



**Stellenbosch**

UNIVERSITY  
IYUNIVESITHI  
UNIVERSITEIT

# **Design, model and build a USAR robot platform**

Mechatronic Project 478  
Final Report

Author: Ronan Wells  
22961305

Supervisor: Mr. Wayne Swart

2023/03/17

Department of Mechanical and Mechatronic Engineering  
Stellenbosch University  
Private Bag X1, Matieland 7602, South Africa.

# Plagiarism declaration

I have read and understand the Stellenbosch University Policy on Plagiarism and the definitions of plagiarism and self-plagiarism contained in the Policy [Plagiarism: The use of the ideas or material of others without acknowledgement, or the re-use of one's own previously evaluated or published material without acknowledgement or indication thereof (self-plagiarism or text-recycling)].

I also understand that direct translations are plagiarism, unless accompanied by an appropriate acknowledgement of the source. I also know that verbatim copy that has not been explicitly indicated as such, is plagiarism.

I know that plagiarism is a punishable offence and may be referred to the University's Central Disciplinary Committee (CDC) who has the authority to expel me for such an offence.

I know that plagiarism is harmful for the academic environment and that it has a negative impact on any profession.

Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully (acknowledged); further, all verbatim copies have been expressly indicated as such (e.g. through quotation marks) and the sources are cited fully.

I declare that, except where a source has been cited, the work contained in this assignment is my own work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment. I declare that have not allowed, and will not allow, anyone to use my work (in paper, graphics, electronic, verbal or any other format) with the intention of passing it off as his/her own work.

I know that a mark of zero may be awarded to assignments with plagiarism and also that no opportunity be given to submit an improved assignment.

Signature: .....

Name: ..... Student no: .....

Date: .....

# Executive summary

<b>Title of Project</b>
Design, model and build a USAR robot platform
<b>Objectives</b>
Create a model to describe the kinematics of a Load Intuitive Module (LIM). Build a prototype Urban Search and Rescue (USAR) device which uses LIMs to climb stairs. Validate the model using the prototype.
<b>What is current practice and what are its limitations?</b>
The current practice for USAR platform ranges widely, but the most successful platforms use tracks with paddles for locomotion. These devices are effective but very expensive, so there is a need for low cost expendable USAR robots.
<b>What is new in this project?</b>
This project will introduce a model to describe a less expensive stair climbing robot platform using LIMs.
<b>If the project is successful, how will it make a difference?</b>
The model developed in this project can be used to inform future USAR designs.
<b>What are the risks to the project being a success? Why is it expected to be successful?</b>
The main risk to this project is that it does not build a working prototype in time. This risk will be mitigated through careful planning and consideration of previous pitfalls.
<b>What contributions have/will other students made/make?</b>
In 2013, Matthew Wilson developed the LIM system as a masters project at the University of Cape Town (UCT). Further development on the system was done in final year projects at UCT by students Jordan Haskel, Murray Buchanan, and Richard Daniel Powrie in 2017, 2018, and 2019 respectively.
<b>Which aspects of the project will carry on after completion and why?</b>
USAR devices using LIMs as a platform can be designed, built and tested.
<b>What arrangements have been/will be made to expedite continuation?</b>
All calculations, designs, and code will be made available to future students.

# Table of contents

<b>Plagiarism declaration</b>	<b>ii</b>
<b>Executive summary</b>	<b>iii</b>
<b>Table of contents</b>	<b>iv</b>
<b>List of figures</b>	<b>vi</b>
<b>List of tables</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Objectives . . . . .	2
1.3 Motivation . . . . .	2
<b>2 Literature review</b>	<b>4</b>
2.1 USAR Robots . . . . .	4
2.2 Load-Intuitive Modules . . . . .	5
2.2.1 Wilson's LIM robot . . . . .	6
2.2.2 Haskel's Theseus . . . . .	7
2.2.3 Buchanan's Ascender . . . . .	8
2.2.4 Powrie's Di-Wheel robot . . . . .	10
2.2.5 Gearing . . . . .	12
2.2.6 Control requirements . . . . .	13
<b>3 2D simulation</b>	<b>14</b>
3.1 Limitations . . . . .	14
3.2 Configuration . . . . .	14
3.3 Observations . . . . .	15
3.3.1 Rear wheels . . . . .	15
3.3.2 Mounting obstacles . . . . .	15
3.3.3 Rolling and flipping . . . . .	17
<b>4 Maths Model</b>	<b>20</b>
4.1 Overview and objectives . . . . .	20
4.2 Definition of motions . . . . .	20
4.3 Core equations and assumptions . . . . .	22
4.4 Boundary conditions . . . . .	24
4.5 Solving technique . . . . .	27

4.6	Required torque . . . . .	28
4.6.1	Sizing components to reduce torque requirement . . . . .	30
4.7	Required coefficient of friction . . . . .	31
<b>5</b>	<b>Design Requirements</b>	<b>33</b>
5.1	Stakeholder requirements . . . . .	33
5.2	Engineering Requirements . . . . .	34
<b>6</b>	<b>Design</b>	<b>35</b>
6.1	Selection of design parameters . . . . .	35
6.2	Body design . . . . .	35
6.3	Motor selection . . . . .	36
6.4	Control circuits and code . . . . .	36
<b>7</b>	<b>Drake Simulation</b>	<b>38</b>
7.1	Overview and objectives . . . . .	38
7.2	Selection of simulator . . . . .	38
7.3	CAD to simulation pipeline . . . . .	38
7.4	Using the simulator . . . . .	39
7.5	Validation of simulation . . . . .	39
<b>8</b>	<b>Experiment</b>	<b>40</b>
8.1	Voltage-torque calibration . . . . .	40
8.1.1	Variables . . . . .	40
8.1.2	Method . . . . .	41
8.1.3	Results . . . . .	41
8.2	Torque measurement . . . . .	42
8.2.1	Variables . . . . .	42
8.2.2	Method . . . . .	43
8.2.3	Results . . . . .	44
<b>9</b>	<b>Analysis</b>	<b>46</b>
9.1	Quantitative comparison . . . . .	46
9.2	Assessment of error . . . . .	47
<b>10</b>	<b>Conclusions</b>	<b>50</b>
<b>A</b>	<b>ECSA Outcome Self Assessment</b>	<b>51</b>
	<b>List of references</b>	<b>52</b>

# List of figures

1.1	Systems layout of Wilson's LIM device (Wilson, 2013) . . . . .	1
1.2	Powrie's "Di-Wheel" robot (Powrie, 2019) . . . . .	2
2.1	A tracked USAR robot, with paddles for obstacle climbing (Delmerico <i>et al.</i> , 2019) . . . . .	4
2.2	ANYmal, a legged USAR robot (Delmerico <i>et al.</i> , 2019) . . . . .	4
2.3	Systems layout of Wilson's LIM device (Wilson, 2013) . . . . .	5
2.4	Wilson's Robot (Wilson, 2013) . . . . .	6
2.5	Wilson's Algodox simulation of stair climbing (Wilson, 2013) . . . . .	6
2.6	Wilson's half assembly climbing a stair (Wilson, 2013) . . . . .	7
2.7	Haskel's Theseus (Haskel, 2017) . . . . .	7
2.8	Buchanan's Ascender (Buchanan, 2018) . . . . .	8
2.9	Buchanan's Ascender climbing a step (Buchanan, 2018) . . . . .	9
2.10	Powrie's Di-Wheel Robot (Powrie, 2019) . . . . .	10
2.11	The Di-Wheel robot falling due to unsynchronised LIMs (Powrie, 2019) . . . . .	10
2.12	Powrie's Di-Wheel robot climbing a step (Powrie, 2019) . . . . .	11
2.13	Two different climbing motions using different gear ratios (Powrie, 2019) . . . . .	12
3.1	Initial Algodox LIM system . . . . .	15
3.2	LIM climbing with wheel contact . . . . .	16
3.3	LIM climbing with LIM frame contact, note how the contact point slips between 1 and 2 . . . . .	16
3.4	LIM climbing with robot body contact . . . . .	17
3.5	Algodox LIM system with grousers on frame . . . . .	17
3.6	Algodox LIM system approaching obstacle horizontally . . . . .	18
3.7	Algodox LIM system approaching obstacle vertically . . . . .	19
4.1	Free body diagram of the LIM frame (component 1) . . . . .	22
4.2	Free body diagram of the LIM frame (component 1) . . . . .	23
4.3	Free body diagram of the gears (components 2-5) . . . . .	23
4.4	Relationship between LIM angle and required torque for Stage 1 motion . . . . .	29
4.5	Relationship between LIM angle and required torque for Stage 3 motion . . . . .	30
4.6	Relationship between LIM angle and required torque for Stage 4 motion . . . . .	31

4.7	Relationship between LIM angle and required torque for Stage 6 motion . . . . .	32
8.1	Relationship between motor voltage and stalling torque. . . . .	43
8.2	Relationship between torque and motion for Stage 1, with 0, 0.5, and 1 representing "None", "Partial", and "Full" respectively. . . . .	45
9.1	Required torques for each stage determined experimentally . . . . .	46
9.2	Required torques for each stage from different sources. . . . .	47

## List of tables

5.1	Stakeholder Requirements . . . . .	33
5.2	Functional Requirements . . . . .	34
5.3	Performance Requirements . . . . .	34
8.1	Torque experiment results for Stage 1 . . . . .	44
9.1	Percentage error of required torque for each stage of motion in the model and simulation. . . . .	48
A.1	ECSA outcome self assessment . . . . .	51

# Chapter 1

## Introduction

### 1.1 Background

During disasters such as earthquakes, Urban Search And Rescue (USAR) robots are used to detect victims in hazardous environments where first responders would otherwise be put at risk. Advanced USAR robots can explore and map the environment while overcoming obstacles, and deliver supplies to victims who cannot be immediately evacuated. USAR robots were first used in the aftermath of the September 11 attacks on the World Trade Centre, where they had limited success as they would frequently get stuck or damaged. Since then, designs for USAR robots with many different locomotion methods have been considered and compared in competitions such as the RoboCup Rescue Robot League and the DARPA Robotics Challenge. At present, USAR robots are typically only successful at surveillance; due to the extreme conditions in disaster zones and the urgency of rescue operations, first responders will rarely consider using USAR robots.

Another problem limiting the use of USAR robots is cost; USAR robots are prohibitively expensive so rescue organisations use them sparingly. There is a need for low-cost, expendable USAR robots. In 2013, Matthew Wilson proposed an automatically-shape-shifting platform that uses a Load Intuitive Module (LIM) in the place of regular wheels, shown in Figure 1.1 (Wilson, 2013). The LIM system uses a two outer "minor wheels" placed on a central hub that can be rotated as a "major wheel". The minor wheels are geared to the central hub such that they drive the vehicle, however if they experience high resistance, for example from hitting an obstacle, the torque will cause the major wheel to rotate instead, flipping one of the minor wheels over the obstacle to automatically climb it. This is a strong concept for an inexpensive stair-climbing robot as it only uses a single motor for both normal driving and climbing obstacles.

"LIMed" robot platforms (platforms using LIMs for locomotion) were built individually

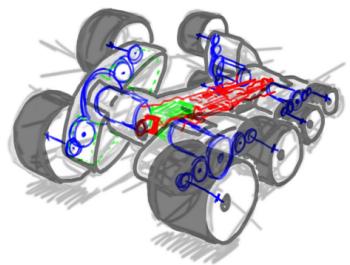


Figure 1.1: Systems layout of Wilson's LIM device (Wilson, 2013)

by four final year students at UCT (Wilson, 2013), (Haskel, 2017), (Buchanan, 2018), and (Powrie, 2019). One of these robots is shown in Figure 2.10. These platforms show some success in climbing a single step, albeit inconsistently. Powrie noted that a mathematical model that accurately describes the kinematics of the system could be developed to optimise the design of LIMed robots. This project is a continuation of these students' work.

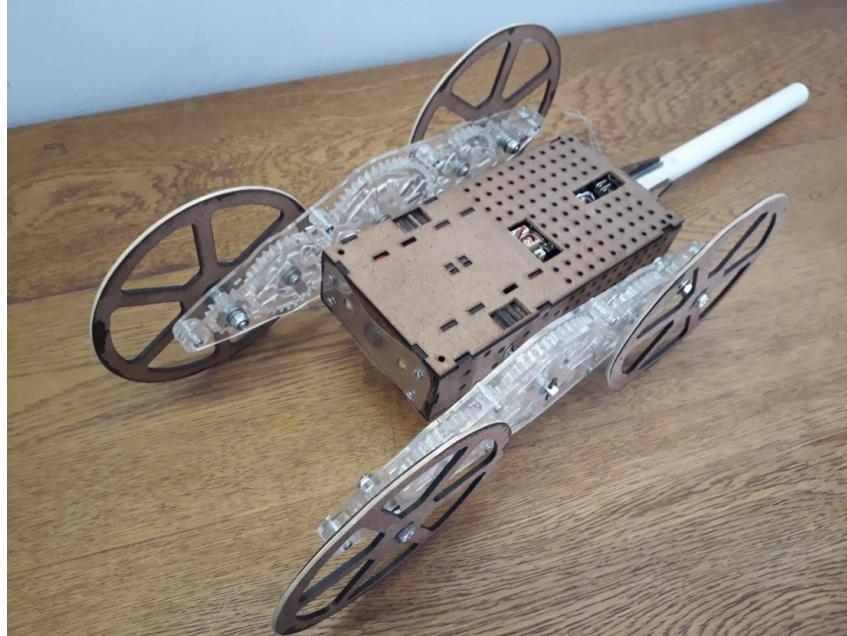


Figure 1.2: Powrie's "Di-Wheel" robot (Powrie, 2019)

## 1.2 Objectives

The aim of this project is to create a model that can be used to inform future USAR designs. The following objectives were identified to meet this aim:

1. Create a model to describe the kinematics of a LIM.
2. Build a prototype USAR device which uses LIMs to climb stairs.
3. Validate the model using the prototype.

This project does not intend to create a fully functioning USAR robot, but rather a prototype of the platform that a USAR robot may use.

## 1.3 Motivation

Many designs for USAR locomotion exist, however highly capable devices are prohibitively expensive. There remains a need for USAR devices that are both

affordable and effective, and LIMed USAR robots could fill that role in the near future. Designers will be able to use the model produced in this report to design and optimise LIMed USAR robots. These robots use fewer actuators so could be cheaper than existing USAR robots. Lowering the cost of USAR robots is a priority as it makes them more accessible to rescue organisations.

