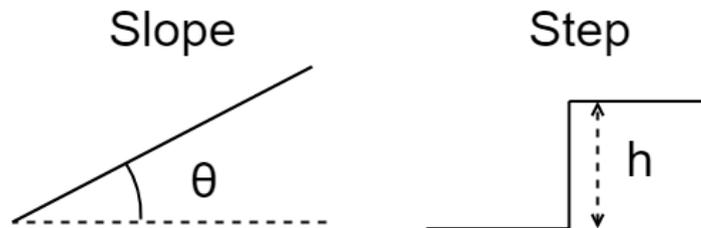


Figure 26. Diagram of slopes and step

The two obstacles used were a slope and a step. These were chosen as they are simple to model mathematically and give a measure of how a robot can perform on smooth and jagged terrain. The two robot designs can be seen in Figure 23 where one is just wheels and the other has a tail to balance it.

In this section the mathematical derivation of several inequalities is produced. If the inequality holds true then Cylindabot should be able to make it up the obstacle. For version one of the robot this analysis focuses on weight position and momentum. The second version has a tail added, this means the moment of rotation between the wheels and the tail becomes how the inequalities are derived.

3.5.1 Wheel with Weight Offset

The two approaches being analysed here are, (i) whether a robot can make it up the obstacle slowly without a run up; (ii) calculate the velocity needed to make it up by momentum alone.

Table 8. Mathematical terms

Term	Definition	Term	Definition
F	Friction	R	Radius of wheel
G	Ground reaction	r	Radius of mass
μ	Frictional coefficient	g	Gravity constant
θ	Slope or step angle	v	Velocity
M,m	Mass	σ	Modulus of restitution
h	Height of step	D	Drive force

Firstly there needs to be enough friction to be able to grip the slope. By using the Coulomb friction formula and balancing different forces the below is derived. Equation 5 is not specific to Cylindabot design but would be true for any vehicle going up a slope.

Equation 5. Friction based inequality

$$F > \mu G \quad [1]$$

$$G = Mg \cos \theta \quad [2]$$

$$Mg \sin \theta < \mu Mg \cos \theta \quad [3]$$

$$\tan \theta < \mu \quad [4]$$

For continuous climbing of a slope, the centre of mass has to be further forward than the point of contact. Once it is, the robot will roll up the slope and the motors could keep the mass at this point so it continues to climb.

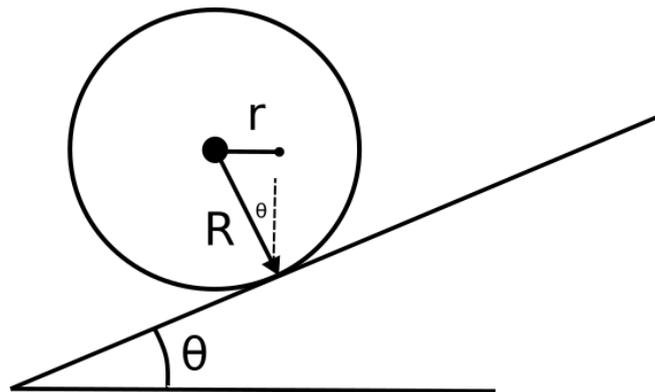


Figure 27. Wheel on slope to show balance point

Equation 6. Mass position inequality for climbing a slope

$$\frac{r}{R} > \sin \theta$$

Where r = radius of the centre of mass and R = radius of wheel.

Steeper sections might be able to be crossed with a run up. The kinetic energy would be converted into gravitational.

Equation 7. Energy inequality for climbing a slope

$$\frac{1}{2}mv^2 > mgh \quad [1]$$

$$v > \sqrt{2gh} \quad [2]$$

For slow ascension of a step, the dynamics are similar to a slope. The centre of mass needs to be further forward than the point of contact. The angle of contact however is now dependent on the wheel size as well. Only the maximum step height needs to be tested as any progress made up the step will then effectively make it up to a smaller step.

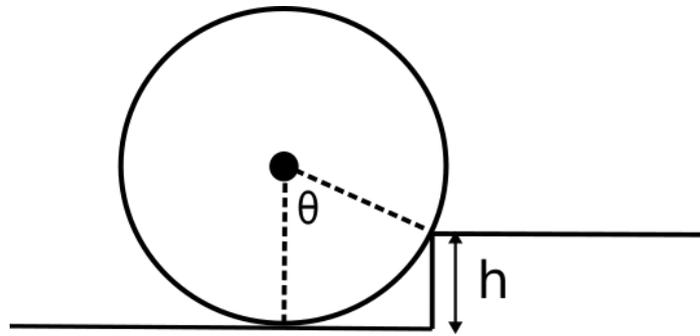


Figure 28 Simple wheel against a step to show relation to central axis angle

Equation 8. Mass position inequality for climbing a step

$$\frac{r}{R} > \sin\theta = \frac{\sqrt{h^2 - 2hR}}{R} \quad [1]$$

$$\Rightarrow r > \sqrt{h^2 - 2hR} \quad [2]$$

For fast ascension of a step, the momentum could be built up and then the step could be bounced over. For this, the robot hitting the step could be treated as an impact. The velocity will be split between the velocity into the corner of the step and perpendicular to it. For simplicity, the robot is being treated as a point mass so that rotational inertia can be ignored.

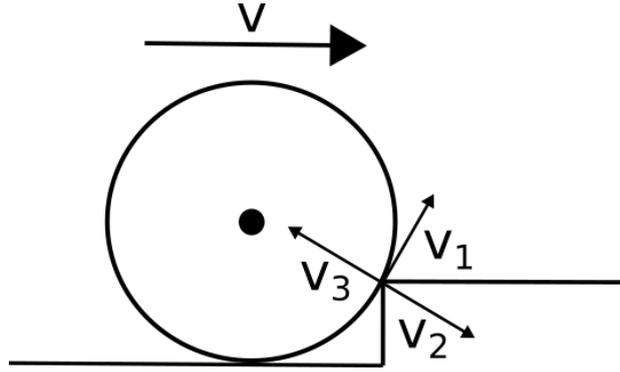


Figure 29. Velocities at point of impact with the step

Equation 9. Splitting up the velocities for a step

$$V = \text{Forward Velocity} \quad [1]$$

$$v_1 = V \cos \theta \quad [2]$$

$$v_2 = V \sin \theta \quad [3]$$

$$v_3 = \sigma v_2 = \sigma V \sin \theta \quad [4]$$

Here v_3 is the rebound velocity caused by the impact and σ is the modulus of restitution between the surface and the wheel. These now need to be converted back into vertical and horizontal velocities v_y, v_x .

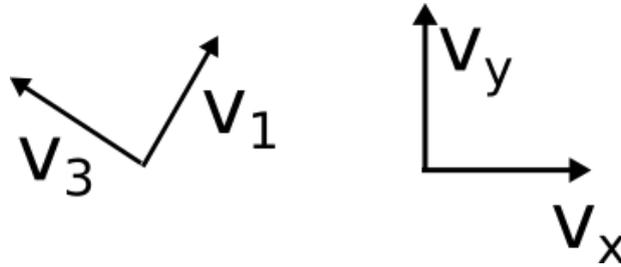


Figure 30. Velocity vectors being transformed back into forward coordinate frames

Equation 10. Derivation of fast step climbing inequality

$$v_x = V(\cos^2 \theta - \sigma \sin^2 \theta) \quad [1]$$

$$v_y = V(1 + \sigma) \sin \theta \cos \theta \quad [2]$$

$$\frac{1}{2} m V_t^2 = v_x^2 + v_y^2 \quad [3]$$

$$\text{if } c = \cos \theta \text{ \& using } \sin^2 = 1 - \cos^2 \quad [4]$$

$$V_t^2 = V^2((1 + \sigma)c^2 - \sigma)^2 + (1 + \sigma)^2(c^2 - c^4) \quad [5]$$

$$V_t^2 = V^2((1 - \sigma^2)c^2 + \sigma^2) \quad [6]$$

$$\text{Requirement: } \frac{1}{2} m V_t^2 > mgh \quad [7]$$

$$\Rightarrow V^2 > \frac{2gR(1-c)}{(1-\sigma^2)c^2 + \sigma^2} \quad [8]$$

If the above inequality is true, then the robot has enough kinetic energy to make it up the height of the step after colliding with the step. Using energy like this should provide a more robust measure as it takes into account all potential ways that the robot could bounce up the step. These inequalities for wheeled robots without a tail first have variables substituted into them and then are compared against simulated results.

3.5.2 Wheels and Curved Tail

The addition of a tail allows the robot to apply more torque through its wheels, though this creates an extra drag behind the robot. It also means that there is another point of contact with the floor, making a more stable platform. It is now not the centre of mass of the robot that moves it forward, but using torque, D applied to the surface by the wheels. This means that friction plays a much more crucial role to the mechanics of the robot's movement. The most common model for friction is $F = \mu R$ where R is the reaction of the surface and μ if the coefficient of friction. This equation represents the maximum friction that is able to be created by the surface. To reduce confusion with wheel radius, the two reaction forces here will be denoted as G_1, G_2 . To simplify the model for mathematical purposes, the radius of the tail is put as the same as the main wheels. The mass is split as M , the mass of the body of the robot, and m , the mass of the tail. Both masses are on the centre line of the robot.

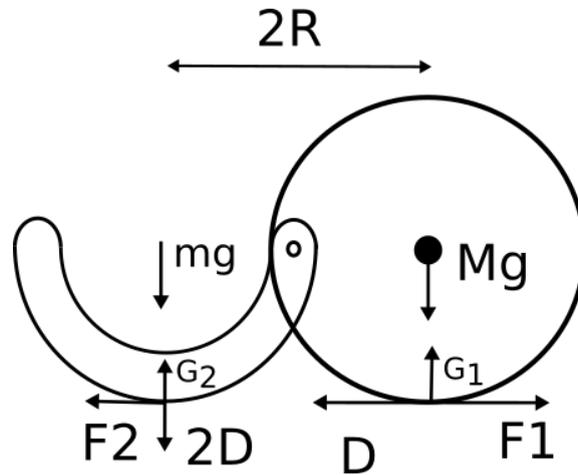


Figure 31. Forces on tailed robot for a flat surface

Equation 11. Inequality for tailed robot on a flat surface

$$F_1 \leq \mu_1 G_1 \quad [1]$$

$$F_2 = \mu_2 G_2 \quad [2]$$

$$G_2 = 2D + mg \quad [3]$$

Now if you want the robot to move on a flat surface.

$$D + F_2 < \mu_1 G_1 \quad [4]$$

$$(1 + 2\mu_2)D + \mu_2 mg < \mu_1 Mg \quad [5]$$

Hence a slip condition for acceleration:

$$D < \frac{\mu_1 Mg - \mu_2 mg}{1 + 2\mu_2} \quad [6]$$

This can be simplified to a move condition as $D \rightarrow 0$:

$$\mu_1 M > \mu_2 m \quad [7]$$

For a flat surface like this, equality is almost always going to hold true. Mass at the centre of the robot (M) is going to be heavier than the tail (m) so $M > m$ and the friction of the wheels (μ_1) will be greater than the tails friction (μ_2).

For a slope, the moments of rotation around the two points of contact on the slope are used to calculate the dynamic requirement. The $\odot G_1$ refers to the moments around the G_1 point of contact in an anti-clockwise direction.

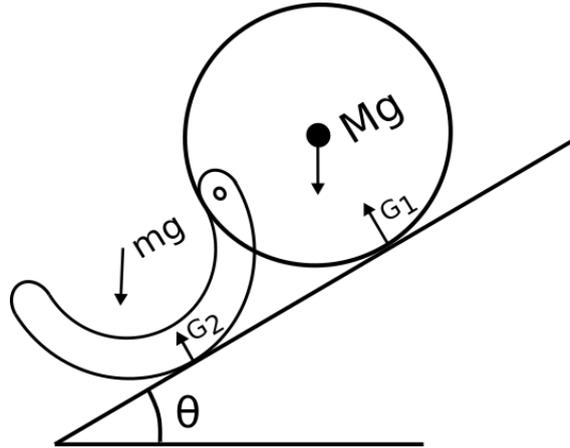


Figure 32. Slope with a tail with forces for dynamics

Taking moments around G_1 and using γ to represent $\tan^{-1}(0.5)$ which is the angle between the centre of the tail and wheel contact point:

Equation 12. Inequality for a slope with a tail

$$\odot G_1 : MgR \sin \theta + \sqrt{5}mgR \cos(\theta - \gamma) \quad [1]$$

$$\odot G_1 : 2RG_2 \quad [2]$$

$$G_2 = \frac{1}{2}Mg \sin \theta + \sqrt{5}mg \cos(\theta - \gamma) \quad [3]$$

Balancing forces perpendicular and parallel to the slope:

$$\nearrow : G_1 + G_2 = (M + m)g \cos \theta \quad [4]$$

$$\nearrow : D + F_2 + (M + m)g \sin \theta < \mu_1 G_1 \quad [5]$$

$$D + \mu_2(2D + G_2) + (M + m)G_2 < \mu_1((M + m)g \cos \theta - G_2) \quad [6]$$

$$D < \mu_1(M + m)g\cos\theta - (\mu_1 + \mu_2)G_2 - (M + m)g\sin\theta \quad [7]$$

Interestingly here, if the drive force is large, it breaks this equality which implies slippage between the wheel and the ground. The final inequality assumes that it is desirable for the wheel not to slip. Later on this drive force is set to zero to find the maximum angle that the wheel could grip on while maintaining speed.

For steps, a similar approach of using moments around the wheel contact point can be applied. The added complication is that the robot might be at an angle up the step as well as the step being a height. So a new variable δ represents the angle of the robot on the step.

Equation 13. Angle of robot on step

$$\cos\theta = \frac{R + 2R\sin\delta - h}{R}$$

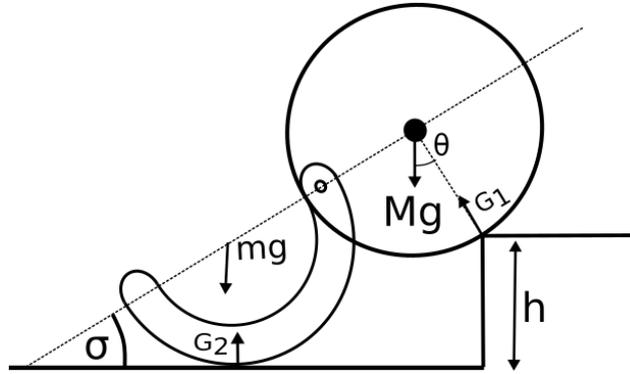


Figure 33. Forces on wheel and tail on a step obstacle.

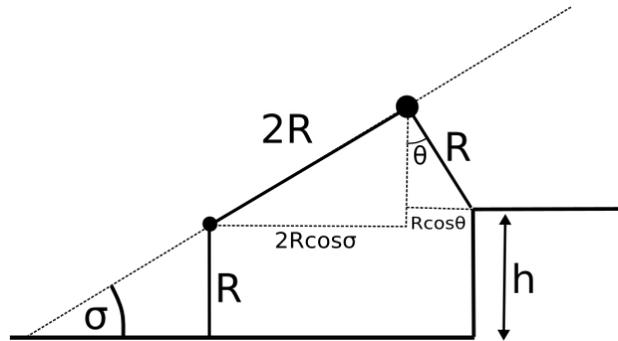


Figure 34 Angle and position diagram to determine moments.

Equation 14. Inequality for climbing a step with a tail.

$$\odot G_1: MgR\sin\theta + \sqrt{5}mgR\cos(\theta - \gamma) \quad [1]$$

$$\odot G_1: G_2(2R\cos\delta + R\sin\theta) + F_2h \quad [2]$$

$$F_2 = \mu_2(G_2 + 2D\cos\delta) \quad [3]$$

$$G_2 = \frac{MgR\sin\theta + \sqrt{5}mgR\cos(\theta - \gamma) - 2\mu_2D\cos\delta}{2R\cos\delta + R\sin\theta + h\mu_2} \quad [4]$$

$$\mu G_1 + G_2 \sin\theta > M_t g \sin\theta + \mu G_1 \cos\theta \quad [5]$$

If this equality is true, then the robot should be able to make it up the step. The key parts being the ground reaction forces and the frictional coefficient. The inequalities for a tailed robot in the next section have values inserted and then later are compared to simulated results.

3.5.3 Mathematics Results

The inequalities above are correct in theory but to get relevant results from all of them, parameters need to be substituted into them. The most important being the radius of the centre of mass, r and frictional coefficient, μ . Due to their significance, they both are given two possible values. This should inform potential designs by producing potential capabilities and multiple data points to test the validity of the inequalities.

Equation 15. Variable for inequalities

$$R = 75mm \text{ (wheel radius) [1]}$$

$$r = \frac{R}{2} \text{ or } \frac{R}{4} \text{ (mass offset) [2]}$$

$$\mu_1 = \mu_2 = 0.5 \text{ or } 1 \text{ (friction) [3]}$$

$$\sigma = 0.1 \text{ (modulus or restitution) [4]}$$

$$M_t = M + m \text{ (total mass) [5]}$$

$$m = 0.1M_t \text{ (tail mass) [6]}$$

$$M = 0.9M_t \text{ (centre mass) [7]}$$

Firstly, the friction required to stop the robot just sliding down the slope, ignoring drive forces. With the given frictional coefficient, this is the maximum theoretical angle that could be traversed.

Equation 16. Frictional limit results

$$\tan\theta < \mu = 0.5 \quad [1]$$

$$\theta < 26.57^\circ \quad [2]$$

If the frictional coefficient was increased to $\mu = 1$ then the maximum angle would increase to 45° . Which basically makes friction practically irrelevant for making it up the slope. Without a tail, inequality in Equation 6 was used with the two centres of mass:

Equation 17. Centre of mass results

$$\frac{r}{R} = 0.5 > \sin\theta \quad [1]$$

$$\theta < 30^\circ \quad [2]$$

$$0.25 > \sin\theta \quad [3]$$

$$\theta < 14.48^\circ \quad [4]$$

The first of these is not possible with $\mu = 0.5$ but is a theoretical limit for this design if the frictional coefficient was increased.

For a step, the first condition is for the frictional limit of a simple two wheeled robot from Equation 5:

Equation 18. Friction limits for a step

$$\begin{aligned} \mu &= 0.5: \\ \theta &< 26.57^\circ \quad [1] \\ \frac{2}{\sqrt{5}} &< \frac{R-h}{R} = \frac{75-h}{75} \quad [2] \\ h &< 30 - 6\sqrt{5} = 7.92\text{mm} \quad [3] \end{aligned} \qquad \begin{aligned} \mu &= 1: \\ \theta &< 26.57^\circ \quad [4] \\ h &< 22.0\text{mm} \quad [5] \end{aligned}$$

Next the condition for the centre of mass being able to get the robot up the step (Equation 8).

Equation 19. Centre of mass results

$$\begin{aligned} r &= \frac{R}{2}: \\ \theta &< 30^\circ \quad [1] \\ \frac{\sqrt{3}}{2} &< \frac{75-h}{75} \quad [2] \\ h &< 10.1\text{mm} \quad [3] \end{aligned} \qquad \begin{aligned} r &= \frac{R}{4}: \\ \theta &< 14.48^\circ \quad [4] \\ 0.968 &< \frac{75-h}{75} \quad [5] \\ h &< 2.40\text{mm} \quad [6] \end{aligned}$$

The following paragraphs consider a robot with a tail. The inequality Equation 12 as $D \rightarrow 0$ and $\mu = 0.5$ can be used:

Equation 20. Tail on a slope results

$$\begin{aligned} 0 &< 0.5\cos\theta - 0.45\sin\theta - 0.05\sqrt{5}\cos(\theta - \gamma) - \sin\theta \quad [1] \\ \tan\theta &< \frac{0.4}{1.5} \quad [2] \\ \theta &< 14.9^\circ \quad [3] \end{aligned}$$

When recalculated with $\mu = 1$, the mathematics gave an angle of $\theta = 21.8^\circ$ to have a tail. This result is not too different from that for a centre of mass at a quarter of the wheel's radius. So a tail does not significantly help to climb a slope.

For a tailed robot there are more floating variables D and σ on a step. To make them solvable $D \rightarrow 0$ and $\delta \rightarrow 0$ can be set:

Equation 21. Tail on a step results

$$\frac{0.9}{2} + \frac{\sin\theta + 0.2\cos\theta}{\sin\theta} \sin\theta > \sin\theta + \frac{\sin\theta + 0.2\cos\theta}{2 + \cos\theta} 0.5\cos\theta$$

When solved the result is $\theta < 26.9^\circ$ which translates to a step height of 16.2mm. For the results of the fast step ascension the speed of the robot is needed so in the next section it is estimated using the time to traverse a flat surface.

3.5.4 Comparison with Simulation

In this section these mathematical results are to be compared with a similar set up with a V-rep simulation. This simulates the problem in three dimensions and hence other outcomes are possible.

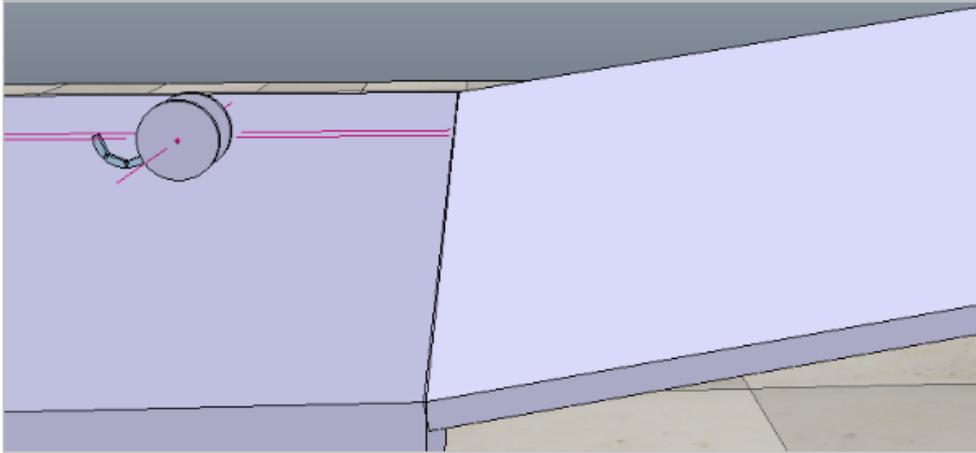


Figure 35. Simulation setup for a slope. The cuboid primitive was angled and positioned to keep contact point with the starting tile.

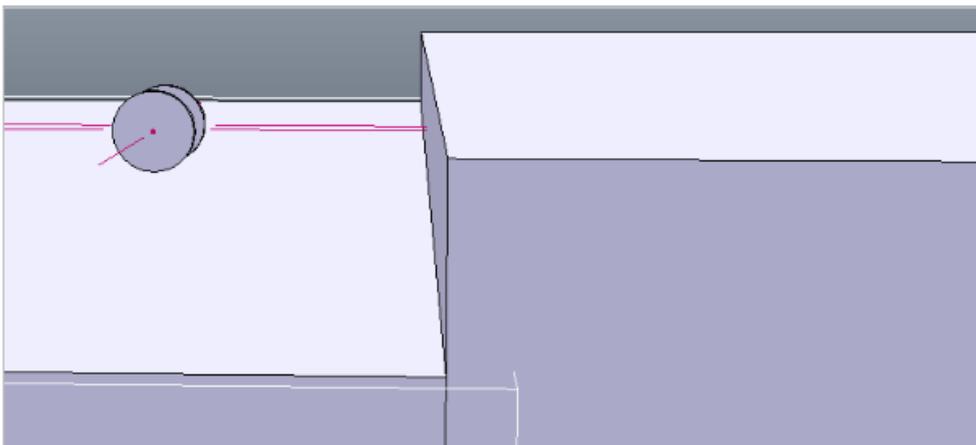


Figure 36 Simulation set up for a step. The cuboid primitive was positioned at the correct height.

The sim setup is the robot attempting the obstacle at different heights/angles. Each time the robot has 15 seconds to make it over the obstacle which changes in increments of one. The slope varies from 0 to 45 degrees and the step from 0 to 49 mm with 10 runs executed at each of these difficulties. This resulted in 300 runs for each result given in the tables below. These ranges of difficulty were determined by a few simple runs and over-estimating the maximum the robot could do. To vary the centre of mass within the wheel from a quarter of the wheel radius to half the radius, the offset mass needed changing from 0.9kg to 4.5kg.

Table 9. Which mathematical prediction were actually useful

	Slopes	Steps
Slow Traversal	✓	✗
Fast Traversal	✗	✓

The results here are estimated by adding together the number of successful runs and using that to calculate the middle of the failure region. Often, the robot went from succeeding every time to majoritively failing with a small increase of obstacle difficulty. The speed of the robot was input into the V-rep simulation as a value of intrinsic target velocity. This means that the exact speed of the actuators is non-deterministic as it is a relationship between the inertia of the wheels, torque resistance to the robot's movement and the wheel's current angular speed. This is complicated further by the unstable rock of the robot as it drives.

The results for a slope match well to the mathematical predictions. The mathematical predictions are a maximum and therefore results just below the predictions are to be expected.

Table 10. Simulation and maths results of a slope for wheeled robot.

Mass position or Tail	Frictional Coefficient μ	Simulation Target Speed			Mathematical Prediction for slow traversal ($^{\circ}$)
		1	2	3	
		Maximum Slope Angle ($^{\circ}$)			
Quarter $\frac{R}{4}$	1	10	13	14.1	14.8
	0.5	10	13	14	14.8
Half $\frac{R}{2}$	1	20	28	29	30
	0.5	20	26	26	26.6
Tail	1	20.9	21.9	21.6	21.8
	0.5	13.5	13.8	13.7	14.9

The controller used for the slope was a proportional gain controller that was tuned to the target speed of 3. This could explain why the lower speeds were not as close to the prediction.

The results for the tail were less affected by the target speed of the robot; this could be because the tail allows the movement to be more stable. It does create a triangle or balance of the two wheels and the tail. With these results matching it is promising that they may translate to the real world. Both of these results are theoretical tests that still need to be tested on actual hardware (see Chapter 5).

The mathematical results for the slow ascension of the step turned out to be unreliable. It turns out that speed was far more important than previously thought. Therefore the actual speed of the robot in m/s was calculated using the average time to complete the distance without a step. When these were plugged into Equation 10 they gave the calculated predictions in the table.

There are two possible reasons for the inaccuracy of these predictions. The speed used here is only an estimate of the speed as the robot tended to oscillate in its movement. Secondly, the method used here assumed the energy was enough to clear the entire step rather than enough to get up some of the step and for the rest to be crossed slowly. The problem is knowing the state of the robot in terms of weight positions and rotational speed after this fast transition.

Table 11. Simulated and maths results of a step for wheeled robot

Mass position or Tail	Frictional Coefficient μ	Simulation Target Speed			Mathematical Prediction for slow traversal (mm)
		1	2	3	
		Maximum Step Height (mm)			
Quarter $\frac{R}{4}$	1	15.8	22.8	39.4	2.40
	0.5	11	15.2	26	2.40
Half $\frac{R}{2}$	1	22	32.7	33.1	10.1
	0.5	11	13.7	24	7.92
Fast traversal predictions	Velocity used (m/s)	0.073	0.146	0.222	N/A
	Height (mm)	16.7	25.5	32.2	N/A

There are no simulated results for a tail given here as it was not able to traverse a 1mm step. When the simulations were observed, it was the tail that was getting stuck on even the smallest step. The curved tail was created using four convex sections that were then grouped together. An unknown property of the interaction between this complex shaped tail and the step means that it got stuck at this point. In the real world it would just slide

over, so somethings in the physics here are not being modelled properly. For this and other reasons a straight tail was adopted for the future designs of Cylindabot.

Here, mathematical limitations were derived in the form of inequalities. They were then compared to simulated results. The predictions for a slope turned out to be accurate, whereas the predictions for steps were not. For both obstacles friction was an important factor in the robot's ability, which is one of the reasons for the rubber grip added to the physical design of Cylindabot.

3.7 Changing Leg Angles

With the wheel only experiments completed, the focus turned to deploying the legs. These simulated experiments were done with the straight tailed robot, Figure 37. The legs were positioned at 10 degree increments and moved out to fully deployed then to retracted again over 280 degrees of movement.

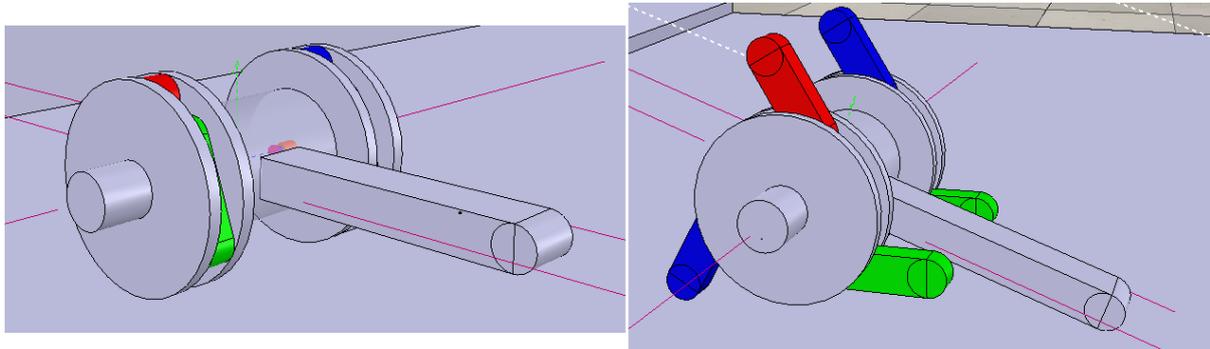


Figure 37. Simulation model with legs set to 0 degrees and 90 degrees.

There are two obstacles being tested in this analysis, a step and a slope. The reason for these two obstacles was that it simplifies the problem into two parts. If a robot is going across a smooth surface, then it is the steepest part that determines whether it is capable of being traversed. Similarly, on rough terrain there will be discrete jumps between surfaces which here are to be represented with a single step.

The simulation is done in a V-rep simulator using the ODE physics engine which was chosen due to reliability when using a cylinder on its side. Most of the variables used are set to the default; the ones that were altered will be explained here. Firstly, the mass and inertia were increased to allow the robot to move more stably. The torque of the leg motors was increased so that they could handle the forces being applied to them. The robot was tested with the legs at different angles from 0 to 280 degrees (Figure 38). At each end of this rotation, the legs are retracted inside the wheel and at 140 degrees the legs are fully extended.

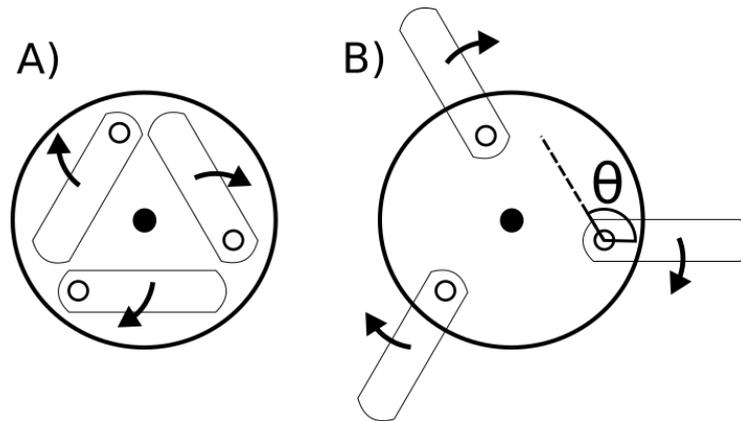


Figure 38. Leg angle movement. A) Legs at 0 degrees. B) Legs at 120 degrees with θ denoting the leg angle

The ODE physics engine was used, as initial tests of a cylinder showed it to be the most stable of the available physics engines. This was determined by trial and error with the physics engines and observing the behaviour. With the other engines, when a cylinder was placed on its sides it would increasingly rock back and forth then finally roll off the screen. One feature that was built into this design was only using grouped primitive shapes as the interactions with them are more stable and reliable. This is instead of using more complex shapes. Grouping shapes together in V-Rep means that they act like a single object, both for the hierarchy and physics interactions. The legs, for example, are made from two cylinders and a cuboid. Another feature of simulations like this is they do not handle concave shapes well. The final simulation setup was to make the step or slope static and give the robot a shorter amount of time to traverse it.

At the start of these simulations the hypothesis was that legs would help the robot climb a step and that they would hinder going up a slope. The robot used at this point had a main body part that was 12 cm wide and a straight tail that was 20 cm long. This is relevant as both were varied later on in this work. The legs are extended from the three points in the wheel and their position is determined as an angle relative to a fully retracted position. All the legs on each wheel are retracted or deployed to the same angle, as then in the real robot only one actuator would be required to move the three legs.

The controller for this tried to get the robot to head in the direction of the obstacle by using differential speed of the wheels and a proportional controller. If the robot ended up flipping upside down, it would invert the direction of wheel motion and turn the robot towards the obstacle instead of putting itself the original way up again. The symmetric design means that it will function the same either way up, however the direction the legs are facing will be opposite. The desired angle for these legs is set at the beginning of the run and does not change while the robot is moving.

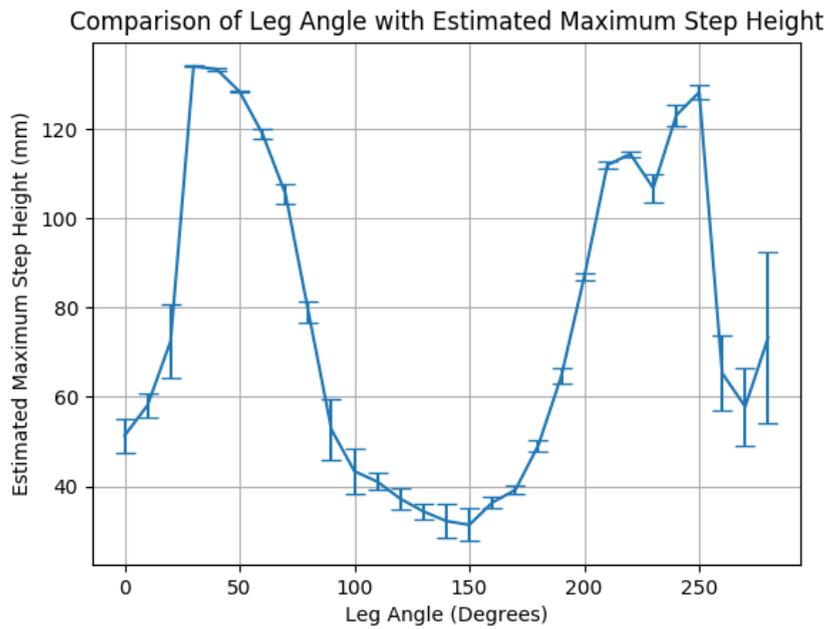


Figure 39. Simulated result of the straight legged robot attempting to traverse a step obstacle. Each point from 1200 simulation runs. Error bars calculated by the difference between estimated maximum height and the mean doubled.

The symmetry in Figure 39 is to be expected as the legs move from retracted through to fully deployed to finally being retracted at the other end of its range of movement. What was not expected was the big decline in the middle. When watching simulation runs, it turns out that the robot was often falling over to the side or over backwards.

Table 12. Detailed results for a step obstacle

Leg Angle	Successes	Time-outs	Falls	Estimate Max Height (mm)	Mean doubled (mm)
0	338	862	0	51.33	55.12
10	369	821	0	58.17	60.87
20	465	734	1	72.50	80.69
30	835	342	23	134.17	134.28
40	831	362	7	133.50	133.74
50	801	393	6	128.50	128.69
60	744	446	10	119.00	120.20
70	663	530	7	105.50	107.60
80	504	670	26	79.00	81.39
90	346	812	42	52.67	59.48
100	290	882	28	43.33	48.34
110	276	909	15	41.00	42.90
120	253	936	11	37.17	39.60

Leg Angle	Successes	Time-outs	Falls	Estimate Max Height (mm)	Mean doubled (mm)
130	236	953	11	34.33	36.06
140	223	966	11	32.17	35.92
150	218	971	11	31.33	35.00
160	248	935	17	36.33	37.58
170	265	920	15	39.17	40.04
180	325	857	18	49.17	50.43
190	418	760	22	64.67	66.36
200	552	640	8	87.00	87.90
210	702	497	1	112.00	112.69
220	717	468	15	114.50	115.16
230	617	515	14	106.83	109.90
240	769	399	32	123.17	125.49
250	800	347	53	128.33	129.81
260	422	768	10	65.33	73.77
270	377	812	11	57.83	66.58
280	470	718	12	73.33	92.53

3.6.1 How does the robot fall on a step?

There is the question of what caused this dip in step performance when the legs were deployed. To explore this, several step heights were observed in the simulation with the legs deployed. In this region of dip, the robot often failed to get up the step in a similar manner.

This dip occurred in one of three scenarios:

1. The legs would gain purchase on the step and then the robot would flip over backwards.
2. Once up the step the weight of the robot would be on the tail. The robot body was above the step with the legs barely gripping the step due to the weight being on the tail.
3. The main body of the robot would make it up the step but the tail was preventing it from completing the obstacle. The tail was at a steep angle and the robot was not able to get enough purchase on the step to pull it over.

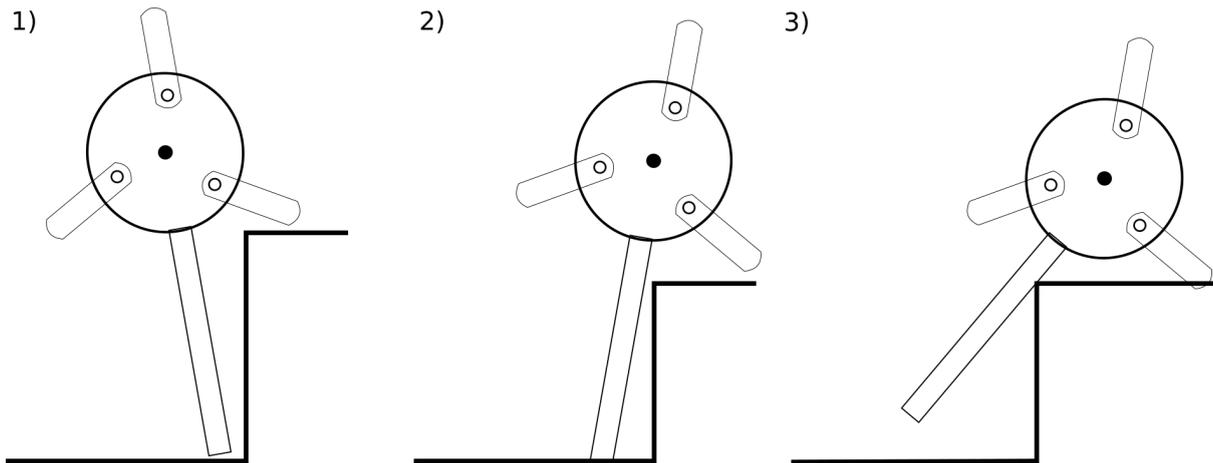


Figure 40. Scenarios of robot climbing a step.

To use the terminology from [2], it is failing in the surmount phase rather than the approach phase. This means that the robot is able to hook its legs onto the step but not make it onto the top of the step. The key factor of importance for all three scenarios of failure is the tail and more specifically, the length of the tail.

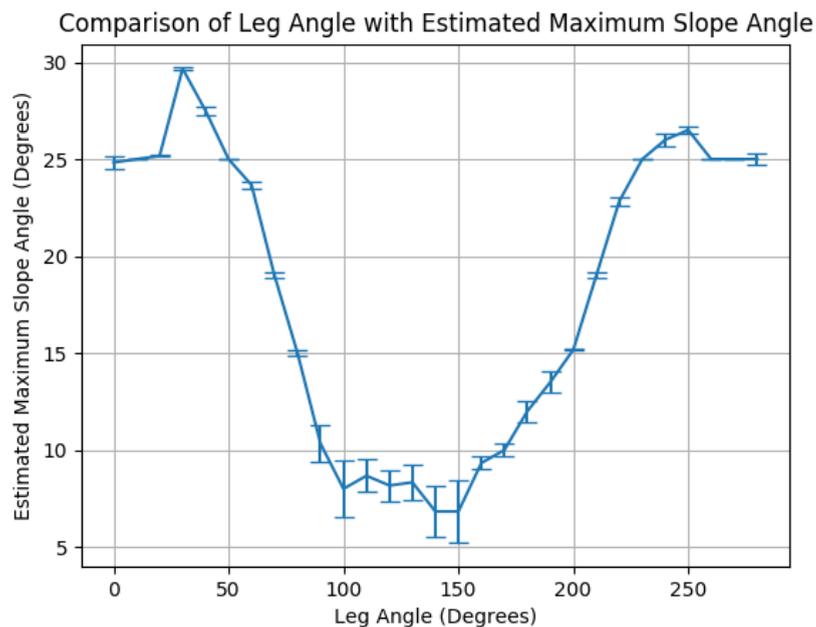


Figure 41. Simulated result of the straight legged robot attempting to traverse a step obstacle. Each point from 80 simulation runs. Error bars calculated by the difference between estimated maximum height and the mean doubled.

A similar result for the slope occurred with a dip in the middle of results, Figure 41. The key difference here is that using legs was significantly worse than not using them. At this point it was realised that maybe increasing the width of the body of the robot and the length of the tail might allow the robot to become more stable. The changing tail length was then applied to the task of ascending an inclined slope (Section 3.8). The main reason

for the reduced number of simulation runs is the number of increments needed to be used for slopes. The slopes were increased by 5 degrees on each run and each difficulty was tested 10 times. The maximum slope angle that was tested ended up being 35 degrees. Giving only 80 simulation runs per point in the data.

Table 13. Detailed results for a slope obstacle.

