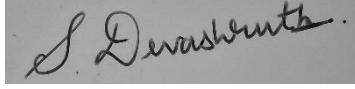


# Coursework Declaration and Feedback Form

|  |  |
|--|--|
| Student Number: <b>2715678s</b>  | Student Name: <b>Devashrutha Saiprasad</b>   |
| Programme of Study: <b>MSc in Electronics and Electrical Engineering</b>   |  |
| Course Code: <b>ENG5059P</b>   | Course Name: <b>MSc Project</b>  |
| Name of <u>First</u><br>Supervisor: <b>Dr Euan McGookin</b>  | Name of <u>Second</u><br>Supervisor: <b>Sondipon Adhikari</b>  |
| Title of<br>Project: <b>Steering redesign of Bogie Runt rover</b>  |  |
| <b>Declaration of Originality and Submission Information</b>   |  |
| <i>I affirm that this submission is all my own work in accordance with the University of Glasgow Regulations and the School of Engineering requirements.</i><br><br>Signed (Student) :  | <br>E N G 5 0 5 9 P |
| Date of Submission:  |  |

|  |                                       |
|--|---------------------------------------|
| <i>Feedback from Lecturer to Student – to be completed by Lecturer or Demonstrator</i> |                                       |
| Grade Awarded:<br>Feedback (as appropriate to the coursework which was assessed):      |                                       |
| Lecturer/Demonstrator:   | Date returned to the Teaching Office: |



# University of Glasgow

## Steering redesign of Bogie Runt rover

Devashrutha Saiprasad

2715678S

Supervised by:

Dr Euan McGookin

and

Sondipon Adhikari

Dissertation submitted in partial fulfillment of the requirements for the degree of

**MASTER OF ELECTRONICS AND ELECTRICAL ENGINEERING**

James Watt School of Engineering

College of Science and Engineering

University of Glasgow

Glasgow

March 2024

## Abstract

This dissertation details a thorough analysis and design process for a rover utilising a steerable rocker-bogie suspension framework. A mechanism renowned for its effectiveness in extraterrestrial exploration vehicles, as seen in its use in NASA's Mars rovers. The research begins with a detailed description of the rocker-bogie design, highlighting its structural nomenclature and engineering principles, such as the differential mechanism, centre of gravity (CoG) considerations, static stability factor (SSF), and the dynamics of lateral and longitudinal stability, which are crucial for maintaining stability, traction and maneuverability on uneven terrains. The literature survey extensively covers the various steering mechanisms—skid, explicit, crab, and Ackerman steering—providing a foundational understanding of their applications and limitations in rover design. A critical analysis of the Bogie-Runt rover, conducted through SolidWorks simulations, underpins the subsequent design phase, offering insights into the practical challenges and considerations involved in rover development.

In the systematic design segment, the dissertation outlines the iterative process of designing an explicit steering rover, from rocker-bogie link lengths to the intricate details of the steering mechanism and motor mounts, all illustrated through detailed SolidWorks 3D diagrams. This section showcases the technical specifications of each component and the rationale behind design choices, ensuring the rover's adaptability and functionality. The steering mechanism used to drive the rover is discussed in detail, including the wheels' relative velocity and angles. The electrical and sensor design chapter addresses the critical aspects of power regulation, distribution, and integrating actuators with the control systems, specifically the Arduino Due and the inertial measurement unit, to achieve precise movement control and environmental interaction. Circuit diagrams further explain the electrical framework, providing a blueprint of the rover's operational circuitry. Software and programming tie the mechanical and electrical components together, detailing the programming logic and algorithms that drive the rover's movements and sensor responses, emphasising the seamless integration of hardware and software for optimal performance. Finally, testing phases validate the rover's design objectives, focusing on stability, straight-line performance, minimum turning radius, on-point turning, climbing ramped surfaces, tilt angle, and manoeuvrability over obstacles. These tests align the theoretical models with actual performance, confirming the rocker-bogie system's effectiveness in these terrains and ensuring the design meets the specified engineering criteria. The entire project database is available on my [GitHub](#) page.

## Acknowledgements

I want to thank my supervisors, Dr Euan McGookin and my secondary supervisor, Sondipon Adhikari, for their constant support and guidance throughout this dissertation. Their expertise and insights have been invaluable. Their willingness to give time has been very much appreciated.

I would also like to thank the James Watt School of Engineering for providing me with the 3D printing facilities, environment for designing and testing my project.

Lastly, I thank my family and friends for their continuous support and encouragement throughout the dissertation.

## Table of Contents

|   |            |
|---|------------|
| <i>Abstract</i> .....   | <i>i</i>   |
| <i>Acknowledgements</i> .....   | <i>ii</i>  |
| <i>Table of Contents</i> .....  | <i>iii</i> |
| <i>List of Figures</i> .....  | <i>v</i>   |
| <i>List of tables</i> .....   | <i>vii</i> |
| <b>1      <i>Introduction</i>.....</b>  | <b>1</b>   |
| 1.1 Background .....  | 1          |
| 1.2 Objectives .....  | 3          |
| <b>2      <i>Literature survey</i>.....</b>                                   | <b>4</b>   |
| 2.1 Introduction to the rocker-bogie suspension .....                         | 4          |
| 2.2 Engineering aspects of the rover and rocker-bogie design.....             | 5          |
| 2.2.1 Rocker-Bogie design approach.....                                       | 5          |
| 2.2.2 Differential Mechanism.....   | 5          |
| 2.2.3 Centre of Gravity (CoG) .....   | 7          |
| 2.2.4 Static Stability Factor (SSF).....                                      | 7          |
| 2.2.5 Lateral and longitudinal stability.....                                 | 8          |
| 2.2.6 Traction .....  | 8          |
| 2.3 Steering Mechanisms .....   | 8          |
| 2.3.1 Skid steering .....   | 8          |
| 2.3.2 Explicit steering .....   | 9          |
| 2.3.3 Crab steering .....   | 9          |
| 2.3.4 Ackerman steering or Side pivot steering .....                          | 9          |
| 2.4 Analysis of the Bogie-Runt Rover .....                                    | 10         |
| <b>3      <i>Systematic design of the directional steering rover</i>.....</b> | <b>12</b>  |
| 3.1 Design of the rocker-bogie link lengths .....                             | 12         |
| 3.2 Design of the rocker-bogie pivots.....                                    | 13         |
| 3.3 Design of the bogie pivot to the rocker pivot bracket .....               | 14         |
| 3.4 Design of the steering mechanism .....                                    | 14         |
| 3.5 Design of the servo mount.....  | 16         |
| 3.6 Design of the electrical box.....   | 16         |
| 3.7 Rocker and bogie links.....   | 17         |
| 3.8 Design of the differential bar .....                                      | 18         |
| 3.8 Steering configuration .....  | 19         |
| 3.9 Final assembly and preliminary testing .....                              | 21         |