# Chapter 2

## Literature review

### 2.1 USAR Robots

In both natural and man-made disasters, USAR operations are critical for reducing casualties. Robots can be deployed in USAR operations to compliment human and canine rescuers. Robots have the advantage of being able to be deployed in scenarios too small or too dangerous for humans, and aerial robots such as quadcopters are extremely effective at quickly mapping terrain and providing situational awareness to teams. Other emerging applications of USAR robotics is remote fire fighting, victim interaction and extraction (Delmerico *et al.*, 2019).



Figure 2.1: A tracked USAR robot, with paddles for obstacle climbing (Delmerico *et al.*, 2019)



Figure 2.2: ANYmal, a legged USAR robot (Delmerico *et al.*, 2019)

In order to perform USAR operations, robots need some form of locomotion. For ground robots, this typically involves either tracks, wheels, or legs. (Delmerico *et al.*, 2019). Tracked robots with actuated paddles for obstacle climbing, such as the one shown in Figure 2.1, have been found to perform extremely well. This is evident by their representation in the winners of the Robocup Rescue Robot League (RRL), an event in which teams compete to produce robots for versatile USAR operations (Sheh *et al.*, 2016). Wheeled robots are generally the simplest and easiest to repair, but can get stuck more easily in uneven terrain. Legged robots provide the advantage of not needing a continuous path, and rapid developments in optical sensors and control systems are enabling them to be even more viable, one of these robots is shown in Figure. Wheel-leg hybrid systems will use legged motion for navigating difficult

terrain, and wheels when on smooth ground (Delmerico *et al.*, 2019).

## 2.2 Load-Intuitive Modules

A Load-Intuitive Module (LIM) refers to a wheel system proposed by Matthew Wilson, shown in Figure 2.3 (Wilson, 2013). The LIM system uses a two outer "minor wheels" placed on a central hub that can be rotated as a "major wheel". The minor wheels are geared to the central hub such that they drive the vehicle, however if they experience high resistance, for example from hitting an obstacle, the torque will cause the major wheel to rotate instead, flipping one of the minor wheels over the obstacle to automatically climb it. The system is referred to as "Load-Intuitive" because it will intuitively climb over obstacles in response to increased load on the wheels. LIMs are designed to be used in low cost USAR robots, allowing them to climb over objects without the need for many actuators.

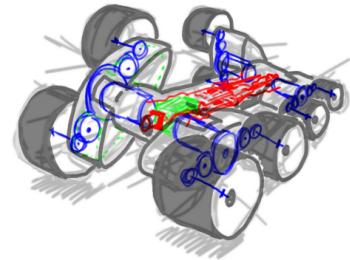


Figure 2.3: Systems layout of Wilson's LIM device (Wilson, 2013)

One advantage LIMs provide over existing locomotion methods is that they can climb obstacles higher than their profile, meaning they can enter low voids while rolling, and climb relatively tall obstacles by flipping over them. Another advantage is that LIMs require minimal actuation, one motor can be used to drive both the rolling and flipping motion, which will reduce costs when compared with other designs.

"LIMed" robot platforms (platforms using LIMs for locomotion) were built individually by four final year students at UCT (Wilson, 2013), (Haskel, 2017), (Buchanan, 2018), and (Powrie, 2019). These platforms show some success in climbing a single step, albeit inconsistently.

### 2.2.1 Wilson's LIM robot

Wilson designed and built the first LIM robot in 2013, shown in Figure 2.4. This robot was designed as a prototype for a low cost USAR star-climbing robot. At first Wilson considered only using LIMs for the front set of wheels, with the rear set using regular wheels. However, after performing a 2D simulation in Algodoor shown in 2.5, he concluded that using LIMs for the rear wheels was necessary as regular wheels provided little to no support to the climbing motion after the first step, presumably because the rear wheel would stop making contact with the stairs. Using LIMs for the rear wheels means they will be able to climb as well, and can always apply a forward force on the body.

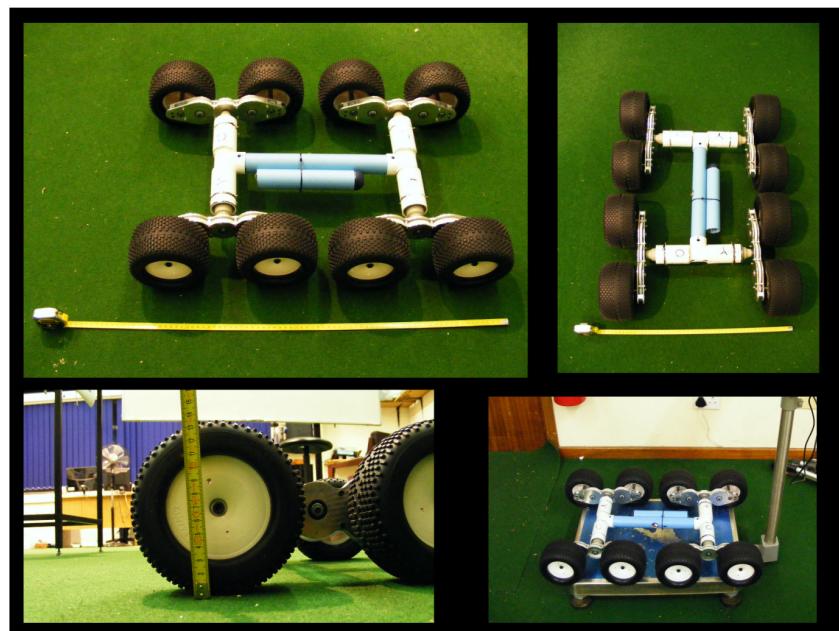


Figure 2.4: Wilson's Robot (Wilson, 2013)

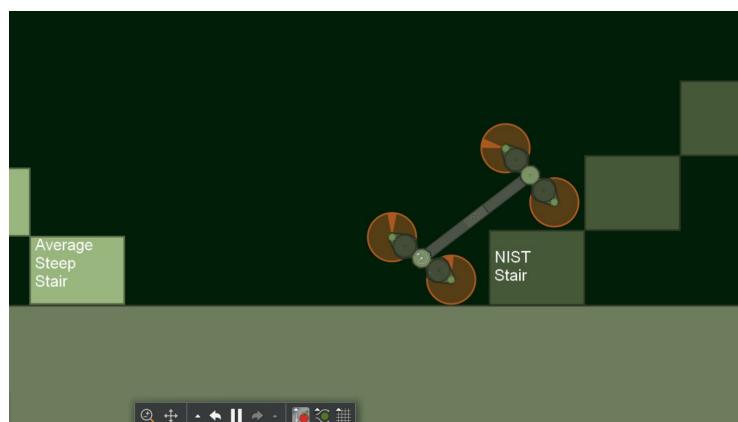


Figure 2.5: Wilson's Algodoor simulation of stair climbing (Wilson, 2013)

Wilson's robot had some limitations that prevented him from performing extensive tests. Chiefly, it was unable to climb stairs as the motors would stall upon encountering an obstacle. To validate the LIM concept in spite of this issue, Wilson split the robot in half and tested stair climbing using only the front LIMs and the chassis dragging behind as a tail. This "tail-dragging half assembly" was able to climb a single step as shown in Figure 2.6. Wilson's project ran out of time before he was able to solve the climbing motion of the complete robot, however he was able to confirm that the LIM system can climb at least a single stair in the half assembly configuration (Wilson, 2013).

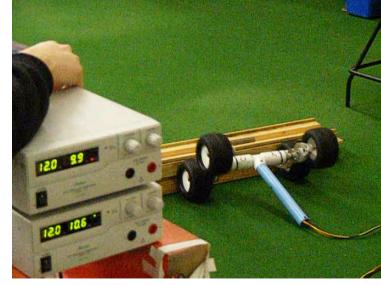


Figure 2.6: Wilson's half assembly climbing a stair (Wilson, 2013)

### 2.2.2 Haskel's Theseus

Haskel designed and built a LIMed robot to further test the concept, which he named "Theseus", shown in Figure 2.7. Unlike Wilson, Haskel assumes that using LIMs for rear wheels is not necessary for the stair climbing motion, and instead chooses to use a dragging tail to provide counter torque, similar to the tail-dragging half assembly used by Wilson. Theseus is much smaller and lighter than Wilson's robot.



Figure 2.7: Haskel's Theseus (Haskel, 2017)

Haskel tested different concepts for the tire tread, dragging tail, and gear ratios. However, none of his configurations could consistently climb a step. In the majority of step-climbing attempts, Theseus' LIMs would flip over to mount the step, but it would not be able to pull itself up. This can be attributed to a lack of grip or a lack of torque. Haskel intended to do further work on the project,

however he ran out of time due to component shortages and protests at UCT (Haskel, 2017).

### 2.2.3 Buchanan's Ascender

Buchanan designed and built "Ascender", a robot platform using LIMs for locomotion, shown in Figure 2.8. Buchanan iterated on the design several times in order to reduce mass and increase torque. The intention was build a drivetrain that could be combined with the electronics of Haskel's Theseus to produce a successful stair climbing robot. As such, the Ascender does not include any electronic control systems, and is instead controlled externally by power supplies connected to the motors.

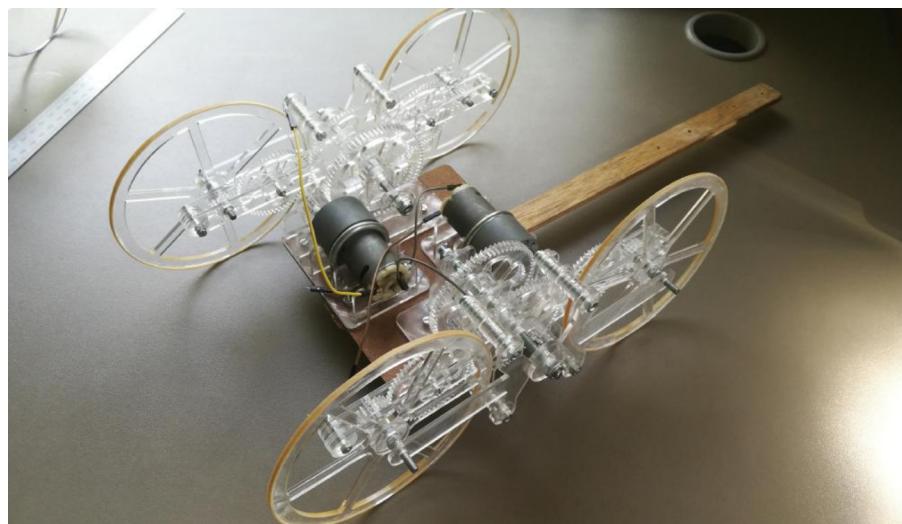


Figure 2.8: Buchanan's Ascender (Buchanan, 2018)

Buchanan's testing showed that the Ascender was able to climb a single step of 120 mm in 6 out of 10 attempts, and a step of 140mm in 2 out of 10 attempts. Buchanan noted a flaw in the design; after the LIMs flip over as part of the climbing motion, the body of the robot would lodge itself onto the the edge of the step and the wheels would spin freely, a phenomenon referred to as beaching. The LIMs would then spin until the top wheel makes contact with the top of the step, from there it would either grip and pull the robot up the step as intended, or it would dislodge the body and the robot would fall off the step. A successful climb is shown in Figure 2.9. Buchanan also reported that the Ascender was fragile to the point that it broke during the testing. Buchanan did not test the Ascender's ability to climb a staircase, but he concluded that it would be able to as a staircase is simply repeated single steps. (Buchanan, 2018)

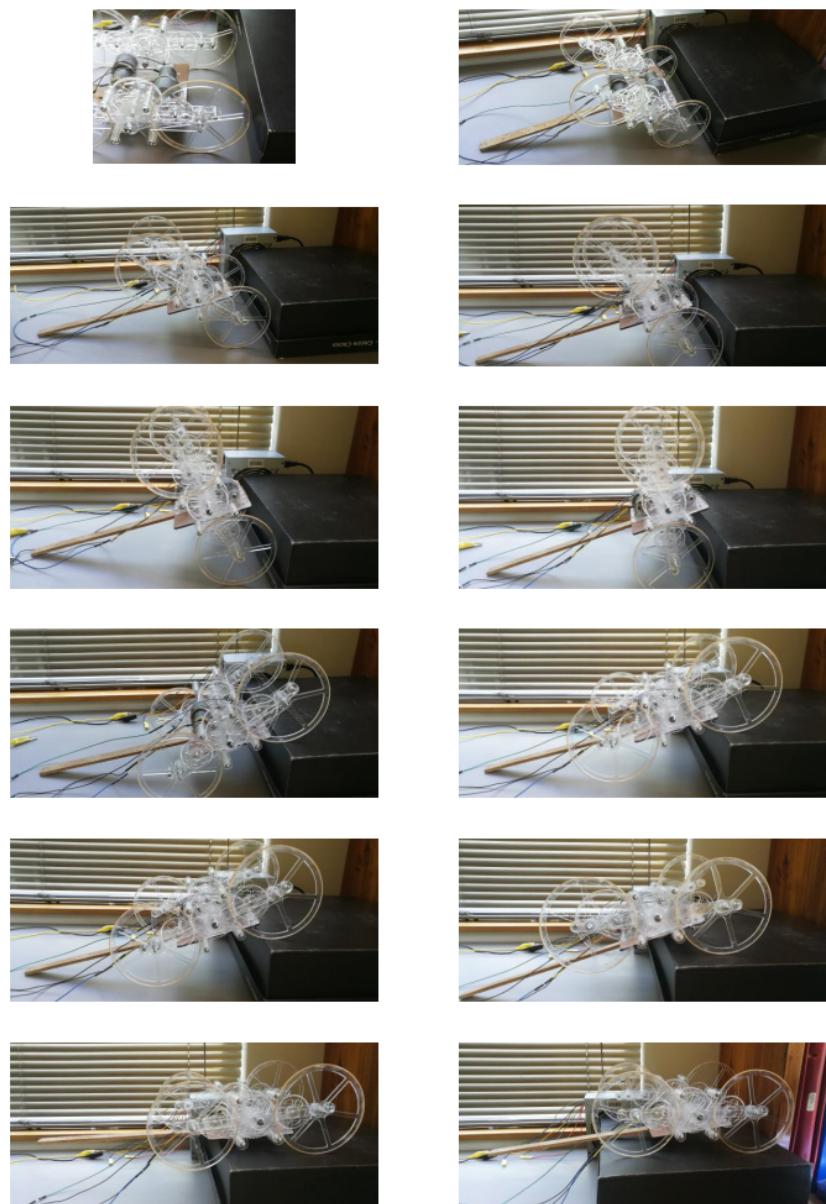


Figure 2.9: Buchanan's Ascender climbing a step (Buchanan, 2018)

### 2.2.4 Powrie's Di-Wheel robot

Powrie developed a robot using LIMs, however in his report he referred to LIMs as Di-Wheels. His reason for renaming them is that the behaviour of the LIMs does not only respond to external loads on the wheels, it also depends on the torque applied by the motors. He chose the name "Di-Wheel" in reference to a similar design by the name of "Tri-Wheel", which used three minor wheels instead of two, developed by Smith *et al.* (2015). Powrie's Di-Wheel robot is larger and more robust than Buchanan's Ascender, while being lighter than Wilson's LIMed robot. It is shown in Figure 2.10.