Leg Angle	Successes	Time-outs	Falls	Estimate Max Slope Angle (°)	Mean doubled (°)
0	59	31	0	24.5	25.42
10	60	30	0	25	25.00
20	60	30	0	25	25.00
30	70	14	6	30	30.00
40	64	17	9	27	27.19
50	60	23	7	25	25.00
60	58	15	17	24	24.14
70	48	32	10	19	19.17
80	40	39	11	15	15.00
90	30	52	8	10	10.67
100	26	60	4	8	8.85
110	26	63	1	8	8.85
120	25	64	1	7.5	8.40
130	26	63	1	8	8.46
140	25	64	1	7.5	9.60
150	27	61	2	8.5	9.26
160	28	59	3	9	9.29
170	28	60	2	9	9.64
180	33	54	3	11.5	12.42
190	38	52	0	14	14.47
200	41	44	5	15.5	15.61
210	48	36	6	19	19.17
220	57	25	8	23.5	23.68
230	60	26	4	25	25.00
240	62	27	1	26	26.13
250	63	23	4	26.5	26.67
260	60	30	0	25	25.00
270	60	30	0	25	25.00
280	60	30	0	25	25.00

3.8 Size Analysis

Here the relative size of Cylindabot is explored. The size of the wheels are kept as a constant. The width of the robot and the length of the tail are varied as shown in Figure 42. The robot leg angle is tested from legs fully retracted (angle = 0), to fully deployed (140) and retracted in the opposite direction (280).

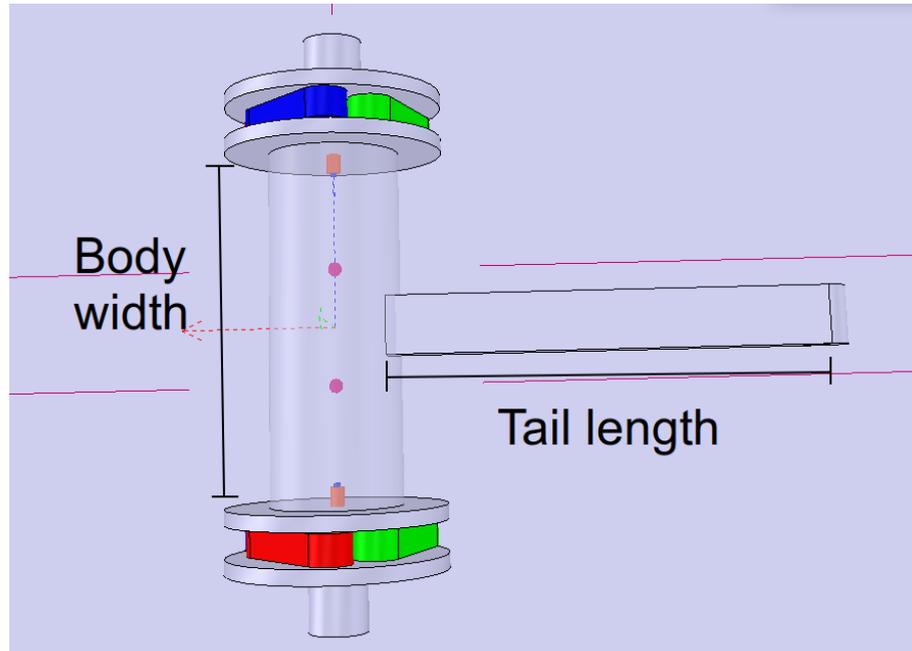


Figure 42. Dimensions to that were varied in the course of this work. Body width is the size of the main cylinder in the middle of the robot. Tail length is the length of the cuboid than is then grouped with a cylinder to make the tail

3.7.1 Width Comparison

The first attempt to improve the design was to make the main body of the robot wider to prevent it from falling over sideways and inverting the robot. Thanks to the symmetric design when the robot flips upside down, all that needs to be changed is the direction the wheels turn in. The original length was 12 cm and the two other lengths tried were 18 and 24 cm which were chosen to give a ratio of scale +50\%-100\%.

Figure 43 shows two differences; firstly a marginal improvement in the middle of the leg angles and secondly a decrease in the step height at large leg angles. The improvement in the middle is believed to be caused as the wider body makes the robot more stable when legs are fully deployed. Interestingly, there is a switch of position at the end of the graph due to the robot being less stable when narrow. Once the legs are fully deployed, if the robot is narrower it can flip over. Then if flipped over it acted like it was again at a small angle 30-50 degrees and was able to catch on the step again.

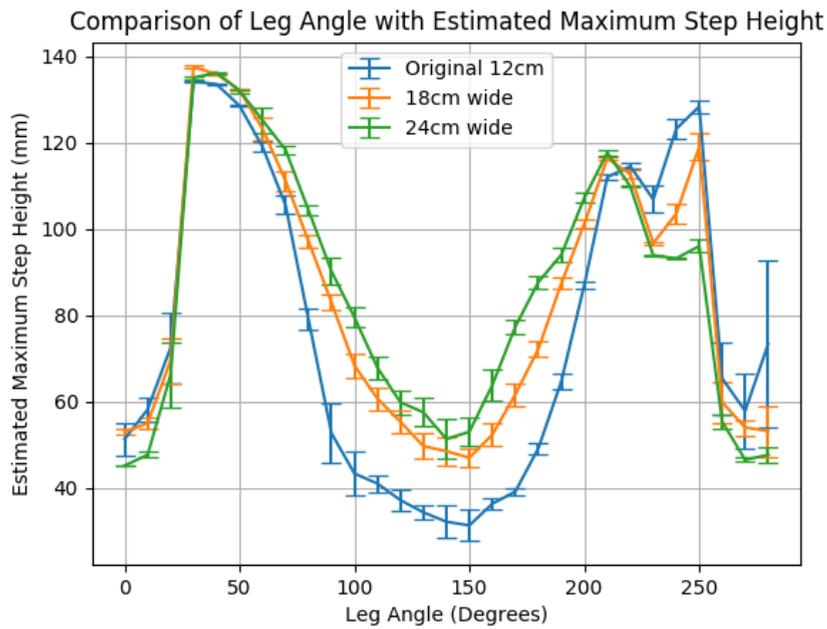


Figure 43. Width variation of the robot on a step obstacle. 30 simulation runs for each point using a fixed height algorithm.

For the reasons of stability the length of 24 cm was used for later simulations. This would also give more space for electronics/actuators in the centre of the robot and physical support to handle torque applied to a longer tail. The need for a longer tail for step traversal is laid out in the next section.

3.7.2 Tail Comparison

The changing tail length was then applied to the task of ascending an inclined slope. The graph in figure 44 shows that the longer tail reduces the unstable failure when the legs are deployed. The difference is that the legs do not have a significant effect on the robot's ability to climb the step. In fact when fully extended, they still have a negative effect on the estimated maximum slope the robot can climb. The reduction in slope ability is because the movement is more unstable and the legs move the centre of mass further from the slope.

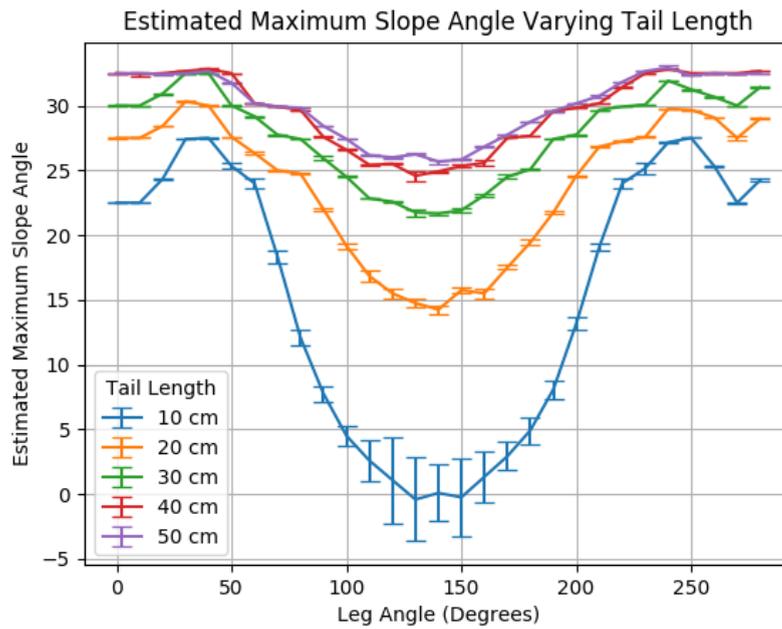


Figure 44. Slope performance while changing leg angles and tail length with robot width of 24cm. Fixed height algorithm with 30 simulations per point and increments of 2.5 degrees.

The increase in the tail length when combined with a wider body had a dramatic effect on the robot's obstacle ability to climb a step. When the legs are fully extended a peak in the performance is seen rather than a lull. In fact, the maximum obstacle height for the fixed positions had to be increased to allow for this.

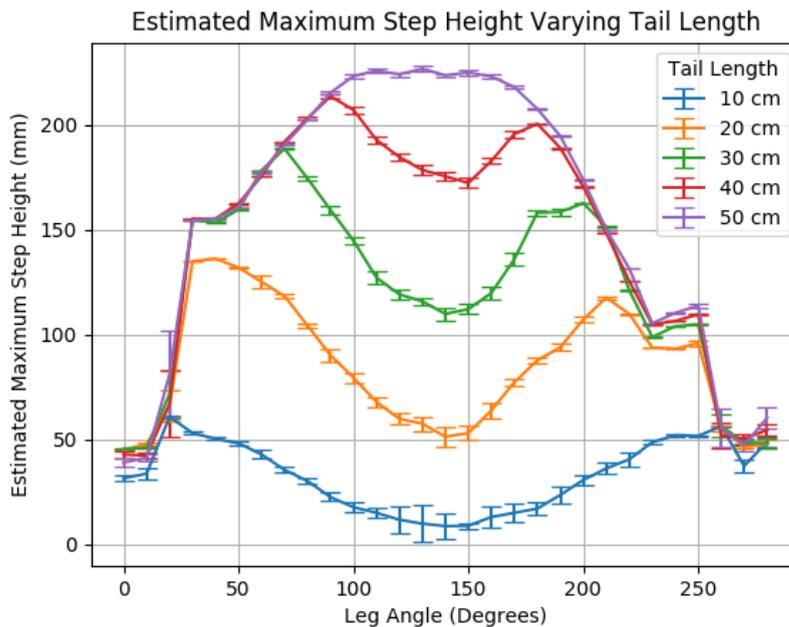


Figure 45. Step climbing ability when varying the tail length for the straight tailed robot with width of 24cm. Fixed height algorithm with 30 simulations per point and increments of 5 mm.

3.7.3 Optimal Size and Comparison

The next challenge is how to fairly compare these different sizes of robot. Making the robot bigger has been shown to improve its obstacle traversal. To find an optimal design, the size of each robot needs to be taken into account; for this the height and bounding box of each size has been used (Table 14). The bounding box describes the volume of a cuboid box that would be required for the robot to fit in. This will allow a better comparison between the different widths and tail length. In the literature, just the height of the robot is used for this type of comparison [13], [37], [98]. This would not allow for a differential comparison as the body width and tail do not affect the height.

Table 14. Size optimisation comparison

Tail length (cm)	Max Step Height (mm)	Leg Angle (°)	Bounding Box (m^3)	Step Ratio (step/robot)	Step Height/ Bounding Box (m^{-2})
10	60.8	20	0.0165	0.41	3.7
20	136.2	40	0.024	0.91	5.7
30	188.7	70	0.0315	1.26	6.0
40	213.7	90	0.039	1.42	5.5
50	226.8	130	0.0465	1.51	4.9

The penultimate column takes the maximum step height and divides it by the diameter of the wheel to give the step height ratio. A similar comparison was done with the slope results which yielded that a shortened tail was considered optimal. The longer tail was used when legs were deployed which only helped with steps. What it did not take into consideration was the number of failures when legs were deployed.

Several examples of robots with transformable wheels can be found in scientific literature. One of the most recent is the WheelLeR robot [9] which deployed a passive leg system. The drive of the motors was applied to a cog so that the legs extend when driven in one direction and retracted when driven in the other. A passive mechanic is applied in [66], a catch that protrudes from the wheels is used to release the legs when the wheels are driven in the correct direction. In the Impass robot [37] the legs are actuated linearly through the axle rather than the rotation used in this work and achieve the highest step ratio of the robots discussed here. The LEON robot [12] has two wheels that transform into legs and augment four other legs to allow it to walk as a hexapod. Lastly the Whlegs robot [99] has four Whlegs made up of three legs in each Whleg that are represented in

simulation by cuboids. The fact these legs do not deploy does put it at a disadvantage when it comes to the step ratio comparison.

Table 15. Comparison with other platforms

Robot	Step Ratio	Slope Angle
Wheeler	1.2	15
Wheel Transformer	1.63	-
Impass	1.73	-
LEON	0.53	35
Wheg	0.52	30
Cylindabot (V2)	1.5	32

In Table 15, the step ratio is calculated by taking the maximum step that can be conquered, divided by the robot's height (legs not deployed). The results show that this design keeps up with the abilities of other robots currently developed. Interestingly none of the transforming wheel designs were simulated. This is potentially because of the need for concave shapes which complicates the physics interaction that would occur.

3.7.4 Remapped Data - Tail length comparisons

Given the symmetric data from the size analysis graphs, the data was remapped to use the leg radius rather than the leg angle. This means that a leg angle of x would be plotted in the same horizontal position as $280 - x$ as they have the same radius. The difference between these two angles is whether the leg is angled towards or away from the way the Wheg is rotating. In Figure 46 the two slope curves line up which means the direction of leg makes little difference. For shorter tails, partially deploying the legs has a positive effect on climbing a slope and then a negative effect once deployed further. Once a longer tail is used this increase vanishes and fully deploying the legs had less of a detrimental effect. The tail length is a more important factor for climbing slopes than leg deployment.

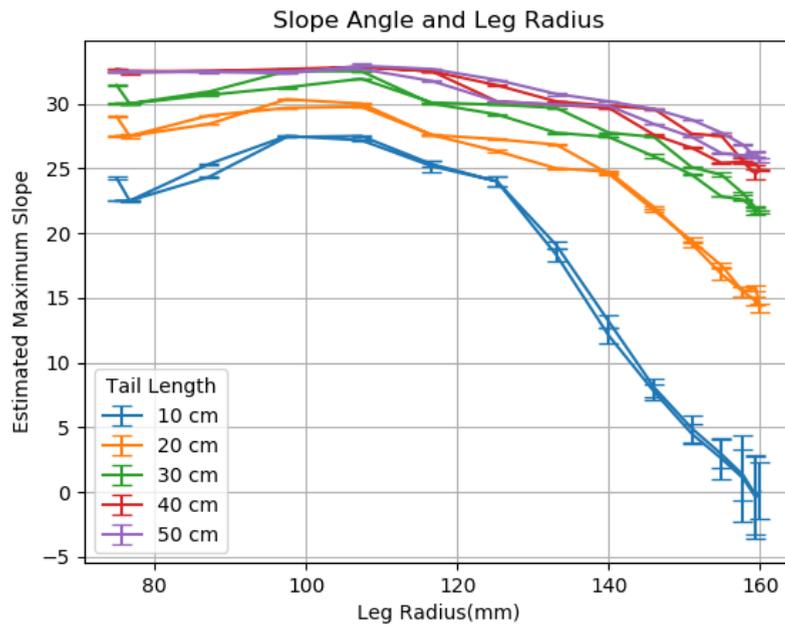


Figure 46. Remapped slope data for straight tailed robot with leg angles replaced with leg radius. Fixed height algorithm with 30 simulations per point and increments of 2.5 degrees.

When this same remapping is applied to the step data, the direction of the legs does have a noticeable effect. Cylindabot is more capable of climbing a step when the legs are angled into the step. This is because the foot of the leg is able to hook onto the top of the step. Whereas this is not possible when angled away from the step.

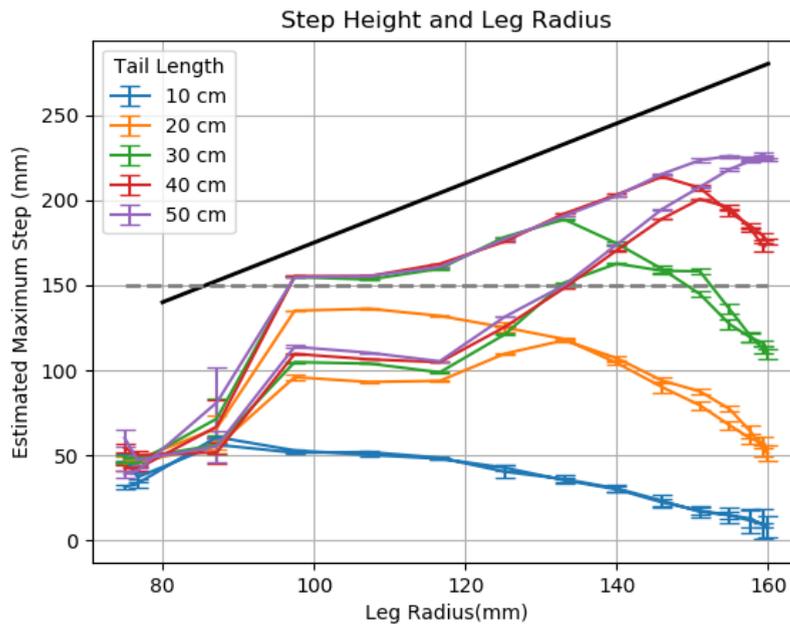


Figure 47. Remapped step data for straight tailed robots with leg angles replaced with leg radius. Fixed height algorithm with 30 simulations per point and increments of 5 mm. The dashed grey line is the height of the robot and the black solid line is the geometric reach of the legs.

3.9 Summary

This chapter has detailed the simulation environment used throughout this research for modelling and testing ideas. CoppeliaSim aka V-Rep was chosen to simulate a model of Cylindabot because of its ease of use and integration with physics engines and languages. A method of shrinking obstacles was unsuccessful for testing a robot's ability to move around an environment. Instead the obstacles were fixed at a range of heights and an estimated maximum difficulty was calculated.

Mathematical inequalities were derived from mechanical dynamics for a wheeled only version of Cylindabot. Simulated data was applied against mathematical predictions for slope and step. These predictions were only accurate for slopes and not for steps. Frictional coefficient was important for both and therefore grip on the wheels needs to be maximised in future designs.

The size analysis data gives important information that can be used to inform the design of the physical prototype. Firstly, the tail length is important and therefore needs further investigation. The legs should be designed to point into the step to hook onto it.

From the work undertaken and reported in this chapter, a number of physical improvements were required in the Cylindabot design, these were: Rubber was added to the legs to improve friction. A long straight tail was used to make full use of the legs when

deployed. The feet of each leg have been designed to hook onto obstacles. These features are discussed in more detail in Chapter 4. The other contribution of this chapter is the development and testing of the simulation test setup. In Chapter 5, this setup is improved using a new robot model based on the physical prototype and a beermat algorithm to improve simulation efficiency.

Chapter 4: Physical Cylindabot Design

Design process, CAD and final prototype of Cylindabot

4.1 Introduction

The Cylindabot was designed to adapt to different environments by having wheels that transform into legs (known as Whegs). Various aspects of this were inspired from other robotic platforms. At every stage of the design process, simplicity and structural strength were prioritised to make a robust platform that was quick to produce, assemble and repair. The version of Cylindabot described in detail in this chapter is the result of a number of iterations to produce a robot that fulfils the design requirements and functions well.

In this chapter the design of the Cylindabot will be explained and how the design progressed into its final version, as well as thoughts on possible future improvements. First, the premise of the robot and its requirements. The chapter continues by outlining where certain aspects of the platform were inspired by previous work in the field. The scale models of the Cylindabot and the initial Whieg design are then described and lessons learnt before construction of a full size robot was achieved are highlighted. The chapter finishes with an analysis of the Cylindabot part and whole design.

The development of the Cylindabot facilitates adaptive and intelligent design while minimising the number of actuators, thus simplifying the system and maintenance. Being able to change its mode of locomotion means that it can adapt to the terrain. Intelligent design means that the focus of solving the movement task is on the morphology of the robot rather than the controller.

There are several reasons why limiting the number of actuators is preferable: Firstly it means the robot can be lighter which means it can survive falls, uses less energy and needs less powerful motors. An analogy for this is the Eden Project in Cornwall where they were able to use clear inflatable pillows instead of glass. The pillows were light enough to not need as much steel and in turn make the entire structure even lighter. So when fewer motors are used and the power requirements are reduced, in turn the battery can be lighter. Fewer motors and a smaller battery means the core of the robot is lighter so less plastic is required to hold the parts. All of which leads back to motors that can be lighter and less powerful. This lightweight mentality also makes it safer for deploying round humans and means the robot is a smaller payload for deployment for planetary exploration for example: “Actuators are the components that most limit the performance of legged vehicles, complicate their control, and increase their cost. “ [100]

Secondly, the control problem is simplified if there are fewer motors. Whether being deployed by remote human control or autonomously, this simplifies the system and maintenance. As a human will either have limited control or need training for controlling a more complex robot. For example, when controlling the four legged mechanical horse in [101] the operator needed regular breaks from the fatigue of controlling the machine. If the robot needs to act autonomously then the control needs to be robust enough to allow this.

Before beginning, a set of design requirements were devised. These lay out simple premises that will practically make it easier in the long run. Design requirements:

1. Mechanically robust to survive collisions - not getting stuck when upside down or breaking when it falls
2. Minimal actuation - simple for control and weight efficiency
3. Simple assembly for prototyping and repair - allows quick development
4. Manufacturable using additive 3D printing - fast prototyping
5. Maximise terrain traversal ability - able to climb over a range of terrain.

4.2 Background and Inspiration

The Cylindabot design, like any platform, was inspired by other robots by taking the best features and trying to overcome their weaknesses. In this section we have a look at existing platforms that are closely related to Cylindabot. While each design is quite capable, Cylindabot is designed to improve on them and allows scientific exploration.

The original inspiration for the platform was the Scout mini robot [49] which used cogged wheels to roll and a retractable tail to be able to jump. It was its symmetry and ability to survive falls that made it interesting. The idea was that it allowed the robot not to get stuck when it falls or jumps in an unknown environment. The main issue with the Scout robot was that its wheel diameter was only 40 mm, so small hindrances would have to be jumped over. This would lead to a slow, uncontrolled and inefficient locomotion.



Figure 48. Left to right: Scout [49], Impass [37], Whег robot [100], Wheeler [9].

The wheel/leg design known as a Wheg is from the aptly named Wheg robot [100]. The continuously rotating set of legs allows good terrain capabilities and minimal actuation. The set of legs act like individual ones landing in a repeating pattern. The R-Hex [34] used single curved legs in a similar fashion. The thin curve of these legs allowed them to be springy to take impacts of movement or falls. This feature was included in later designs of the Cylindabot legs. The problem with Wheg robot and R-Hex is the lack of a rolling mode of locomotion, so the movement is quite jerky. It is hypothesised that this will make them less capable on sloped terrain and proved for the Cylindabot in Chapter 6.

Individually activated leg operation was performed in the Impass robot [37]. The Impass activated each leg linearly rather than in a rotational manner and has impressive step climbing capabilities. This was because it could extend each leg individually to reach for the step and retract them to remain balanced when climbing over them. Each spoked Wheg had four degrees of freedom, each of which needed to be powerful enough to lift the robot. With a total mass of 50kg (without batteries or computing onboard), this meant that the actuators have to be quite strong. This is too heavy for the Cylindabot application both from a fabrication and safety standpoint. The plastic 3D printing fabrication generates parts quickly but they are not as strong as other materials. The Cylindabot is approximately three quarters of a kilogram, so if it was to lose control it would not be dangerous.

To achieve this a limit actuation, a similar approach to Wheeler robot [9] was used. It uses an internal cog attached to each leg that can either deploy/retract them or turn the full Wheg assembly. The Wheeler is probably the closest to the current Cylindabot platform. This design provides the best ratio for the internal cog and the legs. In terms of mechanical design it is significantly smaller than Cylindabot. The motors are directly connected to Whegs which creates a mechanical weakness and Cylindabot has a much improved clearance due to its large wheels. The Wheeler robot is controlled by an arduino which means it lacks the processing power and has a small battery, both of which would limit it in autonomous applications. Although Cylindabot is still a proof of concept, upgrading its battery and processing power now, means that adding sensors and navigation control in the future would not require a redesign.

4.2.1 Novelty

It is important to state why this design is novel compared to what already exists;

- The composite rubber belt over plastic legs for improved grip. Wheeler was able to have the grip included in the print process and others have it attached. The friction fit of the ridged rubber is unique
- The curved springy legs to absorb impacts and falls. Inspired by the RHex legs, this is the first time impact absorption has been used on deployable legs

- Aluminium tube used as an axle which allows for sensing on the far side of each Whег without the need for a slip ring. The tube acts like an electrical conduit that wires can be routed through
- The ability to lock the legs and tail to allow experiments to be consistent and repeatable
- Herringbone gears provide smoother transmission, and due to 3D printing there is no additional complexity to manufacture.

4.3 Initial Designs

There were two versions of the initial design which were tested in simulation and had scale models built for each. Using what was learnt from these, the main design was conceived.

4.3.1 Version 1

The original design of the cylindabot was to have the legs rotate out from the outside of the wheels on each side. These were to be deployed by either servos or motors. To facilitate this, the wheels would still rotate around a tube to allow power and control to make it to the outside of the wheel. This connection would then pass through a slip ring and then servos would deploy the legs. At this early stage it was undecided whether it would deploy them individually or together.

The tail was planned to be a solid arc that wrapped around the main body of the robot. It would then rotate out by 180 degrees using a servo. In early simulation tests, it was realised that it was too short to allow use of the legs on anything other than a flat surface. When the legs plus tail were deployed, as the Whегs turned the robot would be lifted up to the point the tail and legs were close together making this movement unstable.

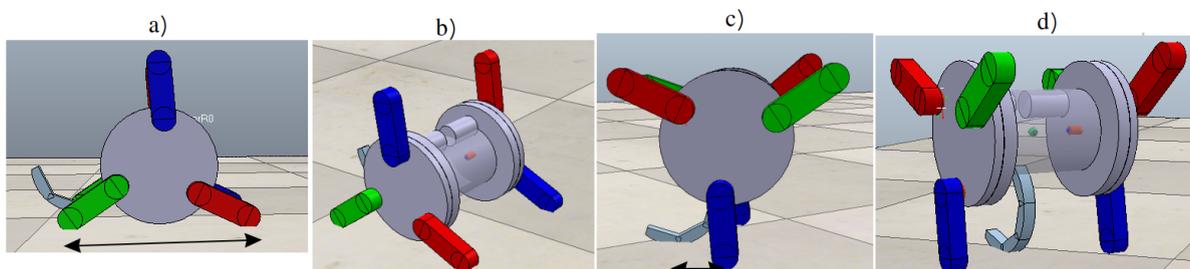


Figure 49. Version 1 of the robot in simulation. a) & b) show that the robot is stable as there is a long distance between the legs and tail. c) & d) show a short distance between them hence the robot was prone to fall backward when traversing obstacles.

4.3.2 Version 2

Therefore this tail design was abandoned for the straight tail that came from the centre of the robot. The straight tail was able to be significantly longer however it has the drawback making retracting the tail within the core of the robot more complex. If the tail remained rigid then as it is retracted it would then protrude out of the other side of the robot. The only way the tail could be retractable without this is with some sort of collapsible mechanism.

The legs, instead of being on the outside of the wheel, were sandwiched between two thinner wheels. The main advantage of this 'sandwich' Whieg design was mechanical strength as the axis of the leg rotation is held at both ends. This significantly reduces the torque required to withstand the impacts of a Whieg movement gait. The main problems with this design were the wheels required a lot of plastic once scaled up and once fully deployed the leg contact angle with a surface meant it rarely gripped the surface.

A comparison in simulation is made in Chapter 6 between this version and the final design. The new design outperformed this one for a grid obstacle of size 5x5 height points for both slopes and steps. A grid obstacle is defined by a matrix of heights over a one metre square.

The scale models were designed to realise the design into a physical object. They were both produced on a 1:5 scale so the wheels had a diameter of 3cm. All points of rotation were achieved using screws which the plastic part rotated about.



Figure 50. Scale models. On the left version one with exterior legs and a curved tail. On the right, version 2 which has the legs in a wheel sandwich and a straight tail.

4.3.3 Number of Legs

The number of legs contained within the Whieg was chosen to be three. More than three legs would reduce the reach of the legs and height of the obstacle it could then manoeuvre across as found in [9]. Two legs within the Whieg would lead to the main

body of the robot impacting the ground with each movement, making the gait quite unstable and unpredictable.

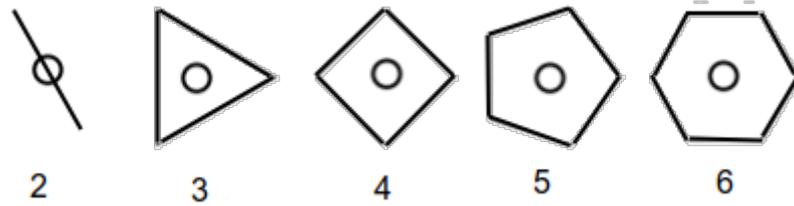


Figure 51. Different numbers of legs represented as polygons.

If the number of legs is tackled geometrically in a similar way to the approach theory from [2] then each Wheg configuration can be seen as a polygon (Figure 51). The approach theory is whether or not a Wheg will be able to reach the top of a step as it approaches. Each vertex of the polygon is seen as the foot of a leg. Once viewed like this an analysis of reach ratio, impact ratio and protected area can be found. The reach is the height that could be reached by the legs divided by the diameter of the circumcircle. The impact ratio is the height of the fall caused by movement divided by the radius of the circumcircle. Finally the protected area is the proportion of the circumcircle that is protected from impacts by the Wheg polygon. A circumcircle is the circle that passes through all the vertices of a polygon.

Another way of looking at this analysis is to take the vertices of the polygons as points the legs rotate about. From this a ratio of leg deployment length can be calculated, which is a measure of how much reach the deployment. All the values in Table 16 are ratios and as such have no units and can be applied generally to any size of Wheg.

Table 16. Number of legs analysis. h = height of reach, D = circumcircle diameter, R = circumcircle radius, r = inscribed circle radius, AP = area of polygon, AC = area of circumcircle, l = length of polygon side

Number of legs	Reach ratio (h/D)	Impact ratio ($(R-r)/R$)	Protect area (AP/AC)	Deployment length (l/R)
2	1.0	1	0	2
3	0.87	0.5	0.41	1.73
4	0.71	0.29	0.64	1.41
5	0.77	0.19	0.76	1.18
6	0.75	0.13	0.83	1
∞	0.5	0	1	0

Which number of legs to pick off this data becomes an optimisation problem. In the case of Cylindabot, the reach was prioritised over impact and protected area. Three legs gave a reasonable value for these considerations. The final line of the table is there as a reference to how a wheel would perform on these metrics. A similar analysis was applied to the Whег I robot in [100] and found three legs also to be optimal.

4.4 First Whег Assembly

The first full Whег assembly prototype was made to physically test out the design before implementing it to a full design (Figure 52). All the pieces were single extrusions apart from the interlocking gear. Two interlock gears slide together at the centre of the design and transfer force from the motors to the middle of the Whег. When the leg gear rotates round these interlock gears, they deploy or retract. Each leg was 10mm thick and the Whег forks were 5mm thick, this spacing was kept in later designs.

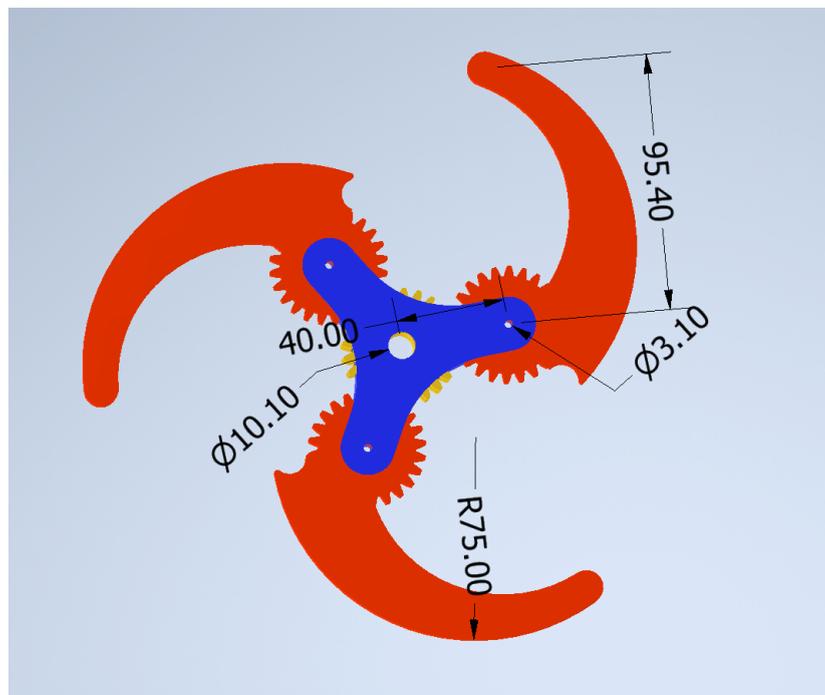


Figure 52. Technical drawing of initial Whег assembly. Red: legs, blue: Whег forks and yellow: interlock gears. Spur gears were used at this stage.

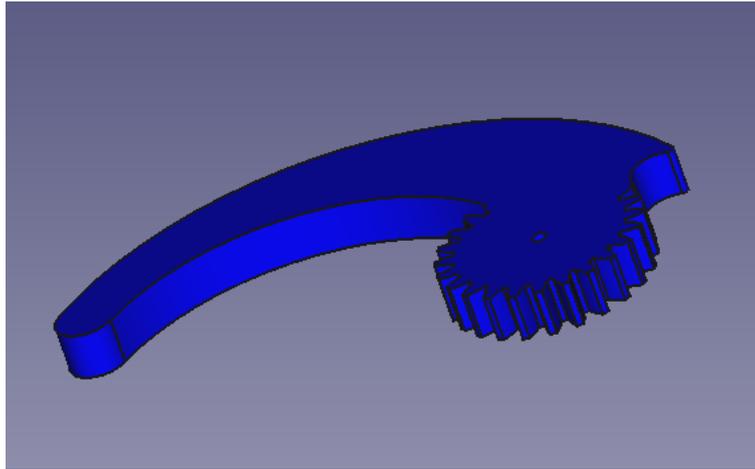


Figure 53. Single first Whieg CAD design, generated from a single extrusion of a spur gear and leg arch.

When printing there was a skirt left on the plastic which normally was easy to remove. However in this case it was around the edge of the gears and the large surface area made it quite time consuming to remove. These spur gears even after this did not move smoothly enough hence the later change to herringbone gears.

The interlocking gear translates drive to the centre of the Whieg assembly so that they can be deployed and retracted. It took several iterations to get the space and the friction fit of these to work properly. Even though it was only a single extrusion, the main shape of the Whieg included a gear, the leg and a pockets on the other side. There are 24 teeth on the gear as at this radius; each is thick enough for torque requirement and ability to be printed.

$$\begin{aligned}
 &\text{Equation 22. Torque requirement and shear stress calculation} \\
 &Mass = 750g, \text{ Lever length} = 135cm \\
 &Static Torque = 0.75 \times 10 \times 0.135 \approx 1.01Nm \\
 &Tooth radius = 2cm, \text{ Tooth area} = 20mm^2 \\
 &Force on teeth = 1.01/0.02 = 50N \\
 &Shear stress = hr2.5 N/mm^2 = 2.5 MPa
 \end{aligned}$$

The results from Equation 22 show that a shear stress of 2.5 MPa is required and from [102] PLA plastic has the strength to handle 33 MPa. This gives a safety factor of over 13 which means it should be strong enough to handle dynamic impact.



Figure 54. Initial Whieg design, physical. On the left with legs deployed and on the right as a wheel.

The legs were two arcs meeting at a rounded point. This arc came out smooth from the printer and therefore did not provide enough grip for most applications. It was considered that added grooves to this would provide grip. This was rejected as this would only provide grip on soft surfaces like carpet or soil. It would not affect the robot's ability to gain traction on hard flat surfaces. Another problem was that the contact area was quite narrow which could mean the robot could sink or get stuck on soft ground.

The Whieg assembly did successfully deploy between legged and wheeled modes. It was possible to flick it between the two modes. The impact of stopping when retracting the legs was taken at the foot of each leg. So when in wheel mode the arc of the leg would be under more stress. Later when the thickness of plastics in these legs was reduced, a small curved wedge was added inside the gear. This acted as a second point to stop the rotation of the leg and improved strength in the wheel configuration.

Making this physical prototype of the retractable Whieg was crucial for the continuation of the design and several improvements were found. What was learnt from this design:

- Plastic parts can be considerably thinner
- Grip needs to be added to the outside of each leg
- Gear motion needs to be smoother
- Harsh impact at the end of travel
- Higher surface contact area needed.

4.5 Overall Design

To explain the overall design, the assembly has been split into three sections (Figure 55). The core of the robot which contains the control PCBs, battery and tail. The drive train that houses the motor and aluminium tube that acts as an axle. The Whieg assembly which has three legs held together by two forks and driven by interlocking gears.

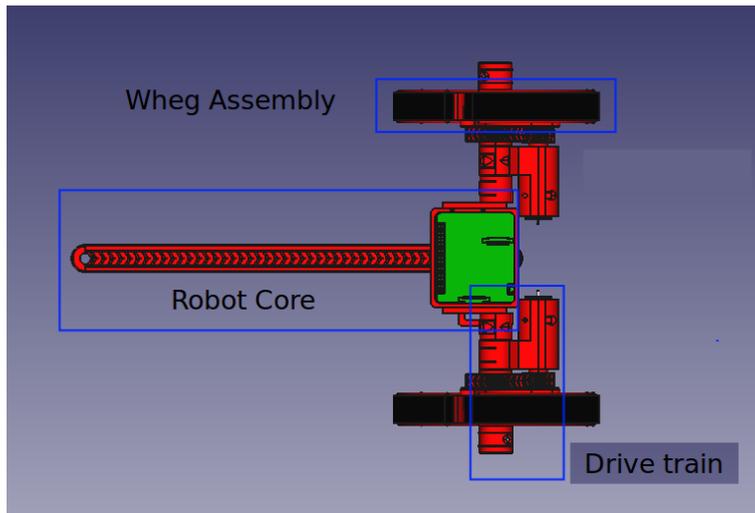


Figure 55. Top view of robot assembly in FreeCAD

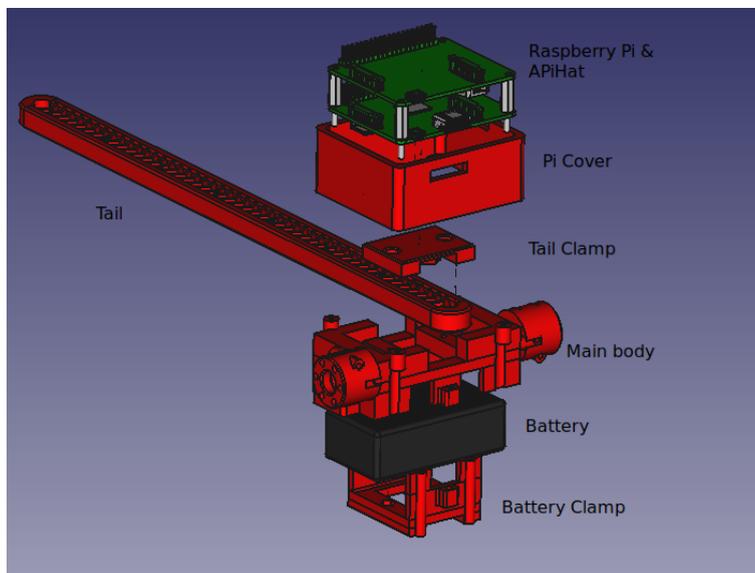


Figure 56. Exploded view of robot core

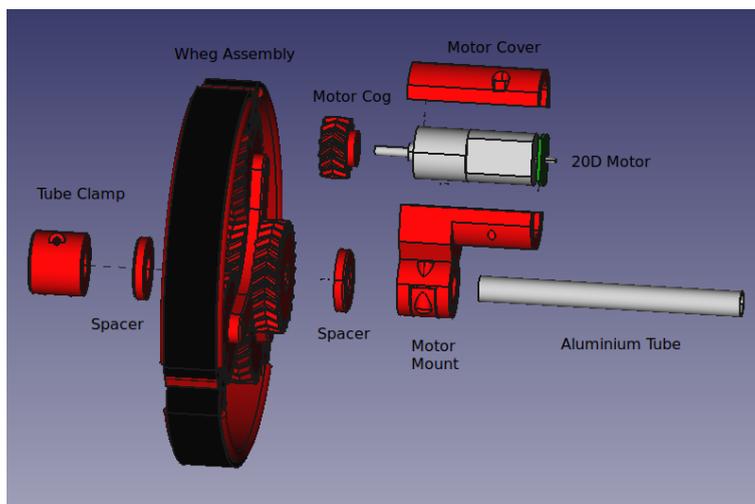


Figure 57. Exploded view of drive train

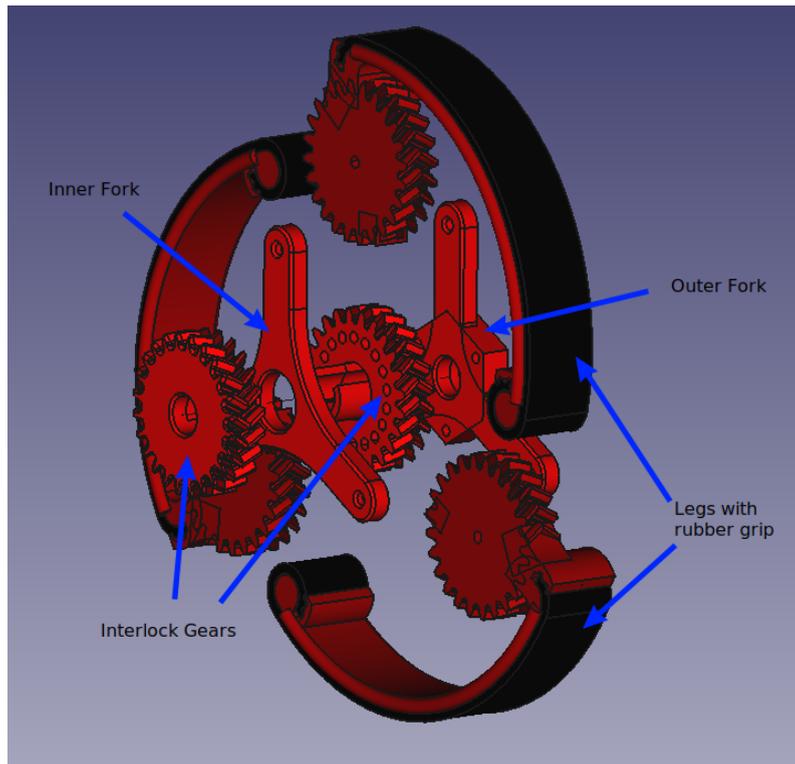


Figure 58. Exploded view of Whieg assembly

4.5.1 How do the Whiegs deploy/retract

In this section the method that the legs within the Whieg deploy or retract is explained. When deployed, the three legs are out so the robot can climb over rougher terrain, and, when retracted, the Whieg acts like a wheel. In Figure 59 the legs are retracted and about to be deployed on the left of the diagram. The drive of the motors is going clockwise which translates to the legs being pushed anti-clockwise so they deploy. On the right the legs are about to be retracted. Here the drive on the interlock gear causes the legs to collapse into the wheel shape. In both cases, if the drive of the motors is flipped it actively works to keep the Whieg in its current configuration. The gear ratio was sourced from the Wheeler robot [9] so that in theory it would deploy on flat ground. In practice it was observed that the legs only deployed when the robot was stopped by an obstruction.

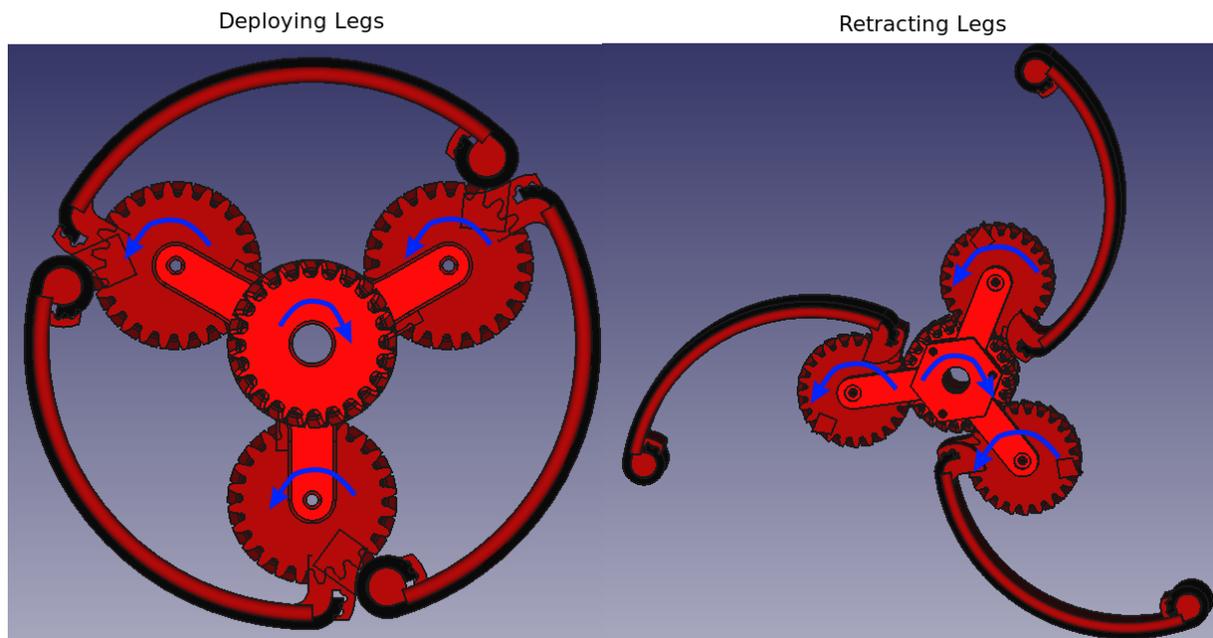


Figure 59. Diagram explaining Whieg deployment. Blue arrows represent torque on gears.

4.6. Main CAD designs (3D printed parts)

For this section the design of the Cylindabot is split into four parts; legs, tube clamps, drivetrain and main body & tail. The legs make up a third of each wheel. The tube clamps hold the robot together and to the aluminium tubes. The drive train encompasses the motor mounts and everything that is rotated by them.

4.6.1 Legs

The legs needed to act as both a third of each wheel and a leg while the robot is walking Figure 60. This required them to be strong enough not to break, have good grip with the surface and flexible to absorb impacts. The main arc of the leg is deliberately thin so as to give it a reasonable amount of flex. At the point where it meets the gear a large fillet is used as this is the point under the most torque. The Herringbone gear was imported as a single part and the rest of the leg added to this. The side of the leg is rounded and has an edge to keep the rubber grip in place. At each end the rubber bends into a slide in clamp to hold it securely. The rubber was chosen as the ridges on it would allow superior grip on the range of terrain and allow it to be held by these clamps. Without the bend in the rubber, there would be plastic in contact with ground hence slipping on smooth surfaces or low friction surfaces. At the foot end of the leg this has to make a full 270 degree bend to account for when the legs are fully deployed.

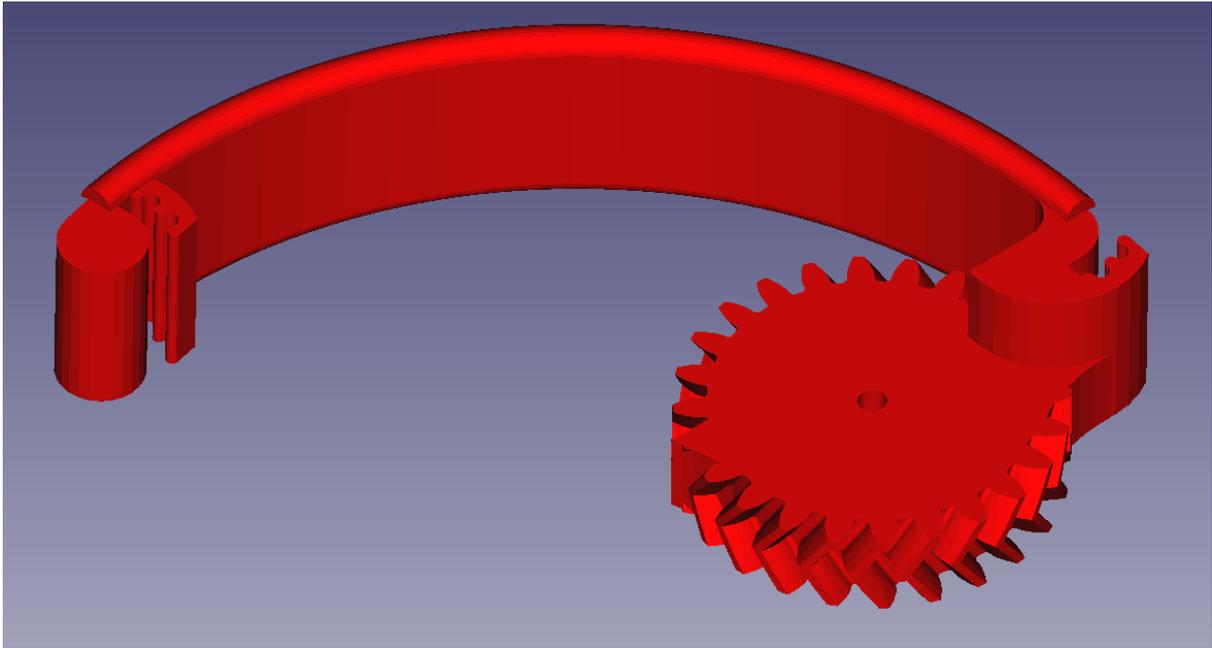


Figure 60. Single Leg, much more complex design compared to initial. Herringbone gear has small slopes added within the cog to prevent turning too far. The main leg is now an extrusion and a revolution. At each end there is a slide in clamp for the rubber grip.

4.6.2 Tube clamps

The tube clamp is a doughnut of plastic with a single screw that is able to clamp onto the aluminium tube (Figure 61). This tube is a crucial part of the structural integrity of the robot as they act as the spine of the robot. The clamps are repeated six times in the robot, three on each side. They are integrated into the main body, and the motor mount, as well as used to cap the end of each Whег assembly. The clamp works with a single M3 screw that pulls a finger of plastic that tightens onto the tube to lock it into position. Using just one screw like this makes the robot quick to assemble for prototyping and repair. The rings of plastic at either end that gives the clamp its mechanical strength and friction based fitting just needs to prevent it slipping along the tube it clamps to.

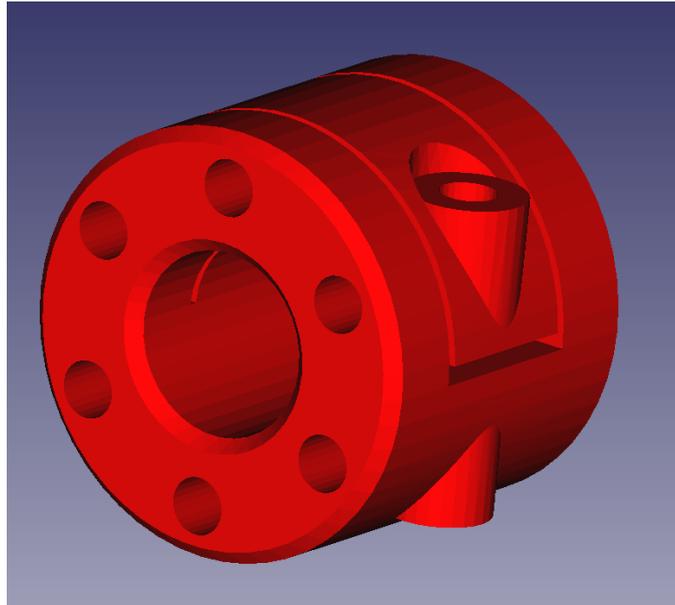


Figure 61. Tube clamp, a single M3 screw is able to tighten onto the aluminium tube using this

4.6.3 Drivetrain

Herringbone gears:

With the lessons learnt from the first Whег design, a change to herringbone gearing was made. A herringbone gear is a double helical gear that makes a V-shape along the face. The two main advantages for this design are a smoother power transfer and no side slip. The power transfer is smoother than a spur gear as the next tooth engages before the previous one has finished. The V-shape makes the slide slip of a spur gear impossible and the side forces usually attributed to a helical gear are cancelled out by the two halves.

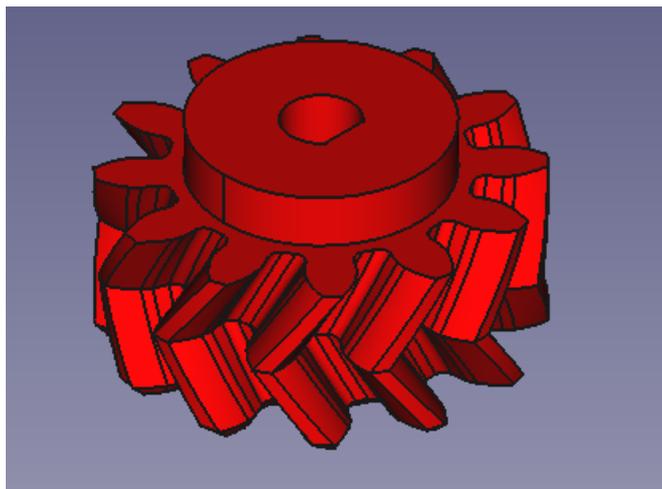


Figure 62. Motor cog, D-shaped hole to fit the motor and half the diameter for a 2:1 ratio.

The production of a herringbone gear can be quite complex, however as these designs are being 3D printed, they can be easily fabricated. There is no difference to the production time as each layer is still a spur gear. Usually 3D printing is not good for this sort of spiral

thread as the small grooves caused by different layers catch on each other. In this case though, the nature of the herringbone means there is no motion in that direction anyway.

The small herringbone gear that is attached to the motor shaft does not need to be held in place as the herringbone grooves prevent it sliding in this direction. The D-shaped hole that sides onto the motor had to go through several iterations before it fitted properly (Figure 62).

Interlocking gears:

The interlocking gears transfer the drive of the motors into the heart of the Wheg assembly and means that the Wheg is able to transform (Figure 63). On each side of Cylindabot, two interlocking gears slide together using round tongue and grooves. These tongues are 10mm long but only insert 5mm into the other interlocking gear which gives space for the Wheg forks to fit between them. These tongues are reasonably thin but all four of them would have to snap simultaneously for the system to fail.

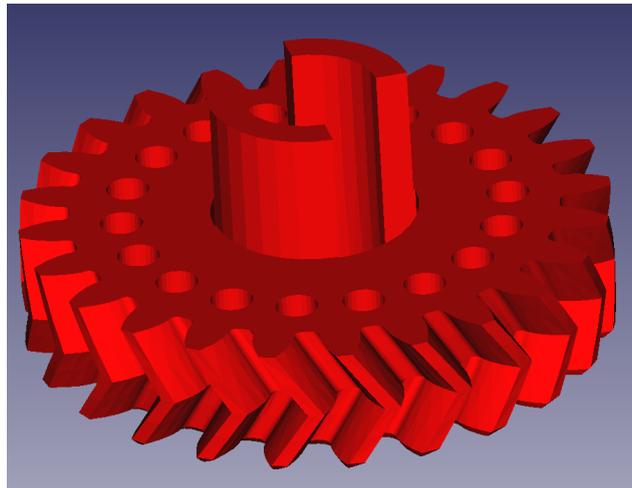


Figure 63. Interlocking gear, two interlocking tongues and multiple holes for locking the leg angle.

Motor mounts:

The motor was held in by two cup plates that screwed together (Figure 64, Figure 65). These screws turn straight into the plastic of the other half and are held in by friction. The front has a recess and hole so that a bolt can directly screw into the motor. The next section cups round the cylindrical section of the motor. The last section slides over the flat side of the motor so that it cannot rotate when activated. The far end is left open so that the encoder plate and power wire are easy to access. Although located away from the main axis of the Wheg, this mount will still have to deal with force of the motor and any back drive caused by the movement of the Wheg. For this reason the 20D motors have to be securely held by the mount and why the walls of these parts are relatively thick (3.5mm on the sides and 5-6mm at the ends).

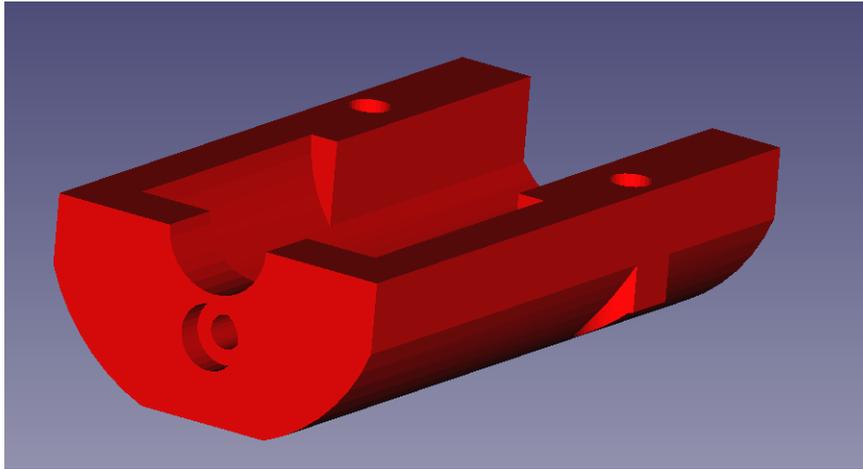


Figure 64. Motor top cover, small recess for motor front screw

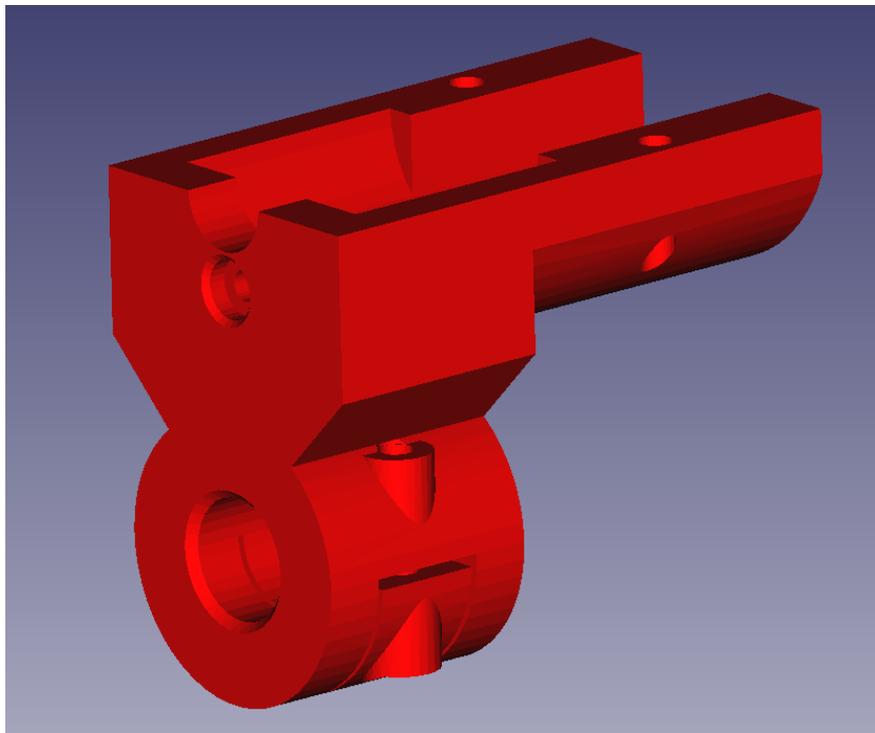


Figure 65. Motor mount, a combination of the top cover and a tube clamp

Wheg forks:

These hold the three legs in a sandwich with the interlock gear in the middle (Figure 66). The Wheg forks were reduced in thickness since the first design and, due to their geometry, are strong enough to handle the movement of the robot. The inter one has a larger hole so that the interlocking gear can fit through it.

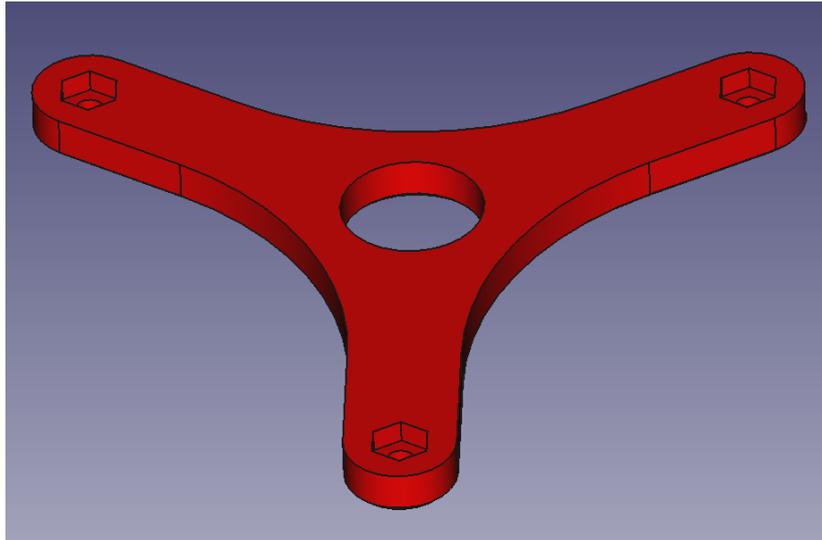


Figure 66. Inner Whег fork, wide hole to find around interlocking gear

Whereas the outer fork only needs to fit around the aluminium tube (Figure 67). The outer fork includes three holes to be able to lock the legs at different angles. This allows rigorous analysis of different levels of deployment of the legs. Although a locking mechanism was considered from [103], it was decided that a simpler design would be sufficient.

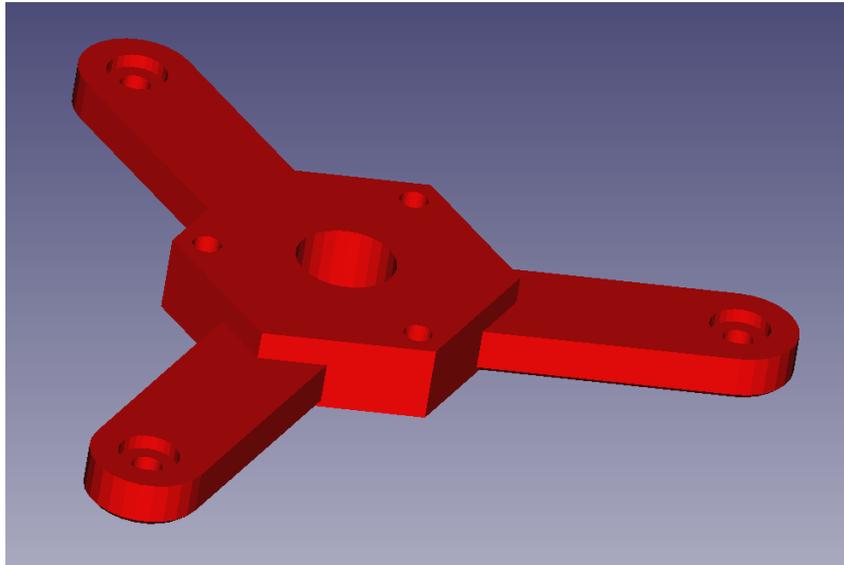


Figure 67. Outer Whег fork, smaller hole for the aluminium tube only and three holes in the hexagon for locking the legs at different angles.

4.6.4 Main body and Tail

Main body & Battery clip:

The main body acts as a nexus between the tail, battery, raspberry pi and the aluminium tubes which hold the Whег assembly. The tube clamps at either end have to take the majority of the mechanical impacts. For this reason, they have a large area of contact between them and the rest of the main body. There are a few holes to allow wiring

through from outside the Wheg assembly (Figure 68). Another small hole allows a bolt between the tube clamp on the main body to the tube clamp on the motor mount. The other five smaller holes do not pass through the tube clamp and are for plastic pins that take torque force along the axis of the aluminium tubing.

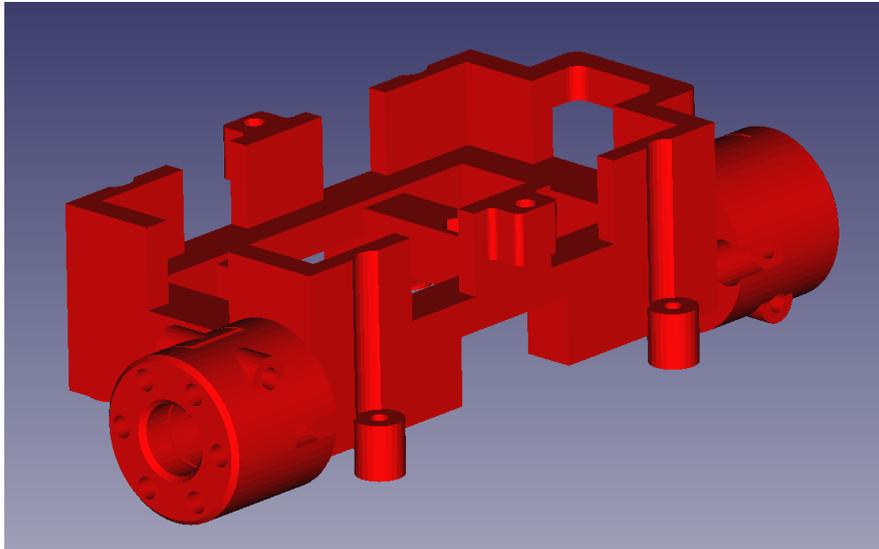


Figure 68. Main body. Longest print time and most complex part of the design. Holds the battery on top, the tail in the middle, mounts to the Raspberry Pi below and has a tube clamp on each side.

The top of the main body holds the battery which is held in by a battery clip (Figure 69). These parts slide together so only two bolts are required. There is a loop of plastic at the far end to hold the wires coming out of the battery out of the way. The middle of the main body is where the tail is attached so that it is in the centre of the robot.. The battery clip also has a secondary tail mount for earlier testing of the tails vertical position.

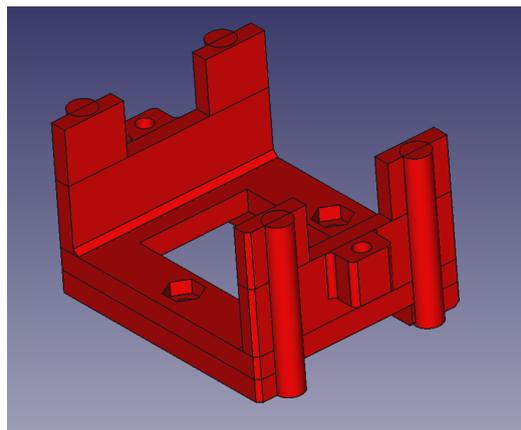


Figure 69. Battery clamp, also can hold the tail in an alternate position

At the bottom are four cylinders that are positioned for the mounting holes in the Raspberry Pi A+ and APiHat. Round grooves needed to be removed above them to allow bolts to be screwed into them. The bolts are M2.5 because of the Raspberry Pi design and this is the only place in the design where M3 bolts are not used.

Pi and APiHat Cover:

The electronics needed to be protected using this cover (Figure 70). The bottom half holds the Raspberry Pi 3 A+ and has to make space for the various ports but not access. The upper half holds the APiHat where all the motors, battery and sensors need to connect.

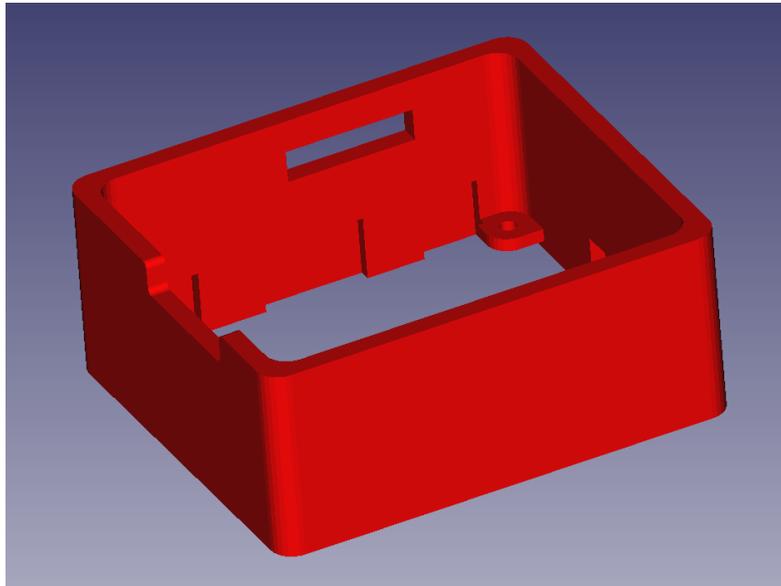


Figure 70. Pi and APiHat cover, designed to protect the Pi and hat while still giving access.

Tail and Tail mount:

The tail is as long as the print bed would allow (Figure 71). The reason for this will be explained by results in later chapters. The herringbone rack was put in so that the same cogs that move the Whegs could also drive the tail. In the final version it is simply clamped in place. Another important aspect of the tail is that it needs to not be caught on terrain as the robot moves within an environment. This is why the herringbone gear rack is recessed and the ends are curved.

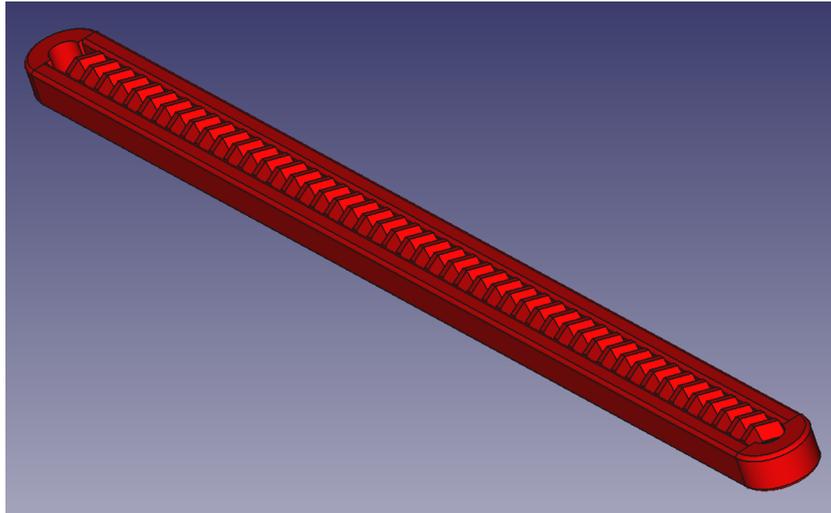
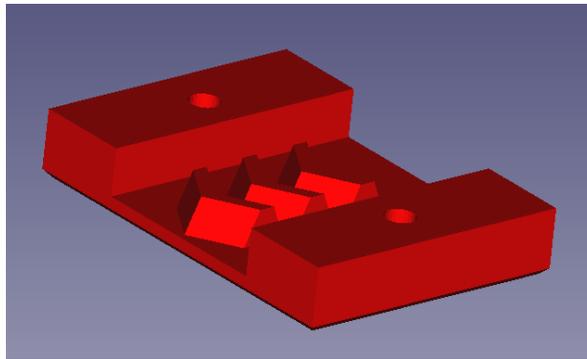


Figure 71. Tail, as long as possible given the print bed available and has nothing to catch on obstacles as the robot moves.



72. Tail clamp, hooks into herringbone gears in the tail.

4.7 Electronic Components

The electronics are all built within the centre of the robot and consist of the battery, controller, controller interface and motors (Figure 73). The battery is a 11.1V Parrot lithium battery, which is stepped down to power a Raspberry Pi 3 A+. This battery allows for long run times. The Pi is powerful enough to perform SLAM or utilise a camera for navigation. Attached to the Pi is APiHat [104] from York Robotics Lab which allows motors to be controlled and any sensors to be connected. The APiHat has a range of other features including built in Arduino, multiple I2C buses and sensor input ports. This hat streamlined the electronic design as all components became plug and play.

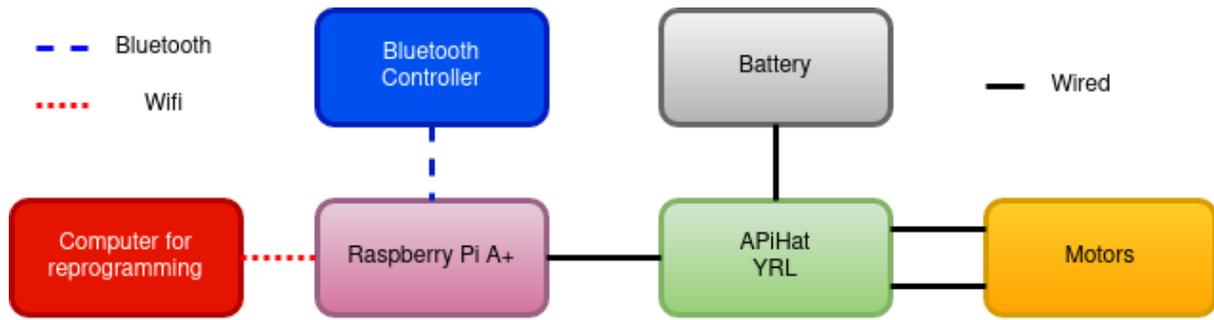


Figure 73. Electronics connections - dashed lines represent wireless connection

The robot is controlled by a Bluetooth joystick that supplies the speed of the two wheels as a Cartesian coordinate (Figure 74). The coordinates x , y were between -8 to 8 , were scaled to motor speed limits and then directly applied to motor speed for the left, right respectively. The aluminium tubes allow electrical wiring to be on either side of the Whegs without a slip ring. Without this the Whegs cause blind spots for sensors. This, combined with the improved computing power, means that autonomous development is more achievable without redesign.

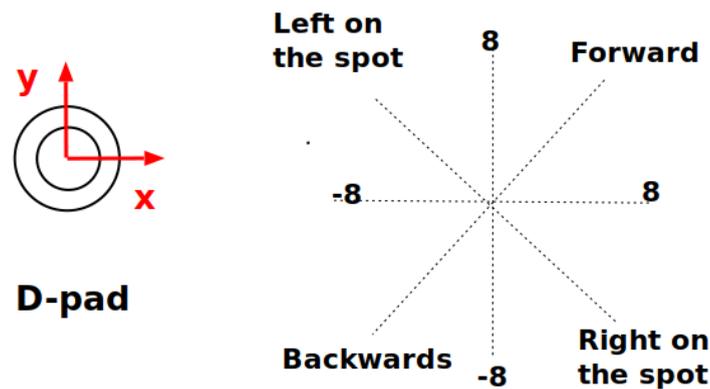


Figure 74. Bluetooth direction pad control. These directions flip if the robot is inverted when moving between different modes of locomotion

The robot initially used micro-metal Pololu motors. They were expected to not have enough torque to lift the robot but were used to test it in wheeled mode. They were soon replaced with Pololu 20D motors with a 1:488 gear ratio and a stall torque of $21 \text{ kg cm} = 2.1 \text{ Nm}$ which is roughly double the required torque calculated in Equation 22.

4.8 Complete Robot

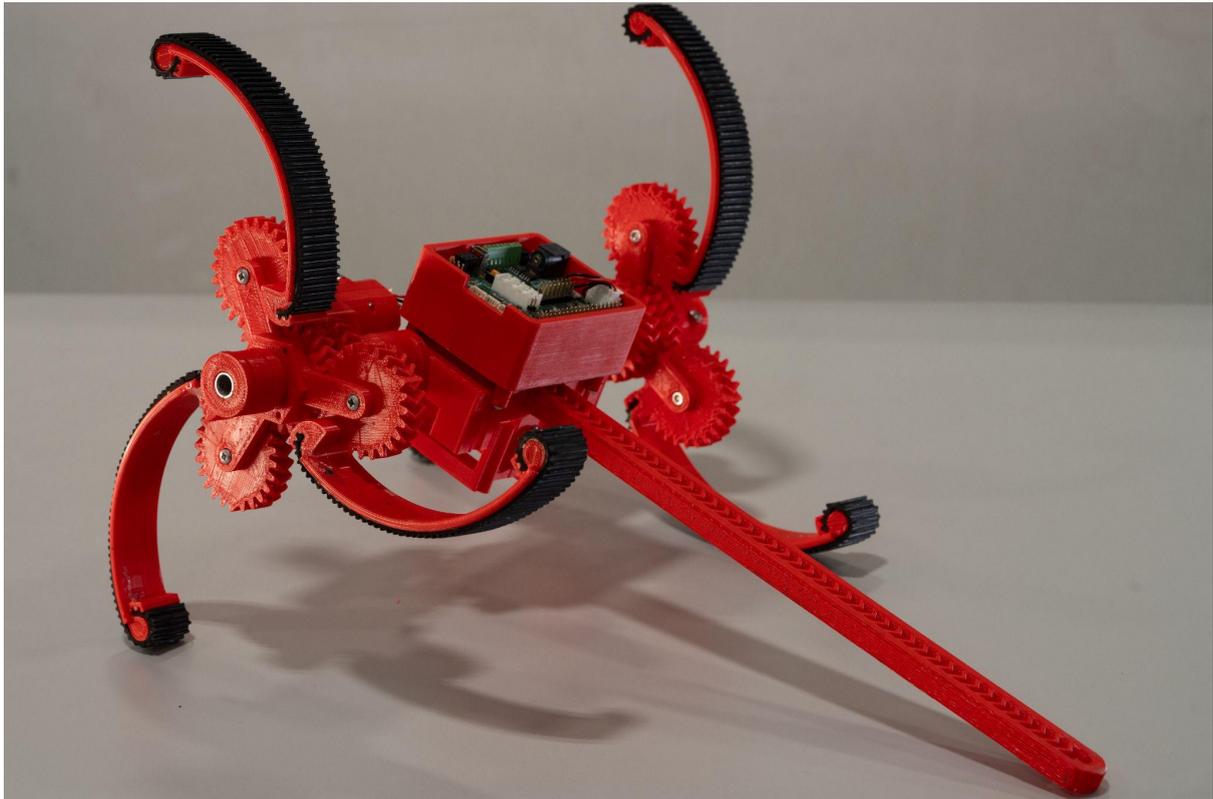


Figure 75. High quality image of Cylindabot.

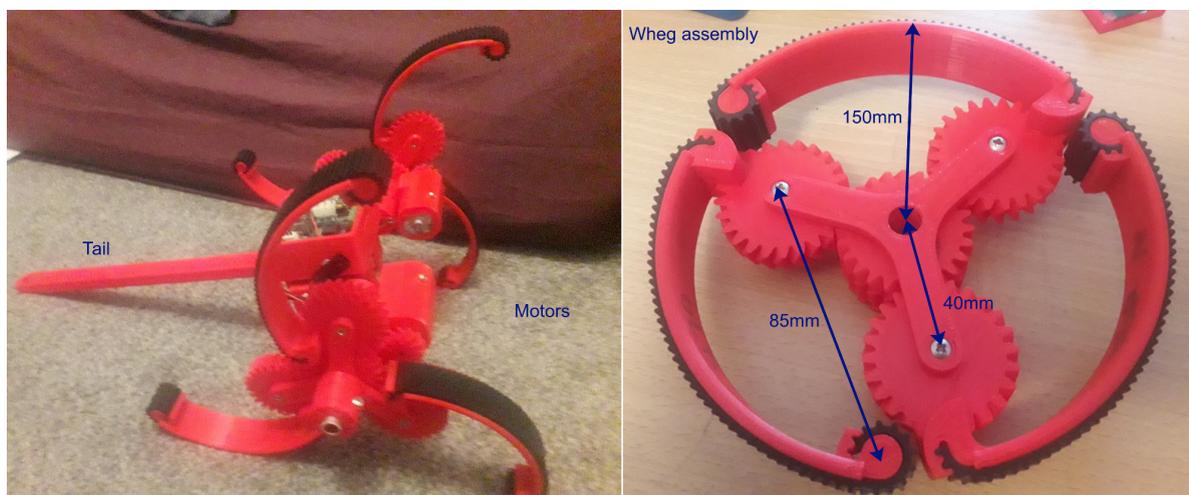


Figure 76. Side view and Whieg assembly.

The full robot is printed out of a red PLA plastic and weighs around 750g. Black ridged rubber belts cover the outside of each leg for grip. Each Whieg assembly has three legs and rotates around an aluminium tube which acts as an axle.

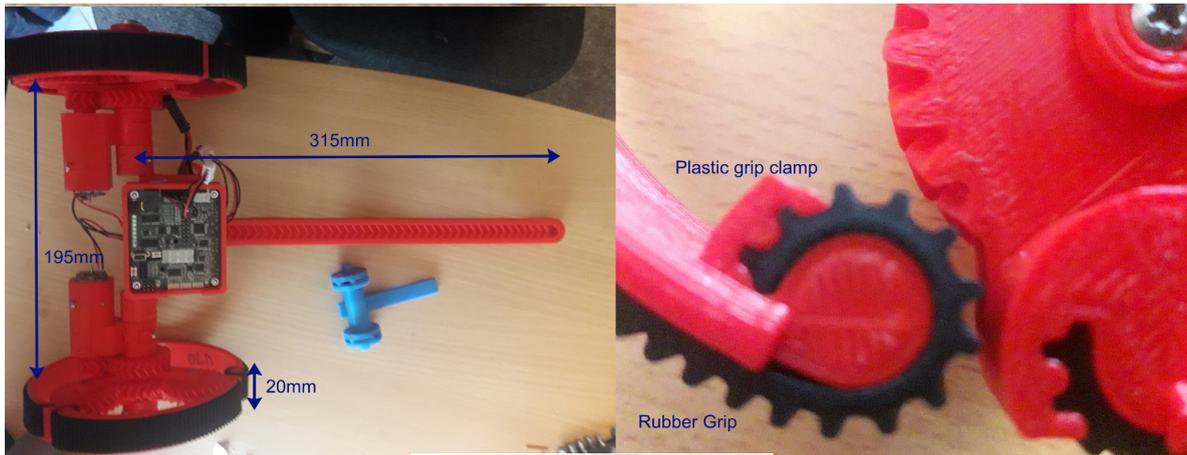


Figure 77. Top view and close up on rubber belt clamp

Figure 77 provides a comparison of the full robot and the second scale model. The width of the Whlegs were to match the width of the rubber belts that gave them grip. When the Whlegs are used in wheel mode the legs rest on the cog of the next one round the circle to reinforce the wheels integrity.



Figure 78. Battery housing, Aluminium tube axle/leg lock and APiHAT

As this is a prototype, much of the APiHat and battery is exposed (figure 78). In a later design these would be covered to protect them from collisions and the elements. Although these leg lock holes are small they are crucial for rigour experimentation. They mean that the same leg angle can be reliable and repeatedly tested.