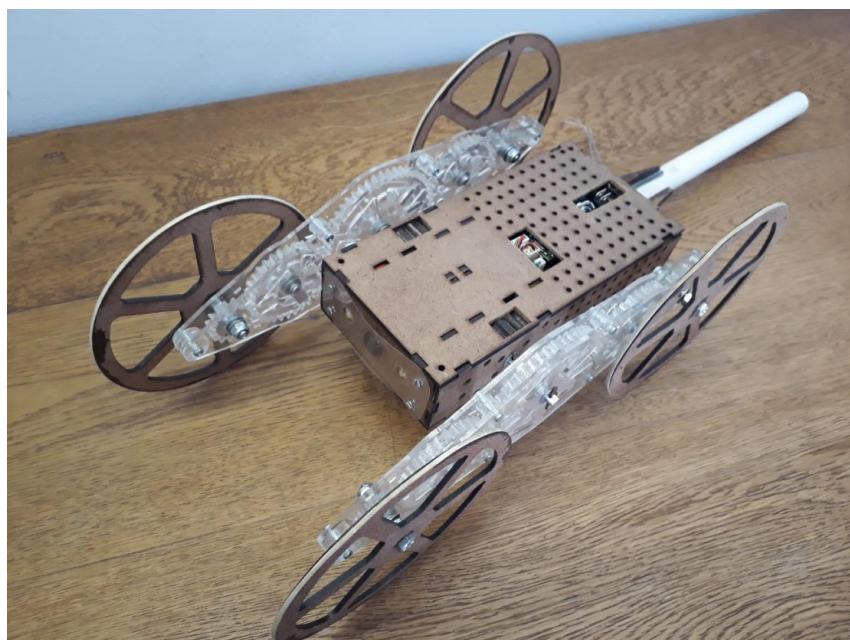


Figure 2.10: Powrie's Di-Wheel Robot (Powrie, 2019)

The Di-Wheel robot was successful in climbing a single step of 220 mm, shown in Figure 2.12. Further testing was not performed as noise from the robot's motors would interfere with the control system, preventing untethered driving. Powrie ran out of time before he was able to solve this issue. Powrie also found that when both motors are powered on, one of the LIMs would flip first, putting all the weight on the other LIM so preventing it from flipping. The result is that the robot would fall on its side, as seen in Figure 2.11 (Powrie, 2019).



Figure 2.11: The Di-Wheel robot falling due to unsynchronised LIMs (Powrie, 2019)

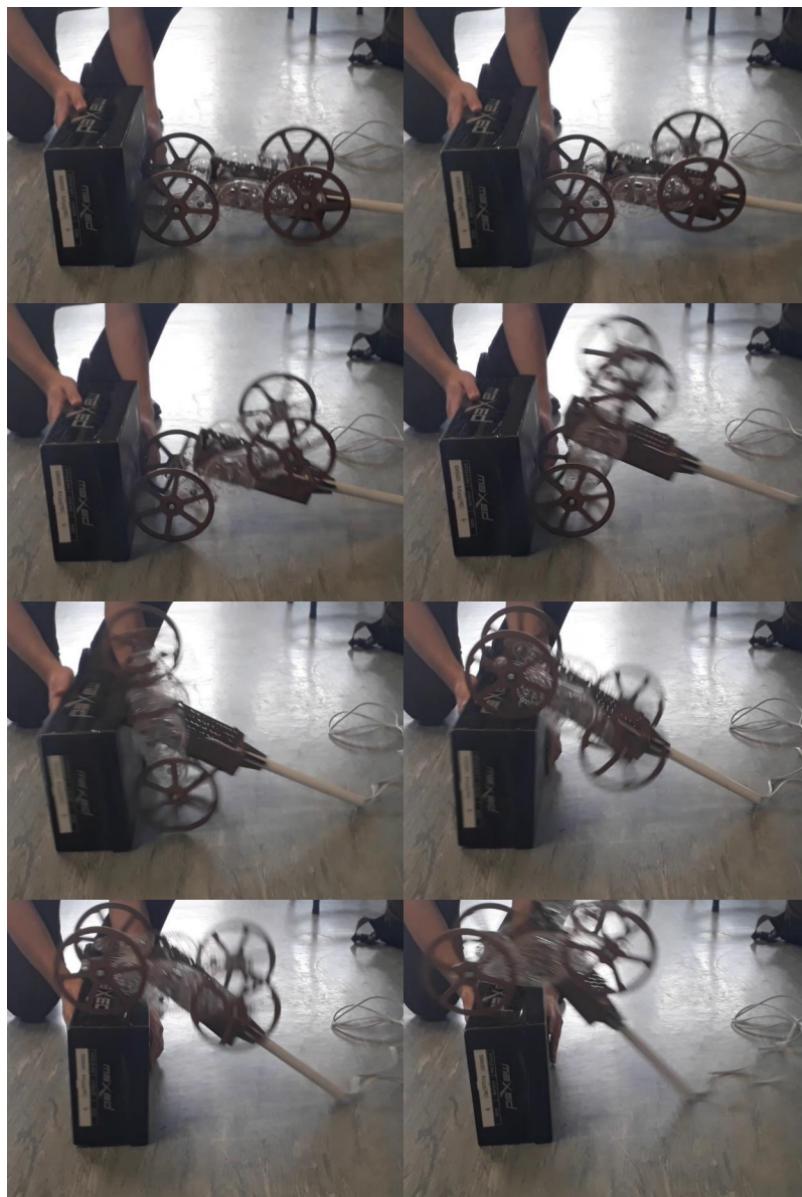


Figure 2.12: Powrie's Di-Wheel robot climbing a step (Powrie, 2019)

### 2.2.5 Gearing

The gear ratios of the LIMs will affect its motion significantly. The rotational speed of the central gear will be translated into both the speed of the LIM frame and the speed of the wheels:

$$\dot{\theta}_{central} = \dot{\theta}_{frame} + \frac{N_A}{N_C} \dot{\theta}_{wheel} \quad (1)$$

where  $\dot{\theta}_{central}$  is the angular speed of the central gear,  $\dot{\theta}_{frame}$  is the angular speed of the LIM frame,  $\dot{\theta}_{wheel}$  is the angular speed of the outer wheels relative to the frame,  $N_A$  is the number of teeth on the outer gears, and  $N_C$  is the number of teeth on the central gear.

When both the wheels and the LIM frame aren't constrained, the system is under-actuated and its motion is non-trivial. However, when the LIM frame isn't flipping (i.e. normal driving on a flat plane),  $\dot{\theta}_{frame} = 0$ , therefore:

$$\dot{\theta}_{wheel} = \frac{N_C}{N_A} \dot{\theta}_{central} \quad (2)$$

When the wheels have encountered an obstacle, such as a step, friction will prevent them from turning,  $\dot{\theta}_{wheel} + \dot{\theta}_{frame} = 0$ . In this case:

$$\dot{\theta}_{frame} = \frac{\dot{\theta}_{central}}{(1 - \frac{N_A}{N_C})} \quad (3)$$

This means that during flipping motion, if  $\frac{N_A}{N_C} > 1$ , then the LIM frame will flip in the opposite direction to the rotation of the central gear, so the front wheel will roll up the side of the obstacle (Wilson, 2013). This is ineffective for climbing steps as the LIM will never mount the step, but rather continue rotating backwards until it returns to the starting position (Haskel, 2017).

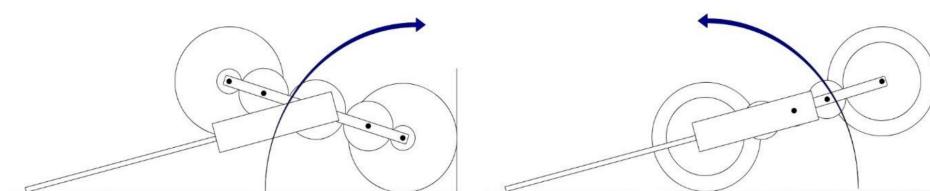


Figure 2.13: Two different climbing motions using different gear ratios (Powrie, 2019)

If  $\frac{NA}{NC} < 1$ , then the LIM frame will rotate forward, the rear wheel will flip over and mount the obstacle. This allows the LIM to climb stairs as intended. The two cases are shown in Figure 2.13, with the left showing the case when  $\frac{NA}{NC} < 1$ , and the right showing the case when  $\frac{NA}{NC} > 1$ .

### 2.2.6 Control requirements

LIMs are considered "Load intuitive" because of their ability to adapt to terrain mechanically. Open-loop control is ideal in this case, as an operator need only turn the motor on and the LIM will drive forward if it can, or attempt to climb an obstacle if it is obstructed (Wilson, 2013). However, Powrie found that his robot was not suited to open loop control. When full voltage is provided to the motor, the LIMs would flip even if when is on a flat plane. Powrie's calculations suggest that whether the LIM flips or not is largely dependent on the torque applied to it, a low torque results rolling, and a high torque results in flipping. His report suggests that LIMs only responds to terrain intuitively for a "medium torque" (Powrie, 2019). In this case a medium torque would be defined as a torque that results in rolling when the LIM is unobstructed, and flipping only when it is obstructed. This indicates that it may be necessary to have a control system that manages the torque provided to the LIMs to ensure that they do not flip on flat terrain if the motors are sufficiently powerful.

Powrie also found that when climbing a step, one LIM could flip first, putting weight on the other and preventing it from flipping, as seen previously in Figure 2.11 (Powrie, 2019). This suggest that a control system is needed to roughly synchronise the LIMs, if one is ahead of the other, more torque should be provided to the trailing LIM to correct its motion.

# **Chapter 3**

## **2D simulation**

In order to create an accurate model of the LIM system, it is important to understand how it functions. Previous reports have given some insight into this, but none of them have demonstrated a LIM robot that can climb consecutive steps. Wilson performed a simple simulation of a LIM robot in Algodoor, and concluded that the robot would need LIMs for the rear wheels in order to support consecutive stair climbing (Wilson, 2013), however all of the subsequent projects simply used a dragging tail instead of rear wheels. There is a need to resolve this inconsistency in past work, and to gain insight into the function of the LIM system. To do this, another 2D simulation using Algodoor is performed.

### **3.1 Limitations**

Algodoor is a two dimensional physics sandbox (Gregorcic and Bodin, 2017). Initial testing with the software showed that limitations on the physics engine prevent accurate simulation of gears with teeth at a centimetre scale. This means it is impossible to accurately simulate a LIM device at the scale that they would be used in reality. The simulation can be scaled up to avoid this issue. However, this prevents an accurate simulation of the kinematics of the system.

Additionally, it was found that Algodoor does not allow for the accurate simulation of an electric motor. In a typical electric motor, the available torque will decrease as the speed increases. This nuance is not present in Algodoor, so it cannot be used to provide an accurate simulation of the motor requirements. Despite these limitations. Algodoor is still useful as a tool to roughly test the motion of LIMs, and to determine how it would interact with steps. The advantage of Algodoor over other simulation methods is its ease of use, it only takes a few minutes to build a LIMed robot in Algodoor.

### **3.2 Configuration**

To improve the accuracy of the Algodoor physics engine, the simulation frequency is set to 1200 and all objects are scaled up 100 times. A basic LIM system with

a dragging tail is set up, using gear and wheel dimensions from Powrie (2019), shown in Figure 3.1.

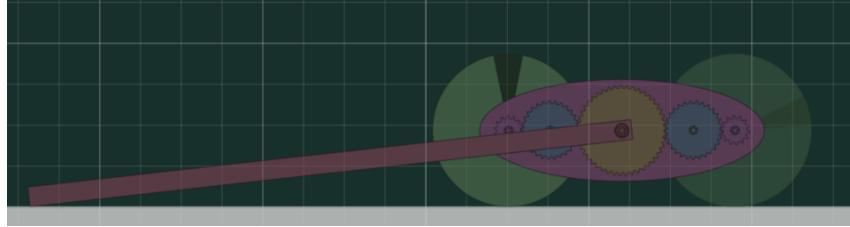


Figure 3.1: Initial Algodox LIM system

### 3.3 Observations

#### 3.3.1 Rear wheels

Simulations suggest that rear wheels are not necessary for successful consecutive stair climbing. If the motor torque is sufficient, the LIM will be able to climb steps with only a dragging tail for counter torque. It should be noted that adding a motorised rear wheel, with or without LIMs, does provide a supporting force to the front LIMs during flipping motion, so if the frontal motor cannot provide sufficient torque, rear wheels should be considered in the design.

#### 3.3.2 Mounting obstacles

There are three ways in which a LIM can mount an obstacle after flipping up to it. The first is that the wheel collides directly with the obstacle, shown in Figure 3.2. This happens when the obstacle is taller than a certain threshold based on the geometry of the LIM, and can result in the wheel bouncing off of the obstacle and failing to pull itself up. In this case a controller may be used to limit the speed of the flipping motion to ensure that the wheel is not going fast enough to bounce off the obstacle when it collides.

The second way is that the LIM frame collides with the obstacle and mounts it, then the LIM continues to rotate until the wheel makes contact with the surface of the obstacle to pull the robot forward. This case is shown in Figure 3.3. Note that the LIM frame can slip on the edge of the obstacle, which may result in failure to climb.