4.9 Wear and Tear Report

Cylindabot was extensively tested and in this section any damage to the fabric of the robot is documented. The majority of the design held up well.

Rubber grips fraying before being glued down:

Originally the rubber grip was just slid into place and held by friction. The grip mostly bends at either end where it is clamped in place. This naturally puts more strain on these parts. When the legs flex under impacts the force translated up and put these sections under more stress. So they began to split and break at these points. The solution was to glue the rubber grip down to the legs. The legs don't actually flex at the end points of the plastic. Once glued down the strain of the curve and the leg flexing acted on different

parts of the rubber grip. Even if the rubber does split in the future the glue will mean that it will continue to function.

One leg broken accidentally:

The only plastic part to break while in use was one of the legs that was broken by the user when picking up the robot. It snapped at the joint between the leg and the herringbone cog. Obviously this could not have been prevented. There is a small amount of wear on this point on all the legs and it needs to be strengthened if the robot is redesigned.

Motor cog:

The cog that connects to the motors D shaft still functions, however it has become looser over time. This is the point of the greatest torque applied by the 20D motors. It can rotate a small amount in each direction. To correct this in a later design a small metal plate could be glued into the plastic to meet the flat side of the D shaft.

4.10 Summary

The Cylindabot was a successful design that was able to be constructed into a physical prototype. It was able to fulfil the design requirements. The majority of the parts were 3D printed using PLA to allow rapid prototyping. Each Whег assembly consists of three legs, each of which has ribbed rubber grip. The Whегs rotate around an aluminium tube that acts as both a conduit for electronics on the other side of the Whегs and an axle that strengthens the design. The core of the robot consists of two motors, control circuit boards, a battery and a tail (for balance). The control circuit board consists of a Raspberry Pi A+ and an APiHat board which allows the rest of the electronics to plug in easily.

This prototype will allow any simulation to both, have real life verification and the reality gap can be measured. In this case verification means that the work is grounded in a design that actually exists, as such, is more tangible and applied. The robot is able to transform from using large radius wheels to three legged Whегs. More importantly these legs and the tail can be locked at different positions.

Although the tail is adjustable, it is still limited to a maximum length of 315mm from the Whег axle. The advantage of a longer tail is that the robot would be more stable when using the legs. The disadvantage would be that the tail could get caught on terrain or limit Cylindabot's ability to navigate tight spaces. This is due to the differential drive of the robot that means that its point of turn is near the front and hence the tail makes a wide sweep when the robot makes a tight turn. This could be mitigated by the robot always driving forward when turning, but this would then limit its turn radius.

There are plans to improve the design by adding sensors, moving the core parts around and making the tail retract. In later chapters the Cylindabot is simulated and physical experiments undertaken. This is where the capabilities of the design is scientifically tested to give a general theory of this style of design.

Chapter 5: Steps and Slopes

Cylindabot tested on slopes and steps in hardware and simulation

5.1 Introduction

The ability of a robot to move is fundamental for permitting the robot to explore its environment, interact with its surroundings, and ultimately perform useful tasks. Autonomous robotic systems are limited by the robot's ability to move, often in challenging environments or unknown environments. These challenges increase significantly once robots are taken outside the laboratory and placed in the real world.

A robot is likely required to be able to manage a range of terrains. The ability to be able to act autonomously in an uncontrolled environment increases the difficulty. Many platforms are designed to function on a specific terrain type (e.g., indoor research laboratory, flat grass, or dry rocky areas). However, an important aspect of ensuring flexibility for a platform across a range of environments and surfaces is to permit adaptation of the robot to the surface it is operating on. Hybrid robots allow different interactions with the ground, for instance, the Cylindabot described in this thesis can both roll smoothly using a form of wheel and walk with legs that can extend from these wheels.

The Whег design allows a wheel to transform from a round wheel to a three legged structure. The term Whег refers to a combination of wheel and leg design which is continually rotated. The benefits of this style are inspired by [8]. This might be considered as simply a wheel of spokes without a rim. In [2], a six spoked wheel is analysed against its performance to move up a step, and an approach/surmount theory is developed. This theory is applied to the results in this chapter. The term Whег is used in this thesis to emphasise continuous rolling and discrete strides but a rimless spoked wheel is another way of describing it.

The Cylindabot robot is designed to allow a high degree of mobility with a minimal number of actuators. The high degree of mobility means that it can traverse smooth and jagged terrains in unknown environments. The minimal actuation means that the robot is lightweight and is easier to control efficiently. The level of deployment of the legs and tail can be locked to investigate how the capabilities change.

In this chapter, the Cylindabot design is presented, experimental results illustrating the robot's capabilities are compared to the theory developed for the design, and the results are considered against similar platforms in the literature. First, the design of the

Cylindabot is discussed with an emphasis on inspiration and practical decisions made. The theory of linear bounds from [2] is then clarified. The simulation setup and results are then summarised, followed by paired results from physical experiments. Finally, the Cylindabot is compared with the current state-of-the-art hybrid locomotion platforms.

Contributions of the chapter:

- A novel Cylindabot design which can transform into different levels of deployment is presented. Using these levels, on both a physical platform and simulation model, an analysis of the effect that this transformation has on the robot's ability on a slope and single step is undertaken.
- An experimental demonstration that both the wheel and leg states are optimal depending on the terrain through the use of hypothesis testing is presented.

The hypotheses are cast on the robot's capability on a step or a slope and will be tested against the extreme deployment of the legs and tail.

Step — H0 (null): Leg and tail deployment has no effect on the robot's maximum ability to traverse a step. H1 (alternative): Having longer legs and tail improves the robot's maximum ability to traverse a step.

Slope — H2 (null): Leg and tail deployment has no effect on the robot's maximum ability to traverse a slope. H3 (alternative): Having shorter legs and a longer tail improves the robot's maximum ability to traverse a slope

5.2 Background

There can be a wide variety of terrains that need to be traversed when deploying an autonomous robot. In order to adapt to these different terrains, a robot should be able to change its locomotion. Such a change can be achieved by either varying its control approach or its physical representation that interacts with the surface. The change in control requires the robot's movement to have several DOF (degrees of freedom), and the focus is on how these can be coordinated. Another approach is to change the fundamental way the robot moves, referred to here as a hybrid robot (see Table 17).

Significant advances have been made in walking robots; for example, the Boston Dynamics Spot, which was based on high radius, low gearing motors. This technology was made possible by the Cheetah robot [5]. The importance of these motors is that they allow control of the back drive, damping, and elastic properties of the joint, which previously was performed with hydraulic or pneumatic joints. The SLIP model of Raibert [105] has been taken to new heights by the Salto-1P platform [71]. It is unclear if the agility of the Salto-1P can be scaled up to a larger platform, as is used by drone propellers for maintaining roll and yaw stability. The R-Hex robot [35] was developed by multiple institutions and has been improved/tested on a range of terrains [61], [106], [107].

Although achieving impressive results, they still only have a single type of interaction with terrain. A manipulator was used on a similar design [108] to change over what was attached to the actuator, which were referred to as shoes.

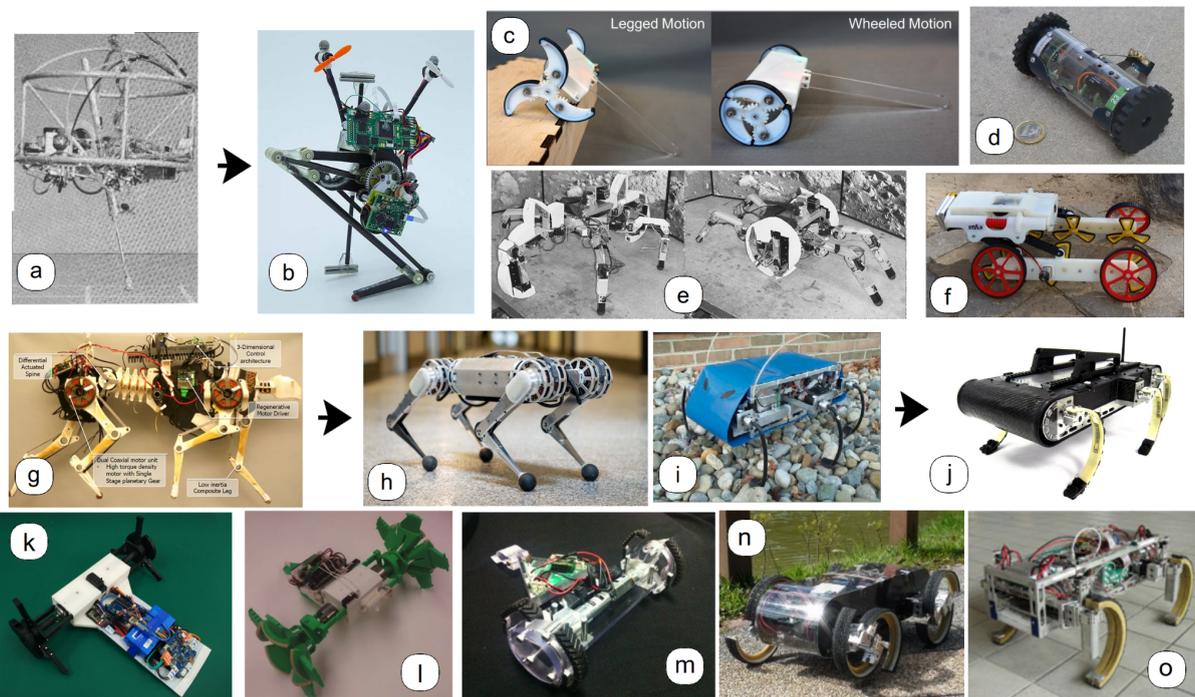


Figure 79. Existing platforms. Raibert's One legged hopper (a) [105] which led to Salto-1P (b) [71]. Wheeler robot © [9]. Scout robot (d) [49]. LEON (e) [12]. Rising STAR (f) [58]. MIT Cheetah (g) [5] leading to MIT Mini Cheetah (h) [109]. Original R-hex (i) [34] leading to the more advanced X-RHex (j) [61]. T-shaped Whег robot (k) [64]. Passive Wheel robot (l) [65]. Wheel Transformer (m) [66]. TurboQuad (n) [59]. Quattroped (o) [60].

The problem with many hybrid platforms is that they require several active degrees of freedom to move in a hybrid way. Adding this extra actuation leads to a more complex control, higher weight, and the need for stronger motors. This usually limits the mobility of the platform and the distance it can travel due to battery life. Secondly, with these modes of locomotion there are the questions of which is best for a given situation and how much should the robot transform.

Certain platforms have aimed for this target of limiting the active degrees of freedom while maximising the robots diversity of locomotion. The scout robot [49] uses two large wheels for rolling and a sprung tail that allows it to jump. The Rising Star platform [58] uses two motors for rows of Whęgs and two other motors for changing sprawl angle plus body position. The Wheeler robot [9] is a similar platform to the Cylindabot with wheels that transform into legs based on the direction of drive. In their work, the authors experimented with different numbers of limbs on the Whęg and gear ratios for deploying the legs which informed this new design. One issue is that the need for wheels was not clearly demonstrated, hence the need to be hybrid was not either. It is stated that wheels

allow the Wheeler robot to move faster, however with a more powerful motor the opposite is found for Cylindabot.

Other transformable dual Wheg systems have been developed with testing focused on the improvement of the legs provided. The T-shaped leg robot [64] uses a horizontal pull bar similar to the spokes in [2]. This mechanism is unable to control how much the legs were deployed which is an important feature both [2] and Cylindabot have. The transformable wheel in [65] was passive and the movement slid along the main axle of the Wheg. In [64]–[66] the reason for the wheel mode was to allow smoother motion on flat surfaces. Instead of using two wheels that transmute into legs or Whegs, the following robots use four instead. Both TurboQuad [59] and Quattroped [60] are four wheeled robots that transform into Whegs by splitting the wheels in half. The TurboQuad moves them linearly to make a two arced Wheg, whereas the Quattroped folds the wheel over to make a single arc.

Table 17. Robot Locomotion Modes

Robot	DoF	Roll	Walk	Jump	Year
Scout [49]	3	x		x	2000
R-Hex [35]	6		x	x	2001
Wheg [99]	6		x		2007
Wheel Transformer [2]	5		x		2007
Impass [37]	8		x		2009
LEON [12]	18	x	x		2010
Cheetah [5]	12		x	x	2013
Salto-1P [71]	5			x	2018
Rising Star [58]	4	x	x		2018
Wheeler [9]	2	x	x		2019
Cylindabot [110]	2	x	x		2021

The issue with the range of configurations these platforms have, is how a robot should approach a locomotion task. The problem being addressed in this work is can the capabilities of this robotic platform be properly modelled. Therefore, in effect, can a simple conversion be made from the design/configuration of a robot and the size of

terrain it can handle. It is impossible for a single platform to traverse every possible obstacle, therefore ideally a robot should be able to determine what locomotion it is capable of performing and have the ability to adapt to the most optimal configuration for that terrain. Additionally, this testing validates the need for a robot's ability to adapt to the environment which is often not fully illustrated

5.3 Linear Constraints for a Step

Understanding the theory of why a design has specific capabilities means future applications can be predicted. The predictive model can then be applied to where a robot could be successfully deployed, what route it should take in an environment or how a more appropriate robot could be designed. This section explains how the approach and surmount theory can be used to bound the possible step height.

In [2], a geometric approach was used to bound the ability to climb a single step. The robot had wheels made of 6 legs/spokes with a tail, where the length of these legs plus the length of tail were varied. This means that retracting the legs does not transform them into a smooth wheel. The reason given for retraction was to fit through small gaps during disaster rescue.

The theory split the ability to traverse a step into two phases: approach and surmount. The approach phase was the ability of the robot to achieve a leg purchase on the step to lift itself up. This assumed the Wheg would be able to rotate at the base of the step. In the three legged case the length of the side of the equilateral is made by the ends of each leg. The surmount was then the ability to pull the rest of the robot and tail over the step. This was calculated by subtracting the leg radius from the tail length and was determined by whether the tail remained on the ground when the legs lifted the robot up the step.

The geometric procedure simplifies the robot legs and tail into lines for ease of calculation. It assumes that the top of the step and the foot of the leg will have ideal friction between. There is no consideration of energy or torque so whether the motor will be powerful enough to lift the robot. Because of this it is a theoretical maximum that could be traversed with a given leg and tail length. Hence it is a linear constraint similar to those found in linear programming theory.

Two linear constraints are generated by Equation 23 and Equation 24. In these equations H is the height of the step, R is the radius of the wheels and L is the length of the tail from the centre of the Whegs. Here, k depends on the number of legs in the Wheg, $\sqrt{3}$ for three legs and $\frac{3}{2}$ for the six legs. Only six legs were tested in [2]. Equation 23 is based on

whether the end of the leg can reach the top of the step. Equation 24 can be simplified to whether the tail is longer than the step plus the legs.

Equation 23. Approach inequality

$$H_1 < kR$$

Equation 24. Surmount inequality

$$H_2 < L - R$$

This theory creates a linear bounding region for the robot's ability to climb a step as shown in Figure 80. If the robot's current configuration and step height is below both these lines then it is possible for it to climb the step. This theory was tested against a six legged hardware platform in [2] and a ratio for three legged robots only theoretically proposed. The approach/surmount theory will be validated by the data gathered with the Cylindabot for both simulation and hardware. Showing there is little reality gap for this sort of obstacle traversal limit, which is the robot reaching the top of the step (Equation 23). The blue line is whether it can get over the step once it reaches the top (Equation 24). The k factor has been set to the proposed value for three legs and the tail to 285 mm.

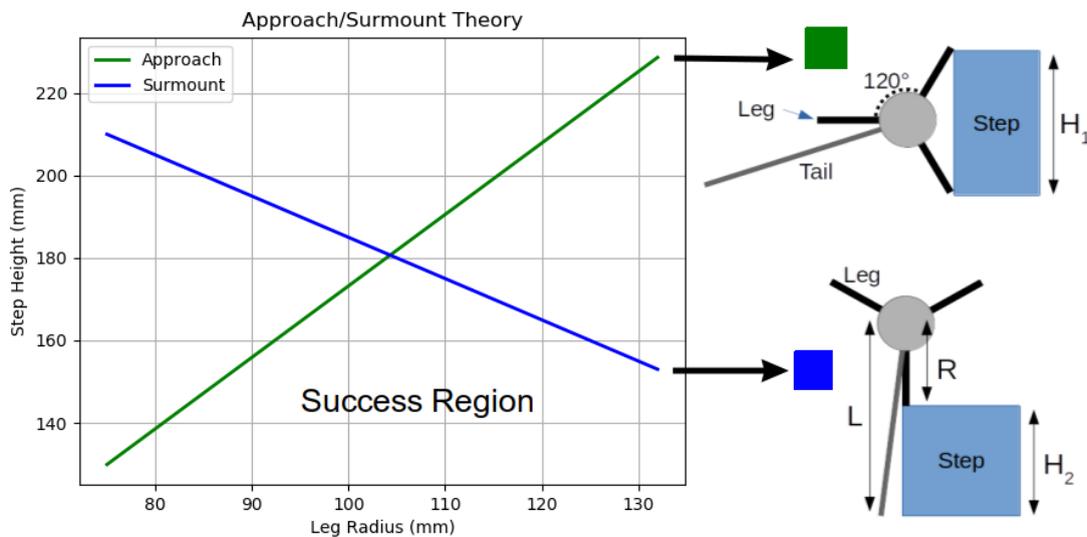


Figure 80. Approach/Surmount theory graph and diagrams of each. The green line is the approach limit which is the robot reaching the top of the step (Equation 23). The blue line is whether it can get over the step once it reaches the top (Equation 24). The k factor has been set to the proposed value for three legs and the tail to 285 mm.

5.4 Simulation Setup

In order to assess the feasibility of the approach, the first stage was to develop and test the design in simulation. Being able to simulate allows for a high degree of experimentation before construction, thus saving time and construction costs. Work in this section outlines

the simulation approach, along with experimental results that have helped inform the development of Cylindabot.

The simulation setup uses CoppeliaSim (previously known as V-Rep) allowing the use of a Python API and easy to access GUI. The code was split between a Lua script, which handled creating the terrain feature, and a python program that controlled the simulation and the robot's movement. A simulation scene was designed to use two tiles with a square metre gap between them. In this gap a terrain obstacle is generated (Figure 81). The two basic obstacles used here are a slope and a step. The slope is a height field and the step is a simple static cuboid at a given height. There are two main differences between this simulation environment and the one used in earlier chapters. Firstly, a Newton physics engine is used as it proved to be more reliable with the new robot model and the slope being represented by a height field rather than an angle cuboid. The choice of height field was due to a programming bug at the time when using an angled cuboid.

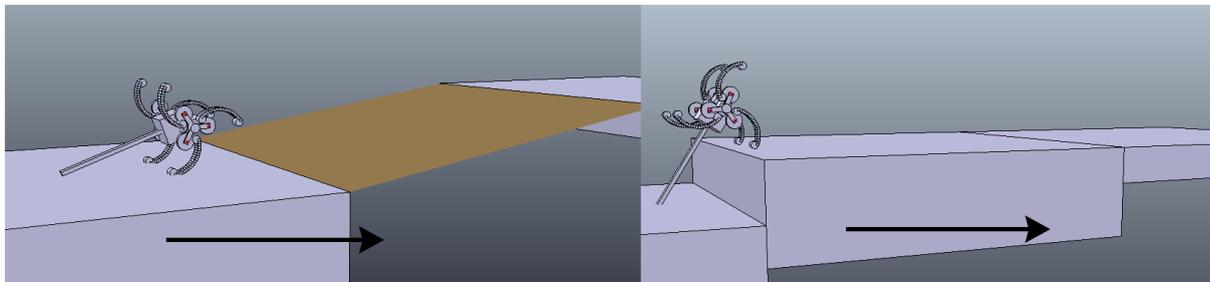


Figure 81. Simulation environment. Left: slopes using a heights field, Right: single step with a single cuboid primitive. The obstacle is added between the start and finish tiles. Arrows show direction of travel.

The controller for the robot is a simple differential drive that is attempting to move the robot to the centre of the finishing tile. It takes the robot's position to give a target angle direction and uses the robot's orientation to adjust wheel speeds to head in that direction by the turn angle. This is a proportional controller that subtracts from the wheel speed on the side it needs to turn towards, as shown in Equation 25. It is possible that this becomes negative, allowing the robot to turn on the spot if facing more than 90 degrees from the target direction. The controller was part of the main python loop and ran at approximately 100 Hz.

Equation 25. Cylindabot subtractive controller

$$\textit{Inside Wheel Speed} = \textit{Speed} - \min(\textit{Speed} \times 2, \frac{\textit{Speed} \times \textit{TurnAngle}}{45})$$

The selection of difficulty of the obstacle uses a method nicknamed the 'beer mat' method. If the robot is successful the result is recorded and the difficulty is increased or if it fails then difficulty is decreased (both in increments of 1 mm). To initiate this in the correct region the difficulty is increased to a point the robot has failed at least once in an increment of 1 cm. Whilst [64] had to repetitively test at difficulties both too easy and too

difficult, the 'beermat' algorithm allowed the robot to be repeatedly tested at the correct difficulty while reducing the number of runs required.

The robot model is created using only primitive shapes (spheres, cuboids, or cylinders) as the physics interactions are more realistic. The curved legs are created using 20 cuboids that are then joined together (Figure 82) these are capped by cylinders, one at foot end and two at the other.

The main body, leg holders and tail were also made with a combination of cuboids and cylinders. These primitives were then composited together using a group function when they were not meant to move relative to each other. Revolute joints were used to connect the legs to the Wheg assembly and the whole Wheg to the main body. Finally a linear joint was used to control the tail position. This was all designed to match the physical design as closely as possible, while still giving robust physical interactions. In section 5.5 the close correlation between simulated and hardware results are shown. The two main parameter changes made to achieve this were setting the strengths of the revolute joints high enough to take Wheg based movements and setting the frictional coefficient to 1.35 (see Table 18).

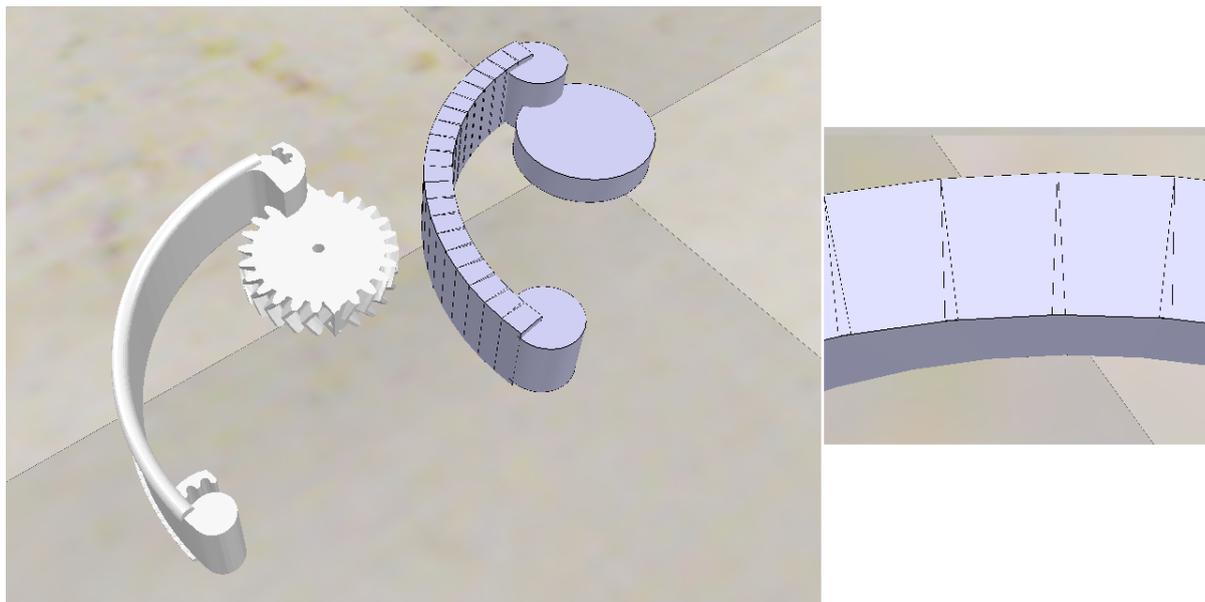


Figure 82. Left: Comparison of a single leg in the simulation with the STL of the printed leg. Right: Magnified section of curved leg to show intersection cuboids.

Two geometrically simple obstacles are analysed here: a single step and a slope. In the graphs, a small icon, in the top right corner, is used to distinguish the difference (i.e. step or slope). The step can be analogous to a jagged surface. The slope is used to represent smooth surfaces and interactions. A smooth surface could be measured by its steepest point. By this argument, a robot's ability on a step and a slope can represent its ability on a range of terrains.

Table 18. Simulation parameters.

Physical Part	Mass (kg)	Inertia/Mass ($m^2 \times 10^{-3}$)		
		x	y	z
Main Body	0.431	1.603	0.835	1.859
Tail	0.060	0.041	0.900	3.4
Inside Whег hub	0.008	0.345	0.345	0.687
Outside Whег hub	0.017	0.579	0.362	0.213
Single Leg	0.090	0.090	3.459	3.04
Actuators	Limit		Parts	Frictional Coefficient
Main Motors	2.5×10^4 Nm		Legs	1.35
Leg Position	2.5×10^3 Nm		Other parts	1.0
Tail Linear	5×10^1 N			
Time step	50ms		Physic Engine	Newton

5.5 Simulation Results—Single Steps

The ability to climb up a single step is a good measure of the robot's capacity to negotiate unfavourable terrain by using its legs. In this case, that step is an increasingly taller metre square cuboid. The theoretical hypothesis is that increasing Whег deployment and tail length should improve the height of the step that can be traversed. The results here are given over 50 successful runs using the 'beer mat' method. Initially the step results were mapped against the angle of the leg along the x-axis as this is how the legs deploy (by rotation). Due to the approach and surmount theory in [2] they were remapped to the radius of the legs from the axle of the wheel. To fully realise the potential step climbing, longer tail lengths were added, not seen in the slope or hardware results.

In Figure 83, the upward trend line for all tail lengths are illustrated. At various points the leg radius reached a breakpoint where its ability to transverse the step trailed off. The downward trends are parallel to each other. This demonstrates the approach and

surmount lines, which helps support the theoretical model. The maximum deployment of the leg and tail fall just short of an urban environment standard. In the UK the maximum rise of a step is 220 mm.

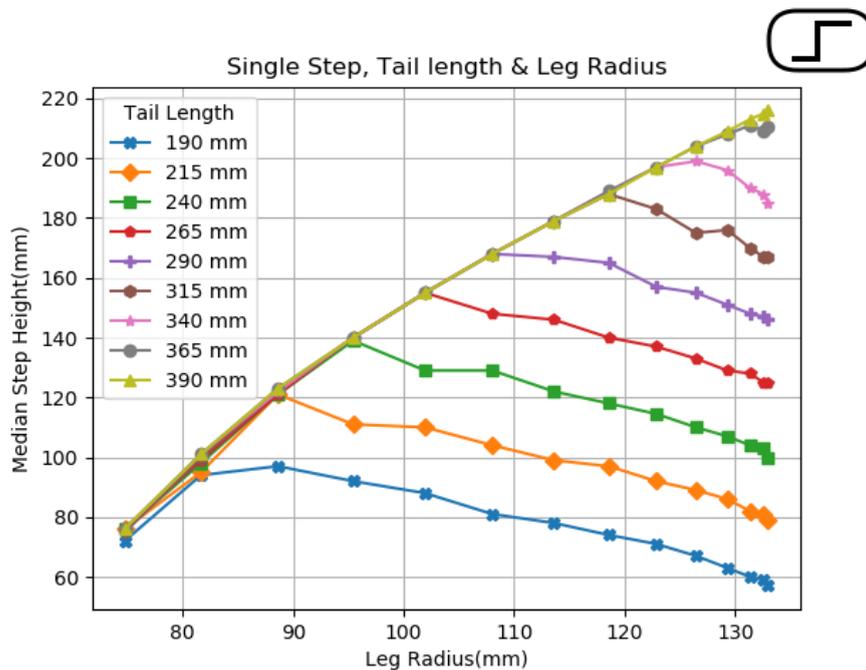


Figure 83. Simulation results for climbing a single step using the 'beermat' algorithm. Median shown given from 50 successful results. Colours represent tail length and the plot is leg radius against step height.

The upward trend shows where the edges of the Whieg catch on the step and are able to lift the robot (1. approach). This is not the same as completely climbing the step and there are three outcomes which depend on the length of the tail. If the tail is shorter than the leg radius and step height, then the robot will fall over backwards (2. fall backwards). Next, if the tail is only a little bit longer than the legs, they will not be able to get enough purchase on the step and the weight of the robot will be in the tail (3. no grip). Finally, if the tail is long enough the robot will be able to pull itself up the step (4. surmount). These are shown in Figure 84.

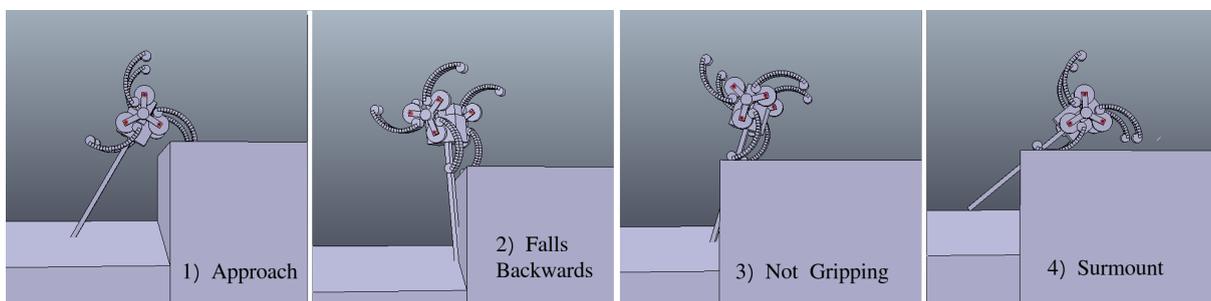


Figure 84. Four phases of attempting a single step: (1) Approach, (2) Fall backward, (3) No grip, (4) Surmount or Successful traversing the step.

Statistics on Step Data:

To test the statistical significance of these data (Figure 83) two algorithms were applied, the T-test and Cohen effect size. These were carried out for a reference configuration of the robot with the legs retracted into a wheel at a radius of 75 mm and tail set to 190 mm. Three different states were tested: fully deployed legs with retracted tail, tail extended but leg retracted, and both legs/tail fully deployed. The P-values were effectively zero and the effect size reached 73.2. The data show that increasing the tail length and leg radius concurrently improve the performance on steps. Hence the ideal configuration of the robot is to have both the tail and legs extended when going over step or a jagged environment.

~~Hypothesis 0 (H0 null). Leg and tail deployment has no effect on the robot's maximum ability to traverse a step.~~

Hypothesis 1 (H1 alternative). Having longer legs and tail improves the robot's maximum ability to traverse a step.

5.6 Simulation Results—Flat Slopes

Slopes are used to mimic smooth interaction with different terrain features. It is expected that the legs will be a hindrance in getting up a slope as they will move the mass of the robot further away from the surface. Results shown in Figure 85 show that the deployment of the legs has a negative impact on the robot's slope ascending ability. The different tail lengths still produce parallel lines. The main feature that causes failures here are the legs slipping against the slope. Similar to the steps results, these needed to be remapped to a different domain to give linear results. In Figure 85, the midpoint of the leg radius was used instead of its maximum. This remapping mostly affects the results when the legs are fully deployed. As at this point the maximum radius does not significantly increase by the amount of curve changes. The curved nature of the legs is salient to the robot's ability to navigate slopes. To calculate this mid-point radius, an additional python script was used in addition to the one that produced the graphs.

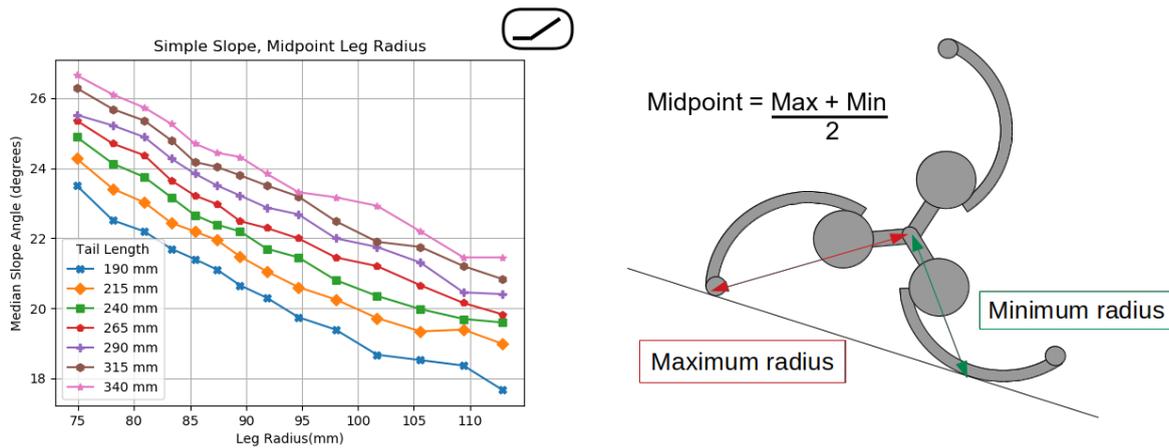


Figure 85. Slope results remapped to mid-point radius which is the average of the maximum and minimum contact point radius. Median of 50 successful results using 'beermat' algorithm to change difficulty.

Statistics on Slope Data:

In the same way as with steps, a statistical analysis was undertaken between the fully retracted configuration of the robot and different parts being deployed. This time the tail only reached 340 mm in length. The P-values were again extremely small, meaning that we can reject the null hypothesis. The effect size is largest when the tail is fully deployed and the Whegs are retracted into wheels.

For both a step and a slope it is better to have the tail as long as possible. The data shows that when the legs are retracted the robot is more capable on slopes which is of key importance. It means there is a need to use wheels for certain terrain types. Both configurations are useful depending on the situation.

~~Hypothesis 2 (H2 null). Leg and tail deployment has no effect on the robot's maximum ability to traverse a slope.~~

Hypothesis 3 (H3 alternative). Having shorter legs and a longer tail improve the robot's maximum ability to traverse a slope.

5.7 Physical Tests

The next step is to see if these simulation results match up with the real world. The previous results need to be validated on hardware to see if there is any reality gap between the two sets of results. One of the key differences between the simulated and real robot was the legs that make up each Whieg. The legs are able to flex under impacts and there are small ridges in the rubber of the legs. Both of these elements were not able to be simulated.

For the experiment, a step and slope were needed to be constructed (Figure 86). Due to physical limitations, they were not able to be configured to the same increments as simulation. This means that changes in step height or slope angle were bigger in the

physical experiment compared to simulation. The robot was human operated rather than autonomously using its position and the success or failure was due to a human observation. The robot's movement was controlled using a Bluetooth joystick.

The Cartesian position of the analogue stick was fed directly to the motor speeds. As only two motors were required in the design, this allowed full control of movement. The data produced is more discrete because of physical limitations. What this means is that the increments that the robot and obstacles can be configured in are larger. So there are fewer points on the graph and their relative inaccuracy is increased. The legs can only be locked into eight positions with 20 degrees between the positions that are not the end positions. The step height and slope adjacent size were altered in increments of 1 cm. This is represented in the y-axis of the graphs being measured in centimetres. After both sets of data are plotted, the reality gap is measured using the R-squared method.



Figure 86. Physical experiment setup. For the step 2 cm/1 cm thick planks were used to adjust height. The same material was used against a bench and slotted decking for the slope. The fixed height increments and slotted decking mean repeatability of the environment was not an issue. Arrows show direction of travel.

5.5.1 Single Step Results

Considering figure 87, the trend is still upward until it peaks. This peak is caused by the length of the tail when the robot starts to fall backwards. One key difference in the hardware results is that, when the legs are fully retracted, the robot is not able to surmount a step as high as it did in simulation. The green and blue lines represent the approach and surmount limits respectively. The green line is a little higher than the actual results as this was a theoretical maximum. It assumed infinitely thin straight legs and they always gained purchase on the step. It is likely that the curve of the legs prevented it reaching this value. The error bars for the simulated data shows the full range of results which have a relatively small variance. The simulated and physical results follow each other well and it is probable that it is simply the accuracy constraints that cause the disparity.

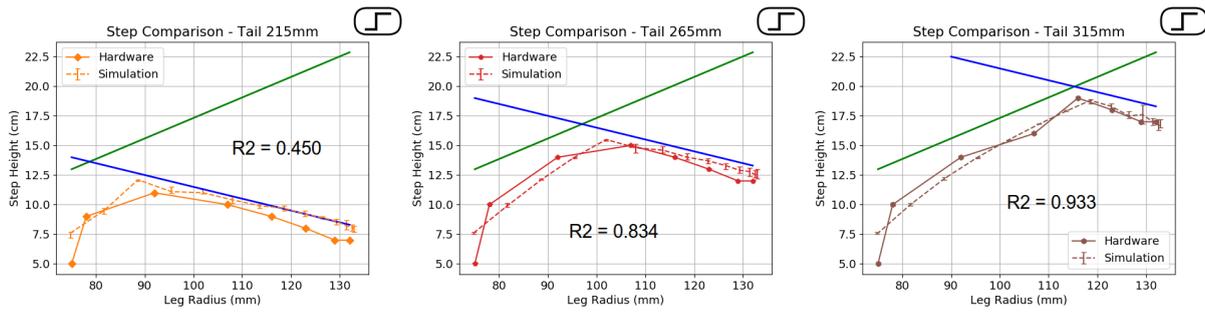


Figure 87. Hardware results for a single step compared with simulation results. The three graphs are for different tail lengths. The dashed lines are the simulation results from figure 83 with the total range added as error bars. Green and blue lines represent approach and surmount theory, respectively, from section 5.3. The R2 is the R-squared statistical values to measure the reality gap.

Table 19. Physical results for a step. Highlighted in yellow is the optimal leg radius.

Tail length (mm)	Leg Radius (mm)							
	75	78	92	107	116	123	129	132
315	5	10	14	16	19	18	17	17
265	5	10	14	15	14	13	12	12
215	5	9	11	10	9	8	7	7

5.5.2 Simple Slopes Results

The biggest divergence between simulation and hardware, Figure 88, is that the legs are fully retracted. This backs up the assumption that smooth continuous drive is better for a slope, i.e. a wheel or track. Again, the simulation and hardware results match each other for the majority of the plot with the exception of full retracted legs. This reality gap however does follow the hypothesis that wheels suit a slope. There is no trend line presented as more work is needed that takes into account the friction between the robot and the surface of the slope.

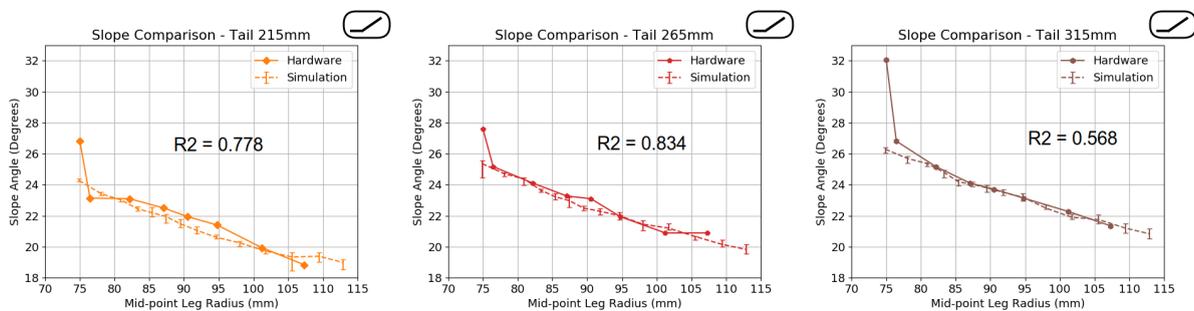


Figure 88. Hardware results for a simple slope compared to simulation. The three graphs are for different tail lengths. The dashed lines are the simulation results from Figure 8 with the total range added as error bars. The R2 is the R-squared statistical values to measure the reality gap.

In Figure 87 and Figure 88, the R-squared values (R²) are given for how close the hardware results match the simulation. The main deviation occurs when the legs are fully retracted into a wheel. For the calculation of these values the linear interpolation was used for \hat{y} off the simulated results and the mean of the hardware results for total error. The generated values show a strong to fair correlation between the simulated and hardware results. In particular, the results for steps when the tail is set to 315 mm. This shows that the simulation for these specific obstacles was able to accurately model the maximum difficulty that could be traversed.

Table 20. Physical results for a slope including tan calculations. Highlighted in yellow is the optimal leg radius.

Tail length (mm)/ Slope tan or angle(°)		Leg Radius (mm)							
		75	78	92	107	116	123	129	132
315	tan(θ)	47/75	47/93	47/100	47/105	47/107	47/110	43/105	43/110
	θ	32.07	26.81	25.17	24.11	23.71	23.14	22.27	21.35
265	tan(θ)	47/90	47/100	47/105	43/100	29/68	29/72	29/76	29/76
	θ	27.57	25.17	24.11	23.27	23.10	21.94	20.89	20.89
215	tan(θ)	47/93	47/110	29/68	29/70	29/72	29/74	29/80	29/85
	θ	26.81	23.14	23.1	22.5	21.94	21.4	19.93	18.84

5.8 Discussion

Several examples of robots with transformable wheels can be found in scientific literature. One of the most recent is the WheelER robot [9] which deployed a passive leg system. The drive of the motors was applied to a cog so that the legs extend when driven in one direction and retracted when driven in the other. A passive mechanic is applied in [66]; a catch that protrudes from the wheels and when the wheel is driven in one direction it releases the legs. In the Impass robot [37] the legs are actuated linearly through the axle rather than the rotation used in this work and achieve the higher step ratio. The LEON robot [12] has two wheels that transform into legs and augment four other legs to allow it to walk as a hexapod. Lastly, the Whegs robot [99] has four Whegs made up of three legs in each Wheg that are represented in simulation by cuboids. The fact these legs do not deploy does put it at a disadvantage when it comes to the step ratio comparison.