The third way is that the body of the robot, presented here as an extension of the tail, will mount the obstacle. This is shown in Figure 3.4. When the body has beached onto the obstacle, seen in Figure 3.4.2, there is nothing resisting the motion of either the wheels or the LIM, so they can accelerate quite quickly. If

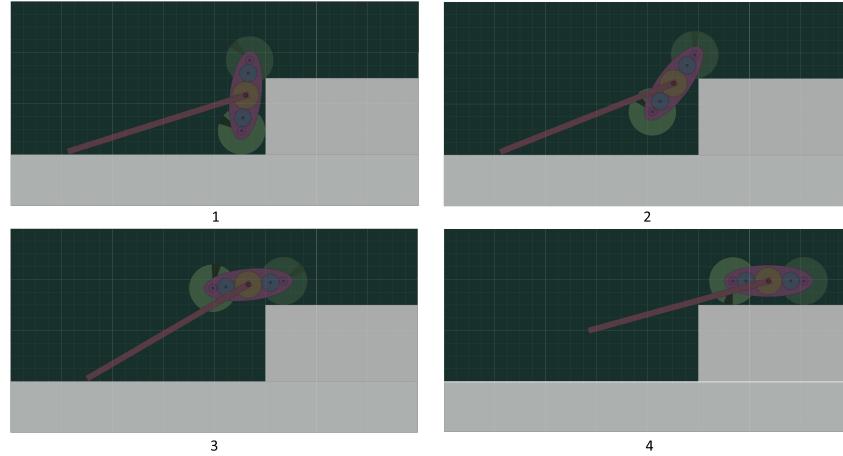


Figure 3.2: LIM climbing with wheel contact

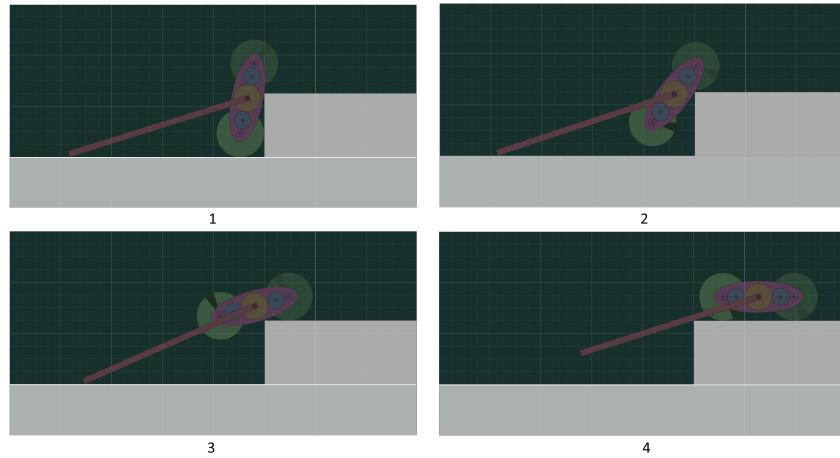


Figure 3.3: LIM climbing with LIM frame contact, note how the contact point slips between 1 and 2

they move too fast, the wheel can bounce off the obstacle when it makes contact, dislodging the body so that it falls back down to the initial position. Buchanan (2018) found that his Ascender followed this motion, which caused it to fail many of its climbing tests. He mentions that this can be avoided by moving the LIM axle to the end of the body, so that the body does not protrude beyond the LIM frame during climbing motion.

Each of these climbing methods has its flaws, however the case where the LIM frame collides with the obstacle is preferred as it reduces the chance that the wheel will bounce off the obstacle. To address slipping, grousers can be added to the robot's body (Robillard, 2019). The updated model with grousers can be seen in Figure 3.5.

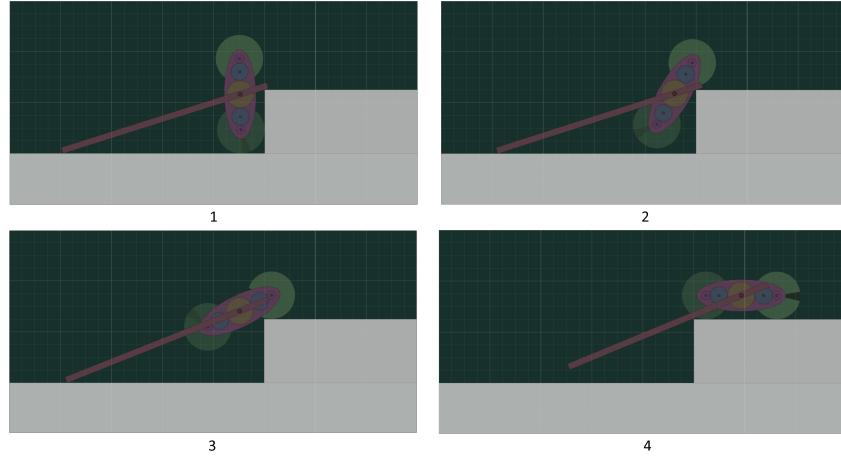


Figure 3.4: LIM climbing with robot body contact

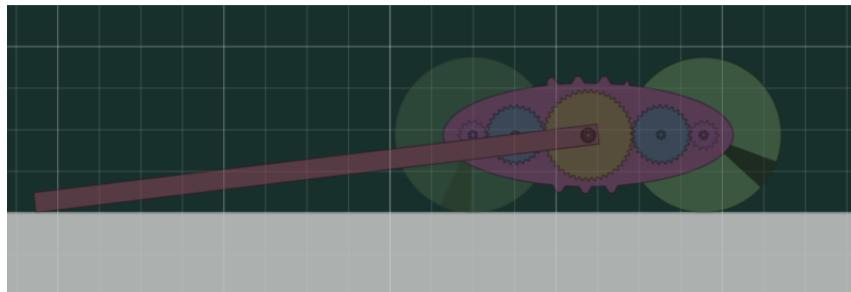


Figure 3.5: Algodoor LIM system with grousers on frame

### 3.3.3 Rolling and flipping

It was observed in simulation that the LIM would either roll or flip depending on the torque applied to it. There appeared to be a very small range of "medium torque" at which the LIM was truly load intuitive, if the motor torque was too weak it would never be able to flip over the obstacle and if it was too strong it would always flip and never roll, which hinders movement on flat terrain. However, this may simply be a result of inaccuracies of the simulation, as scaling and poor motor physics could significantly affect this motion.

In reality, as an electric motor increases in speed, the torque available will decrease proportionally. This means that once the LIM is rolling at speed, it will no longer have enough torque to flip itself unless it is stopped by an obstacle. This suggests that too much torque will only cause unintended flipping when the LIM is at rest on a flat plane.

Another observation made was that if the LIM rolls at speed into the obstacle, it will have to absorb all the energy of the impact. None of the translational kinetic energy of the robot is transformed into rotational kinetic energy of the LIM for flipping. This results in quite an inefficient system. One option to mitigate

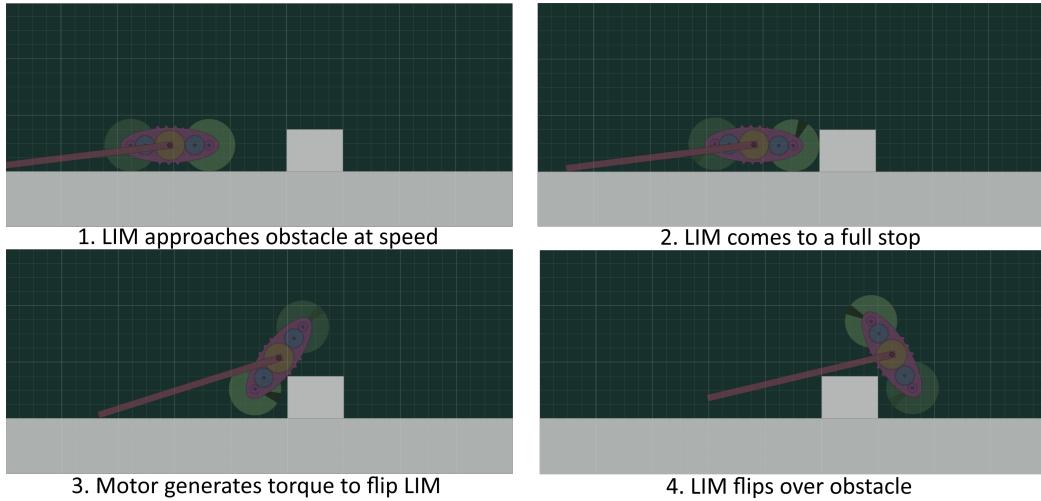


Figure 3.6: Algodoor LIM system approaching obstacle horizontally

this would be to implement a control system that solves the inverted pendulum problem in order to stand the LIMs upright on one wheel during normal operation. When the lower wheel encounters an obstacle while rolling, the LIM will readily flip over it without having to stop. The LIM could then switch to horizontal rolling when the robot needs a lower profile to enter a void. This solution would only be effective if a robust control system can be developed. These approaches are shown in Figures 3.6 and 3.7.

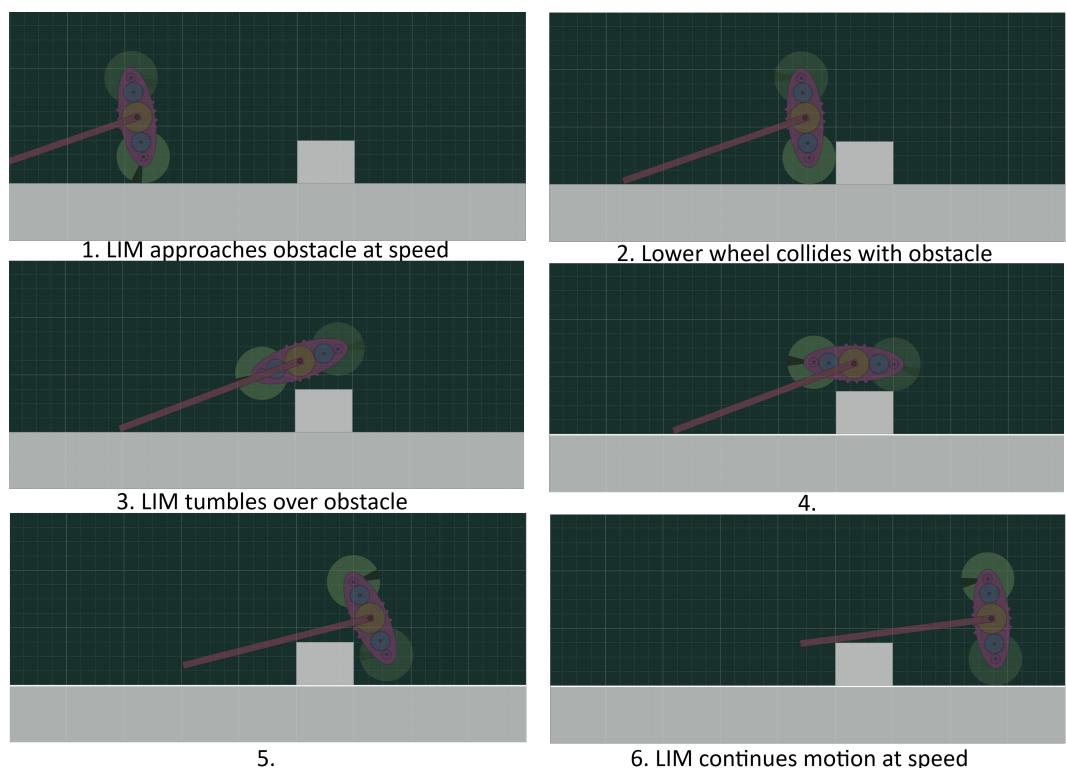


Figure 3.7: Algodoo LIM system approaching obstacle vertically

# Chapter 4

## Maths Model

### 4.1 Overview and objectives

This project develops a system of equations to analytically model the climbing motion of the device, which will be referred to as the, "Maths Model" henceforth. The intention behind this model is to allow designers to input parameters such as gear ratio, mass, wheel size, tail length, and motor torque, then determine whether the specified device will be able to climb steps. The equations can also be reversed to solve for specific parameters, such as the motor torque required to lift a device with certain properties. The maths model is developed using the MATLAB symbolic toolbox.

### 4.2 Definition of motions

As a LIMed device climbs, the tail and wheels come into contact with different surfaces on the stairs and ground, which changes how the device moves. The overall climbing motion is broken down into sequential stages which are modelled individually. These stages are defined as follows:

#### Stage 0: Rolling

When the LIMed device does is on a flat plane with no obstacle, it simply rolls forward. If the motor torque is high enough, the LIMs will flip even without an obstacle. However, DC motors lose torque as they gain speed, so the rolling motion prevents the motors from producing enough torque to flip the LIMs.

#### Stage 1: Lifting

When the front wheel of the LIM comes into contact with the first step, it is blocked by a step and fixed in place. The tail pushes against the ground and the

LIM starts rotating up the step. This motion ends when the LIM is vertical.

## Stage 2: Flipping

Once the LIM is vertical, the bottom wheel starts to roll backwards as the top wheel falls forward onto the step. The distance that the bottom wheel rolls depends on the speed of the LIM and the height of the step. If the frame of the LIM hits the edge of the step, the device may slip backwards until the top wheel makes contact with the step. This motion ends when the top wheel is on the step.

## Stage 3: Climbing

The front wheel rests on the step while the back wheel is on the ground. The tail pushes against the ground and the back wheel lifts while the front wheel simultaneously rolls forward on the step. This motion ends when the front wheel reaches the next step.

Stages 1 to 3 will repeat until the tail leaves the ground. After this point the tail will push against the edge of the previous steps.

## Stage 4: Lifting from step

Similar to Stage 1, but the tail pushes against the edge of a the previous step, which now applies a force pulling the device backwards. Assuming friction on the wheel is sufficient, the LIM starts rotating up the step. This motion ends when the LIM is vertical.

## Stage 5: Flipping from step

Similar to Stage 2, but the tail pushes against the edge of a the previous step, which now applies a force pulling the device backwards. This motion ends when the top wheel is on the step.

## Stage 6: Climbing from step

Similar to Stage 3, but the tail pushes against the edge of a the previous step, which now applies a force pulling the device backwards. The back wheel lifts while the front wheel rolls forward on the step. This motion ends when the front wheel reaches the next step; however, due to the tail force pulling the device backwards, the back wheel will continue to lift past the horizontal while front

wheel rolls forward.

### 4.3 Core equations and assumptions

The maths model consists of a system of equations that are solved simultaneously. These equations are derived from the free body diagrams of each component, and only consider movement in two dimensions. The components in question are the LIM frame, the sun gear, the idler gears, and the planet gears including the shaft and the wheels, and the body of the device. These components make contact with the surroundings and the other components, which exert equal and opposite forces on each other. As the maths model only considers movement in two dimensions, the LIMs are assumed to move synchronously. To simplify the equations, only one LIM is modelled, and its mass and moment of inertia is doubled.

Consider the device climbing a step as visualised in Figure 4.1. Each component is assigned a number, where the LIM frame is 1, the sun gear is 2, the idler gears are 3 and 4, the planet gears, including the wheels, are 5 and 6, and the body of the device, including the tail, is 7. Powering the motor causes a torque between

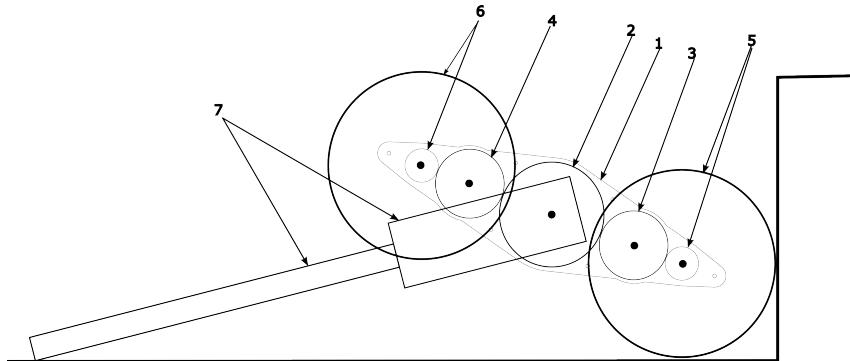


Figure 4.1: Free body diagram of the LIM frame (component 1)

the body of the device and the SUN gear. The components of the device that make contact with the environment, namely the front wheel and the tail, will be subject to external reaction forces, which are dependent on the stage of motion. These forces and torques will propagate internally through the gears, causing a forward motion. The free body diagrams for each component are shown in Figures 4.2 and 4.3.

Each component exerts a force on connected components, which exert an equal and opposite force back. For example, the the sum of forces equations for the sun gear are,

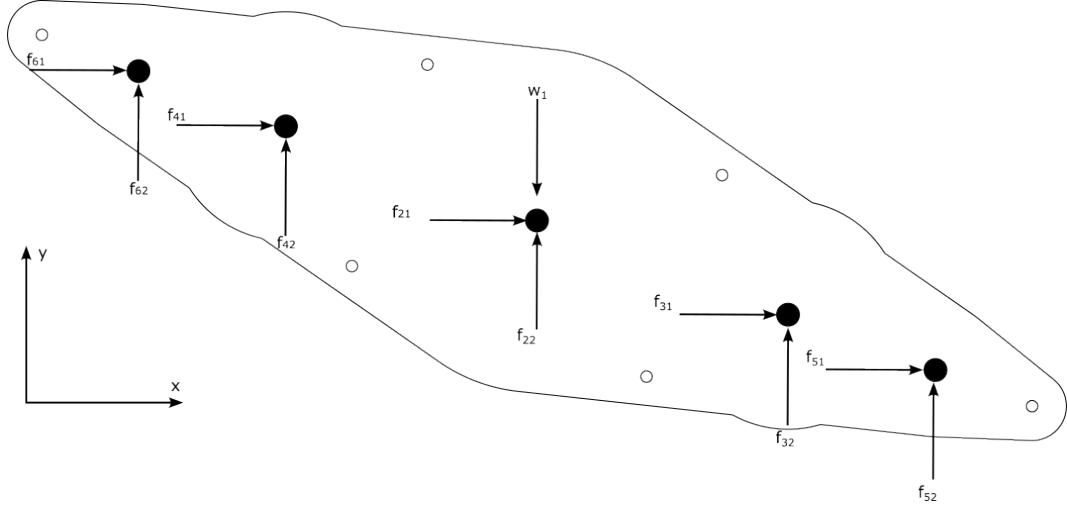


Figure 4.2: Free body diagram of the LIM frame (component 1)

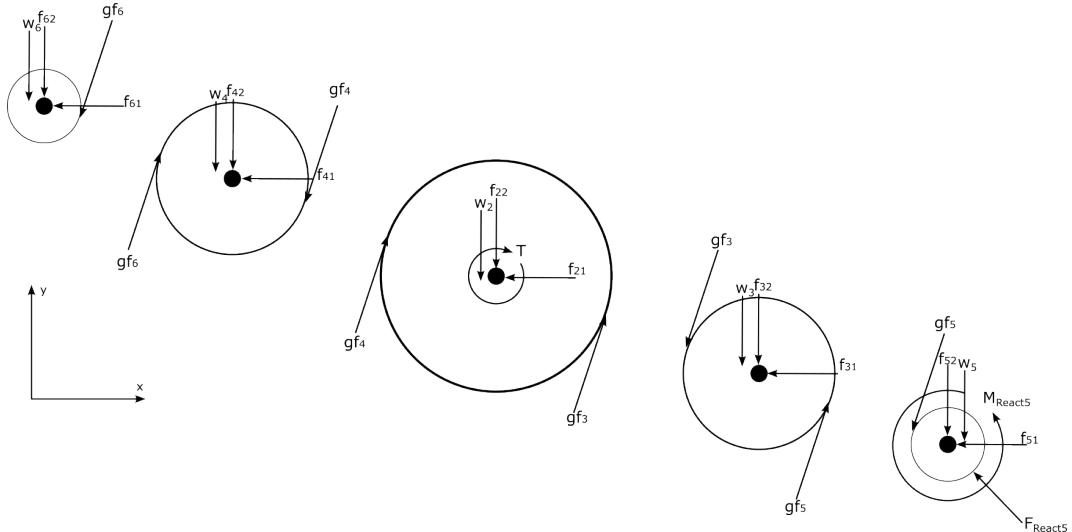


Figure 4.3: Free body diagram of the gears (components 2-5)

$$m_2 \vec{a}_2 == \vec{w}_2 - \vec{f}_2 + \vec{gf}_3 + \vec{gf}_4 \quad (4.1)$$

Where  $m_x$  refers to the mass of component  $x$ ,  $\vec{a}_x$  is the acceleration vector of component  $x$ ,  $\vec{w}_x$  is the weight vector of component  $x$ ,  $\vec{f}_x$  is the force vector that component  $x$  applies to component 1, and  $\vec{gf}_3$  and  $\vec{gf}_4$  are the force vectors that components 3 and 4 respectively apply to component 2 through the gears. This can be expanded to,

$$m_2 \begin{bmatrix} a_{21} \\ a_{22} \\ a_{23} \end{bmatrix} = \begin{bmatrix} w_{21} - f_{21} + gf_{31} + gf_{41} \\ w_{22} - f_{22} + gf_{32} + gf_{42} \\ w_{23} - f_{23} + gf_{33} + gf_{43} \end{bmatrix} \quad (4.2)$$

where vector positions 1, 2, and 3 denote the x, y, and z dimensions respectively.

This equation can be further simplified by restraining movement to the x and y dimensions, and substituting  $\vec{w}_2 = [0 \ m_2g \ 0]',$

$$m_2 \begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix} = \begin{bmatrix} -f_{21} + gf_{31} + gf_{41} \\ m_2g - f_{22} + gf_{32} + gf_{42} \end{bmatrix} \quad (4.3)$$

Similarly, the sum of moments equation for the sun gear is derived

$$I_2 \ddot{\theta}_2 = T - r_2|gf_3| + r_2|gf_3| \quad (4.4)$$

where  $I_2$  is the moment of inertia of component 2,  $\theta_2$  is the angle of component 2,  $T$  is the moment exerted by the motor on component 2, and  $r_2$  is the radius of component 2. Note that in this project, all angles and moments are given in the clockwise direction.

The full list of equations can be found in appendix ??? and in the matlab code.

## 4.4 Boundary conditions

Depending on the stage of the climbing, components of the device will come into contact with different parts of the steps and at different angles. To model climbing in each stage, a set of equations defining the contact is given for each stage.

### Stage 1: Lifting

The front wheel, referred to as component 5, is locked in place by the reaction forces from the step, this leads to the condition,

$$\vec{a}_5 = \vec{0} \quad (4.5a)$$

$$\vec{v}_5 = \vec{0} \quad (4.5b)$$

$$\dot{\theta}_5 = 0 \quad (4.5c)$$

$$\ddot{\theta}_5 = 0 \quad (4.5d)$$

where  $\vec{v}_x$  refers to the velocity of component  $x$ .

The front wheel must be placed into a two dimensional coordinate system. In stage 1, the front wheel can be on the ground or on one of the steps, so long as the tail can reach the ground, and the front wheel will always be pressed against the edge of the next step. Placing the origin at the start of the steps, the position of the wheel can be defined as,

$$\vec{s}_5 = \begin{bmatrix} \text{StepWidth} \cdot N - r_w \\ \text{StepHeight} \cdot N + r_w \\ 0 \end{bmatrix} \quad (4.6)$$

where  $\vec{s}_5$  is the position vector of the centre of the front wheel,  $N$  is the number of the step on which the wheel is placed, with  $N = 0$  placing the step on the ground, and  $r_w$  is the radius of the wheel.

The end of the tail pushes against the ground, which leads to the following conditions,

$$s_{7end2} = 0 \quad (4.7a)$$

$$F_{react71} = -\mu_{tail} F_{react72} \quad (4.7b)$$

$$-\ddot{\theta}_7 = \frac{(|\vec{l}_7|^2 - l_{72}^2)a_{12} + l_{72}v_{12}^2}{|\vec{l}_7|^3(1 - \frac{l_{72}^2}{|\vec{l}_7|^2})^{3/2}} \quad (4.7c)$$

where  $s_{7end2}$  is the y position of the endpoint of the tail,  $F_{react71}$  and  $F_{react72}$  are the reaction forces exerted onto the tail by the ground in the x and y dimensions respectively,  $\mu_{tail}$  is the coefficient of friction between the tail and the ground,  $\vec{l}_7$  is the position of the end of the tail relative to the motor axle, and  $l_{72}$  is the component of  $\vec{l}_7$  in the y dimension.

Equation 4.7c is derived from,

$$-\theta_7(t) = \arcsin \frac{l_{72}(t)}{|\vec{l}_7|} \quad (4.8a)$$

$$\frac{d^2}{dt^2} - \theta_7(t) = \frac{d^2}{dt^2} \arcsin \frac{l_{72}(t)}{|\vec{l}_7|} \quad (4.8b)$$

where  $\theta_7$  and  $l_{72}$  are treated as functions of time,  $t$ . This relation can be seen in figure ???.  $v_{12}$  and  $a_{12}$  refer to the velocity and acceleration of the motor axle in the y dimension, and are equivalent to  $-\dot{l}_{72}$  and  $-\ddot{l}_{72}$  respectively.

?????? verify

## Stage 2: Flipping

Once the LIM is vertical, there is no longer a force pushing the bottom wheel forward into the step, instead it rolls backwards. This leads to a new condition,

$$\vec{a}_5 = \begin{bmatrix} \ddot{\theta}_5 r_w \\ 0 \\ 0 \end{bmatrix} \quad (4.9a)$$

$$\vec{v}_5 = \begin{bmatrix} \dot{\theta}_5 r_w \\ 0 \\ 0 \end{bmatrix} \quad (4.9b)$$

$$s_{51} = \theta_5 r_w; \quad (4.9c)$$

$$M_{react5} = -F_{react51} r_w; \quad (4.9d)$$

where  $M_{react5}$  is the reaction moment on wheel 5 as a result of the reaction forces. The tail is still on the ground, so will still be subject to Equations 4.7.

### Stage 3: Climbing

In Stage 3, the front and back wheel have swapped. To simplify the movements, component 5 will always refer to the wheel that makes contact with the steps, and component 6 will refer to the wheel that does not.

The front wheel is rolling and the tail is on the ground, which are the same conditions as in Stage 2. As such, Stage 3 is subject to Equations 4.9 and 4.7.

### Stage 4: Lifting with tail on step

Stage 4 is similar to stage 1, except the tail is on the step. As such, this motion is subject to Equations 4.5 because the wheel is locked, and a set of new conditions because the tail is on the edge of the step,

$$u_1 = \text{StepWidth} \cdot (N_2 - 1) - s_{11} \quad (4.10a)$$

$$u_2 = \text{StepHeight} \cdot N_2 - s_{12} \quad (4.10b)$$

$$\tan(-\theta_7) = \frac{u_2}{u_1} \quad (4.10c)$$

$$-\ddot{\theta}_7 = \frac{u_1^2(a_{11}u_2 - 2v_{11}v_{12}) + u_2^2(a_{11}u_2 + 2v_{11}v_{12}) - a_{12}u_1^3 - u_1u_2(-2v_{11}^2 + 2v_{12}^2 + a_{12}u_2)}{(u_1^2 + u_2^2)^2} \quad (4.10d)$$

$$F'_{react71} = -\mu_{tail} F'_{react72} \quad (4.10e)$$

$$\vec{F}_{react7} = \begin{bmatrix} -\cos(\theta_7) & -\sin(\theta_7) & 0 \\ \sin(\theta_7) & -\cos(\theta_7) & 0 \\ 0 & 0 & 1 \end{bmatrix} \vec{F}'_{react7} \quad (4.10f)$$

where  $N_2$  is the number of the step that the tail makes contact with,  $u_1$  and  $u_2$  are the displacement from the motor axis to the point of contact with the step. Equation 4.10e is the second derivative of 4.10d.  $\vec{F}'_{react7}$  is the vector of reaction forces in a rotated coordinate system such that  $F'_{react72}$  is the normal force between the tail and the step.

### Stage 5: Flipping from step

Stage 5 is similar to stage 2, except the tail is on the step. As such, this motion is subject to Equations 4.9 because the wheel is locked, and Equations 4.10 because the tail is on the step.

### Stage 6: Climbing from step

Stage 6 is similar to stage 3, except the tail is on the step. As such, this motion is subject to Equations 4.9 because the wheel is locked, and Equations 4.10 because

the tail is on the step.

## 4.5 Solving technique

The MATLAB symbolic toolbox comes with the `solve()` function, which can be used to solve sets of non-linear simultaneous equations. However, it struggles with this large set of equations that contain trigonometric functions, often stalling. This project develops a function that breaks down the set of equations into smaller, easier to solve sets, then substituting these solutions into the remaining equations. Figure ??? shows the flowchart for this function.

The function loops through the equations until they have all been solved, or it cannot find any more solutions. First, it identifies any trivial equations that contain only one variable, such as  $a = 1$ , and solves them. This will also pick up equations with more than one solution, such as  $\sin a = 1$ ; to resolve this, the solver will look for inequalities that contain the relevant variable, in this case  $a$ , and apply them to the solution. For example, if you were to provide the function with the set of equations:

$$a = 0.5 \quad (4.11a)$$

$$\sin(b) = a \quad (4.11b)$$

$$b >= \frac{\pi}{2} \quad (4.11c)$$

$$b <= \pi \quad (4.11d)$$

The first loop would identify Equation 4.11a as it only has one variable. It wouldn't find any inequalities that describe the variable  $a$  so it would simply send the equation  $a = 0.5$  to MATLAB's `solve()` function. It then records the solution and substitutes it into the remaining equations,

$$\sin(b) = 0.5 \quad (4.12a)$$

$$b >= \frac{\pi}{2} \quad (4.12b)$$

$$b <= \pi \quad (4.12c)$$

The second loop identifies Equation 4.12a as it only has one variable. It looks for inequalities that describe  $b$  and finds Inequalities 4.12b and 4.12c. It then sends all three to MATLAB's `solve()` function, which returns the solution for  $b$ . The function then outputs a structure with the fields,

$$\text{solution.a} = 0.5 \quad (4.13a)$$

$$\text{solution.b} = \frac{5\pi}{6} \quad (4.13b)$$

Once the function can no longer find trivial solutions, it looks for sets of equations that contain the exact same variables, such as equations of the form,

$$a - 2b = 0 \quad (4.14a)$$

$$b - a + 2 = 0 \quad (4.14b)$$

and sends them to be solved. For this to work, there must be at least as many equations as there are variables in the set. This function will also identify and delete duplicate equations.