In Table 21 the step ratio is calculated by taking the maximum step that can be surmounted, divided by the robot's height (legs not deployed). This means these results

are agnostic to the size of the robot and are relative to the design capabilities. The only robot to outperform the Cylindabot was the Impass robot with its linearly pushed legs able to reach a taller step. Although it lacks the ability to roll smoothly which will hinder its movement on smooth surfaces.

Overall Cylindabot has been able to perform well against similar robotics platforms when using the benchmark of the step height a robot can climb over. Here, that was converted into step ratio to make for a fair comparison. However, although slopes are still a simple obstacle, the frictional coefficient is difficult to standardise between different institutions and could be the reason it is less commonly documented. Further investigation is needed to see if the linear trends in Figure 88 can be converted into a general theory including the frictional coefficient of the slope.

Table 21. Robot ability comparison (Data taken from respective papers).

Robot	Step Ratio	Slope Angle	Ref
Wheeler	1.2	15	[9]
Wheel Transformer	1.46	-	[66]
Impass	1.73	-	[37]
LEON	0.53	35	[12]
Wheg	0.52	30	[99]
Road Runner	0.015	7	[38]
T-Shaped Wheg	1.33	-	[64]
QRoSS	0.7	22	[111]
Cylindabot	1.43	32	[110]

There are two key limitations to the Cylindabot platform; the lack of sensors and the dynamic control of the leg/tail deployment. Firstly, although the Cylindabot is large enough and has processing power to act autonomously, it does not have the sensing capability to do so. Once the robot is able to determine what is around it will then have to determine which mode (wheeled or legged) it will traverse the environment, like the action planner in [12]. Secondly, the physical prototype can only change between full wheel or leg mode without human assistance and the tail has to be set to different positions manually unlike the variable diameter robot from [2]. Although it is not clear if they achieved full dynamic reconfiguration to achieve the under/over capabilities of the Rising STAR [58]. It has been shown that wheels or legs are suited to different obstacles

so it is reasonable to assume that being partially deployed between the two might be advantageous. An environment made of a mix of smooth and jagged surfaces, for example, where the robot would use wheels on the smooth sections and legs for jagged bits.

The reality gap for these two obstacles has been shown to be relatively small. This validates the use of the simulation for further experimentation in particular terrains that would be difficult to construct for physical experiments. Simulation has the advantage of being a more efficient way of obtaining data both in time and resources.

5.9 Summary

Steps and slopes are simple geometric obstacles that are often found in real environments and, hence, can act as a benchmark to assess a robot's capabilities to move within an environment with various terrains. Steps and slopes also were chosen for how they can test jagged and smooth environment interactions. In jagged terrain, a robot's gait reach allows it to traverse a section of terrain. On the other hand, the amount of grip a robot has determines its performance on a smooth terrain. Steps and slopes are defined by single parameters and are able to test these qualities.

This chapter discusses the novel robot design from the previous chapter, the Cylindabot. This uses an adaptive locomotion mechanism combining wheels and legs. Real results, and simulations, testing its ability to navigate slopes and steps are compared. Cylindabot's design enables it to locomote in various environments. The step height ability adheres to the approach/surmount theory. On a slope, results were also made linear by calculating the midpoint leg radius. This work suggests that the theory introduced in the chapter can be used in robot designs of any size to determine their abilities to manage slopes and steps within a given environment. This can be backed up by simulation runs before moving to hardware experiments. The experimental results presented in this chapter indicate that the Cylindabot is competitive with similar platforms and outperforms all robots if slopes and step performance are considered together. Finally, there is a need for both wheels and legs depending on the terrain for the robot to perform optimally.

In the later chapters, Cylindabot is tested against new obstacle styles. In Chapter 6, a grid of heights are randomly generated to produce a new challenge. In Chapter 7, a set of ridges, a bridge and a gap obstacles are used to find Cylindabot's success rate. This success rate is then applied to solve a map environment with multiple routes to the target. Slopes and steps are revisited in the appendix where the effect of friction is investigated.

Chapter 6: Grid Terrain

Cylindabot simulated traversing obstacle tiles defined by a grid of heights.

6.1 Introduction

The ability to cross a simple obstacle means that a benchmark for the robot's ability can be created. Although the scope is limited, the next type of obstacle that Cylindabot was tested against was grid terrains. A grid terrain is where a matrix of heights define a one metre square tile. This generates a much more realistic terrain landscape. These heights were majoritively randomly, generated with a uniform distribution and the maximum of this distribution was the measure of difficulty. There were a few matrixes of heights that were predefined for use on a map solving task.

The random nature of these terrains is first used in a comparison between the old (Chapter 3) and new versions of Cylindabot (Chapter 5) as well a four wheeled robot design. A set of 10 tiles are then generated to be used in a map solving exercise which was unable to generate significant results. The comparison and 10 tiles both used a grid density five by five. Finally the density of these height grids (number of heights) is varied and the effect on Cylindabot traversal capabilities is investigated.

Once a set of heights was determined, there were then three ways in which these heights could be applied to make simulated obstacle tiles: steps, slopes and plinths. The three types of tiles alter the shape of the top part of the terrain; whether it was flat, or slopes that meet at a point. The first of these was achieved by placing cuboids with the correct heights to form an array of steps. This style of terrain was used to test RHex [35] and it was a scaled up version of one used for a cockroach [112]. The terrain for RHex had fixed heights due to physical limitations, unlike the simulated terrains in this chapter which were randomly generated. The sloped terrain used a height field to create slopes between the fixed heights from the grid. These sloped grids were used in [12] and were referred to as DEMs (digital elevation maps). With 50x50 points over an 8x8m area this DEM formed the entire area that the LEON robot had to traverse. NIST (National Institute of Standards and Technology) created a set of tests for robots [90] including step fields and lattices of ramps. The third and final shape used on this height grid is plinths, which were a fusion between steps and slopes. This terrain has a flat top like a step but has a slope connecting it to the next plinth.

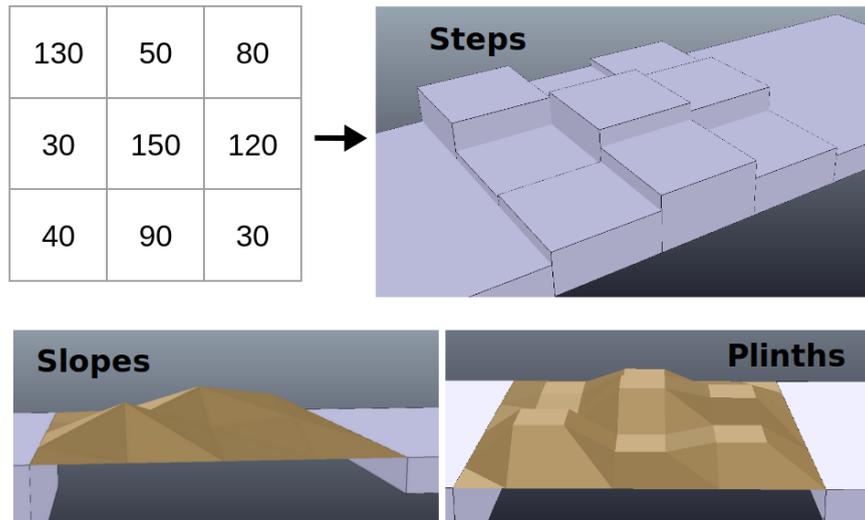


Figure 89. A matrix of heights in mm (top left). The obstacle tiles produced from these heights; steps (top right), slopes (bottom left) or plinths (bottom right).

For the simulation, the random heights were generated in Python before the terrain was generated in CoppeliaSim using a Lua script. The python script ran the simulation, controlled the robot and saved the results. The Lua script only generated the terrain obstacles when the simulation was running. The step type was formed from primitive cuboids that were set to static. Once an object is set to static, it acts like an immovable object, fixed in its position. The slope and plinth style were both made by using a height field object which makes triangular slopes to match the heights matrix given. For both, the edge of the matrix grid was set to zero so that there was a slope meeting the tile the Cylindabot started on. For the plinth each height, randomly generated, was repeated four times to create a flat square at the top/bottom of slopes that connected them.

The first application of these grid terrains was a 5 by 5 grid used to compare two versions of Cylindabot with a four wheeled robot design. Next, 10 fixed designs were generated and analysed. These were used in an attempt to get Cylindabot to solve a map by adapting its leg angle. Finally the size of the grid was varied to see how it affected the difficulty and whether Cylindabot should use legs or wheels.

6.1.1 Sinking grids

While varying the size of the grid, it was noticed that certain grid sizes were giving unusual results. For specific grid sizes, the results were observed to be dramatically lower than others. It was then discovered (by watching the simulation through the GUI) that Cylindabot was sinking through the heights field for these grid sizes and not for others. This bug was consistently occurring for these sizes which were also prime number sizes. The solution discovered was a minor change to the physical size of the height field tile. The dimensions of the tile were increased from one metre by exactly 1 mm. This made the

length of the side 1001 which is divisible by 7, 11 & 13. One possible conclusion of this is that the bug is caused by a rounding error when dividing the grid up, however there is no corroborating evidence to support this conclusion.

6.1.2 Attempt to smooth grids

There was an attempt to find an algorithm to, from a grid of heights, create a smooth surface. The idea was to use cubic splines between the given points that should create a smooth curve. The problem was how to factor in the 2-dimensional nature of the heights. Also, the heights fields were temperamental with certain grid sizes. This concept was never completed and plinths were used instead as a simpler alternative.

6.2 Comparison - Random 5x5 Grid

In this section the final version of Cylindabot is compared to a four wheeled robot and an older version. The test obstacle chosen was a grid of either steps or slopes which had a grid size of 5 by 5. In the simulation the robots started on a flat area and then had to cross a square metre of terrain. The four wheeled robot was designed to model the Husky robot [22] with the alteration of having 150mm diameter wheels. This is so the wheel size matches that of Cylindabot for a fair comparison.

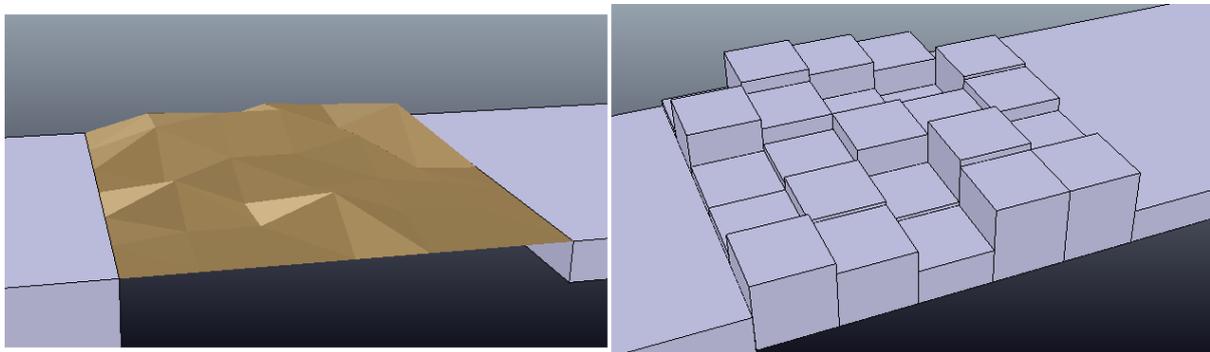


Figure 90. Simulation setup screen shot

The experimental loop used the ‘beermat’ algorithm to determine the maximum obstacle height. The beermat algorithm increased the difficulty when successful and decreased it when Cylindabot was unable to cross a terrain. The 25 heights were randomly generated between zero and this maximum. The random heights meant the results were much more variable than they were with the single slope or step. For both the old and new design of Cylindabot, three different configurations were tested: no legs, half legs and full legs. These represented the legs being set to 0, 70 and 140 on the straight legged robot and 0, 65 and 130 on the new design. All configurations were given a budget of 100 successful runs in the failure region and the obstacle difficulty was changed in an increment of 10mm. This increment and success number were chosen as the beermat algorithm was

able to converge to a failure region. The higher variability of the data meant a larger increment was required.

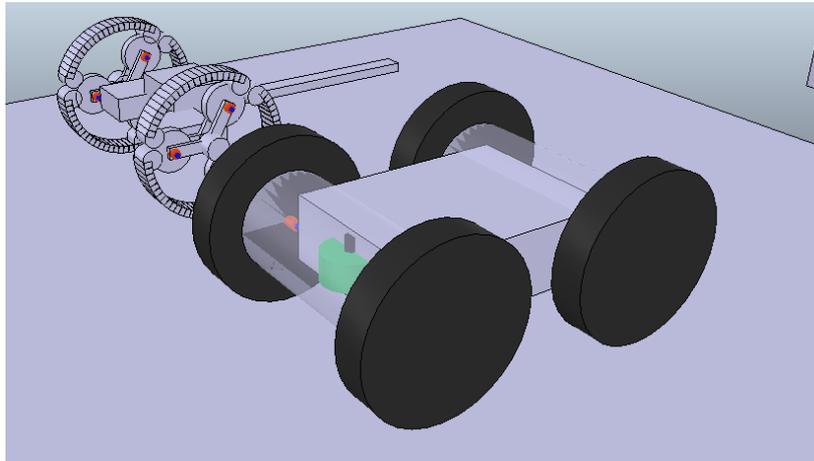


Figure 91. Four wheeled robot next to Cylindabot (New Design)

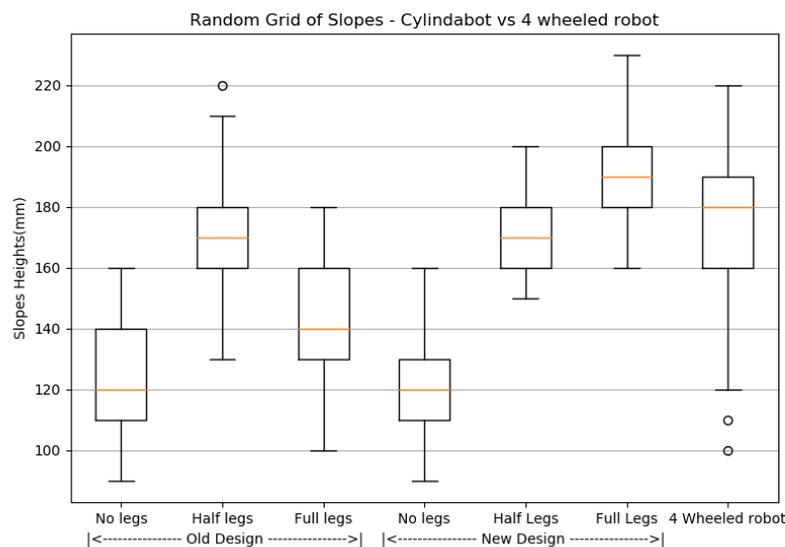


Figure 92. Comparison of different configurations of Cylindabot on sloped grid. The grid heights were determined by a 5 by 5 matrix. Each box plot represent 100 successful terrain crossings

In Figure 92, for a slope, the straight legged version of Cylindabot performed similarly to the new version with curved legs when in wheeled mode. An improvement of a factor of 1.6 occurred when the legs were fully deployed compared to wheeled mode. This is only a slight upgrade in performance from the four wheeled robot. On the other hand, Cylindabot managed to cross statistically significantly higher obstacles when a grid of steps was used. The widest variance is with the four wheeled robot which implies that it is unpredictable on a grid of slopes. A small change in the randomised slope arrangement can mean that the four wheeled robot succeeds or fails. It was unexpected that the legged configuration of Cylindabot would outperform the wheeled setup. However these slopes are 20cm in length so must be jagged enough for legs to be superior.

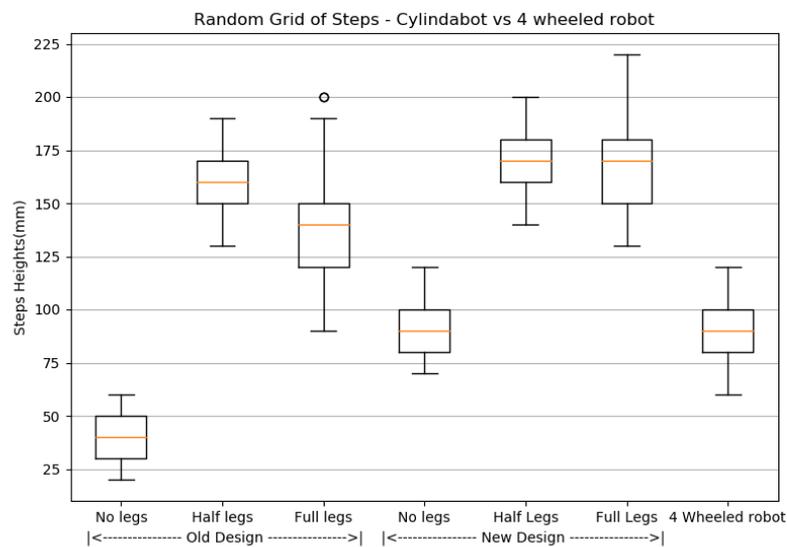


Figure 93. Comparison of different configurations of Cylindabot on stepped grid. The grid heights were determined by a 5 by 5 matrix. Each box plot represent 100 successful terrain crossings

Figure 93 shows a similar grid comparison for a stepped tile. When the legs were retracted into a wheel, the new design outperformed (by a factor of 2.25) the old design due to the small gaps between each section of the wheel. These gaps are between each leg of the Whег and it is the curve at both ends that causes them. The ability of the new robot, when legs are retracted, is roughly equal to that of the four wheeled robot. So in this specific situation the effect of the gaps in a wheel (new design) has a similar effect as having the added push of rear wheels (four wheeled robot). Where Cylindabot really shines in this comparison is when the legs are deployed. Here, with the legs being partially or fully deployed, Cylindabot does statistically significantly better than the four wheeled robot on the steps.

To analyse this data, further statistical tests have been done, specifically T-test and Cohen's effect size (Table 22). The comparisons are all made with the control set of the new robot design with the legs fully extended. This control configuration was chosen as it performed highest on both grids of steps and slopes. For the older design, the best results were used when the legs were half deployed. Every P-value shows a significant difference in obstacle difficulty.

Table 22. Statistical measures for grid comparison of Cylindabot version and a four wheeled design.. The grid heights were determined by a 5 by 5 matrix. P-value from T-test and effect size from Cohen's.

Comparison	Obstacle Type	Statistical Measure	
		P-Value	Effect Size
New vs Old	Slopes	10^{-16}	1.2
	Steps	10^{-4}	0.56
New vs 4 Wheel	Slopes	10^{-5}	0.64
	Steps	10^{-80}	4.52
New Best vs Worst	Slopes	10^{-82}	4.67
	Steps	10^{-83}	4.73

The effect size shows a similar story, with most of the comparisons having an effect size greater than one. The two that did not were the old robot on steps and the four wheeled robot on slopes. This data shows three different results: the new version outperformed the old one, Cylindabot outperformed the four wheeled robot, and legged mode was better for the 5x5 grids.

6.3 Ten Standard Tiles

The next step for these grid terrains was to generate ten different tiles of changing styles and difficulty. This was so that Cylindabot could plan an optimum route through the various terrains. Five of the tiles used the sloped height field (brown) and the other five used the cuboid steps (grey).

The first three of each type of these tiles were randomly generated off of a difficulty rating; easy, medium and hard. The maximum height was increased until a criteria of success rate was met. At this point the height matrix was saved and reproduced for that design. Easy was when at least one configuration of the robot failed, so the success rate was just below 100%. Medium was when the success rate was less than 70% and hard was a rate of less than 30%. These success rate criteria were chosen to differentiate the three difficulties. Two pyramid height matrices were designed where the height increased by a given increment towards the centre of the tile. The final two tiles were a ring of heights with a dip in the middle; called pits and walls. The pyramid, pit and wall heights were calibrated again to less than a 50% success rate. The success rate criteria for all these obstacle tiles were selected so that Cylindabot would be able to cross the tile some but

not all the time. It was hoped that this would mean that Cyclindabot would need to be in an optimal configuration to traverse the environment.

There was an earlier design used for these tiles that was calibrated for the old design of Cyclindabot, however when re-tested with the new design the success rate was too high. The change in design made the robot more capable and hence the tile designs too easy to traverse. As a result, the tiles were recalibrated to be the right difficulty for the new design. For this calibration step, each generated terrain was tested once with each leg angle between 0 and 130 in increments of 10 degrees.

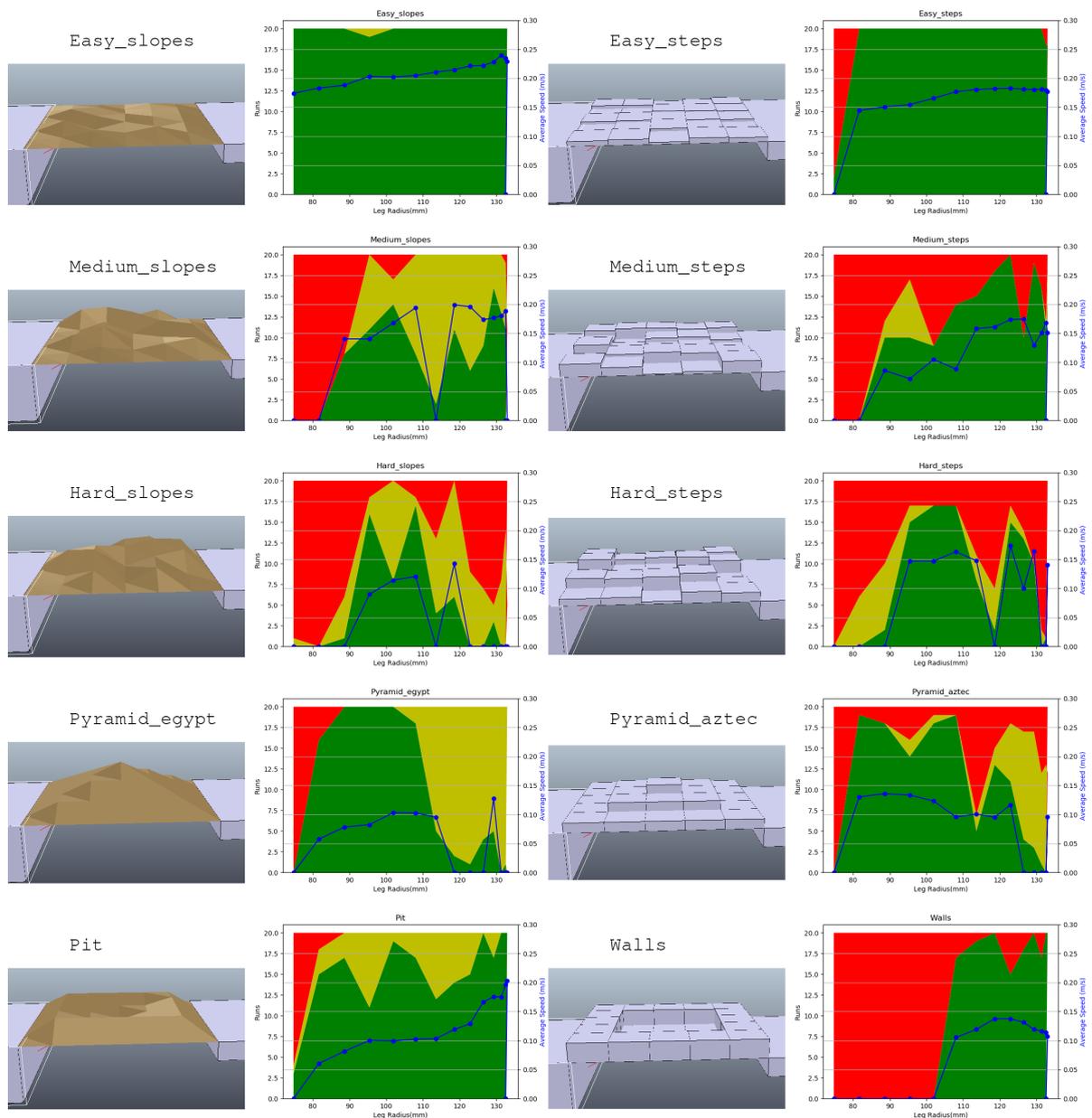


Figure 94. 10 standard terrains for map environment. Green = success, yellow = time out, red = fall, blue = average speed. 20 runs per leg angle.

In Figure 94, each of the tiles were tested against all configurations of leg angle. Both the easy tiles are completed in almost every configuration of Cyclindabot. Note that the

Egyptian style pyramid is not entirely symmetric, with the left hand ridge having flat triangles cut out of it. This is due to how the height field is generated in CoppeliaSim with each square of four corner heights always split into two triangles in the same direction. Hence the right hand ridge being straight and the left being cut into sections.

The sloped terrains had more time out failures, while steps had more falls. Time out refers to it going over the allotted time of 30 seconds, whereas a fall is when the height of the robot becomes negative and it must have fallen off the side. It was counted as a success if it reached the second flat tile.

When successful in crossing the terrain, deploying the legs had a small increase in speed. The cause of this is the increased travel distance per rotation of the Wheel and the fixed target rotational speed. This data is used in the next section to rank the leg angles so the robot could adapt as it went along.

6.4 Checkerboard Map

Nine of the previously mentioned ten tiles were randomly selected and ordered to produce a checkerboard map. The map was 5 x 5 metres with the obstacle tiles creating a diagonal square. This was done so that after each traversal of an obstacle, the robot could move easily before attempting the next. There are a total of 70 possible routes through the map, the majority of which cross three obstacle tiles, with two routes that only cross two obstacle tiles.

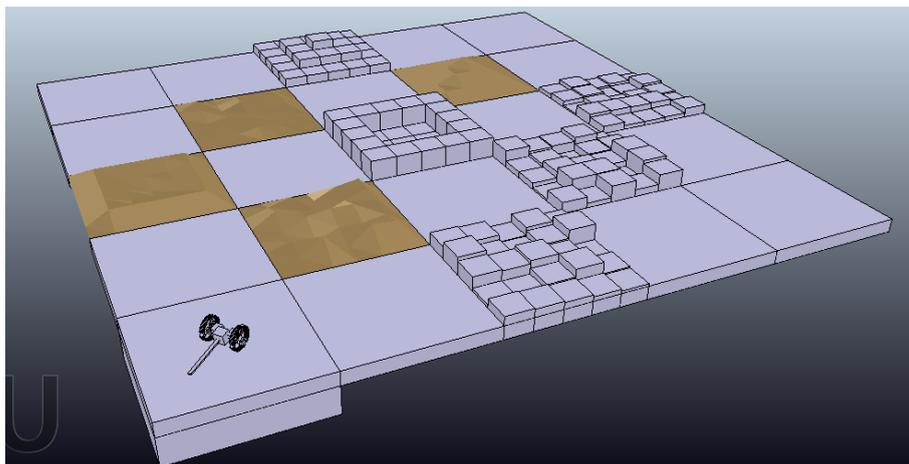


Figure 95. Checkerboard map with 9 obstacle tiles

The method of navigation used to cross the map was waypoints, which were the centres of tiles. A random route was generated then position coordinates were placed in a list of waypoints. Cylindabot used the same controller to head towards each of these target locations (Equation 25). Once reached, the next waypoint became the new target until

the robot had finished the route. The robot was tested 100 times with five different leg angles as well as with the legs adapting to the terrain obstacle.

Table 23. Checkerboard Map Results (Success rate). 100 runs per configuration (* = only 20 runs). Adapt 0/70 refer to adaptation of the leg angles with resetting to a leg angle of 0/70 between obstacles

Leg angle	Success rate (%)	Time out/Error/Falls (%)
0	6	85/9/0
30	26	71/3/0
70	64	36/0/0
100	56	41/3/2
130	51	46/3/2
Adapt 0	48	51/0/1
Adapt 70*	60	40/0/0

The different leg angles were ranked for each of the terrain tiles, first by their success rate and then by the average speed. In each case, only the highest rank leg angle was chosen. Between each of the tiles the leg angle was reset before attempting the next. It was expected that the adapting the leg angle should outperform when they are fixed. The first set of adaptation results (Table 23: Adapt 0) being surprisingly low, it was thought that resetting the leg angle to zero between each obstacle might have been the problem. Therefore a second simulation run was done with the leg angle set to 70 between obstacle tiles. This did improve the adaptation success rate to 60%. While in wheeled mode and coming off a terrain tile it is possible for Cylindabot to flip over. This error could account for the difference between adapt 0 and adapt 70. In Chapter 7, a new algorithm is added to flip the robot the right way up.

Having the reset angle being 70 degrees still did not outperform setting the leg angle to 70 for the entire time. The results here were unsuccessful and did not show that adapting the leg angle in this way improves the success rate. It was discovered in the next section that 70 degrees was optimal for random grids of this density or more and a possible configuration of grid densities is proposed to be used in future work.

6.5 Grid Density

The grid density is the number of points in a grid that define the heights of an obstacle. In this section, the size of this grid is investigated to see how it affects the robots performance. So a terrain tile with a single height will either be a single step or pyramid, whereas when 81 heights are used in a nine by nine, the terrain becomes a more complex terrain landscape as seen in figure 96.

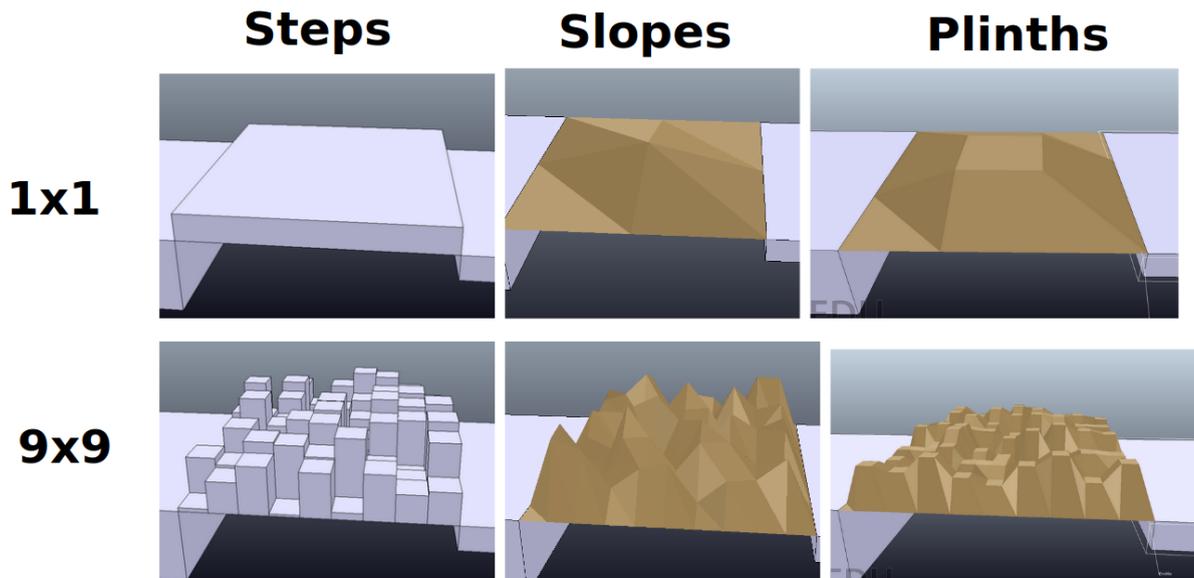


Figure 96. The three different grid styles with grid size/density of 1 or 9.

For a sloped terrain, the size of the grid can be seen as analogous to the roughness of the surface. So the higher the grid number, the more rough the surface will be. It is expected that legs will do better on the stepped terrain. The hypothesis is that as the grid size increases, the legs will have a higher performance over wheels. Three leg angles were chosen for this experiment; zero, 70 and 130. These represent wheeled mode, maximum step performance position and legs fully deployed respectively.

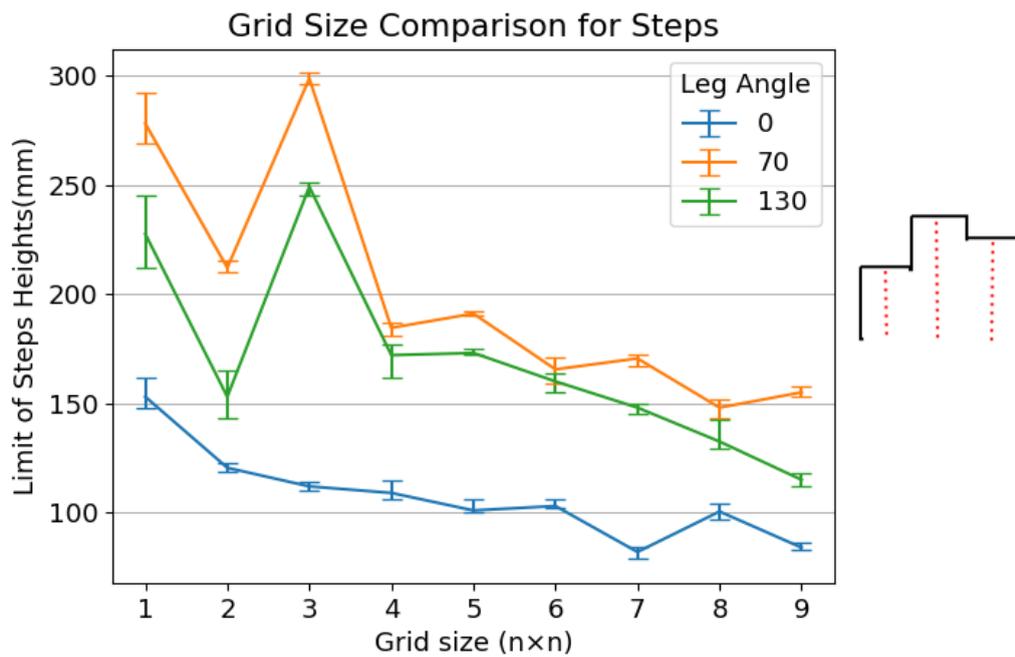


Figure 97. Grid size performance on steps. 100 runs per grid size/angle and error bar show quartile range. Right: diagram show how the red dotted heights generate the obstacles

For steps, the highest variability was when the grid size was one; this is due to only one random height being picked. The heights were selected from a uniform distribution so the difficulty has a higher variance when only a single number is selected. There is a dip in performance on even grid sizes, which is due to the robot having to traverse the tile with unequal footing. Unequal footing in this case being steps of different heights with one Whег on each. Overall, the 70 degrees leg angle was superior for every grid size and the three lines are trending down. This trend shows that step obstacles become more difficult with higher grid density and the trend continues for the other two grid tile types.

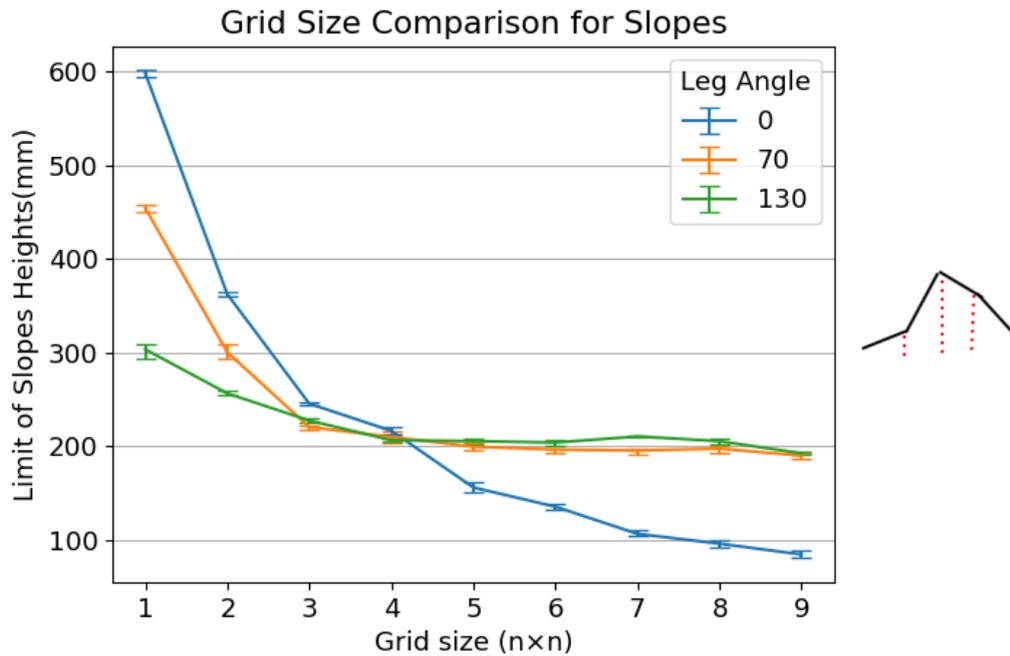


Figure 98. Grid size performance on slopes. 100 runs per grid size/angle and error bar show quartile range. Right: diagram show how the red dotted heights generate the obstacles

The slopes grid results are quite different (Figure 98) from those for steps. Of prime importance is that, for grid sizes lower than four, wheeled mode (leg angle of zero) outperforms using legs (by a factor of 2, leg angle 0 vs 130). This corroborates the results of a single angled slope (Chapter 5). However, as the grid size is increased, the ability of wheels falls faster than that of legs, leading to legs surpassing wheels at a grid size of five. With the exception of grid sizes one and two, the two leg deployed modes seem to accomplish the same results. They also seem to plateau at a height of 200, where the wheeled ability continues to decline. The final difference is that the scale of the graph for slopes is roughly double that of steps and plinths. For plinths, this is logical as the angled parts of the plinths are half the length used in slopes and this ratio is just a coincidence for steps.

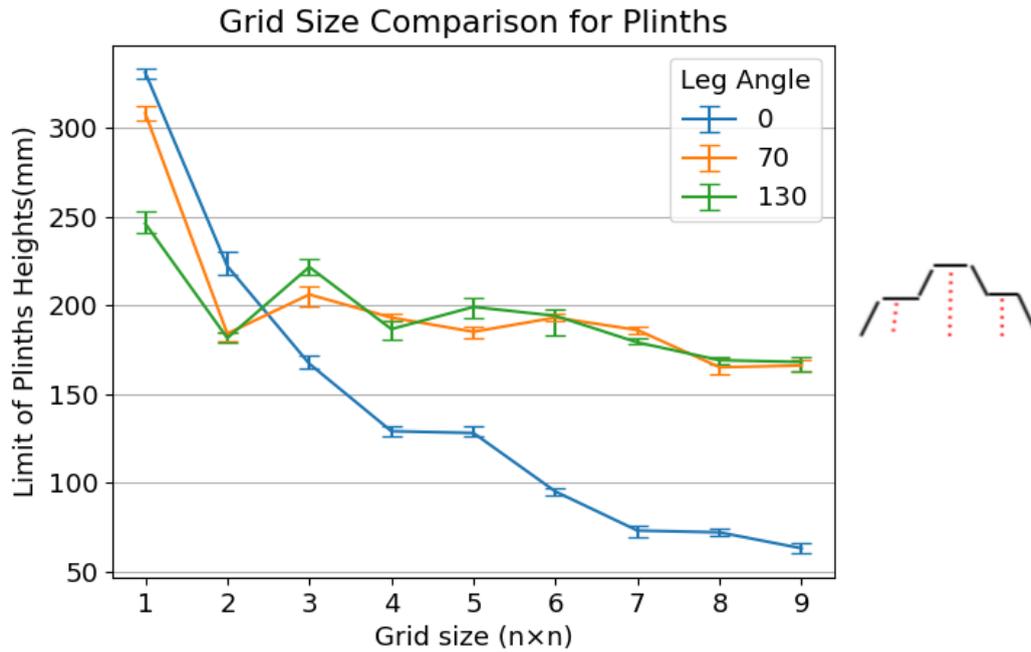


Figure 99. Grid size performance on plinths. 100 runs per grid size/angle and error bar show quartile range. Right: diagram show how the red dotted heights generate the obstacles

The grid of plinths was designed as a compromise between steps and slopes. The results demonstrate features that were present in both of them. In wheeled mode, Cylindabot had the same behaviour as the sloped grid with a curved decline in performance, although the heights are lower for plinths compared to slopes. The lines for legs when deployed on plinths have features from both steps and slopes. There are the same drops for even grids (steps), and the difference between the two leg angles does not have a significant effect (slopes). The 70 degree leg angle is superior to the 130 only for a grid size of one. This shows that the plinth concept was successful in forming a combination of steps and slopes.

The issue with using a five by five grid earlier in the map experiment was that, at this grid size, the leg angle of 70 degrees was equal or superior to the other configurations. A concept of selectively choosing one of each grid size instead of 5x5 was derived but never implemented. This would be choosing slopes, plinths and steps in that order repeated for the grid sizes from 1-9 (so 1,4,7 = slopes, 2,5,8 = plinths, 3,6,9 = steps). Instead, ridge terrains were used for experimenting with Cylindabot path and configuration planning through a map of obstacles. These grid terrain obstacles did provide useful insights to how the complexity of a terrain's design can affect its difficulty.

6.6 Summary

A grid style terrain is a more complex challenge for Cylindabot. It is defined by a matrix of heights that are randomly generated. As such, it is stochastic in nature and

unpredictable. There are three styles in which these heights can be translated into a simulated terrain tile; steps, slopes and plinths.

In the first section of this chapter, Cylindabot was shown to outperform its old design as well as a four wheeled design. This comparison was done on a five by five grid size for both steps and slopes.