Once the function can no longer find sets of equations that use the exact same variables, it will attempt to find sets of equations that fully describe a set of variables, such as,

$$a - 3b = 0 \quad (4.15a)$$

$$b - c + 2 = 0 \quad (4.15b)$$

$$a - 2c = 0 \quad (4.15c)$$

Finally, if it can't find any fully defined sets up to a certain size, it will attempt to solve the complete set of remaining equation using the `solve()` function. If the function is unable to solve all the equations, it will return a structure containing the solutions it has found, and a vector containing the remaining equations.

## 4.6 Required torque

When designing a LIM robot, it is essential to size the motor and gearbox to be able to provide enough torque to lift the LIMs. To do this, one must first determine which stage of motion requires the most torque. To find the minimum torque required to cause forward movement, we solve for the torque that results in an overall acceleration of 0 when the velocity is also 0, essentially adding the conditions:

$$\vec{a} = \vec{0} \quad (4.16a)$$

$$\vec{v} = \vec{0} \quad (4.16b)$$

$$\ddot{\theta}_1 = 0 \quad (4.16c)$$

$$\dot{\theta}_1 = 0 \quad (4.16d)$$

$$(4.16e)$$

which simplifies the dynamic equations into static equations.

The set of equations is then solved for a range of positions in each stage.

### Stage 1: Lifting

During the lifting motion, the angle of the LIMs,  $\theta_1$ , increases from  $0^\circ$  when the LIM is horizontal to  $90^\circ$  when the LIM is vertical. Figure 4.4 shows that the required torque is highest when the LIM is horizontal, which in this particular device is 1.014 Nm, or 0.507 Nm per motor.

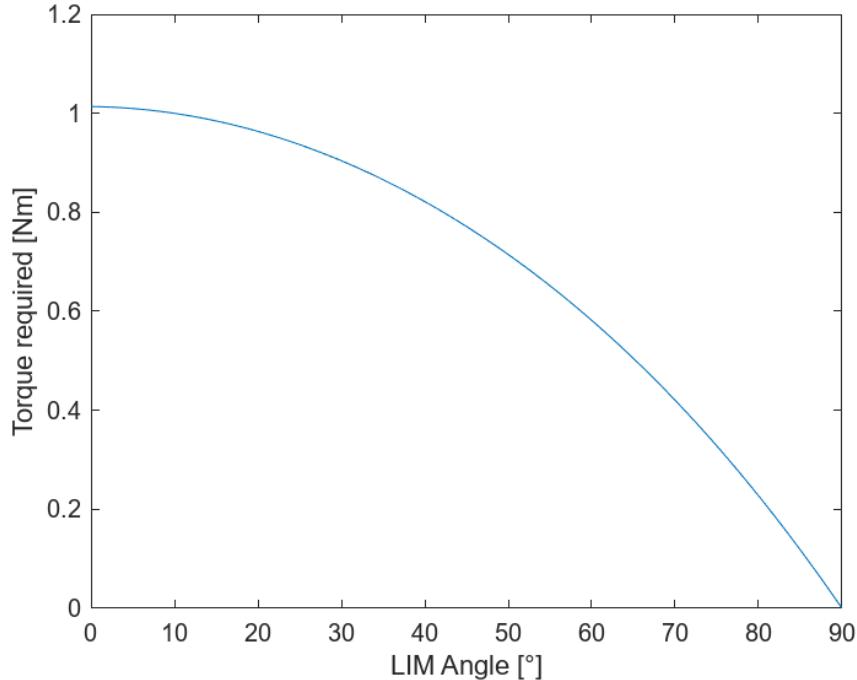


Figure 4.4: Relationship between LIM angle and required torque for Stage 1 motion

### Stage 2: Flipping

During the flipping motion, the device does not need any torque to move. Even without a torque input, it will simply fall onto the next step.

### Stage 3: Climbing

During the climbing motion, the LIMs lift from the starting position, in this case  $-42.5^\circ$  to the horizontal  $0^\circ$ . In addition, the front wheel can roll forward, instead of solving for the torque that results in an overall forward acceleration, we solve for the torque that causes a positive vertical acceleration in the back wheel. Instead of Equations 4.16, we have:

$$a_{62} = 0 \quad (4.17a)$$

$$\vec{v} = \vec{0} \quad (4.17b)$$

$$\dot{\theta}_1 = 0 \quad (4.17c)$$

$$(4.17d)$$

Figure 4.5 shows that the required torque increases as the LIM lifts, until the maximum of 0.970 Nm, or 0.485 Nm per motor when the LIM is horizontal.

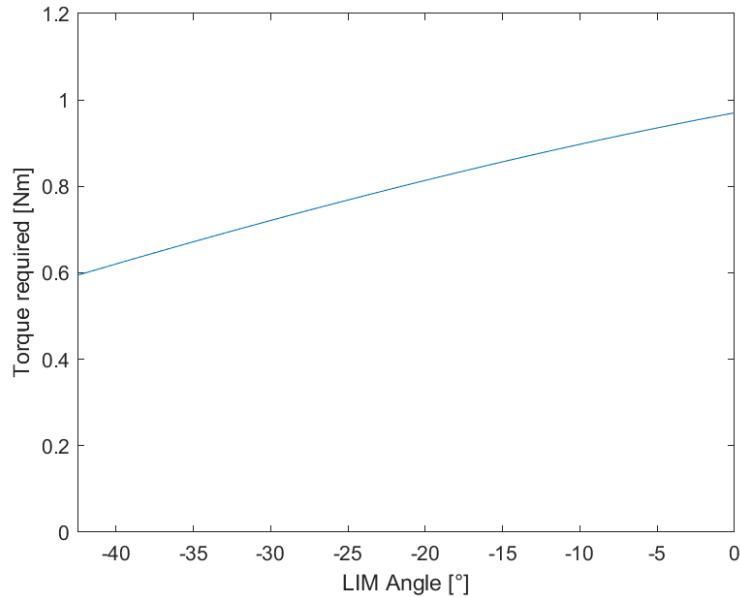


Figure 4.5: Relationship between LIM angle and required torque for Stage 3 motion

### Stage 4: Lifting from step

Similar to Stage 1,  $\theta_1$  increases from  $0^\circ$  when the LIM is horizontal to  $90^\circ$  when the LIM is vertical. Figure 4.6 shows that for this device, the required torque is highest when the LIM is at  $26.4^\circ$ , which gives a required torque of 1.295 Nm, or 0.647 Nm per motor.

### Stage 5: Flipping from step

During the flipping motion, the device does not need any torque to move. Even without a torque input, it will simply fall onto the next step.

### Stage 6: Climbing from step

Similar to Stage 3,  $\theta_1$  increases from  $-42.5^\circ$  to  $0^\circ$ , and Equations 4.17 are used instead of Equations 4.16. Figure 4.7 shows that for this device, the required torque is highest when the LIM is horizontal, which gives a required torque of 1.228 Nm, or 0.614 Nm per motor. These plots show that the highest required torques will be in Stages 4 and 6, and that the particular device modelled requires an output torque of at least 0.647 Nm on each motor.

#### 4.6.1 Sizing components to reduce torque requirement

So far this section has only modelled a device with fixed parameters, such as gear ratio, wheel size, mass, and tail length, however a designer may be interested in

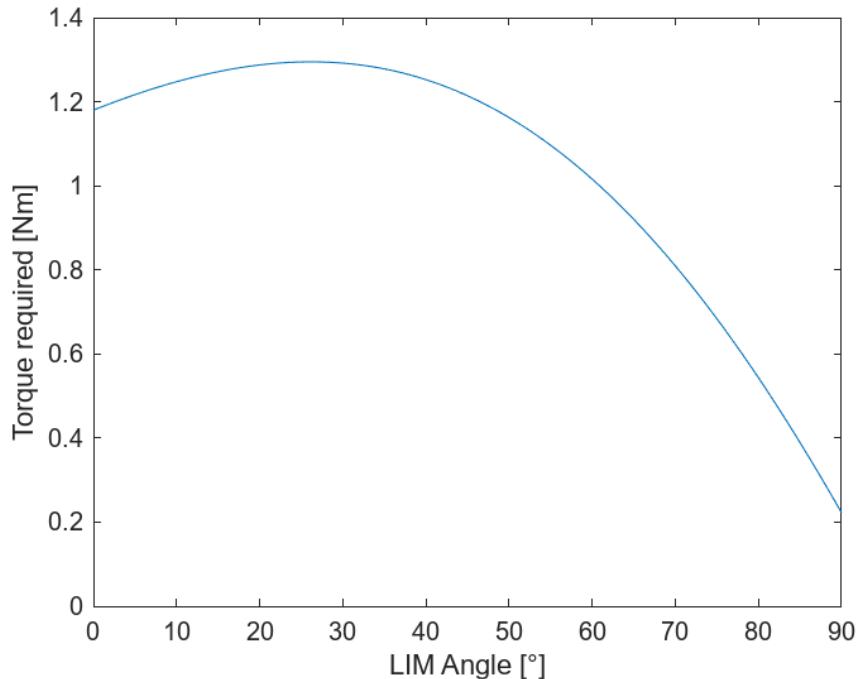


Figure 4.6: Relationship between LIM angle and required torque for Stage 4 motion

varying these parameters. The model can be used to inform the design process by showing the effect that changing certain parameters has on the required torque. For example, Figure ??? shows how changing the gear ratio affects the torque required.

## 4.7 Required coefficient of friction

When the device climbs steps, it is possible that the front wheel, component 5, will slip. Applying the coulomb approximation of friction to the model, Slipping will occur when  $|F_{51}| > \mu_5 F_{52}$ , where  $\mu_5$  is the coefficient of friction between the wheel and the step. This coefficient can be increased by using certain materials, such as rubber, for the contact surface of the wheels. However, the coefficient of friction is also dependent on the material of the step itself, which a designer would not have control over. For this reason, designers may want to modify the design to reduce the required coefficient of friction.

Similar to the required torque, the coefficient of friction required to prevent slipping can also be calculated across the motion, and the highest value is in Stage 4. The parameters can be

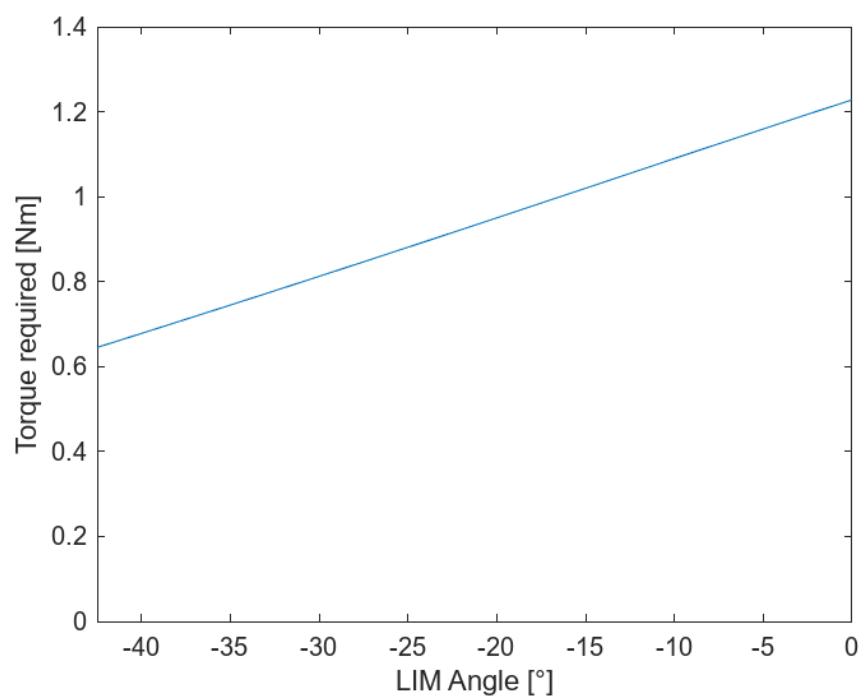


Figure 4.7: Relationship between LIM angle and required torque for Stage 6 motion

# Chapter 5

## Design Requirements

The objective of the design phase is to design a LIMed platform that can be used to validate the mathematical model. The requirements for this design are documented in this section.

### 5.1 Stakeholder requirements

This project is done at the request of Justin Pead, who supervised previous LIM projects at UCT. For the purposes of this section, this project's supervisor, Mr. Wayne Swart, acts as an intermediary between the stakeholder and the designer.

Table 5.1: Stakeholder Requirements

Number	Stakeholder	Description	Priority
SR1	Justin Pead	The device must be able to climb stairs	High
SR2	Justin Pead	The device must roll on its wheels when it is not obstructed, and climb over obstacles when it is obstructed.	High
SR3	Justin Pead	The device must use LIMs for locomotion	Must-have
SR4	Justin Pead	The device should be described by the model	Must have
SR5	Justin Pead	The device should be relevant to USAR applications	High
SR6	Wayne Swart	The design and construction of the device must demonstrate the relevant graduate attributes	Must have
SR7	Wayne Swart	The budget for the project is R5000	Must have

## 5.2 Engineering Requirements

In this section, the stakeholder requirements (SRs) are expressed as functional requirements (FRs) and performance requirements (PRs). FRs describe the actions that the system must perform, while PRs are measures of how well the system performs the functions.

Table 5.2: Functional Requirements

Number	Description	Relevant Stakeholder Requirement
FR1	Climb obstacles	SR1, SR2
FR2	Roll	SR2
FR3	Accept user control	SR5

Table 5.3: Performance Requirements

Number	Description	Target	Range	Unit	Relevant Stakeholder Requirement
PR1	Height of obstacles that can be climbed	Maximise	200+ <sup>1</sup>	mm	SR1, SR2
PR2	Climbing success rate	Maximise	90 - 100	%	SR1, SR2
PR3	Top speed	Maximise	1+	m/s	SR2
PR4	Acceleration	Maximise	1+	m/s <sup>2</sup>	SR2
PR5	Height Clearance	Minimise	50 - 300	mm	SR5
PR6	Cost	Minimise	0 - 5000	ZAR	SR6

---

<sup>1</sup>Maximum step rise specified by the SANS10400 building regulation (SAN, 2010)

# **Chapter 6**

## **Design**

In order to validate the model, a device must be designed, built and tested.