Next, a set of fixed terrain obstacles were generated. The six that were randomly generated were graded from easy to hard. The other 4 were designed for a specific style. These 10 obstacle tiles needed to be re-calibrated to the new design so they were at the right difficulty. Unfortunately, when tested in a checkerboard map, adapting the leg angle did not outperform the legs being fixed at 70 degrees (the optimal angle for steps).

It was discovered that when the grid size was varied, it was after the five by five grid size that the obstacle could be treated as a single step. Hence, optimal leg angle was 70 degrees. Leg mode was always superior when it came to a grid of steps, which was expected for all grid sizes. Wheeled mode began as better for slopes, but became surpassed by legs after a grid size of 5 for slopes and 3 for plinths. The two legged modes, of 70 and 130, also converged to become the same ability for slopes and plinths.

Although unsuccessful in adapting to a map for grids, in Chapter 7, Cylindabot is tested against ridges and bridges. In the next chapter the effect of friction on obstacle traversal is investigated.

Chapter 7: Ridges and Bridges Map

Six different obstacles show how Cylindabot can adapt to a map.

7.1 Introduction

Hybrid locomotion robots come with the promise to traverse a range of terrains not possible by robots with a single mode of locomotion. This ability has been shown for single obstacles (Chapter 5) and tiles of heights (Chapter 6) however it has rarely been applied to more complex scenarios. In this chapter Cylindabot is tested in an arena consisting of varied terrain obstacles. The idea is to show that adapting the robot's form of locomotion is more important than changing the possible routes around the environment.

Six different obstacles were used in this task. Originally four obstacles were chosen and once initial testing was complete, two additional obstacles were added. The first four consisted of square ridge, 45 degree sloped ridge, round ridge and a low bridge (Figure 101). The extra two obstacles were a gap and a long slope. The variety of obstacles all have properties that make them unique and have also been used in the literature. In [13], a step, round ridge and a slope were used to compare hybrid robots with single mode of locomotion robots and made the comparison irrespective of size by including the robot height in the calculation. In the context of planetary exploration, [91] compares robots on steps/slopes as well as several qualitative measures. In [3] a variety of robots ability on steps, slopes and gaps for a range of styles of locomotion were compared (however there is a lack of detail on robot specifications).

The obstacle types were chosen because they offer a variety of challenges for the robot to traverse. These obstacles force the robot to adapt to be able to traverse them. Each terrain needs a different strategy, and when they are put together to construct a map, adaptation is required alongside following a route.

The obstacles are analysed in two different ways: the beermat algorithm, and secondly a full grid success rate method. The beermat algorithm increases and decreases the difficulty to determine the maximum obstacle difficulty that can still be successfully transversed. This approach saves simulation time and determines the failure region. The full grid algorithm tests Cylindabot at fixed increments within these calculated limits from the previous algorithm to estimate a success rate for that specific configuration (robot & obstacle). This style of success rate is used for the gaits of the hexapod robot in [113] for climbing over a square ridge obstacle.

After each obstacle was tested, this data was then applied to a map of 24 of them (Figure 107). These maps were randomly generated and at each waypoint, Cylindabot could turn left or right. Three different approaches to how Cylindabot could adapt to this map were applied. The first method was a control test where both the route and leg angle were randomised. The second had a random leg angle set and used the success rate to plan a route. This path planning only used local data. Different local and global path planning algorithms were reviewed in [114]. The final approach randomised the route and had Cylindabot choose the optimal leg angle for crossing that obstacle.

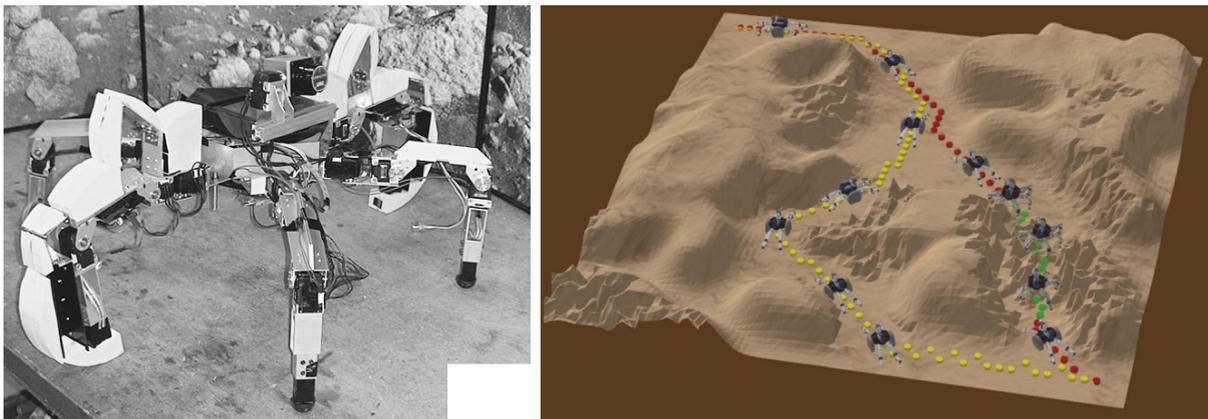


Figure 100. LEON robot in hexapod mode and two paths determined from action planner [12].

Within hybrid locomotion there has been little research into testing robots across a map environment. The only one found at this time was [12] where the LEON robot could both roll with two wheels or walks using a hexapod gait. Terramechanics is used to generate surface contact interaction. An action planner was used to determine two routes across a height field map. The first route had LEON stay in wheeled mode to reduce energy usage whereas the second had it transform to legged mode and take a shorter route. The work with LEON in [12] is groundbreaking and includes physical testing of the platform. The only issue lies in the fact that only two routes were tested on a single map. In this chapter, 90 different simulation runs are applied to 30 different maps.

7.2 Ridges & Bridge

Initially, four different obstacles were decided upon. The choice of obstacles were for two reasons; ease of simulation and unique characteristics. The characteristics of each shape proved to give different optimal leg angles for each of these obstacles. The first three were shape variations on a ridge and the last was a low bridge that the robot had to make underneath. The three shapes were square, slope and round. The square was a cuboid where both the height and depth of the ridge were increased together. The slope was a cuboid partially obstructed by the obstacle tile floor to make a 45 degree wedge. The

round was a semicircle created using a cylinder that again was partially exposed above the floor of the tile. The low bridge was a tall thin cuboid that hung above the floor of the tile (Figure 101). To make it clearer in this chapter a colour code is used for the different obstacles.

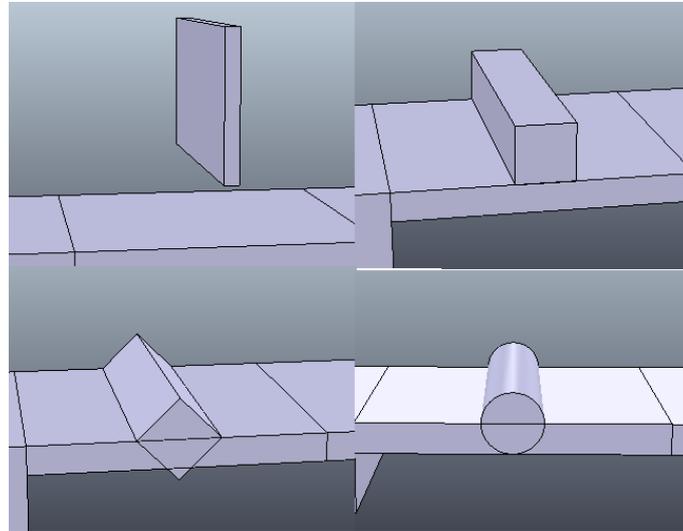


Figure 101. Screenshot of four ridge types: bridge, square, sloped and round.

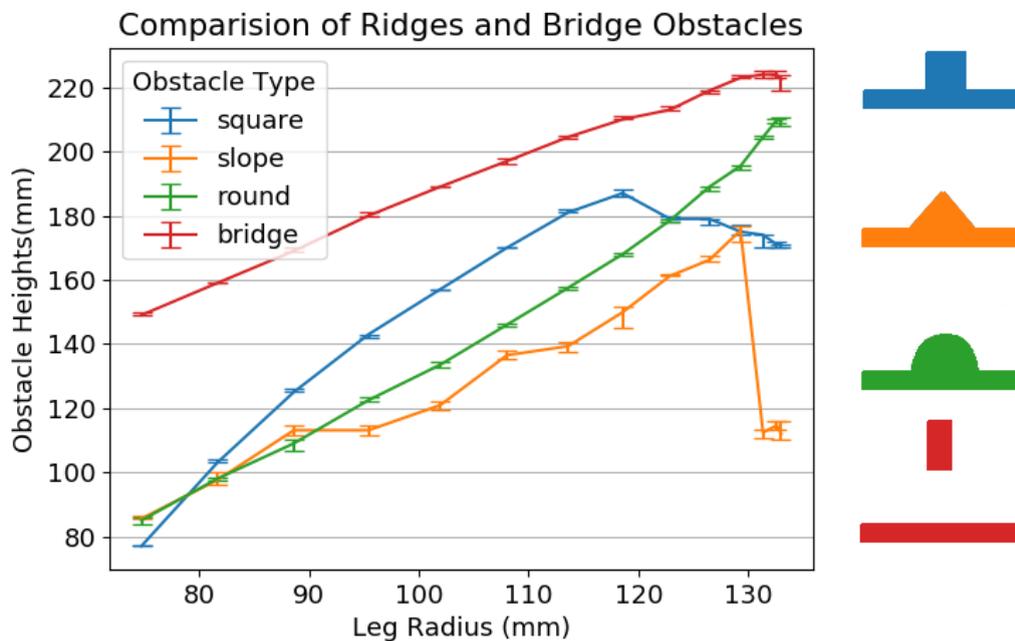


Figure 102. Ridge and Bridge results of obstacle height from the beermat algorithm. 50 successful runs per plotted point. Showing median point and error bar are quantiles.

The beermat algorithm was applied to generate the data in Figure 102. The results for a bridge are obvious. As the leg radius is reduced, Cylindabot can make it under a lower bridge. The interesting result from the bridge is the ratio between the leg radius and the bridge height. The ratio starts at 2 which is what you would expect for the wheel but as

the leg deploys it transitions to become $\sqrt{3}$. The square root of three is the ratio in an equilateral triangle between its edge length and the distance to its centre. This ratio is between the radius of the leg and the maximum height of the Wheg shape when legs are deployed. Future work would have to see if these shape ratios hold up for other shapes other than a circle and triangle.

The square ridge results match the step results both in characteristics and value (see Chapter 5). This means that the step down off the ridge does not have a significant effect. The round ridge on the other hand has a similar climb to its square brother, but without the surmount drop off. The curve in the round ridge's shape means that the robot cannot fall back in the same way as it would on a square step. Finally, the sloped ridge has a sharp drop off near fully deployed legs. By watching simulation runs it was discovered that this is due to the robot twisting in the roll axis and getting stuck on its side. Later experiments show that this occurrence happened for a band of heights then stopped again. If these results were analysed using the metrics in [13], Cylindabot would score high on steps, round ridge and inclines.

7.3 Full Grid Analysis

To be able to apply these obstacles to a path planning algorithm, more information is required. This meant gathering information not simply from the failure region but across a wider range of possible permutations. For the legs, angles were split into five different values 0, 40, 70, 100 and 130 degrees. The number of leg angles was reduced to remove combinations evenly across the range. These angles also include the high and low points from the beermat results (Figure 102). Using this previous data, the maximum and minimum useful values were determined from where the robot should succeed to where it should fail (with all leg angles). For the low bridge this was between 140 - 240 and for the ridges it was between 80 - 220. The beermat algorithm was replaced with a full grid approach where every combination of these variables was tested in simulation 10 times. This generates a success rate out of ten which is good enough for making decisions for path planning.

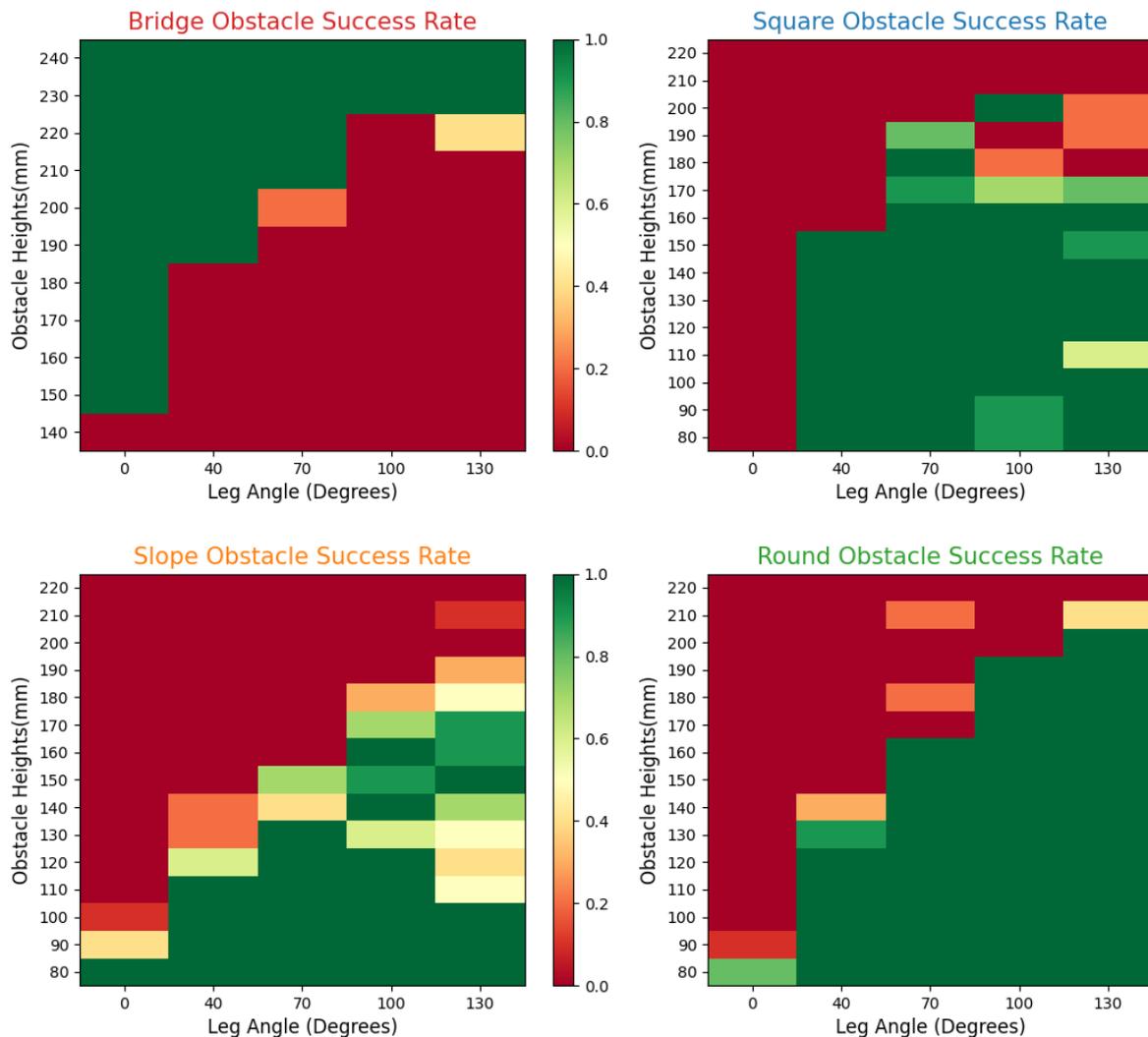


Figure 103. Success rate for Cylindabot crossing ridges and bridges. Dark Green = 100% success and Red = 0% that is all failed attempts. With 10 simulation attempts made at each combination of leg angle and height, which equal 550/750 runs per graph (11/15 heights).

Figure 103 shows the success rate of the bridge and ridges terrains. The bridge graph shows clearly that a higher bridge is less challenging which is the inverse for the ridges. The bridge is the only one of these terrains where the wheeled mode of Cylindabot is optimum, whereas for square ridges the wheeled mode has zero success. There are a few unexpected failures far away from the expected failure region for square ridges. As expected, 70 degrees was the optimal value which correlates with the previous result from the beermat algorithm. The outlier of success rate in this graph occurred when the leg angle was 100 degrees and the square ridge height was 200mm. It flips from having zero successes to a success rate of 100% with only a single increment of difficulty. By watching the simulation play out, it appears that the legs with these parameters catch the step further forwards and the robot is able to pull the tail off the ground successfully without falling backwards. Hence, it supersedes the surmount rule [2] in this niche situation.

Table 24. Overall success rate of leg angles to ridges and bridge obstacles. Yellow elements highlights show optimal leg angle for an obstacle

Leg Angle	0	40	70	100	130
Bridge	0.909	0.545	0.382	0.182	0.218
Square	0.000	0.533	0.78	0.713	0.647
Slope	0.100	0.333	0.473	0.633	0.587
Round	0.060	0.413	0.627	0.800	0.893

The results for the slopes casts further light on the harsh drop off experienced in the beer mat results (Figure 102). When the legs are fully deployed there is a region of failure where the robot flips on its side and becomes stuck. This is due to the point at the top of the slope ridge and the height of the ridge being in a certain region. Being at this point meant that Cylindabot rotated in a roll direction and was unable to move as it was on its side. Once the height was increased, the robot would fall further when dismounting the ridge, land on both Whegs and hence not get stuck.

These four obstacles were used to obtain some initial results for a map environment and preliminary test the map simulation. Three maps were randomly generated and Cylindabot adapted its route, leg angle or neither when attempting to cross it. Further details on how this map system functions can be found later in this chapter. These first simulations were successful with the success rate of adapting the leg angles out performing choosing an alternative route. Adapting leg angle got 49.3% whereas route adaptation got 14.5% and no adaptation only got 1.4%. This was only over 3 maps so it is only preliminary.

It was decided that the set of obstacles should first be expanded before the main set of map simulation experiments were conducted. The wider the set of possible terrains the more universal the results will be. The gap and long slope were chosen as they add two styles of obstacle that were missing from the previous set.

8.3.1 Extra Obstacles - Gap and Long Slope

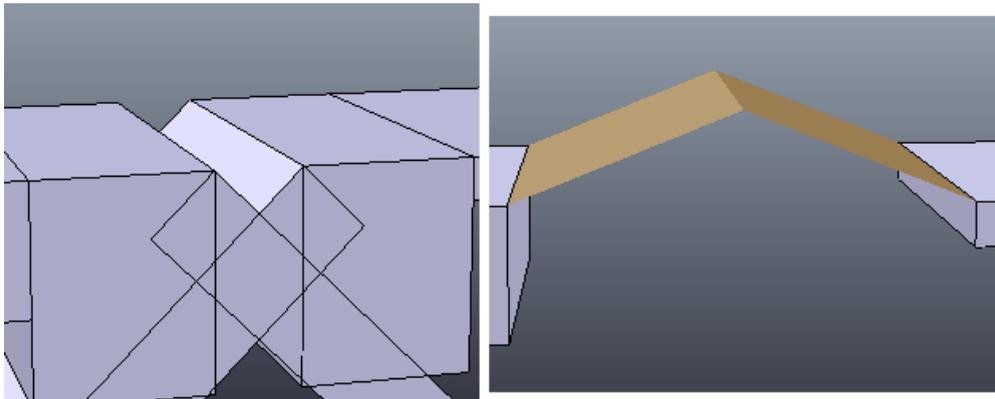


Figure 104. Gap and long slope in simulation. A set of four cuboids were used to create the 45 degree gap. A height field was used for the long slope.

Two extra obstacles were chosen to add to this set of ridges/bridges. These were a long slope and a gap (Figure 104). The slope used a height field similar to that used in Chapter 5, however the slope rises then falls back down. The length of the slope was fixed and the height was set in the middle of the tile. The gap was made by two cuboids set at 45 degrees.

There was some brief simulation work on a square gap traversal. The problem was that the Wheg would become stuck and lead to a physics error in simulation. Either the robot would become stuck in a physically impossible position like inside an obstacle or fly off after a build up of force. The first of these is a potential problem with the Wheg design that it is possible for it to become wedged in certain sized holes or gaps. This is similar to the approach in [62] where large contact points are proposed to be better at crossing gaps. It might be possible to use a square gap if simulation parameters were altered; however it was decided to use the 45 degree slope instead.

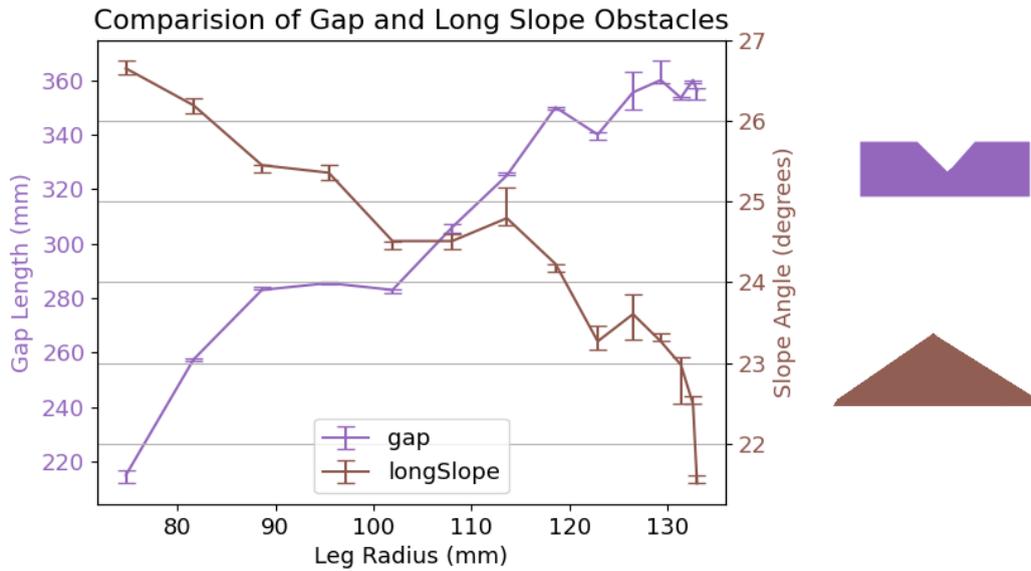


Figure 105. Gap and long slope obstacle height from simulation. Beermat algorithm with 50 successes.

Figure 105 shows what the previous results (Chapter 5) would have predicted for slopes: deploying legs has a negative effect on climbing a slope. The results are less predictable than with a full length flat slope with the results not being strictly decreasing. Both data sets have plateaus before continuing their trends. The gap obstacle has an increasing trend suggesting that deploying the legs is favourable.

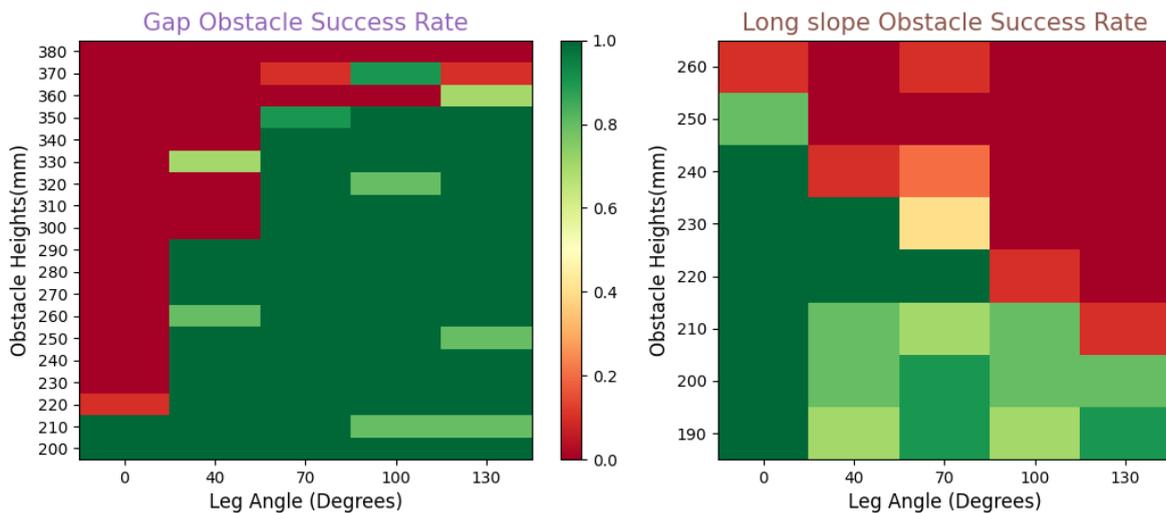


Figure 106. Success rate for Cylindabot crossing gaps and long slopes. Green = 100% success and Red = 0% that is all failed attempts. With 10 simulation attempts made at each combination of leg angle and height (19/9 heights).

To generate the limits of these full grid results in Figure 106, the data from the beermat algorithm was used (Figure 105). Without this information, the computational burden would be increased significantly. The gap data still took twice as many simulation runs to

complete because there are twice as many possible heights due to a wider success range. Leg angles of 70, 100 and 130 degrees showed a similar performance on the gap obstacle.

The long slope has the most stochastic results compared to the other obstacles described in this chapter. There is an overall downward trend as the legs are deployed. However, only in wheeled mode (leg angle 0), was there a solid area of success before failing. It is not clear if this is due to inherent features of the obstacle or that the control algorithm needs improving for the long slope obstacle. There are clear optimal configurations for Cylindabot on both obstacles (Table 25), therefore they will be useful for the map task.

Table 25. Success rate for gap and long slope obstacles

Leg Angle	0	40	70	100	130
Gap	0.11	0.55	0.84	0.87	0.86
Long Slope	0.86	0.55	0.53	0.30	0.23

7.4 Map Setup

The idea behind the map simulation is to give Cylindabot choices in the obstacles it faces or route that it can take. The map is set out like blocks of an American style city, alternating between obstacles and flat tile. At each point the robot has two options for the route it can take: X or Y directions. The ridges and bridges used here only work when approached from the correct compass direction so each obstacle tile has empty space on either side of it and is orientated accordingly. The task of Cylindabot is to move from bottom left to top right. The total size of the map is 7 x 7 metres with each side made up for 4 flat tiles and 3 obstacles across each side. This means that there are 20 possible routes through the map, each crossing 6 obstacles.

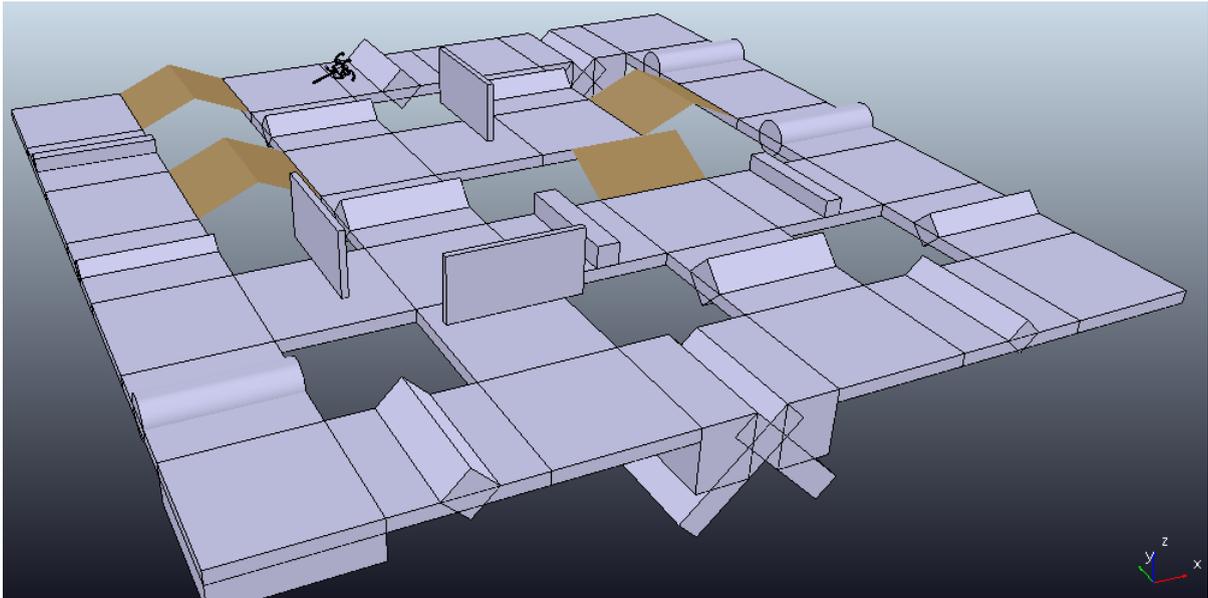


Figure 107. Screenshot of a full map with Cylindabot approaching the fifth obstacle of six.

The simulation setup had to be altered to achieve this new task. Occasionally, after crossing an obstacle, Cylindabot would complete it upside down. Although still able to move, being the other way up might affect its ability to traverse the next obstacle. An algorithm was added to the controller to flip Cylindabot the right way round before it continued across the map. Another python script was added as a watchdog program; it would restart the simulation after a given time or if an error occurred. This was because the API with CoppeliaSim occasionally (once every several hours) dropped messages. The dropped message would lead to the simulation stalling. The watchdog program simply restarted it when needed.

A key python script analysed the map generated and used the information from the success rate data. The program was used in two ways: generating a route or producing a set of leg angles. To generate a route, it used the local information of the two obstacles in front of Cylindabot and chose the one most likely to be successful (using the success rate for that combination of leg angle and obstacle height). If the probabilities were equal, it chose to take the route on the left. For choosing leg angles it chose the most likely to succeed at that obstacle height and if there was a tie, the leg angle with the highest overall success rate on that type of obstacle was chosen. The difficulty of the obstacles was trimmed from the full grid results (Figure 103 and Figure 106). This was so that there were no obstacles that were impossible to cross and all approaches were fair.

7.5 Map Results

Three different approaches were used in the map environment experiments. The most basic was where the route and the leg angle were randomly determined. The route was

created with an even probability of turning left or right rather than taking all possible routes and choosing one. This means that the complete routes through the centre of the map are less likely because they have more branch points. The leg angle was randomly chosen from the five tested leg angles (0,40,70,100,130 degrees) at the start of the task and remained the same for all obstacles. The second approach had a randomly determined leg angle and then used this to make the best turn choice. The success rates of turning left or right were compared. The final method optimised the leg angle of these same success rates and the route was randomly determined.

Table 26. Overall Success Rate of Cylindabot using different Map Technics.

Adaptation	Success Rate
None	6.3%
Adapting Route	17.9%
Adapting Leg Angle	58.0%

The success rate in Table 26 was calculated by having Cylindabot attempt 30 different maps each 30 times with all three adaptation types. The data shows that adapting the legs outperforms the other two methods. Adapting the legs is nine times more likely to succeed than no adaptation and three times more likely that adapting the route.

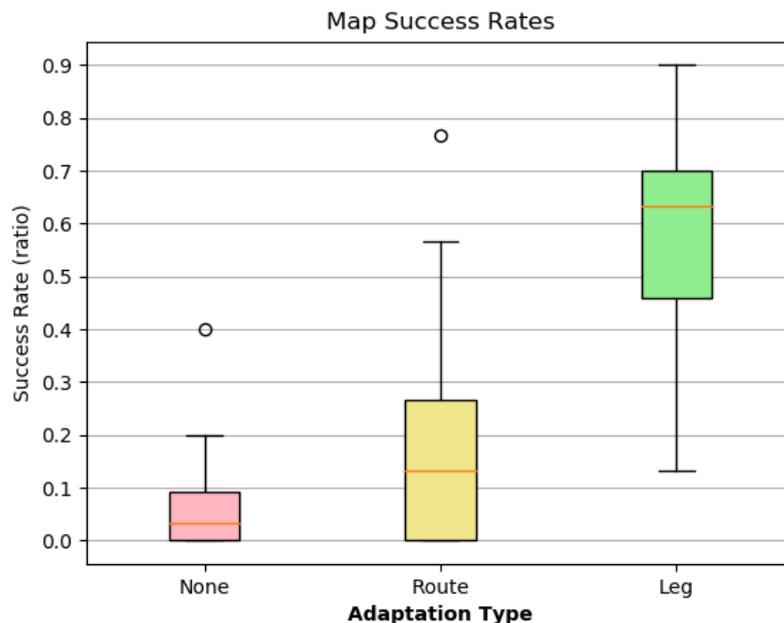


Figure 108. Box plots of success rate on each individual map. Calculated by taking the number of successes and dividing by the 30 attempts.

In Figure 108 the success rate of each map is plotted and shows that leg adaptation has a higher chance of completing the map. There is overlap between the different methods due to certain map configurations being more difficult than others. To demonstrate this difficulty, the number of waypoints reached was plotted in figure 109. Another way of setting waypoints is the path length before Cylindabot either failed or completed the map. The map had a total of six waypoints. The maps were ordered by the median waypoints reached when adapting the route. This order was chosen as the route adaptation had the highest variation and allowed the data to be more clearly displayed.

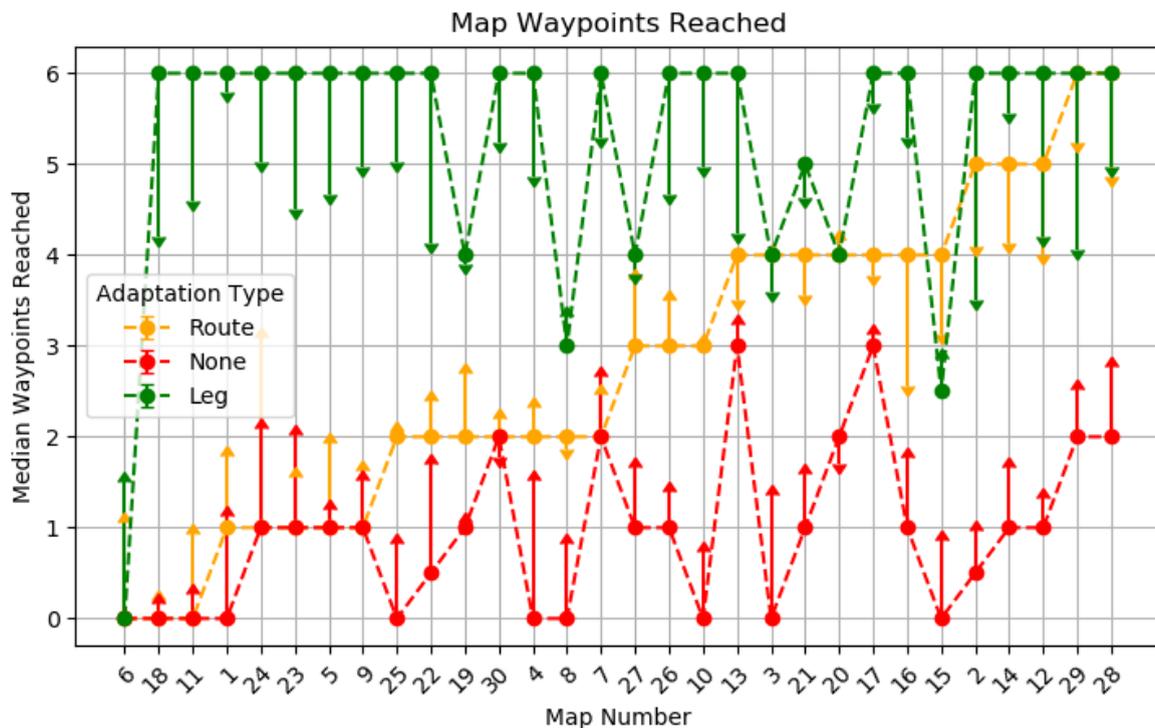


Figure 109. Number of waypoints reached over the 30 maps. Dots represent the median number of waypoints reached and the arrows point to the mean to show data skew. The maps were ordered by the difficulty for the route adaptation.

In figure 109 the hardest map was number six where the median path length was zero for all three approaches. Map fifteen was the only map where the median waypoint of the route adaptation was higher than the leg angle adaptation. For the other twenty-nine maps leg angle adaptation was the same or better. The skew shown by the arrows was larger than expected, showing that the random elements of the method cause a stochastic outcome. The ease of maps was not consistent across the three methods. With the exception of map six, the hardest maps for route adaptation were successful for leg angle adaptation. The medium difficulty maps had drops in performance for leg adaptation and some of the highest points for no adaptation.

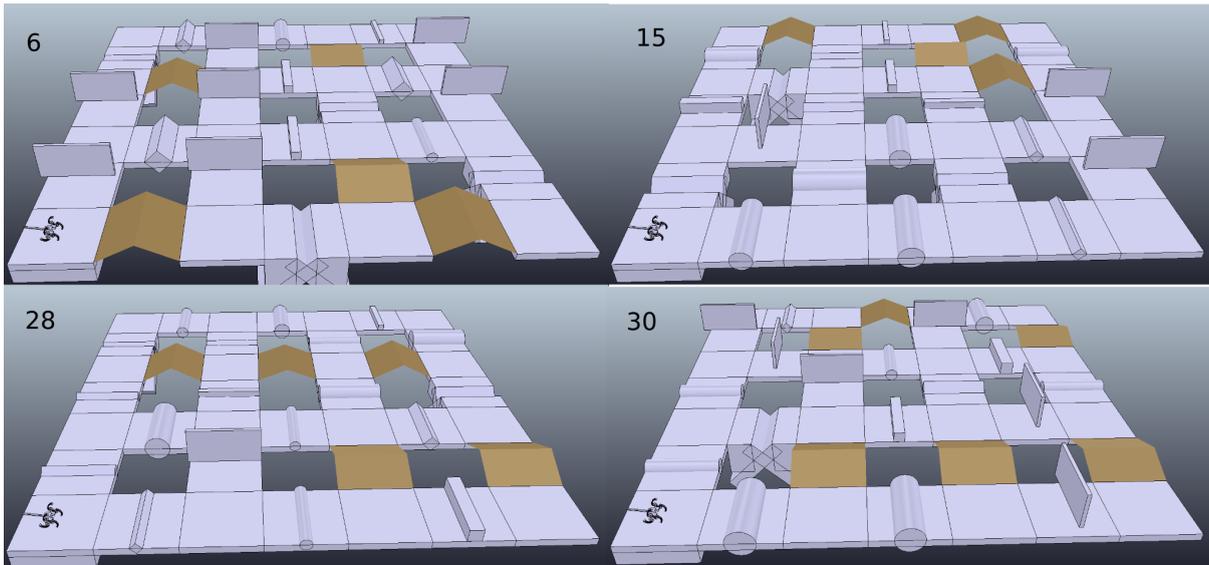


Figure 110. Four different maps numbered 6, 15, 28, and 30. Map 6 was the hardest map whereas map 28 the easiest. Map 15 was the only where route adaptation beat leg angle adaptation. Finally map 30 was a medium difficulty map.

7.6 Summary

In this chapter, six unique obstacles were analysed to determine how best they could be traversed. These obstacles were picked for their variety and unique interaction with Cylindabot. The beermat algorithm was used to find other failure regions for the terrains for all leg angles. The limits generated by this were then applied to a full grid approach to produce the success rate of different configurations. These success rates were then applied to a more complex map environment. Path planning was compared against adapting the leg angle of Cylindabot. Both route and leg angle adaptation used the success rate to optimise decisions. It was found that adapting the leg angle was more successful for almost all of the maps used. Each of the ridges/bridges were selected to give a different challenge to Cylindabot so that a single strategy would be unsuccessful.

This Chapter is the first time that a hybrid robot has shown that adapting its configuration is more successful than planning a different route. Currently the Cylindabot prototype can only transform between two fixed Wheg positions. Combining hybrid locomotion with path planning has rarely been researched hence these results are an important step towards furthering that topic. The only previous work found was the LEON robot [12] which had other features and limitations compared to this chapter's work. There is more research needed to be done in hybrid robots in a map environment (one where choices of route can be made). Cylindabot has already shown that it can adapt to obstacles (Chapter 5 & 6) and now for a map environment. In the next chapter future work will be discussed.

Chapter 8: Conclusion

8.1 Thesis Summary

Hybrid locomotion robots are platforms that have more than one way of moving. They are often limited by having multiple actuators for every mode of locomotion. In the literature there is an array of different robots each with unique characteristics. The problem is there are no standardised benchmarks for comparisons. The promise of hybrid locomotion has not yet been applied to real world applications.

Cylindabot is a new hybrid locomotion robot that can walk and roll with a minimal number of actuators. It achieves this by having Whegs that are able to transform between a large radius wheel and three legs. A fixed straight tail is used behind the robot as a third point of balance. So that multiple configurations could be investigated, the leg angle and tail length could be locked in different positions. The physical prototype's ability on slopes and steps was measured.

A digital model of Cylindabot was generated in CoppeliaSim (VRep) so that the robot could be tested in simulation. On slopes and steps the simulation results were compared to the physical versions and verified that the simulation was accurate, illustrating that there was only a small reality gap. It was concluded that for maximal performance wheels should be used for a slope and legs for a step. Simulations allowed the robot to be tested for a larger number of repetitions and over a wider range of obstacles. Traversal of grid obstacles where the terrain is defined by a matrix of heights were shown to benefit from this adaptation. Grid obstacles were created in simulation as they would be difficult to manufacture for real testing.

Finally, Cylindabot was tested against a map environment where adapting the leg angle was more important than choosing a route. This map consisted of six different ridge or bridge obstacles, chosen to create a diverse set of obstructions. The routes are based on taxicab geometry where the robot has two choices at most intersections while attempting to travel from one corner of the square environment to the other. The leg angle and route adaptation were based on success rates, which were calculated in separate evaluations of each of the six obstacles. The ability to adapt to the obstacle in a given route is shown.

Thesis Contributions

1. Often hybrid robots are designed for the novelty and engineering challenge, not to justify the different configuration. Cylindabot is able to show that both modes of locomotion are optimal for different obstacles.

2. The work reported on steps has shown the approach/surmount theory from [2] for a three legged Wheg to be true which was only geometrically proposed previously.
3. For the majority of the results in this thesis multiple data points are calculated so that trends can be determined rather than the often used single point comparisons. For instance on a step if only the full leg or wheeled modes were tested the optimal leg angle of 70 degrees would have been missed.
4. An investigation of frictional coefficients was made for the first time on hybrid locomotion robots.
5. The analysis of a hybrid locomotion across a map was greatly extended by having more repetition on more maps. Whereas previously in [12], two possible routes on a single map design were compared.

8.1.1 Revisit Hypothesis

“Adapting a ground robot's mode of locomotion shows measurable improvement for traversing a range of obstacles.”

It is believed that this thesis has shown this to be true in the specific case of different obstacles and when traversing a map. This hypothesis was also broken down into three stages: measure, compare and generalise. For steps, a geometric inequality was shown to be true, which gives a general answer for terrain where the reach and balance of the robot is important. Whereas with slopes, resolving how friction affects the robot's capability was not accomplished and hence only the first two stages were completed. Although inconclusive for a grid map, for a map of ridges all three stages were successful.

8.2 Future Work

There is still research and development that could be undertaken with Cylindabot both physically and in simulation. In this section possible future work that could build on the work within this thesis are suggested. The future work is split by which chapter it is most relevant to.

Chapter 4: Design

Sensors:

Throughout the design process it was considered how to add Time of Flight (ToF) sensors and camera to the design. The ToF lasers could be the VL53L0X as they would work on an I2C bus and their address is reprogrammable. If only a few were needed to be connected then they could plug directly into the different I2C buses already set up on a APiHat. If more were required to be connected to a single I2C bus then the shut off pins on the ToF could be used to allow them to be readdressed. At the front a simple Pi Camera could be mounted and connected to the Pi using a ribbon cable. From a high

level perspective, the pitch and roll caused by the Whegged gait would need to be corrected for when using these sensors. Adding these ToF sensors would require no redesign of the electronics as there are several I2C buses built into the APiHat. The only other sensor that might be required is an Inertial Movement Unit (IMU) so that the robot is able to detect its orientation and tell if it has become stuck.

Moving main boards and battery:

To make space in the centre of the robot for suspension and retractable tail, both the main boards and the battery would have to be moved. They are currently in the centre of the robot, mounted on top of one another. Instead they would be split up with each of them being attached to one of the 20D motor mounts. This would mean designing two different motor mounts and a new centre of the robot. Due to the modular design of the rest of the robot this would be all that would need to change. The advantage of this suspension would be damping of vibrations and improved impact absorption. This could affect Cylindabot capabilities on slopes where the legs would slip and shake the robot.

Retractable tail:

A retractable tail would have a range of potential benefits. Firstly, once fully retracted the robot would be more robust to impacts and be able to roll down uncertain drops where a controlled descent would be impossible. Secondly, the robot would be able to fit around tighter corners with the tail partially retracted. Thirdly, it would allow the tail to actually be longer when fully deployed. This would allow the robot to fully realise the potential of the legs which currently has only been achieved in simulation. Finally, new methods of locomotion would be possible. The tail could be retracted underneath the rest of the robot. If deployed slowly it could push it up a step or if released quickly could even allow the robot to jump.

Chapter 5: Slopes and Steps

Identifying obstacle types:

Once sensors are added to Cylindabot the next stage is to use them to identify if an obstacle should be treated as a step or a slope. This could be achieved by having time of flight sensors at different pitch angles so the profile of obstacles could be determined as Cylindabot moves. The aluminium tube axles would be useful for this as sensors could be mounted on the outside of the Whegs and would not turn as the robot moves forward.

SLAM (Simultaneous Localisation and Mapping):

The data from the sensors, camera and IMU could then be combined to allow a SLAM algorithm to be deployed for Cylindabot. This would be first applied to the wheeled mode and then the legged mode of locomotion. The oscillating gait when using the legs would mean the angle of the sensors would change during motion making it harder to have accurate readings.

Chapter 6: Grids

Grid terrain map:

Use the lessons learned from the ridge and bridge map to improve the performance of Cylindabot ability to adapt. Firstly, calculate a success grid for a better understanding of how well the robot can cross in each configuration. Then, use a taxicab geometry for the map as then the robot is approaching terrain tiles from the same orientation that it was tested in. Finally, use the results from grid density so that each grid size is used once where different configurations are optimal (1,4,7 = slopes, 2,5,8 = plinths, 3,6,9 = steps). These changes should mean that the adaptation of the leg angle should be more successful.

Chapter 7: Maps

Combine route and leg angle adaptation:

Expand the current algorithm to be able to choose the best leg angle and direction at the intersection points of the map. This would continue to use the success rate data. Adding this feature should increase the overall performance while not changing the architecture of the code considerably.

Global route planning:

The current algorithm uses the success rate of the local obstacle in front of Cylindabot. Instead the success rate of all the possible routes could be calculated so that all steps of the journey are taken into account. The probabilities of traversing each obstacle would be multiplied together. This could be done with or without leg angle adaption.

Different shaped maps:

The map is a square taxicab geometry which is only one way these maps could be arranged. This shape gives a binary choice and leads to the number of routes being from Pascal's triangle. One possible arrangement is to have stages where three possible routes split and then join back together. The number of routes in this case would be powers of three. Also in the arrangement global and local route planning would be the same as the routes joint back together.

Arbitrary terrain and pathing:

For arbitrary terrain three developments would be needed. First, these new terrains would have to be generated in a way that gives different challenges. Secondly, the terrain areas would have to be categorised into smooth, jagged or impassable; smooth being optimal for wheels and jagged being best handled by legs. Thirdly, an optimal path would have to be found, taking into account transform time, speeds in different configurations and

amount of risk taken. To begin with, the plan was to generate a terrain that had a 2D profile so that the categorising into smooth, jagged or impassable could begin.

8.3 Final Discussion

Robotic locomotion has made leaps and bounds of progress over the last few decades to the point where walking and wheeled locomotion has become fully autonomous in unknown environments. Hybrid robots will soon make this transition from a controlled environment to real world applications. The work in this thesis is to quantitatively measure Cylindabot's ability to cross terrain so that, when it or similar robots are taken into the field, those predictions can be used to navigate the environment.

Cylindabot is designed to either work in an environment with humans or without. The robot is capable of navigating an urban environment. The size of Cylindabot was chosen so that it is large enough to have capabilities on stairs and other obstacles in a normal city environment. It is small enough that it does not pose a safety concern if it collides with someone and could be carried in case of an error, crash or bug. Putting it in a class of collaborative robots would mean it would not have to work completely independently.

The core lessons learnt from designing Cylindabot were design modularity, simplicity and simulation integration. Having a modular design meant each component was quick to manufacture or customise due to small print volume. The overall simplicity meant the focus was on the data collection rather than dealing with design complexity. Simulating several models of Cylindabot before prototyping and then doing further experimentation in simulation was vital to the results.

Once a benchmark for comparing hybrid robots is established in the field then different platforms can be compared to one another. During this work it is shown that a robot's ability on a slope or a step can be translated to a range of different terrains. The reach and balance required for a step is different from the grip and low centre of mass needed for slope. For steps and slopes to be used as a benchmark they need to be a fair comparison when applied to different robots. This is easier to achieve with a step as dividing by the height of the robot accounts for its size. To use a slope the length and surface needs to be consistent (1 metre horizontally and MDF for Cylindabot). The next generation of hybrid robots should focus on measuring and realising the benefits of changing their modes of locomotion rather than the novelty.

After this, standardised design for hybrid locomotion will be mass produced instead of prototypes. Examples of this include, the four wheeled design such as the Clearpath's Husky for rolling robots and the wide radius motor quadruped like Boston Dynamic's

Spot for walking. The greater availability of hybrid robots will lead to more experimentation, results and applications farthing research in this area.

Appendix

In this section results that were not able to generate clear conclusions are presented. The first set of data is the traversal of terrain defined by two parameters. The second is where the frictional coefficient on Cylindabot is varied to investigate the change in step and slope climbing ability.

A.1 Double Parameter Terrain

For the majority of this thesis obstacles were defined by a single parameter: height, which for slopes was often measured as an angle. In this section the obstacle's difficulty is defined by two parameters: height and depth. The three obstacles chosen for this are multiple steps, curved top step and a sloped step. These were chosen to extend the understanding of a single step or slope from Chapter 5. The multiple step has the application of staircases which are then compared to building standards. Curved top step and sloped step if deep enough become a slope in characteristics. The curve and slope on these steps has an interesting effect on the sumount condition from [2].

These simulated experiments generated a rich data set and created more future questions than conclusions. Each of the data sets have been split into two halves so the trends could be clearly shown as there is a wealth of overlapping points. The data was plotted against both height and angle of the height over the depth. For each, the appropriate measurement was chosen to best show the data correctly.

A.1.1 Multi-step

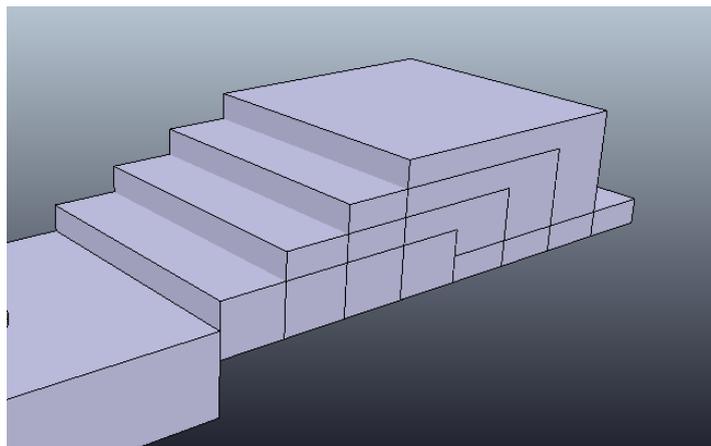


Figure 111. Screenshot of a Multi-step

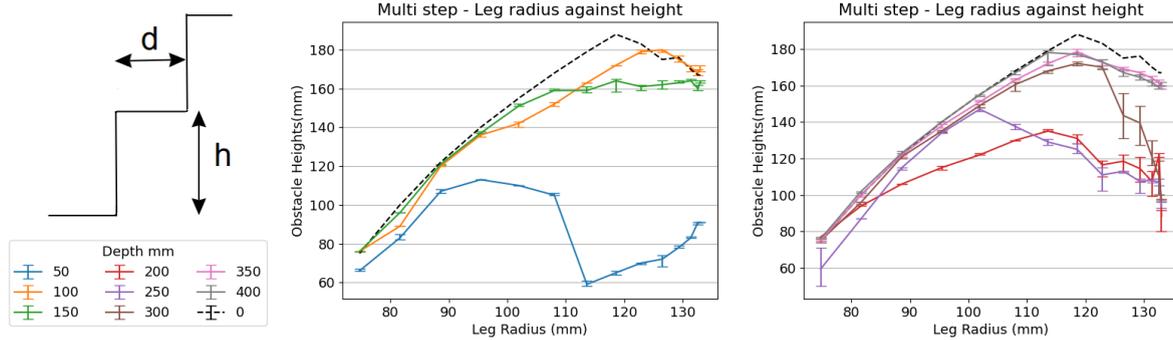


Figure 112. Multi-Step heights and depths. Beermat algorithm used with 50 successful results per point. The tail was set to 315mm. The black dashed lines are the result for a single step.

There are four distinct phases to this data. The first is the shortest step where the step height first rises, plateaus and drops before rising again. A theory for this drop off at a radius of 110mm is that leg position the centre of mass will be further back than the tail. The next phase is for step depth of 100 and 150 mm, where height climbs but does not have a previous surmount break off points. The next phases have earlier and more extreme break points for depths 200-300 mm. A theory for these breakpoints is due to the end tail being over the edge of step rather than on a flat surface. The final phase data matches a single step; this is where the depth is longer than the tail. Further simulation is needed to investigate phases one to three. One thing that would have to be changed is to limit the number of steps. In the current setup, the limit of the number of steps is the length of the one metre obstacle tile, so for the 50 mm depth there are 20 steps (Figure 113). Cylindabot was success at a climbing stair like this, showing that with these flat sections it is able to traverse a much steeper incline than without.

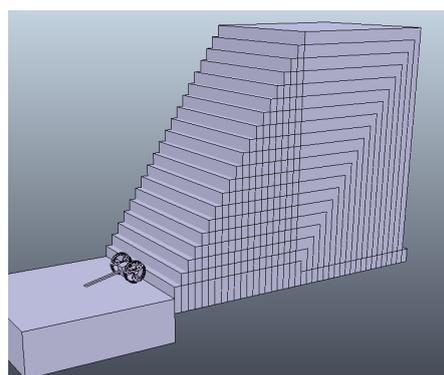


Figure 113. Short depth of 50mm causes 20 steps which might be an unfair comparison

When these results are compared to the England regulations for staircases (Table 27), Cylindabot is not able to climb all types of stairs. If the depth is equal or greater than 300 then the robot can climb a step shorter than 170mm. This covers most of the general access types to stairs. However, the reduced depth for private staircases means that

Cylindabot would struggle on most of them. One solution for this problem would be to add a motorised extendable tail to Cylindabot. This would allow the tail to be retracted, be on the current step and then extended as it climbs the next step. To be able to climb the tallest steps from private regulations the tail would need to be ~390mm long as shown in Figure 83. A new control algorithm would have to be designed for Cylindabot to synergise the movement of the Whegs and tail.

Table 27. Stair regulations for England taken from [115]. The rise is the height of each step and the going is the depth.

English Stairs regulations (mm)	Rise		Going	
	Min	Max	Min	Max
Private	150	220	220	300
Utility	150	190	250	400
General Access	150	170	250	400

A.1.2 Curved Step

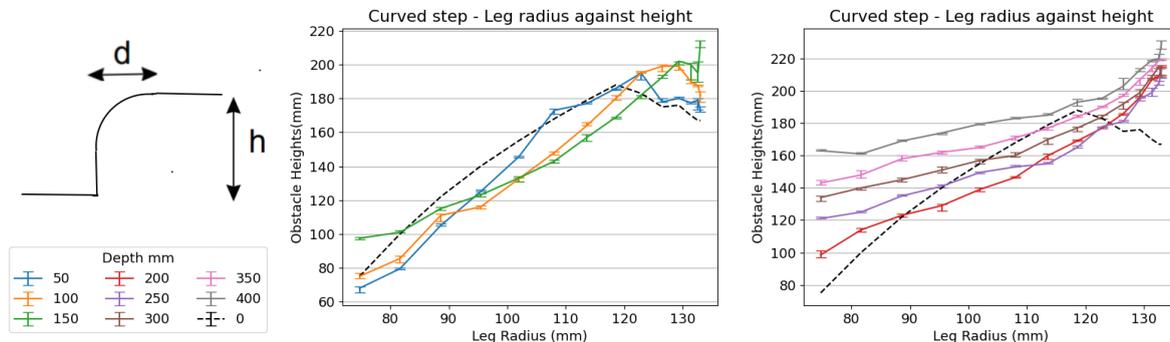


Figure 114. Curved step. Beermat algorithm used with 50 successful results per point. The tail was set to 315mm. The black dashed lines are the result for a single step.

The first three depths (50-150) reacted in a similar manner to the original single step, with the difference being the drop off point occurred later in deployment of the legs. This is due to the centre of mass being further forward so the robot is less likely to fall backwards. Once the depth was increased (200-400) there was no drop off as the legs were fully deployed. Another feature was that the points were more spread out for low leg radius than when the legs were more deployed. At these depths, the height is smaller than the depth so the start of the obstacle will be a slope. More analysis is needed to understand the relationship between the depth and height difficulty. The data was re-plotted against the angle of the curve when it met the flat surface but there were still

discrepancies. One possible theory is that the reach of the legs could be calculated and used to generate the angle on the curve it can reach.

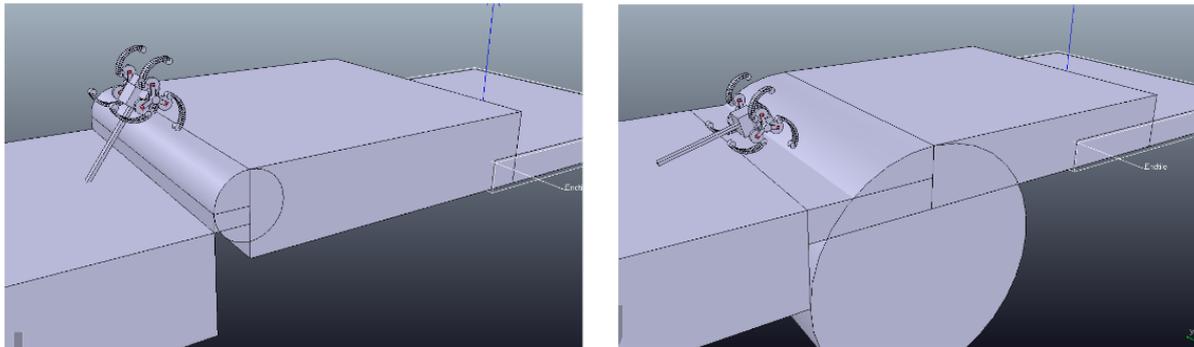


Figure 115. Screenshots of curved steps. Left: step taller than depth, Right: depth longer than height.

A.1.3 Sloped Step

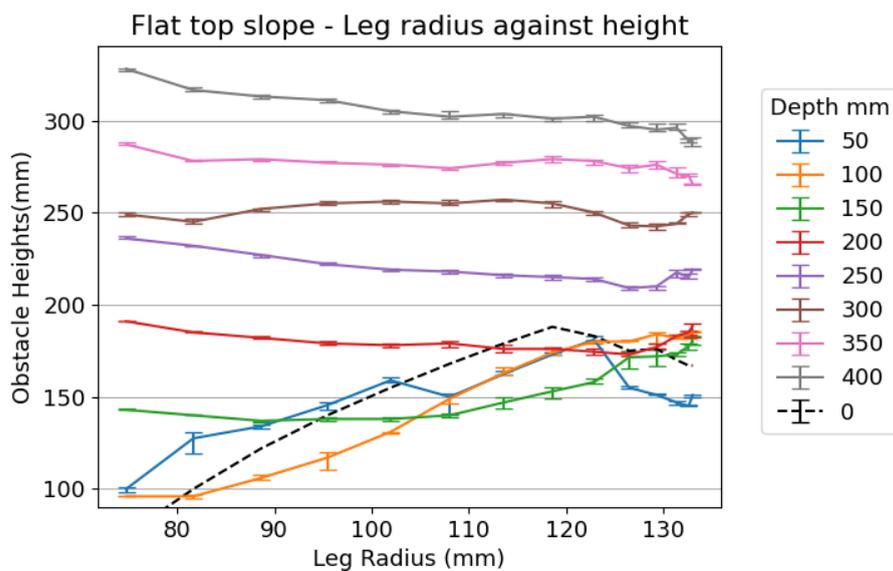


Figure 116. Sloped Step using height as y-axis. Beermat algorithm used with 50 successful results per point. The tail was set to 315mm. Dash line is single step data.

When the sloped step is plotted against the height (Figure 116) it rises significantly higher than previous steps in this chapter. This is because it is being climbed like a slope. The dashed line is the single step results from Chapter 5. The depth of the 50 mm line follows a similar trend to this single step rising then dropping off. Whereas the 100 and 150 mm depths are mono-increasing. There is a section of where 50 and 100 mm overlap precisely, the cause of this is unknown. At this point it was decided to replot this data against the slope angle rather than the step height.

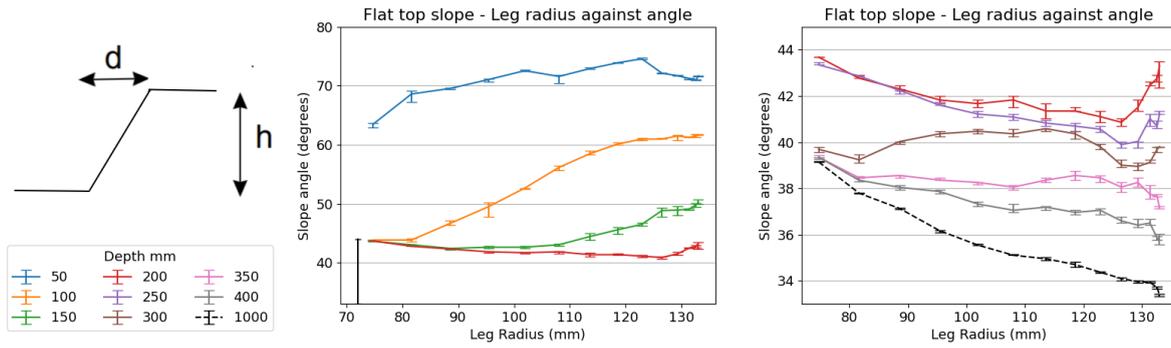


Figure 117. Sloped Step using angle as y-axis. BeerMat algorithm used with 50 successful results per point. The tail was set to 315mm. Depth of 200mm repeated between two graphs and a solid black bar shows the scale of the right hand graph in the left hand graph. The dashed line is the result for a 1 metre flat cuboid slope.

The sloped step data in Figure 117 was plotted against the slope angle and separated between the low/high depths to clearly show the results. The hierarchy of the lines have flipped when plotted against angle rather than height. There are points when Cyllindabot is in wheeled mode. Depths of 100-250 mm meet at roughly 43.5° and 300-1000 mm meet at roughly 39.5° . It is believed that this split is caused by whether the tail is on the floor before the slope or on the slope itself. After a depth of 200mm the wheeled mode always outperformed legs. This matches the sloped grid results (Chapter 6), when grid size is less than 5, hence the slope depth is $1000/5 = 200$ mm. The maximum reach of the Wheg can be calculated by $130 \times \sqrt{3} \approx 225$ mm. The ability to reach the top of the sloped step explains the rise in performance for the depths of 200-300 mm and why higher depths do not have this increase.

These slope results for a one metre cuboid are significantly higher than the previous results for a flat slope from Chapter 5. The friction of Cyllindabot's Whegs was set to 1.35 for both of these data sets. It was decided that it was both simpler and more practical to vary the frictional coefficient of Cyllindabot rather than the environment. Different models of Cyllindabot could be saved easily in a similar way to how different sizes were done in Chapter 3. Also, if different robots were to be used in these environments it makes more sense to tune each robot to the simulation scene rather than change in environment each time. The difference between the situations is that the results from Figure 85 (Chapter 5) was with a height field obstacle and the data in Figure 117 was with a cuboid set to the correct angle/length (Figure 118). This means that the physics engine must treat these two objects differently.

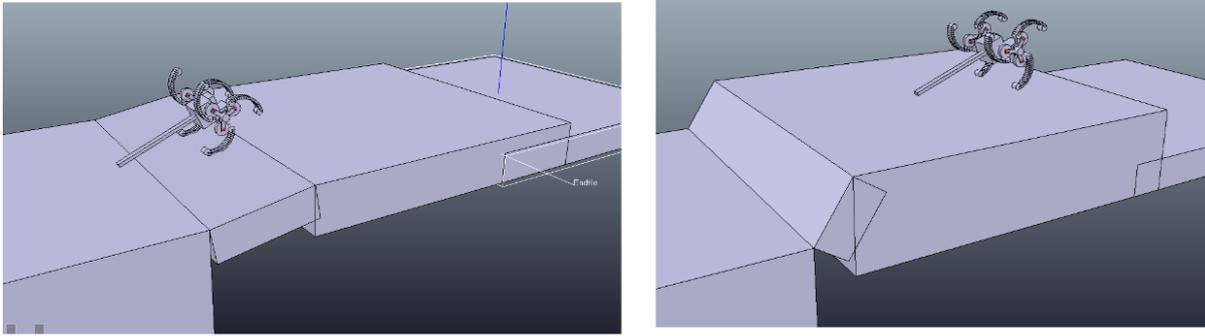


Figure 118. Screenshots of the sloped step. Left showing a configuration that would act like a slope and on the right one that would have to be handled like a step.

A.2 Frictional Coefficient

Cylindabot tested on slopes/steps with different frictions.

A.2.1 Introduction

In earlier simulations the friction coefficient was calibrated to match the physical result completed with the Cylindabot prototype (Chapter 5). This section goes into a more detailed exploration of the effect that changing the friction has on the robot's ability. The frictional coefficient is a ratio between the reaction forces of an object on a surface and the maximum frictional force that can resist movement. It is effectively how grippy the two surfaces are. Simple slopes and steps will be used to test how friction affects the robots ability.

- Friction is more important for traversing slopes and has a significant effect
- Friction has little or no affect Cylindabot's ability on steps
- Two parameter obstacles do not have a simple relationship. More investigation is needed to determine the underlying mechanics.

Coulomb friction is used in CoppeliaSim to model surface interactions, which is the most common model used for friction. To be specific, the Newton physic engine uses Coulomb friction with stiction which is used in [116]. A range of other friction models are compared in [117].

Friction is not a heavily investigated area of robotics. In [118] a two wheeled balancing robot is analysed on normal grip, low grip and frictionless environments. The body

position can affect how much grip is required. In [119] a four wheeled robot is able to move its centre of mass to allow less friction to be required.

A.2.2 Slope maths

In Chapter 5, a geometric limit was shown for step ascension using the work from [2] which is referred to as the approach/surmount theory. With physical and simulation results it was proven that Cylindabot follows these inequalities. There is currently no such theory for robots like Cylindabot climbing slopes. In this section dynamic inequalities are applied to the new design of Cylindabot to attempt this.

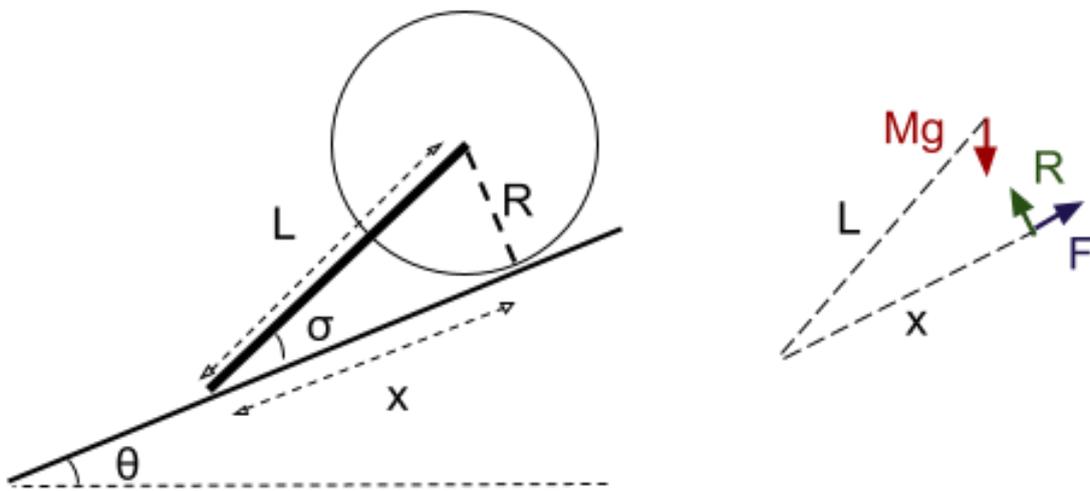


Figure 119. Mathematical model of Cylindabot on a slope and force diagram

An assumption is that the tail of Cylindabot is smooth hence has a negligible frictional coefficient. With that in mind, moments are taken around the tail contact point to calculate the reaction force of the Whegs on the slope.

Equation 26. Kinematics of Cylindabot on a slope

$$\ominus \text{Tail contact: } xR = MgL\cos(\sigma + \theta) \quad [1]$$

$$F \leq \mu R \quad [2]$$

$$F = Mg \sin(\theta) \quad [3]$$

$$\therefore \frac{\mu L}{x} \cos(\sigma + \theta) \geq \sin(\theta) \quad [4]$$

$$\cos(\sigma + \theta) = \cos\sigma\cos\theta - \sin\sigma\sin\theta \quad [5]$$

$$\cos\sigma = \frac{x}{L} \text{ \& } \sin\sigma = \frac{r}{L} \quad [6]$$

$$\mu(x\cos\theta - r\sin\theta) \geq x\sin\theta \quad [7]$$

$$\tan\theta \leq \frac{\mu x}{x + \mu r} \quad [8]$$

$$\tan\theta\left(\frac{1}{\mu} + \tan\sigma\right) \leq 1 \quad [9]$$

A.2.3 Friction on Single Step

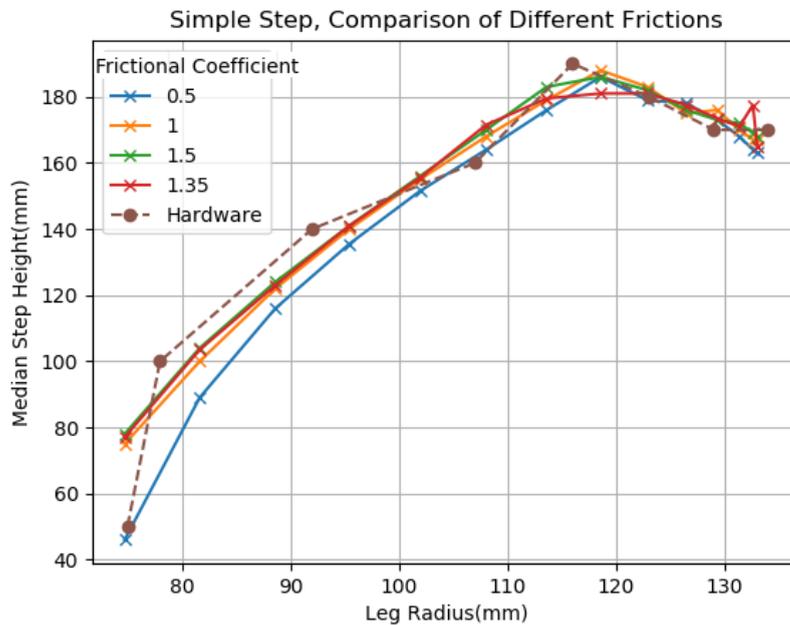


Figure 120. Change in frictional coefficient on step climbing ability. Beermat algorithm applied to 50 successes per point.

Figure 120 shows that changing the frictional coefficient has little effect on Cylindabot's ability to climb a step. There is a slight drop when the legs are retracted when the coefficient is reduced to 0.5. This is due to Cylindabot being in wheeled mode and it needing friction to grip the step. In the other cases the leg is able to catch the flat part of the step and therefore friction is less vital. This lack of reaction was noticed early on to the point that for the single step results from Chapter 5 the friction of 1 rather than 1.35 was used. These results imply that frictional coefficient has little effect on Cylindabot's ability to climb a step; this is not true for slopes.

A.2.4 Friction on Heightfield Slopes

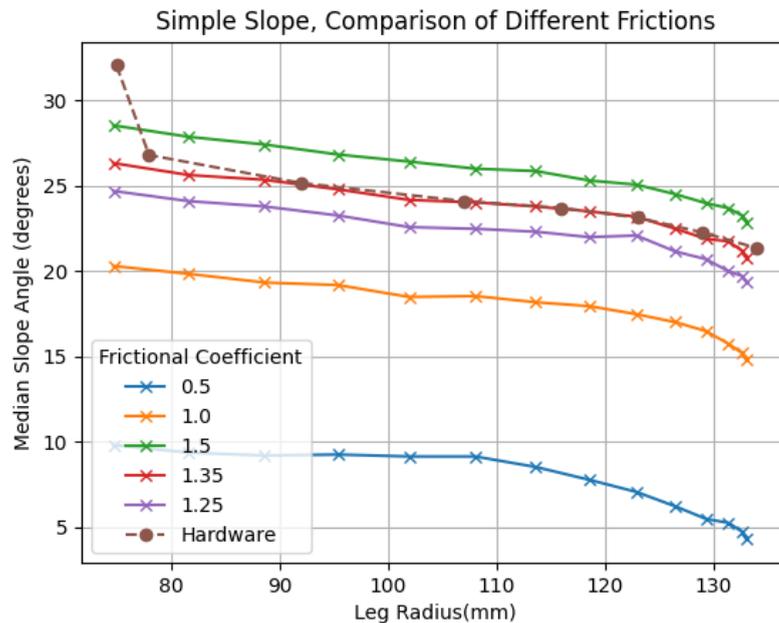


Figure 121. Simulated slope data with varied frictional coefficients. 50 successes for each point using beermat algorithm.

Changing the friction has a significant effect on how well Cylindabot is able to climb a slope. When compared to changing the tail length, reducing the friction from 1.35 to 1 has a larger effect on slope climbing than reducing the tail length from 315mm. The lines are still curved as the data is mapped against the leg radius rather than the midpoint radius. Although at different positions, what are all similar shaped curves showing that the frictional coefficient does not change the relationship between slope angle and leg radius.

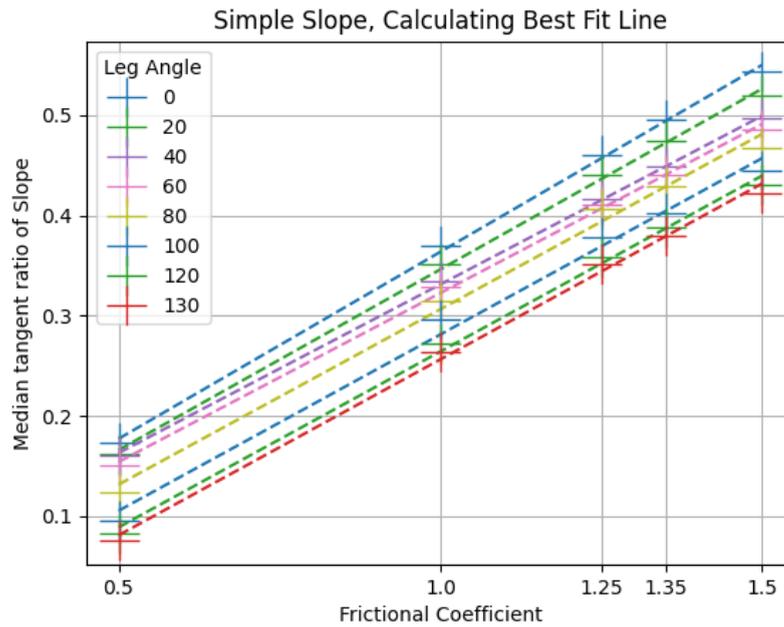


Figure 122. Friction coefficient against tangent ratio. Replotting results from Figure 121. Dash lines are lines of best fit. Not all leg angles plotted for clarity and follow the same trend.

Figure 122 shows that there is a positive correlation between the friction coefficient and the tangent of the slope angle. The variation seems to have little effect on the gradient of these lines, mainly changing the y-intercept. Only half of the tested leg angles were plotted to simplify the graph and the unshown data points also follow this trend.

A.2.5 Friction on a Cuboid Slope

Due to the results on a two parameter slope (Figure 117) being higher than previous results on a slope (Figure 85), Cylindabot was simulated on a one metre cuboid slope with varied frictional coefficients. This was to see if the cuboid slope had a fundamental difference or different frictional coefficient.

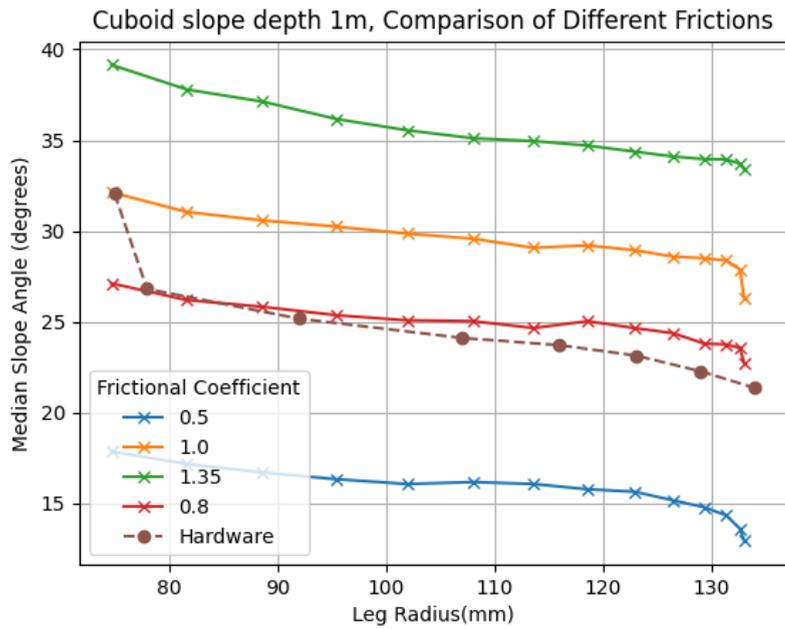


Figure 123. Friction varied on a 1m cuboid slope. Beermat algorithm with 50 successes per point.

The same downward trend can be seen for the cuboid slope in Figure 123 that was previously seen for a height field slope (figure 121). The difference between these graphs is that a much higher slope angle is achieved by a lower frictional coefficient when climbing a cuboid slope. A friction of 0.8 was able to climb the same steepness of slope as both the hardware prototype and the height field slope at a friction of 1.35. This red line (friction = 0.8) does not match the physical data as accurately as the previous data from Chapter 5.

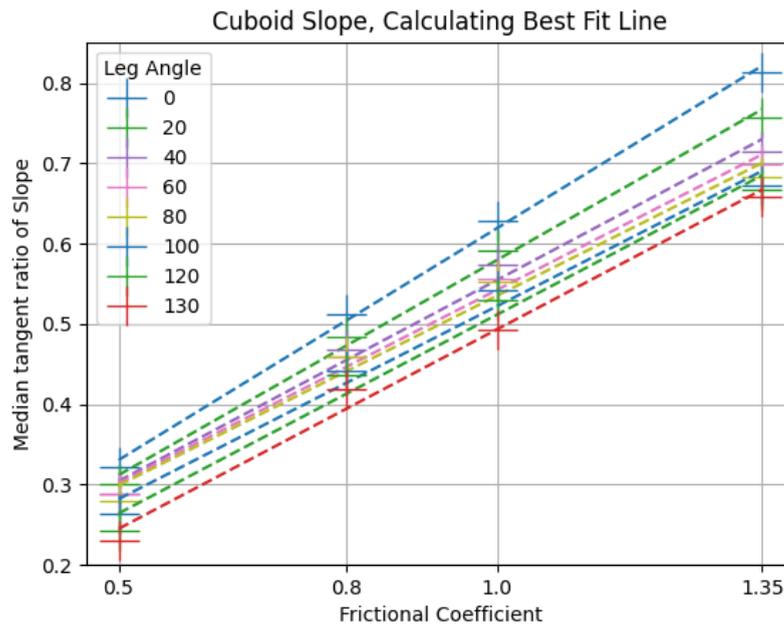


Figure 124. Frictional coefficients varied for a cuboid slope.

Again the cuboid data in Figure 124 shows a similar trend when plotting against frictional coefficients. The problem then comes how to rectify this disconnect between the two data sets. To attempt to consolidate this, a ratio was added to the mu of Equation 26.9 and then non-linear least squares was applied to see if this conversion was consistent. If the ratio was the same for different leg angles then the mathematical model was correct and a conversion factor found. However the mu ratio was not the same for these data sets. The mu ratio is decreasing as the legs are deployed and this trend means that the error is systemic rather than random. Therefore the mathematical model from Equation 26.9 must be incorrect.

Table 28 Non-linear least squares regression for mu ratio on a slope data

Leg Angle (degrees)	Tan (body angle)	Mu ratio (height fields)	Mu ratio (cuboid)
0	0.2308	0.4096	0.6993
20	0.2708	0.3966	0.6682
40	0.3079	0.3819	0.6531
60	0.3392	0.3784	0.6510
80	0.3634	0.3674	0.6530
100	0.3797	0.3423	0.6385

Leg Angle (degrees)	Tan (body angle)	Mu ratio (height fields)	Mu ratio (cuboid)
120	0.3880	0.3242	0.6253
130	0.3890	0.3156	0.5958

Further investigation of these different frictional coefficient interactions is needed. One possible avenue is looking into unrealistic frictions in the simulator to examine the overall trends. Although not applicable to a real robot, it would give important insight to whether these trends are linear or match the mathematical predictions.