### **6.1 Selection of design parameters**

There are many parameters to consider when designing LIMs, such as gear ratios, lengths, and wheel size. Optimising the design is beyond the scope of this project, but the device does need to work to the extent that it can be used to validate the model. Initially, this project intended to use the early versions of the model to inform the design process. However, as the model was not yet validated, significantly changing key parameters could easily result in a device that is unable to climb stairs at all. Previous projects were able to show inconsistent success, albeit inconsistently. Of the previous projects, Powrie's design was the latest and most informed, and is used as a starting point for this project. Powrie's report specifies the dimensions and gear ratios for the LIMs used in this project.

### **6.2 Body design**

The body of the device is the central part that contains the motors and any peripheral devices. Powrie's design for the body includes a camera, batteries, and extra gearing to increase the torque of his motor. These components are superfluous for the objectives of this project, and are not included in the design. Additionally, the body in Powrie's design protrudes forward significantly beyond the axle of the LIMs. This is done to allow for gearing of the axial motors used in the design, but previous students have found that this protrusion causes the body to collide with the step when climbing, which can result in failure to climb the step. This project uses DC motors with worm gearbox outputs, allowing the motor to be placed at a right angle to the axis. This eliminates the need to for significant forward protrusion or additional gearing, and simplifies the construction process. Figure ??? shows the body of Powrie's design, while Figure ??? shows the body of this project's design.

## 6.3 Motor selection

Brushed DC motors often include a gearbox to increase the torque of the output at the cost of speed. For this design, the output shaft must be at a right angle to the motor, which is typically done with a worm gearbox. The most readily available motors that meet these requirements are the JGY-370 12V DC worm gear motors, which come with a range of gear ratios. Initially, the 40 RPM model was selected as it provides similar torque to the EGB-380S with added gearing that Powrie used, however initial tests indicated that additional torque was needed for consistent climbing, so the 6 RPM model was used instead.

Additionally, these motors have a self locking gearbox, which improves the controllability of the device, as shutting off the motors will cause the gearbox to lock and the device to stop moving, rather than allowing the device to fall backwards down the steps.

## 6.4 Control circuits and code

In order to validate the model, the torque of the motors must be variable and reversible. This project uses a ??? H-bridge controlled by an STM32-F303RE Nucleo board to drive the motors. Two potentiometer sliders are used to control the torque of the two motors independently, the sliders are set up as voltage dividers and connected to the ADC channels of the STM32 microcontroller. The microcontroller sends a direction and PWM signal for each motor to the ??? H-bridge, which controls the motor using PWM. The circuit diagram for this setup is shown in Figure ???.

The microcontroller is programmed to power off the motors when the sliders are centred, move the motors forward when the sliders are pushed forward, and move the motors in reverse when the sliders are pulled back. The flow diagram for this code is shown in Figure ???.

Initial tests showed that PWM duty cycle is not proportional to torque output on the motors. Instead, even at low duty cycles, the motors can lift a large load, they just do so slowly. This is likely because the self locking gearbox prevents the axle from ever falling backwards. Consider a DC motor lifting a lever as seen in figure ???.. Typically, when the PWM signal is on the motor outputs full torque and moves forward, then when the PWM signal is off the motor outputs no torque and falls backwards. The PWM frequency is typically high enough that the human eye sees this as a smooth motion. However, because of the self locking gearbox, the axle will lock when the the PWM signal is off rather than moving backwards, resulting in an overall forward motion even at low duty cycles. This is quite useful for controlling the motors as the speed of the motors is proportional to the PWM duty cycle regardless of the load it must lift.

However, for the validation of the model, the motor torque should be variable

and measurable. Brushed DC motor torque is proportional to the current running through it, and the current reduces as the speed of the motor increases. The speed of the motor is also proportional to the terminal voltage. The approximate characteristics of a DC motor are given in Figure ????. The equation describing the torque of a DC motor in relation to motor voltage and speed is then,

$$T = T_{\text{Rated-stall}} \left( \frac{V_{\text{Terminal}}}{V_{\text{Rated}}} - \frac{\omega}{\omega_{\text{Rated-No-load}}} \right) \quad (6.1)$$

where  $T$  is the output torque,  $T_{\text{Rated-stall}}$  is the stalling torque at the rated voltage,  $V_{\text{Terminal}}$  is the supply voltage across the motor terminals,  $V_{\text{Rated}}$  is the motor's rated supply voltage,  $\omega$  is the rotational speed of the output shaft, and  $\omega_{\text{Rated-no-load}}$  is the speed of the output shaft when no load is placed on it at the rated voltage.

To solve for the stalling torque,  $\omega$  is set to 0. This gives the relation,

$$T_{\text{Stall}} = T_{\text{Rated-stall}} \frac{V_{\text{Terminal}}}{V_{\text{Rated}}} \quad (6.2)$$

In order to vary the torque of the device, the terminal voltage must be varied. For this purpose, an LM2596S buck converter is used to vary the supply voltage.

# Chapter 7

## Drake Simulation

### 7.1 Overview and objectives

Instead of building a set of equations to simulate a robot from scratch, designers can use existing tools to simplify and streamline the modelling of a robot's movement. One such tool is rigid body simulation, which involves placing objects in a simulated environment and modelling the weights, joints, and contact forces of each object. This project uses rigid body simulation to model the motion of the device, with the objective to inform designers on how the device will move. By using established simulation software, it is easier to model the device using fewer assumptions than in the maths model, leading to a more accurate system.

### 7.2 Selection of simulator

The simulation programs considered in this project are PyBullet, Gazebo, and Drake. Each of these have their own advantages but Drake was selected due to its focus on accurate multibody and friction simulation.

Drake is a multibody physics engine developed and used by MIT. This project uses Drake's python bindings which are available in the package "pydrake".

### 7.3 CAD to simulation pipeline

To model a robot, drake requires a simulation description format file (SDFormat). Currently, not many CAD programs provide a method to easily generate these files. This project used Onshape, a cloud based CAD program, for this reason. The python library, onshape-to-robot, is then used to build an SDFormat file. The generated SDFormat file is then modified in order to meet Drake's specific requirements, and to apply realistic coefficients of friction to the wheels and tail.

## 7.4 Using the simulator

Once the robot has been loaded into the multibody simulation, it can be driven by setting a torque on the motor joints. Simply setting the torque on the joint to a constant value would cause the device to accelerate continuously. To accurately model the torque output of a geared DC motor, I calculate the torque using the following equation

$$\text{Torque} = \text{StallTorque} * (\text{Voltage}/12.0 - \text{speed}/\text{NLSpeed})$$

This essentially acts as a high-gain proportional speed controller. The motor voltage is set through sliders in the simulator options. The device can now be tested in simulation to determine whether it can climb stairs, and to gain insight into which components make contact with the stairs and when.

## 7.5 Validation of simulation

To validate the simulation, the torque needed for the device to climb at each stage is recorded to be compared with the real world data. In this simulated experiment, the device is placed in front of the step and a motor voltage is applied to it that corresponds to a specific stalling torque. If the device is unable to climb then the experiment is repeated with a higher voltage until the device is able to climb, at which point the torque is read from the simulation to be compared with the measured values from the real device.

Notes on friction

# Chapter 8

## Experiment

In order to validate that the model is accurate for each motion, the torque that causes the device to perform each motion determined experimentally. The equivalent values produced by the model can be compared with the measured torques to validate the model quantitatively. However, the torque produced by a DC motor cannot be measured or set directly. The torque of a DC motor is proportional to the current running through it, and when the motor isn't moving, the current is proportional to the voltage across the motor terminals. The terminal voltage is varied in this experiment.

### 8.1 Voltage-torque calibration

The relationship between terminal voltage and stalling torque is determined experimentally, this will allow subsequent experiments to measure the terminal voltage and calculate the stalling torque. To characterise the motors, an experiment was set up to determine the stalling torque produced at different input voltages. In this experiment, a the motor lifts a lever with a mass attached. The torque required to lift the lever increases as the lever angle increases, until the motor can no longer provide enough torque and stalls. The lever angle which causes the motor to stall can be used to calculate the stalling torque, specifically by using the horizontal displacement of the mass,

$$T_{Stall} = mgs_x \quad (8.1)$$

where  $T_{Stall}$  is the stalling torque,  $m$  is the mass attached to the lever,  $g$  is the gravitational acceleration, and  $s_x$  is the horizontal displacement between the mass and the motor axis. This test is repeated with different motor voltages.

#### 8.1.1 Variables

Independent variable:

- Motor voltage (V)

Dependent variable:

- Distance the weight is lifted (mm)

Controlled variables:

- Motor used
- Lever used
- Power supply used
- Multimeter used
- Ruler used

### 8.1.2 Method

1. Remove LIM from motor axle.
2. Attach the lever to the motor axle.
3. Measure the weight of the mass on a calibrated scale.
4. Drive the motor until the lever is pointing straight down.
5. Attach the mass to the lever.
6. Adjust the voltage of the power supply to the chosen value.
7. Power on the motor.
8. Wait until the lever stops moving.
9. Record the voltage across the motor terminals.
10. Quickly power off the motor. Leaving the motor on while stalling can cause damage to the motor.
11. Measure the horizontal distance that the lever has moved.
12. Drive the motor until the lever is pointing straight down.
13. Repeat steps 6 to 12 with different power supply voltages.
14. Remove the mass from the lever.
15. Repeat steps 3 to 14 with different masses.

### 8.1.3 Results

The measured results, as well as the calculated torque, are shown in Table ??, and visualised in Figure 8.1.

This plot shows a linear relationship between stalling torque and voltage, and gives the formula,

$$T_{Stall} = 0.5166V - 0.3598 \quad (8.2)$$

Mass (g)	Lever (mm)	Voltage (V)	Torque (Nm)
145	97	0.83	0.13797765
145	101	0.92	0.14366745
145	151	1.1	0.21478995
204	154	1.22	0.30819096
204	229	1.56	0.45828396
542	89	1.72	0.47321478
537	106	1.93	0.55840482
542	106	1.8	0.56360412
537	164	2.45	0.86394708
542	171	2.5	0.90921042
542	220	3	1.1697444
542	242	3.25	1.28671884
1289	117	3.5	1.47947553
1289	136	4	1.71973224
1289	160	4.5	2.0232144

where  $V$  is the motor terminal voltage. This experiment did not attempt to measure torques beyond  $2Nm$  as this risked damaging the device and was beyond the expected requirements for the device.

## 8.2 Torque measurement

This experiment aims to determine the torque required to lift the device through each stage of motion. The results of this experiment will be compared with the theoretical torques required provided by the model and the simulation. The device is placed in the starting position for each stage of motion and powered with a certain supply voltage. The motion of the device is then assessed, if the device does not move then it is marked as "None"; if the device starts to move but does not complete the stage of motion, it is marked as "Partial"; and if it completes the stage of motion, it is marked as "Full".

### 8.2.1 Variables

Independent variable:

- Motor voltage (V)

Dependent variable:

- Assessment of motion ("Full", "Partial", or "None")

Controlled variables:

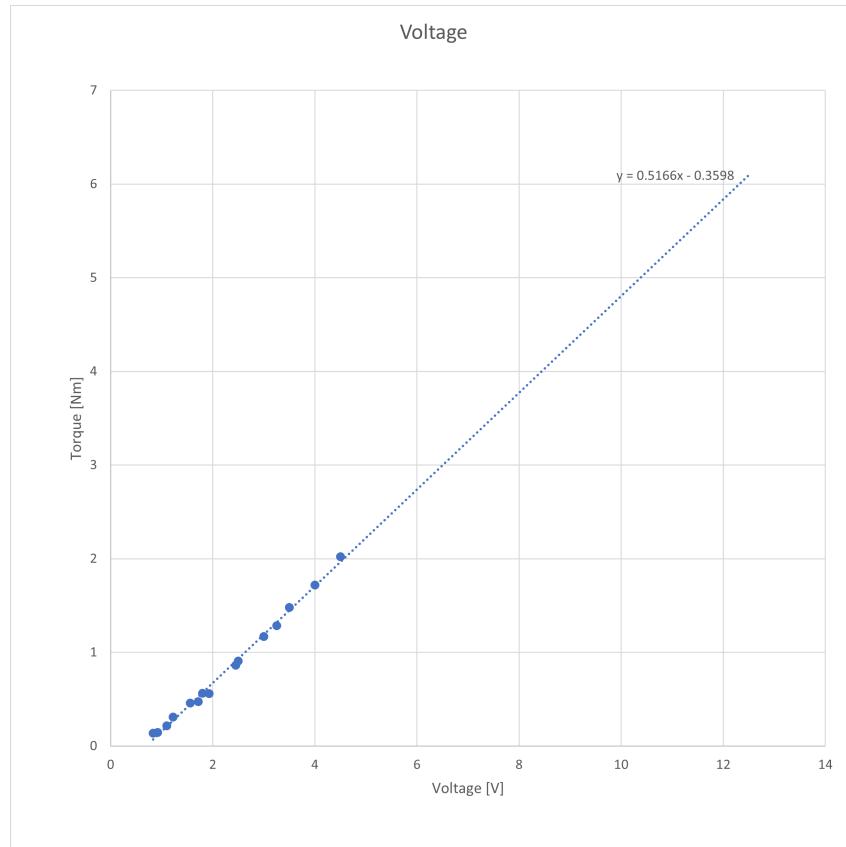


Figure 8.1: Relationship between motor voltage and stalling torque.

- LIMed device
- Stairs
- Multimeter used

### 8.2.2 Method

1. Place device in the starting position for the chosen stage of motion.
2. Adjust the voltage of the power supply to the chosen value.
3. Power on the device.
4. Observe how the device moves and classify the movement into "Full", "Partial", or "None".
5. Record the voltage across the motor terminals.
6. Power off the device.
7. Repeat steps 1 to 6 with a range of power supply voltages.

8. Repeat steps 1 to 7 for the different stages of motion.

### 8.2.3 Results

Table 8.1 shows the measured results for the Stage 1 motion, as well as the torques calculated using Equation 8.2. The data for the other stages of motion can be found in Appendix ???.