Video Links

In the next section all of the academic references are given however here are links and QR codes for certain robots mentioned in this thesis

[Cylindabot](#) [110]



[Impass](#) [37]



[Wheeler](#) [9] -



[Salto](#) [71]



[Scout](#) [49]



[MIT Mini Cheetah](#) [109]



Reference

- [1] L. Pedersen, D. Kortenkamp, D. Wettergreen, and I. Nourbakhsh, "A Survey of Space Robotics," 2003. [Online]. Available: <https://ntrs.nasa.gov/search.jsp?R=20030054507>
- [2] K. Nagatani, M. Kuze, and K. Yoshida, "Development of transformable mobile robot with mechanism of variable wheel diameter," *J. Robot. Mechatron*, vol. 19, no. 3, pp. 252–257, 2007.
- [3] T. Bridgwater, G. Griffiths, A. Winfield, and T. Pipe, "Locomotion Strategies Applicable to the Nuclear Cave Environment: Review and First Experimental Results."
- [4] Y. Li, Q. Huang, J. Gao, L. Zhang, and Y. Tian, "A novel semi-autonomous throwbot for reconnaissance application," in *Proceedings of the 10th World Congress on Intelligent Control and Automation*, Jul. 2012, pp. 3822–3827.
- [5] S. Seok *et al.*, "Design Principles for Energy-Efficient Legged Locomotion and Implementation on the MIT Cheetah Robot," *IEEE/ASME Trans. Mechatron.*, vol. 20, no. 3, pp. 1117–1129, Jun. 2013.
- [6] I. Spectrum, "Spot," Sep. 24, 2019. <https://robots.ieee.org/robots/spotmini/> (accessed Mar. 14, 2022).
- [7] G. M. Nelson and R. D. Quinn, "Posture control of a cockroach-like robot," *IEEE Control Syst. Mag.*, vol. 19, no. 2, pp. 9–14, Apr. 1999.
- [8] R. T. Schroer, M. J. Boggess, R. J. Bachmann, R. D. Quinn, and R. E. Ritzmann, "Comparing cockroach and Whegs robot body motions," Sep. 2004, pp. 3288–3293 Vol.4.
- [9] C. Zheng and K. Lee, "WheeLeR: Wheel-Leg Reconfigurable Mechanism with Passive Gears for Mobile Robot Applications," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 9292–9298.
- [10] Q. Nguyen, A. Agrawal, W. Martin, H. Geyer, and K. Sreenath, "Dynamic bipedal locomotion over stochastic discrete terrain," *International Journal of Robotics Research*, 2018. doi: 10.1177/0278364918791718.
- [11] A. S. Santos, H. Ignacio Perez Azpúrua, G. Pessin, and G. M. Freitas, "Path Planning for Mobile Robots on Rough Terrain," in *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, Nov. 2018, pp. 265–270.
- [12] E. Rohmer, G. Reina, and K. Yoshida, "Dynamic simulation-based action planner for a reconfigurable hybrid leg-wheel planetary exploration rover," *Adv. Robot.*, vol. 24, no. 8–9, pp. 1219–1238, May 2010.
- [13] L. Bruzzone and G. Quaglia, "Review article: Locomotion systems for ground mobile robots in unstructured environments," *Mechanical Sciences*, vol. 3, no. 2. Copernicus GmbH, pp. 49–62, 2012. doi: 10.5194/ms-3-49-2012.
- [14] J. Aguilar *et al.*, "A review on locomotion robophysics: The study of movement at the intersection of robotics, soft matter and dynamical systems," *Reports on Progress in Physics*. 2016. doi: 10.1088/0034-4885/79/11/110001.
- [15] I. Tsitsimpelis, C. J. Taylor, B. Lennox, and M. J. Joyce, "A review of ground-based robotic systems for the characterization of nuclear environments," *Prog. Nuclear Energy*, vol. 111, pp. 109–124, Mar. 2019.
- [16] A. Conduraru (Slătineanu), I. Doroftei, and I. Conduraru, "An Overview on the Design of Mobile Robots with Hybrid Locomotion," *Adv. Mat. Res.*, vol. 837, pp. 555–560, Nov. 2013.
- [17] A. Ellery, "Environment-robot interaction - The basis for mobility in planetary micro-rovers," in *Robotics and Autonomous Systems*, 2004. doi: 10.1016/j.robot.2004.08.007.
- [18] Y. Ota, K. Yoneda, T. Tamaki, and S. Hirose, "A walking and wheeled hybrid locomotion with twin-frame structure robot," 2003. doi: 10.1109/irds.2002.1041669.
- [19] R. H. Armour and J. F. V. Vincent, "Rolling in Nature and Robotics: A Review," *J. Bionic Eng.*

- 2006, doi: 10.1016/S1672-6529(07)60003-1.
- [20] R. Armour, K. Paskins, A. Bowyer, J. Vincent, and W. Megill, “Jumping robots: A biomimetic solution to locomotion across rough terrain,” in *Bioinspiration and Biomimetics*, 2007. doi: 10.1088/1748-3182/2/3/S01.
- [21] K. Yoshida and H. Hamano, “Motion dynamics of a rover with slip-based traction model,” in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, 2002. doi: 10.1109/ROBOT.2002.1013712.
- [22] “Husky UGV - outdoor field research robot by clearpath,” *Clearpath Robotics*, May 31, 2015. <https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/> (accessed Sep. 16, 2022).
- [23] C. West *et al.*, “A Debris Clearance Robot for Extreme Environments,” in *Towards Autonomous Robotic Systems*, 2019, pp. 148–159.
- [24] È. Pairet, P. Ardón, X. Liu, J. Lopes, H. Hastie, and K. S. Lohan, “A Digital Twin for Human-Robot Interaction,” in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Mar. 2019, pp. 372–372.
- [25] H. Gan and W. S. Lee, “Development of a Navigation System for a Smart Farm,” *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 1–4, Jan. 2018.
- [26] T. Rouček *et al.*, “DARPA Subterranean Challenge: Multi-robotic Exploration of Underground Environments,” in *Modelling and Simulation for Autonomous Systems*, 2020, pp. 274–290.
- [27] “UR5 collaborative robot arm.” <https://www.universal-robots.com/products/ur5-robot/> (accessed Sep. 16, 2022).
- [28] F. Zhang, Y. Yu, Q. Wang, X. Zeng, and H. Niu, “A terrain-adaptive robot prototype designed for bumpy-surface exploration,” *Mechanism and Machine Theory*, vol. 141, pp. 213–225, Nov. 2019.
- [29] K. Tadakuma, “Tetrahedral Mobile Robot with Novel Ball Shape Wheel,” in *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechanics, 2006. BioRob 2006.*, Feb. 2006, pp. 946–952.
- [30] K. Tadakuma, R. Tadakuma, and J. Berengeres, “Development of holonomic omnidirectional Vehicle with ‘Omni-Ball’: spherical wheels,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, Oct. 2007. doi: 10.1109/iros.2007.4399560.
- [31] K. Tadakuma *et al.*, “Throwable tetrahedral robot with transformation capability,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2009, pp. 2801–2808.
- [32] D. C. Kar, “Design of Statically Stable Walking Robot: A Review,” *Journal of Robotic Systems*. 2003. doi: 10.1002/rob.10118.
- [33] S. Guccione and G. Muscato, “The Wheeleg Robot,” *IEEE Robot. Autom. Mag.*, vol. 10, no. 4, pp. 33–43, 2003.
- [34] U. Saranli, M. Buehler, and D. E. Koditschek, “Design, modeling and preliminary control of a compliant hexapod robot,” *Proceedings-IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2589–2596, 2000.
- [35] U. Saranli, M. Buehler, and D. E. Koditschek, “RHex: A Simple and Highly Mobile Hexapod Robot,” 2001.
- [36] H. Nozaki, Y. Kujirai, R. Niiyama, Y. Kawahara, T. Yonezawa, and J. Nakazawa, “Continuous Shape Changing Locomotion of 32-legged Spherical Robot,” *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018. doi: 10.1109/iros.2018.8593791.
- [37] J. B. Jeans and D. Hong, “IMPASS: Intelligent Mobility Platform with Active Spoke System,” Aug. 2009, pp. 1605–1606.
- [38] P. A. Bhounsule, E. Ameperosa, S. Miller, K. Seay, and R. Ulep, “Dead-beat control of walking for a torso-actuated rimless wheel using an event-based, discrete, linear controller,” in *Volume 5A: 40th Mechanisms and Robotics Conference*, Charlotte, North Carolina, USA,

- Aug. 2016. doi: 10.1115/detc2016-59563.
- [39] T. F. Nygaard, C. P. Martin, J. Torresen, and K. Glette, “Exploring Mechanically Self-Reconfiguring Robots for Autonomous Design,” *arXiv [cs.RO]*, May 08, 2018. [Online]. Available: <http://arxiv.org/abs/1805.02965>
- [40] T. F. Nygaard, C. P. Martin, E. Samuelsen, J. Torresen, and K. Glette, “Real-world evolution adapts robot morphology and control to hardware limitations,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, Kyoto, Japan, Jul. 2018, pp. 125–132.
- [41] T. F. Nygaard, D. Howard, and K. Glette, “Real world morphological evolution is feasible,” in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, Cancún, Mexico, Jul. 2020, pp. 1392–1394.
- [42] G. Kenneally, A. De, and D. E. Koditschek, “Design Principles for a Family of Direct-Drive Legged Robots,” *IEEE Robotics and Automation Letters*, 2016, doi: 10.1109/lra.2016.2528294.
- [43] M. Hutter *et al.*, “ANYmal - a highly mobile and dynamic quadrupedal robot,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016, pp. 38–44.
- [44] R. W. Xu, K. Chin Hsieh, U. H. Chan, H. Un Cheang, W. K. Shi, and C. Tin Hon, “Analytical Review on Developing Progress of the Quadruped Robot Industry and Gaits Research,” in *2022 8th International Conference on Automation, Robotics and Applications (ICARA)*, Feb. 2022, pp. 1–8.
- [45] M. Raibert, K. Blankespoor, G. Nelson, R. Playter, and B. Team, “BigDog, the Rough-Terrain Quadruped Robot,” 2008, doi: 10.3182/20080706-5-KR-1001.4278.
- [46] S. Kim, J. E. Clark, and M. R. Cutkosky, “ISprawl: Design and tuning for high-speed autonomous open-loop running,” *Int. J. Rob. Res.*, vol. 25, no. 9, pp. 903–912, Sep. 2006.
- [47] M. Kovač, M. Fuchs, A. Guignard, J. C. Zufferey, and D. Floreano, “A miniature 7g jumping robot,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2008, pp. 373–378.
- [48] E. W. Hawkes *et al.*, “Engineered jumpers overcome biological limits via work multiplication,” *Nature*, vol. 604, no. 7907, pp. 657–661, Apr. 2022.
- [49] D. F. Hougen *et al.*, “A miniature robotic system for reconnaissance and surveillance,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, Apr. 2000, vol. 1, pp. 501–507 vol.1.
- [50] K. Kikuchi, K. Sakaguchi, T. Sudo, N. Bushida, Y. Chiba, and Y. Asai, “A study on a wheel-based stair-climbing robot with a hopping mechanism,” *Mech. Syst. Signal Process.*, vol. 22, no. 6, pp. 1316–1326, Aug. 2008.
- [51] A. Drenner *et al.*, “Mobility enhancements to the Scout robot platform,” in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, May 2002, vol. 1, pp. 1069–1074 vol.1.
- [52] B. Kratochvil *et al.*, “Heterogeneous implementation of an adaptive robotic sensing team,” in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, Sep. 2003, vol. 3, pp. 4264–4269 vol.3.
- [53] J. A. Smith, I. Sharf, and M. Trentini, “PAW: a Hybrid Wheeled-Leg Robot,” 2006.
- [54] K. Turker, I. Sharf, and M. Trentini, “Step negotiation with wheel traction: a strategy for a wheel-legged robot,” *IEEE Int. Conf. Robot. Autom.*, 2012.
- [55] M. Guihard, P. Gorce, and J. G. Fontaine, “SAPPHYR: legs to pull a wheel structure,” 1995. doi: 10.1109/icsmc.1995.537952.
- [56] J. M. Morrey, B. Lambrecht, A. D. Horchler, R. E. Ritzmann, and R. D. Quinn, “Highly Mobile and Robust Small Quadruped Robots,” 2003. [Online]. Available: <http://5iorobots.cwru.edu>,
- [57] K. A. Daltorio *et al.*, “Mini-Whegs™ climbs steep surfaces using insect-inspired attachment mechanisms,” *Int. J. Rob. Res.*, vol. 28, no. 2, pp. 285–302, Feb. 2009.
- [58] D. Zarrouk and L. Yehezkel, “Rising STAR: A highly reconfigurable sprawl tuned robot,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1888–1895, Jul. 2018.

- [59] W. H. Chen, H. S. Lin, Y. M. Lin, and P. C. Lin, "TurboQuad: A Novel Leg-Wheel Transformable Robot with Smooth and Fast Behavioral Transitions," *IEEE Trans. Rob.*, vol. 33, no. 5, pp. 1025–1040, Oct. 2017.
- [60] S. C. Chen, K. J. Huang, W. H. Chen, S. Y. Shen, C. H. Li, and P. C. Lin, "Quattroped: A leg-wheel transformable robot," *IEEE/ASME Trans. Mechatron.*, vol. 19, no. 2, pp. 730–742, 2014.
- [61] K. C. Galloway *et al.*, "X-RHex: A Highly Mobile Hexapedal Robot for Sensorimotor Tasks Recommended Citation X-RHex: A Highly Mobile Hexapedal Robot for Sensorimotor Tasks," 2010. [Online]. Available: https://repository.upenn.edu/ese_reports
- [62] K. Tadakuma *et al.*, "Armadillo-inspired wheel-leg retractable module," in *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2009, pp. 610–615.
- [63] K. Tadakuma *et al.*, "Mechanical Design of the Wheel-Leg Hybrid Mobile Robot to Realize a Large Wheel Diameter," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 3358–3365.
- [64] T. Sun, X. Xiang, W. Su, H. Wu, and Y. Song, "A transformable wheel-legged mobile robot: Design, analysis and experiment," *Rob. Auton. Syst.*, vol. 98, pp. 30–41, Dec. 2017.
- [65] Y. She, C. J. Hurd, and H. J. Su, "A transformable wheel robot with a passive leg," in *IEEE International Conference on Intelligent Robots and Systems*, Dec. 2015, vol. 2015-December, pp. 4165–4170.
- [66] Y. S. Kim, G. P. Jung, H. Kim, K. J. Cho, and C. N. Chu, "Wheel Transformer: A wheel-leg hybrid robot with passive transformable wheels," *IEEE Trans. Rob.*, vol. 30, no. 6, pp. 1487–1498, Dec. 2014.
- [67] R. Cao, J. Gu, C. Yu, and A. Rosendo, "OmniWheg: An Omnidirectional Wheel-Leg Transformable Robot," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2022, pp. 5626–5631.
- [68] M. H. Raibert, "Legged Robots," *Commun. ACM*, vol. 29, no. 6, pp. 499–514, 1986.
- [69] M. H. Raibert, M. Chepponis, and H. B. Brown, "Running on Four Legs As Though They Were One," *IEEE Journal on Robotics and Automation*, 1986, doi: 10.1109/JRA.1986.1087044.
- [70] P. Gregorio, M. Ahmadi, and M. Buehler, "Design, control, and energetics of an electrically actuated legged robot," *IEEE Trans. Syst. Man Cybern. B Cybern.*, vol. 27, no. 4, pp. 626–634, 1997.
- [71] J. K. Yim and R. S. Fearing, "Precision Jumping Limits from Flight-phase Control in Salto-1P," 2018. doi: 10.0/Linux-x86_64.
- [72] R. Velázquez and A. Lay-Ekuakille, "A review of models and structures for wheeled mobile robots: Four case studies," in *2011 15th International Conference on Advanced Robotics (ICAR)*, Jun. 2011, pp. 524–529.
- [73] R. P. M. Chan, K. A. Stol, and C. R. Halkyard, "Review of modelling and control of two-wheeled robots," *Annual Reviews in Control*. 2013. doi: 10.1016/j.arcontrol.2013.03.004.
- [74] T. Chen, P. Hazelwood, and K. Stol, "Step Ascent Modelling of a Two-Wheeled Robot," 2012.
- [75] Z. Kausar, K. Stol, and N. Patel, "Performance enhancement of a statically unstable two wheeled mobile robot traversing on an uneven surface," in *2010 IEEE Conference on Robotics, Automation and Mechatronics, RAM 2010*, 2010, pp. 156–162.
- [76] T. Chanthasopeephan, A. Jarakorn, P. Polchankajorn, and T. Maneewarn, "Impact reduction mobile robot and the design of the compliant legs," *Rob. Auton. Syst.*, vol. 62, no. 1, pp. 38–45, Jan. 2014.
- [77] J. P. Grotzinger *et al.*, "Mars Science Laboratory Mission and Science Investigation," *Space Sci. Rev.*, vol. 170, no. 1, pp. 5–56, Sep. 2012.
- [78] K. Yoshida, "Achievements in space robotics," *IEEE Robot. Autom. Mag.*, vol. 16, no. 4, pp. 20–28, Dec. 2009.
- [79] M. G. Bekker, *Theory of land locomotion: the mechanics of vehicle mobility*. Ann Arbor, MI: University of Michigan Press, 1956.
- [80] R. He, C. Sandu, A. K. Khan, A. G. Guthrie, P. Schalk Els, and H. A. Hamersma, "Review of

- terramechanics models and their applicability to real-time applications,” *Journal of Terramechanics*, 2018. doi: 10.1016/j.jterra.2018.04.003.
- [81] G. Sitkei, “Sinkage and rolling resistance of wheels,” *Progress in Agricultural Engineering Sciences*, 2015, doi: 10.1556/446.11.2015.3.
- [82] R. Bauer, W. Leung, and T. Barfoot, “Experimental and simulation results of wheel-soil interaction for planetary rovers,” in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2005. doi: 10.1109/IROS.2005.1545179.
- [83] K. Iagnemma, S. Kang, H. Shibly, and S. Dubowsky, “Online terrain parameter estimation for wheeled mobile robots with application to planetary rovers,” *IEEE Trans. Rob.*, 2004, doi: 10.1109/TRO.2004.829462.
- [84] A. R. Khairuddin, M. S. Talib, and H. Haron, “Review on simultaneous localization and mapping (SLAM),” in *2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Nov. 2015, pp. 85–90.
- [85] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part I,” *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006.
- [86] T. Bailey and H. Durrant-Whyte, “Simultaneous localization and mapping (SLAM): part II,” *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, Sep. 2006.
- [87] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, “Learning robust perceptive locomotion for quadrupedal robots in the wild,” *Sci Robot*, vol. 7, no. 62, p. eabk2822, Jan. 2022.
- [88] A. Broggi, E. Cardarelli, S. Cattani, and M. Sabbatelli, “Terrain mapping for off-road Autonomous Ground Vehicles using rational B-Spline surfaces and stereo vision,” in *2013 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2013, pp. 648–653.
- [89] G. Ishigami, K. Nagatani, and K. Yoshida, “Path Planning for Planetary Exploration Rovers and Its Evaluation based on Wheel Slip Dynamics,” *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 2007. doi: 10.1109/robot.2007.363672.
- [90] S. B. P. Mattson, T. Companion, and M. Hickman, “Guide for Evaluating, Purchasing, and Training with Response Robots Using DHS-NIST-ASTM International Standard Test Methods”.
- [91] A. Seeni, B. Schfer, and G. Hirzinger, “Robot Mobility Systems for Planetary Surface Exploration – State-of-the-Art and Future Outlook: A Literature Survey,” in *Aerospace Technologies Advancements*, InTech, 2010.
- [92] Von Karman and Gabrielli, “What price speed? Specific power required for propulsion of vehicles,” *Mech. Eng.*
- [93] S. Kim and P. M. Wensing, “Design of Dynamic Legged Robots,” *Foundations and Trends in Robotics*, vol. 5, no. 2, pp. 117–190, 2017.
- [94] X. Zhou and S. Bi, “A survey of bio-inspired compliant legged robot designs,” *Bioinspiration and Biomimetics*, vol. 7, no. 4, Dec. 2012, doi: 10.1088/1748-3182/7/4/041001.
- [95] M. Martone, C. Pavlov, A. Zeloof, V. Bahl, and A. M. Johnson, “Enhancing the Vertical Mobility of a Robot Hexapod Using Microspines,” *arXiv [cs.RO]*, Jun. 11, 2019. [Online]. Available: <http://arxiv.org/abs/1906.04811>
- [96] Ackerman E, “Boston dynamics sand flea robot demonstrates astonishing jumping skills,” *IEEE Spectrum Robotics Blog*, 2012.
- [97] R. Woolley, University of York, J. Timmis, A. Tyrrell, University of York, and University of York, “Analysis of two-wheeled robot morphology for a slope environment,” in *UK-RAS Conference for PhD and Early Career Researchers Proceedings*, May 2020. doi: 10.31256/sf6zi7l.
- [98] G. Granosik, “Hypermobile robots - The survey,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 75, no. 1. Kluwer Academic Publishers, pp. 147–169, 2014. doi: 10.1007/s10846-013-9985-5.
- [99] B. K. Taylor, S. Balakirsky, E. Messina, and R. D. Quinn, “Design and Validation of a Whegs

- Robot in USARSim,” in *Proceedings of the Workshop on Performance Metrics for Intelligent Systems.*, 2007.
- [100] R. D. Quinn, J. T. Off4, D. A. Kingsley, and R. E. Ritzmann, “Improving Mobility Through Abstract Biological Principles,” *Intl. Conference on Intelligent Robots and Systems*, 2002.
- [101] R. A. Liston and R. S. Mosher, “A versatile walking truck,” in *Transportation Engineering Conference*, 1968.
- [102] I. Anderson, “Mechanical Properties of Specimens 3D Printed with Virgin and Recycled Polylactic Acid,” *3D Printing and Additive Manufacturing*, vol. 4, no. 2, pp. 110–115, Jun. 2017.
- [103] M. Plooij, G. Mathijssen, P. Cherelle, D. Lefeber, and B. Vanderborght, “Lock your robot: A review of locking devices in robotics,” *IEEE Robot. Autom. Mag.*, 2015, doi: 10.1109/MRA.2014.2381368.
- [104] J. Hilder, *apihat-python: Python Library for the YRL028 APIHAT*. Github. Accessed: Jan. 18, 2023. [Online]. Available: <https://github.com/yorkrobotlab/apihat-python>
- [105] M. H. Raibert H Benjamin Brown and J. Michael Chepponis, “One-Legged Hopping Machine,” 1984.
- [106] S. Roberts *et al.*, “Desert RHex Technical Report: Jornada and White Sands Trip,” 2014. [Online]. Available: <https://www.researchgate.net/publication/304049339>
- [107] M. Barragan, N. Flowers, and A. M. Johnson, “MiniRHex: A Small, Open-source, Fully Programmable Walking Hexapod,” 2018. [Online]. Available: <https://robomechanics.github.io/MiniRHex/>
- [108] R. Kim, A. Debate, S. Balakirsky, and A. Mazumdar, *Using Manipulation to Enable Adaptive Ground Mobility*. 2020.
- [109] B. Katz, J. D. I. Carlo, and S. Kim, “Mini cheetah: A platform for pushing the limits of dynamic quadruped control,” in *Proceedings - IEEE International Conference on Robotics and Automation*, May 2019, vol. 2019-May, pp. 6295–6301.
- [110] R. Woolley, J. Timmis, and A. M. Tyrrell, “Cylindabot: Transformable Whег Robot Traversing Stepped and Sloped Environments,” *Robotics*, vol. 10, no. 3, p. 104, Aug. 2021.
- [111] T. Aoki, K. Asami, S. Ito, and S. Waki, “Development of quadruped walking robot with spherical shell: improvement of climbing over a step,” *ROBOMECH Journal*, vol. 7, no. 1, pp. 1–12, May 2020.
- [112] R. J. Full, “Rapid negotiation of rough terrain by the death-head cockroach,” *Am. Zool.*, vol. 38, p. 81A, 1998.
- [113] D. Goldschmidt, F. Wörgötter, and P. Manoonpong, “Biologically-inspired adaptive obstacle negotiation behavior of hexapod robots,” *Front. Neurobot.*, vol. 8, p. 3, Jan. 2014.
- [114] S. Campbell, N. O’Mahony, A. Carvalho, L. Krpalkova, D. Riordan, and J. Walsh, “Path Planning Techniques for Mobile Robots A Review,” in *2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, Barcelona, Spain, Feb. 2020. doi: 10.1109/icmre49073.2020.9065187.
- [115] DCLG, *Approved Document K: Protection from falling, collision and impact (2013 edition - for use in England)*. London, England: NBS/RIBA Enterprises, 2013.
- [116] P. E. Dupont, “Friction modeling in dynamic robot simulation,” in *Proceedings., IEEE International Conference on Robotics and Automation*, May 1990, pp. 1370–1376 vol.2.
- [117] F. Marques, P. Flores, J. C. Pimenta Claro, and H. M. Lankarani, “A survey and comparison of several friction force models for dynamic analysis of multibody mechanical systems,” *Nonlinear Dyn.*, vol. 86, no. 3, pp. 1407–1443, Nov. 2016.
- [118] D. R. Jones and K. A. Stol, “Modelling and Stability Control of Two-Wheeled Robots in Low-Traction Environments,” Jan. 2010, [Online]. Available: <http://dx.doi.org/>
- [119] B.-S. Sim, K.-J. Kim, and K.-H. Yu, “Development of Body Rotational Wheeled Robot and its Verification of Effectiveness,” *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020. doi: 10.1109/icra40945.2020.9197047.

List of Figures

Figure 1. Diagram showing the different forms of locomotion	p.15
Figure 2. Venn diagram of the morphologies discussed here and what type of locomotion they use	p.17
Figure 3. Husky robot from Clear Path [22]	p.18
Figure 4. Servo wheel [28] and Tetrahedron robot [29] with “Omni-balls”	p.18
Figure 5. a) Original RHex [35], b) RHex 0.9	p.20
Figure 6. a) Mochibot [37] b) Impass [36]	p.20
Figure 7. Variable diameter spoked wheel robot [2]	p.21
Figure 8. Leg adapting quadruped [39]. Left: leg retracted, right: legs extended.	p.21
Figure 9. a) Minitaur [42] b) MIT Cheetah [5]	p.22
Figure 10. a) BigDog [45] b) ISprawl [46]	p.22
Figure 11. a) Jollbot [20] b) 7g Jumping robot [47]	p.23
Figure 12. a) Scout robot [49], b) PAW robot [51]	p.24
Figure 13. a) Mini-Whegs [55], b) Rising Star [56]	p.25
Figure 14. a) TurboQuad [57], b) Quattroped [58]	p.26
Figure 15. Armadillo inspired robot [60]	p.27
Figure 16. a) T-shaped Whег robot [62], b) Passive Wheel robot [63]	p.27
Figure 17. a) Wheel Transformer [64], b) Wheeler robot [9]	p.28
Figure 18. a) Salto-1P [69], b) ATRIAS [10]	p.32
Figure 19. Smooth terrain measured by slopes and jagged terrain measured by steps.	p.38
Figure 20. Graph of different robots comparing their Cost of Transport with their mass. A log graph is used to compress data so comparison can be made. Graph used here from [91] and contains robots not analysed here.	p.40
Figure 21. Slopes and steps diagram	p.42

Figure 22. Simulation architecture for CoppeliaSim	p.47
Figure 23. Wheeled simulation models. Without tail with offset mass and curved tail.	p.48
Figure 24. Curved tail with end mounted legs vs straight tail with encased legs. Early versions of Cylindabot.	p.49
Figure 25. Success Drop-off for a step with a leg angle of 30 degrees. Simulation results every 5 mm with 30 runs at each height.	p.52
Figure 26. Diagram of slopes and step	p.53
Figure 27. Wheel on slope to show balance point	p.54
Figure 28 Simple wheel against a step to show relation to central axis angle	p.55
Figure 29. Velocities at point of impact with the step	p.56
Figure 30. Velocity vectors being transformed back into forward coordinate frames	p.56
Figure 31. Forces on tailed robot for a flat surface	p.57
Figure 32. Slope with a tail with forces for dynamics	p.58
Figure 33. Forces on wheel and tail on a step obstacle.	p.59
Figure 34 Angle and position diagram to determine moments.	p.59
Figure 35. Simulation setup for a slope. The cuboid primitive was angled and positioned to keep contact point with the starting tile.	p.62
Figure 36 Simulation set up for a step. The cuboid primitive was positioned at the correct height.	p.62
Figure 37. Simulation model with legs set to 0 degrees and 90 degrees.	p.65
Figure 38. Leg angle movement. A) Legs at 0 degrees. B) Legs at 120 degrees with θ denoting the leg angle	p.66
Figure 39. Simulated result of the straight legged robot attempting to traverse a step obstacle. Each point from 1200 simulation runs. Error bars calculated by the difference between estimated maximum height and the mean doubled.	p.67
Figure 40. Scenarios of robot climbing a step.	p.69

Figure 41. Simulated result of the straight legged robot attempting to traverse a step obstacle. Each point from 80 simulation runs. Error bars calculated by the difference between estimated maximum height and the mean doubled.	p.69
Figure 42. Dimensions to that were varied in the course of this work. Body width is the size of the main cylinder in the middle of the robot. Tail length is the length of the cuboid than is then grouped with a cylinder to make the tail	p.71
Figure 43. Width variation of the robot on a step obstacle. 30 simulation runs for each point using a fixed height algorithm.	p.72
Figure 44. Slope performance while changing leg angles and tail length. Fixed height algorithm with 30 simulations per point and increments of 2.5 degrees.	p.73
Figure 45. Step climbing ability when varying the tail length for the straight tailed robot. Fixed height algorithm with 30 simulations per point and increments of 5 mm.	p.73
Figure 46. Remapped slope data for a straight tailed robot with leg angles replaced with leg radius. Fixed height algorithm with 30 simulations per point and increments of 2.5 degrees.	p.76
Figure 47. Remapped step data for straight tailed robots with leg angles replaced with leg radius. Fixed height algorithm with 30 simulations per point and increments of 5 mm.	p.77
Figure 48. Left to right: Scout [49], Impass [37], Whег robot [98], Wheeler [9].	p.80
Figure 49. Version 1 of the robot in simulation. a) & b) show that the robot is stable as there is a long distance between the legs and tail. c) & d) show a short distance between them hence the robot was prone to fall backward when traversing obstacles.	p.82
Figure 50. Scale models. On the left version one with exterior legs and a curved tail. On the right, version 2 which has the legs in a wheel sandwich and a straight tail.	p.83
Figure 51. Different numbers of legs represented as polygons.	p.84
Figure 52. Technical drawing of initial Whег assembly. Red: legs, blue: Whег forks and yellow: interlock gears. Spur gears were used at this stage.	p.85

Figure 53. Single first Whieg CAD design, generated from a single extrusion of a spur gear and leg arch.	p.86
Figure 54. Initial Whieg design, physica. On the left with legs deployed and on the right as a wheel.	p.87
Figure 55. Top view of robot assembly in FreeCAD	p.88
Figure 56. Exploded view of robot core	p.88
Figure 57. Exploded view of drive train	p.88
Figure 58 Exploded view of Whieg assembly	p.89
Figure 59. Diagram explaining Whieg deployment. Blue arrows represent torque on gears.	p.90
Figure 60. Single Leg, much more complex design compared to initial. Herringbone gear has small slopes within the cog to prevent over turn. The main leg is now an extrusion and a revolution. At each end there is a slide in clamp for the rubber grip.	p.91
Figure 61. Tube clamp, a single M3 screw is able to tighten onto the aluminium tube using this	p.92
Figure 62. Motor cog, D-shaped hole to fit the motor and half the diameter for a 2:1 ratio.	p.92
Figure 63. Interlocking gear, two interlocking tongues and multiple holes for locking the leg angle.	p.93
Figure 64. Motor top cover, small recess for motor front screw	p.94
Figure 65. Motor mount, a combination of the top cover and a tube clamp	p.94
Figure 66. Inner Whieg fork, wide hole to find around interlocking gear	p.95
Figure 67. Outer Whieg fork, smaller hole for the aluminium tube only and three holes in the hexagon for locking the legs at different angles.	p.95
Figure 68. Main body. Longest print time and most complex part of the design. Holds the battery on top, the tail in the middle, mounts to the Raspberry Pi below and has a tube clamp on each side.	p.96

Figure 69. Battery clamp, also can hold the tail in an alternate position	p.96
Figure 70. Pi and APiHat cover, designed to protect the Pi and hat while still giving access.	p.97
Figure 71. Tail, as long as possible given the print bed available and has nothing to catch on obstacles as the robot moves.	p.98
Figure 72. Tail clamp, hooks into herringbone gears in the tail.	p.98
Figure 73. Electronics connections - dashed lines represent wireless connection	p.99
Figure 74. Bluetooth direction pad control. These directions flip if the robot is inverted when moving between different modes of locomotion	p.99
Figure 75. High quality image of Cylindabot.	p.100
Figure 76. Side view and Whег assembly.	p.100
Figure 77. Top view and close up on rubber belt clamp	p.101
Figure 78. Battery housing, Aluminium tube axle/leg lock and APiHAT	p.101
Figure 79. Existing platforms. Raibert’s One legged hopper (a) [103] which led to Salto-1P (b) [69]. Wheeler robot © [9]. Scout robot (d) [49]. LEON (e) [12]. Rising STAR (f) [56]. MIT Cheetah (g) [5] leading to MIT Mini Cheetah (h) [107]. Original R-hex (i) [34] leading to the more advanced X-RHex (j) [59]. T-shaped Whег robot (k) [62]. Passive Wheel robot (l) [63]. Wheel Transformer (m) [64]. TurboQuad (n) [57]. Quattroped (o) [58].	p.106
Figure 80. Approach/Surmount theory graph and diagrams of each. The green line is the approach limit which is the robot reaching the top of the step (Equation 23). The blue line is whether it can get over the step once it reaches the top (Equation 24). The k factor has been set to the proposed value for three legs and the tail to 285 mm.	p.109
Figure 81. Simulation environment. Left: slopes using a heights field, Right: single step with a single cuboid primitive. The obstacle is added between the start and finish tiles. Arrows show direction of travel.	p.110
Figure 82. Left: Comparison of a single leg in the simulation with the STL of the printed leg. Right: Magnified section of curved leg to show intersection cuboids.	p.111

Figure 83. Simulation results for climbing a single step using the 'beer-mat' algorithm. Median shown given from 50 successful results. Colours represent tail length and the plot is leg radius against step height.	p.113
Figure 84. Four phases of attempting a single step: (1) Approach, (2) Fall backward, (3) No grip, (4) Surmount or Successful traversing the step.	p.113
Figure 85. Slope results remapped to mid-point radius which is the average of the maximum and minimum contact point radius. Median of 50 successful results using 'beer-mat' algorithm to change difficulty.	p.115
Figure 86. Physical experiment setup. For the step 2 cm/1 cm thick planks were used to adjust height. The same material was used against a bench and slotted decking for the slope. The fixed height increments and slotted decking mean repeatability of the environment was not an issue. Arrows show direction of travel.	p.116
Figure 87. Hardware results for a single step compared with simulation results. The three graphs are for different tail lengths. The dashed lines are the simulation results from figure 80 with the total range added as error bars. Green and blue lines represent approach and surmount theory, respectively, from section 5.3. The R2 is the R-squared statistical values to measure the reality gap.	p.117
Figure 88. Hardware results for a simple slope compared to simulation. The three graphs are for different tail lengths. The dashed lines are the simulation results from Figure 8 with the total range added as error bars. The R2 is the R-squared statistical values to measure the reality gap.	p.117
Figure 89. A matrix of heights in mm (top left). The obstacle tiles produced from these heights; steps (top right), slopes (bottom left) or plinths (bottom right).	p.123
Figure 90 Simulation setup screen shot	p.124
Figure 91. Four wheeled robot next to Cylindabot (New Design)	p.125
Figure 92. Comparison of different configurations of Cylindabot on sloped grid. The grid heights were determined by a 5 by 5 matrix. Each box plot represent 100 successful terrain crossings	p.125
Figure 93. Comparison of different configurations of Cylindabot on stepped grid. The grid heights were determined by a 5 by 5 matrix. Each box plot represent 100 successful terrain crossings	p.126

Figure 94. 10 standard terrains for map environment. Green = success, yellow = time out, red = fall, blue = average speed. 20 runs per leg angle.	p.128
Figure 95. Checkerboard map with 9 obstacle tiles	p.129
Figure 96. The three different grid styles with grid size/density of 1 or 9.	p.131
Figure 97. Grid size performance on steps. 100 runs per grid size/angle and error bar show quartile range.	p.132
Figure 98. Grid size performance on slopes. 100 runs per grid size/angle and error bar show quartile range.	p.133
Figure 99. Grid size performance on plinths. 100 runs per grid size/angle and error bar show quartile range.	p.134
Figure 100. LEON robot in hexapod mode and two paths determined from action planner [12].	p.137
Figure 101. Screenshot of four ridge types: bridge, square, sloped and round.	p.138
Figure 102. Ridge and Bridge results of obstacle height from the beermat algorithm. 50 successful runs per plotted point. Showing median point and error bar are quantiles.	p.138
Figure 103. Success rate for Cylindabot crossing ridges and bridges. Dark Green = 100% success and Red = 0% that is all failed attempts. With 10 simulation attempts made at each combination of leg angle and height, which equal 550/750 runs per graph (11/15 heights).	p.140
Figure 104. Gap and long slope in simulation. A set of four cuboids were used to create the 45 degree gap. A height field was used for the long slope.	p.142
Figure 105. Gap and long slope obstacle height from simulation. Beermat algorithm with 50 successes.	p.143
Figure 106. Success rate for Cylindabot crossing gaps and long slopes. Green = 100% success and Red = 0% that is all failed attempts. With 10 simulation attempts made at each combination of leg angle and height (19/9 heights).	p.143
Figure 107. Screenshot of a full map with Cylindabot approaching the fifth obstacle of six.	p.145
Figure 108. Box plots of success rate on each individual map. Calculated by taking the number of successes and dividing by the 30 attempts.	p.146

Figure 109. Number of waypoints reached over the 30 maps. Dots represent the median number of waypoints reached and the arrows point to the mean to show data skew. The maps were ordered by the difficulty for the route adaptation.	p.147
Figure 110. Four different maps numbered 6, 15, 28, and 30. Map 6 was the hardest map whereas map 28 the easiest. Map 15 was the only where route adaptation beat leg angle adaptation. Finally map 30 was a medium difficulty map.	p.148
Figure 111. Screenshot of a Multi-step	p.155
Figure 112. Multi-Step heights and depths. Beermat algorithm used with 50 successful results per point. The tail was set to 315mm. The black dashed lines are the result for a single step.	p.156
Figure 113. Short depth of 50mm causes 20 steps which might be an unfair comparison	p.156
Figure 114. Curved step. Beermat algorithm used with 50 successful results per point. The tail was set to 315mm. The black dashed lines are the result for a single step.	p.157
Figure 115. Screenshots of curved steps. Left: step taller than depth, Right: depth longer than height.	p.158
Figure 116. Sloped Step using height as y-axis. Beermat algorithm used with 50 successful results per point. The tail was set to 315mm. Dash line is single step data.	p.158
Figure 117. Sloped Step using angle as y-axis. Beermat algorithm used with 50 successful results per point. The tail was set to 315mm. Depth of 200mm repeated between two graphs and a solid black bar shows the scale of the right hand graph in the left hand graph. The dashed line is the result for a 1 metre flat cuboid slope.	p.159
Figure 118. Screenshots of the sloped step. Left showing a configuration that would act like a slope and on the right one that would have to be handled like a step.	p.160
Figure 119. Mathematical model of Cylindabot on a slope and force diagram	p.161
Figure 120. Change in frictional coefficient on step climbing ability. Beermat algorithm applied to 50 successes per point.	p.162

Figure 121. Simulated slope data with varied frictional coefficients. 50 successes for each point using beermat algorithm. p.163

Figure 122. Friction coefficient against tangent ratio. Replotting results from Figure 118. Dash lines are lines of best fit. Only three leg angles plotted for clarity. p.164

Figure 123. Friction varied on a 1m cuboid slope. Beermat algorithm with 50 successes per point. p.165

Figure 124. Frictional coefficients varied for a cuboid slope. p.166

List of Tables

Table 1. Categorisation by action	p.16
Table 2. Morphology Comparison	p.29
Table 3. Cost of Transport	p.39
Table 4. Froude number	p.41
Table 5. Jump comparison table	p.43
Table 6. Success Rate of a Shrinking Step	p.49
Table 7. Success rate for a fixed step with a leg angle of 30 degrees	p.50
Table 8. Mathematical terms	p.54
Table 9. Which mathematical prediction were actually useful	p.63
Table 10. Simulation and maths results of a slope for wheeled robot.	p.63
Table 11. Simulated and maths results of a step for wheeled robot	p.64
Table 12. Detailed results for a step obstacle	p.67
Table 13. Detailed results for a slope obstacle.	p.70
Table 14. Size optimisation comparison	p.74
Table 15. Comparison with other platforms	p.75
Table 16. Number of legs analysis. h = height of reach, D = circumcircle diameter, R = circumcircle radius, r = inscribed circle radius, AP = area of polygon, AC = area of circumcircle, l = length of polygon side	p.84
Table 17. Robot Locomotion Modes	p.107
Table 18. Simulation parameters.	p.112
Table 19. Physical results for a step. Highlighted in yellow is the optimal leg radius.	p.117
Table 20. Physical results for a slope including tan calculations. Highlighted in yellow is the optimal leg radius.	p.118
Table 21. Robot ability comparison (Data taken from respective papers).	p.119

Table 22. Statistical measures for grid comparison of Cylindabot version and a four wheeled design.. The grid heights were determined by a 5 by 5 matrix. P-value from T-test and effect size from Cohen's.	p.127
Table 23. Checkerboard Map Results (Success rate). 100 runs per configuration (*=20 runs). Adapt 0/70 refer to adaptation of the leg angles with resetting to a leg angle of 0/70 between obstacles	p.130
Table 24. Overall success rate of leg angles to ridges and bridge obstacles. Yellow elements highlights show optimal leg angle for an obstacle	p.141
Table 25. Success rate for gap and long slope obstacles	p.144
Table 26. Overall Success Rate of Cylindabot using different Map Technics.	p.146
Table 27. Stair regulations for England taken from [113]. The rise is the height of each step and the going is the depth.	p.157
Table 28 Non-linear least squares regression for mu ratio on a slope data	p.166

Acknowledgments

My supervisors (Andy & Jon) for your steadfast support throughout the years of this thesis. Allowing the room to grow independently as a researcher and being there when I needed you. Your feedback was invaluable in slowly shaping this thesis into the jumbled mess it probably still is. Thank you for all your patience and persistence.

James Hilder for the A Pi Hat without which with the Covid situation I would not have been able to make Cylindabot. For teaching me practical skills during my Msc that were key to the success of this thesis.

My parents and Jess for keeping me motivated during the writing process and reading it to check it was completely wrong (which was mostly adding “s” and “,”).