Table 8.1: Torque experiment results for Stage 1

Voltage (V)	Assessment of motion	Torque (Nm)
0.89	None	0.099974
1.08	None	0.198128
1.12	None	0.218792
1.57	None	0.451262
1.68	None	0.508088
1.74	None	0.539084
1.81	None	0.575246
1.87	None	0.606242
1.89	Partial	0.616574
1.98	Partial	0.663068
2.05	Partial	0.69923
2.11	Full	0.730226
2.13	Full	0.740558
2.25	Full	0.80255
2.26	Full	0.807716
2.3	Full	0.82838

This data is plotted in Figure 8.2, which shows that the full motion only happens with torques of at least  $0.73Nm$ , while partial motion only requires  $0.61Nm$ .

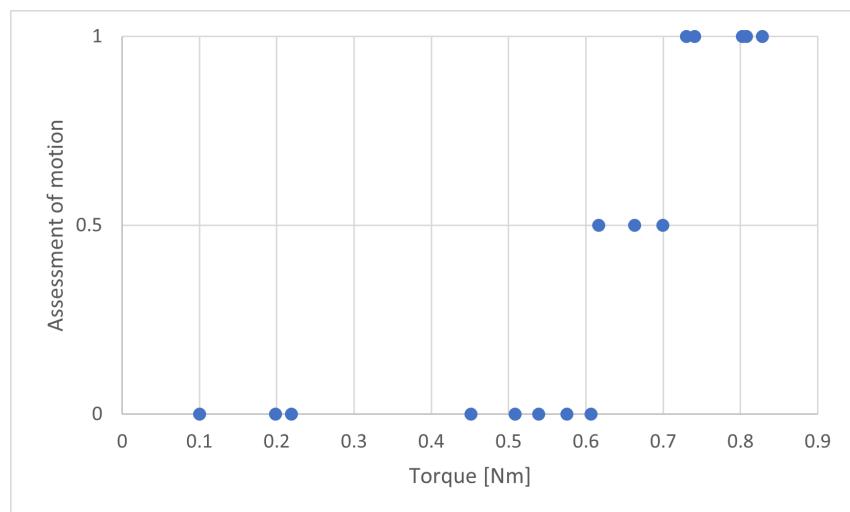


Figure 8.2: Relationship between torque and motion for Stage 1, with 0, 0.5, and 1 representing "None", "Partial", and "Full" respectively.

# Chapter 9

## Analysis

The objective of this analysis is to validate the model and the simulation. The required torque calculated in the maths model and simulated in drake is compared with the experimental data to determine the validity of the maths model and the simulation.

### 9.1 Quantitative comparison

The torque measurement experiment gives two values for each stage of motion, the torque required to cause some partial movement, and the torque required to cause full movement. These values will be referred to as  $T_{Partial}$  and  $T_{Full}$  respectively, and are shown in Figure 9.1.

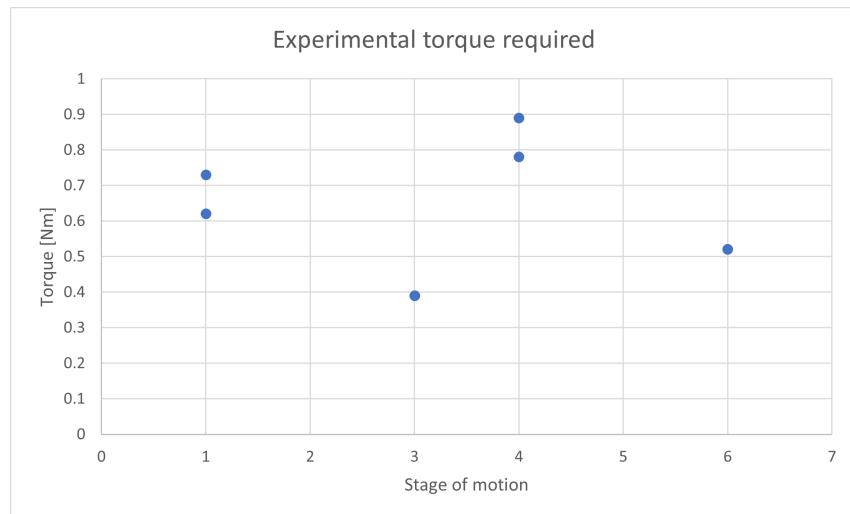


Figure 9.1: Required torques for each stage determined experimentally

In Figure 9.1, there are two points for each stage, the higher of which is  $T_{Full}$  and the lower is  $T_{Partial}$ . In cases where only a single point is present, there was no torque that would cause a partial movement; meaning that if the torque is high enough to start the movement, it is also high enough to complete the movement. Stages 0, 2, and 5 are omitted as the device does not push against

gravity in these stages, so the torque required is negligible.

To determine the required torque in the simulation, the torque measurement experiment is simulated. The torque required across each stage of motion has already been calculated using the maths model, as shown in Figure ????. The required torque at the start of each stage is equivalent to  $T_{Partial}$ , as this is the torque that will cause the motion to start. The highest required torque for each stage is  $T_{Full}$ , as if the device can output this torque it will be able to complete the full stage of motion. The required torque produced by the simulation and the maths model are shown side by side with the experimental data in Figure 9.2.

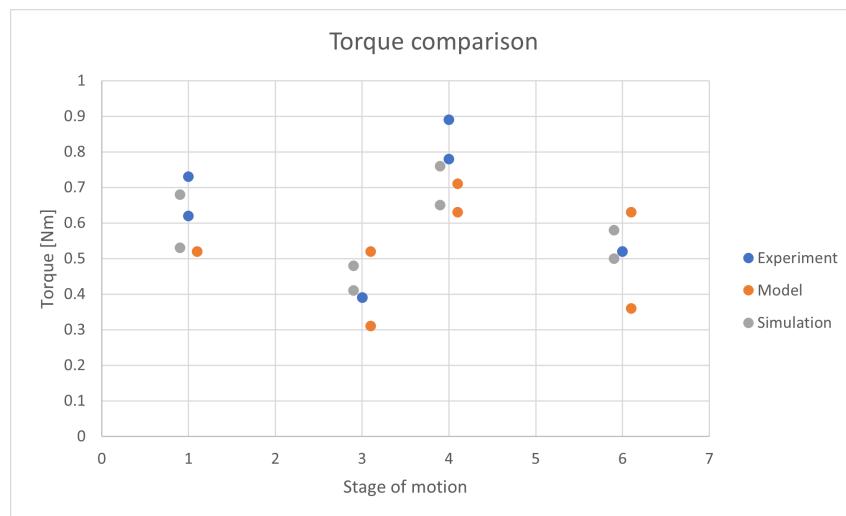


Figure 9.2: Required torques for each stage from different sources.

Figure 9.2 shows that the model, simulation, and experiment all produce similar values. In Stages 1 and 4, the model and simulation both produce a lower required torque than the device actually needs. In Stages 3 and 6, the device needs less torque to complete the motion than the model and simulation predict. To test the accuracy of the model and simulation across a range of design parameters, the experiment and comparison were repeated twice, once with a 60 cm long tail in place of the 30 cm one, and once with a 340 g mass attached to the body of the device. The comparison between these results are shown in Figures ??? and ???.

## 9.2 Assessment of error

Table 9.1 shows the percentage error of the required torque produced by the model and simulation. The simulation is shown to be more accurate than the maths model at every data point, and the point with the highest error for both is

the torque required to cause complete motion in Stage 3.

Table 9.1: Percentage error of required torque for each stage of motion in the model and simulation.

	Stage 1		Stage 3		Stage 4		Stage 6	
	Partial	Full	Partial	Full	Partial	Full	Partial	Full
Model	-16.1%	-28.8%	-20.5%	33.3%	-19.2%	-20.2%	-30.8%	21.2%
Simulation	-14.5%	-6.8%	5.1%	23.1%	-16.7%	-14.6%	-3.8%	11.5%

Identifying the source of the error is not essential for the purposes of informing design, one can simply design the motors to have at least 1.5 times the calculated required torque. However, analysing the source of this error is important in order to improve future versions of the model.

In Stage 1, the model predicts that there is no torque that will cause partial movement, as the device lifts the torque required decreases, so if the torque is high enough to start lifting, it is high enough to complete Stage 1 motion. However, Figure 9.2 shows that there is a range of torques that produce partial motion in the experimental and simulated data. The most likely cause for this is that some assumption made in the maths model is invalid. Although the calculations are done differently, the simulation is subject to the same rigid body and friction assumptions as the maths model, the main discrepancy between the two is that the simulation allows motion in three dimensions while the model is limited to two. As there is a large discrepancy between the results of the model and the simulation, it is likely that the three dimensional movement is what allows the device to get stuck part way during Stage 1. Initial tests showed that if one of the LIMs moves ahead of the other, the device will tip slightly, putting more weight on the LIM that falls behind, this causes that LIM to be unable to lift without excessive torque. An example of this is shown in Figure ??. It is likely that even when the LIMs appear to move in unison, even a slight imbalance can cause the motion to stop part way when the torque is close to the required torque. To address this error, one could update the maths model to allow movement in three dimensions, however this would drastically increase the complexity and the calculation time of the model, and is excessive when tools for three dimensional simulation such as Drake are readily available.

In Stages 3 and 6, the experiment shows that there is no torque that causes partial motion. However, the simulation indicates that there is a small range of torques that cause partial motion, and the maths model indicates that there is a large range of torques that cause partial motion. Both the simulation and the maths model also indicate that the required torque is higher than the experimental value. The reason for this error in the maths model is because it incorrectly

assumes that the frame of the LIM will not come into contact with the step. When the device climbs in Stages 3 and 6, the frame of the LIM slides along the edge of the step while the front wheel pulls the device up the step, this is illustrated in Figure ????. The step supports the frame of the LIM, so that the device needs less torque in order to overcome gravity, which explains why the actual device needs less torque to complete the stage of motion than the model predicted. To address this error, the model could be updated to model the contact between the step and the LIM frame, however this would be difficult to do analytically as the LIM frame has an irregular shape. The simulation does model the contact between the LIM frame and the step, however it still indicates that there should be partial movement and that the required torque is higher than the experimental value. This is likely because the coefficient of friction between the LIM frame and the step is higher in the simulation than in reality. Coefficients of friction in Drake default to 1.0, which is much larger than most real surfaces will have. This error could be eliminated by setting the coefficient of friction of the LIM frame to a lower value.

Another potential source of error comes from the calibration of the motors. The voltage torque calibration experiment in Section 8.1 was used to produce Equation 8.2, which was later used to calculate the experimental required torque from the measured voltage. An error in the calibration method could cause all of the experimental results to be off by a constant factor. The method of this experiment involved using the motors to lift a weight with a lever. Typically in this style of experiment, the motor torque would reach an equilibrium with the moment from the weight and the lever would stop moving. However, as the gearbox on the motors is self locking, the lever cannot move in reverse. This means that if the lever ever lifts beyond the equilibrium point, it will not fall back to the equilibrium point and the recorded lever angle will be higher than the motors could actually output. There are two possible ways that the lever can lift beyond the equilibrium point in this experiment. Firstly, if the lever and the mass have a significant momentum when they reach the equilibrium point, they will move beyond it; this is mitigated by the high gear ratio on the motor which causes it to move quite slowly. Secondly, as the mass is attached to the lever by string, it can act as a pendulum. The swing of the pendulum will sometimes make it easier for the lever to lift, and sometimes make it harder. This combines with the self locking gearbox to cause the lever to inch forward while the pendulum is helping it, then the gearbox locks when the pendulum swings the other way. This is visualised in Figure ????. The net result is that the lever moves slightly beyond the equilibrium point. If the motors incorrectly calibrated to appear to produce more torque than they actually do, it would help explain why the experimental torques are higher than the modelled and simulated torques in Stages 1 and 3. This error could be mitigated by attaching the mass directly to the lever rather than by string.

# **Chapter 10**

## **Conclusions**

# Appendix A

## ECSA Outcome Self Assessment

Table A.1: ECSA outcome self assessment

ECSA outcome	Application
Demonstrate competence to identify, assess, formulate and solve convergent and divergent engineering problems creatively and innovatively.	The design aspect of this project will require creative solutions to overcome the limitations of previous designs.
Application of scientific and engineering knowledge: Demonstrate competence to apply knowledge of mathematics, basic science and engineering sciences from first principles to solve engineering problems.	Producing the mathematical model will require kinematic calculations.
Engineering Design: Demonstrate competence to perform creative, procedural and non-procedural design and synthesis of components, systems, engineering works, products or processes.	A prototype of a USAR platform will be designed and produced.
Engineering methods, skills and tools, including Information Technology: Demonstrate competence to use appropriate engineering methods, skills and tools, including those based on information technology.	The project involves a CAD drawing and simulation of the platform, which will be based on information technology.
Professional and technical communication: Demonstrate competence to communicate effectively, both orally and in writing, with engineering audiences and the community at large.	All deliverables, including the proposal, progress report, final report, and presentation will demonstrate competent and effective communication.
Individual, Team and Multidisciplinary Working: Demonstrate competence to work effectively as an individual, in teams and in multi-disciplinary environments.	This project is done individually, with some input from the project supervisor.
Independent Learning Ability: Demonstrate competence to engage in independent learning through well-developed learning skills.	Research will be done in the literature review.

# List of references

(2010). Sans 10400-n:2010 (edition 3).

Buchanan, M. (2018). Ascender.

Delmerico, J., Mintchev, S., Giusti, A., Gromov, B., Melo, K., Horvat, T., Cadena, C., Hutter, M., Ijspeert, A., Floreano, D., Gambardella, L.M., Siegwart, R. and Scaramuzza, D. (2019). The current state and future outlook of rescue robotics. *Journal of Field Robotics*, vol. 36, no. 7, pp. 1171–1191. <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21887>. Available at: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21887>

Gregorcic, B. and Bodin, M. (2017). Algodoo: A tool for encouraging creativity in physics teaching and learning. In: *The Physics Teacher*, vol. 55, pp. 25–28.

Haskel, J. (2017). A cost effective, tele-operated observation search and rescue robot.

Powrie, R. (2019). Di-wheel robot.

Robillard, D. (2019). Di-wheel robot.

Sheh, R., Schwertfeger, S. and Visser, A. (2016). 16 years of robocup rescue. Available at: <https://doi.org/10.1007/s13218-016-0444-x>

Smith, L.M., Quinn, R.D., Johnson, K.A. and Tuck, W.R. (2015). The tri-wheel: A novel wheel-leg mobility concept. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4146–4152.

Wilson, M. (2013). Development of a low-cost, mid-sized, tele-operated, wheeled robot for rescue reconnaissance.