|                   |  |           |
|-------------------|--|-----------|
| <b>4</b>          | <b><i>Electrical and sensor design</i></b>                           | <b>23</b> |
| <b>4.1</b>        | <b>Power regulation</b>  | <b>23</b> |
| 4.1.1             | Preliminary test results   | 23        |
| <b>4.2</b>        | <b>Power distribution</b>  | <b>23</b> |
| 4.2.1             | Preliminary test results   | 24        |
| <b>4.3</b>        | <b>Actuators</b>   | <b>24</b> |
| 4.3.1             | Geared DC motor  | 24        |
| 4.3.2             | MG90D Servo motor  | 24        |
| 4.3.3             | Preliminary test results   | 25        |
| <b>4.4</b>        | <b>Arduino due and telemetry</b>                                     | <b>25</b> |
| 4.4.1             | Arduino Due  | 25        |
| 4.4.2             | Bluetooth module   | 25        |
| 4.4.3             | MPU-6050 IMU   | 26        |
| 4.4.4             | Optical motor encoder  | 26        |
| <b>4.5</b>        | <b>Circuit diagram</b>   | <b>26</b> |
| <b>5</b>          | <b><i>Software and Programming</i></b>                               | <b>28</b> |
| <b>5.1</b>        | <b>Programming flowchart</b>   | <b>28</b> |
| <b>5.2</b>        | <b>Input function</b>  | <b>29</b> |
| <b>5.3</b>        | <b>Calculating motor speed</b>                                       | <b>30</b> |
| <b>5.4</b>        | <b>Calculating the servo angles</b>                                  | <b>31</b> |
| <b>6</b>          | <b><i>Experimental testing and validation</i></b>                    | <b>33</b> |
| <b>6.1</b>        | <b>Straight line test</b>  | <b>33</b> |
| <b>6.2</b>        | <b>Minimum turning radius test</b>                                   | <b>35</b> |
| <b>6.3</b>        | <b>On point turning test</b>   | <b>38</b> |
| <b>6.4</b>        | <b>Driving over obstacles test</b>                                   | <b>40</b> |
| <b>6.5</b>        | <b>Maximum climbing angle test</b>                                   | <b>42</b> |
| <b>6.6</b>        | <b>Tip over angle test</b>   | <b>44</b> |
| <b>7</b>          | <b><i>Conclusions and future scope</i></b>                           | <b>47</b> |
| <b>References</b> |  | <b>48</b> |
| <b>Appendices</b> |  | <b>50</b> |
| Appendix A        | – Design drawings of the parts with dimensions in mm                 | 50        |
| Appendix B        | -- Codebase  | 57        |
| Appendix C        | – Test data and general information on parts used to build the rover | 64        |

## List of Figures

|  |    |
|--|----|
| Figure 1.1: (a) Lunokhod, (b) Apollo 15 rover, (c) Sojourner, Source-NASA.....   | 1  |
| Figure 2.1: Rocker-Bogie nomenclature, Source-NASA,JPL[4] .....  | 4  |
| Figure 2.2: Differential gearbox assembly in the Mars Exploration Rover (MER), Source-NASA, JPL[7] .....   | 6  |
| Figure 2.3: The components of the differential bar on the Mars Science Laboratory (MSL) rover, Source-NASA, JPL [9]......  | 7  |
| Figure 2.4: (a) Crab steering showing the six-wheeled rover locomotion direction with respect to the heading Source -[15], (b) Ackerman steering mechanism showing the steering pivots and the main axel, Source-Andy Dingley.....   | 9  |
| Figure 2.5: Example of explicit and skid steering towards the right, Source-[14].....  | 10 |
| Figure 2.6: Bogie-Runt designed in SolidWorks with physical dimensions in mm. ....   | 11 |
| Figure 2.7: Rocker-Bogie link lengths of the Bogie-Runt designed in SolidWorks (Units-mm). ....  | 11 |
| Figure 3.1: Rocker-Bogie link lengths of the explicit steering rover designed in SolidWorks (Units-mm).....  | 12 |
| Figure 3.2: Front 15° offset plane with 25mm diameter joint for the front link rod, side and back 0°plane with 25mm diameter joint for rear link rod to the bogie pivot, views of the rocker pivot designed in SolidWorks.....   | 13 |
| Figure 3.3: The bogie pivot's front 0° plane with 25mm diameter joint for the mid link rod, side, and back 0° plane with 25mm diameter joint for the rear link rod, views designed in SolidWorks. ....   | 13 |
| Figure 3.4: The bogie pivot to rocker pivot bracket, side, front and top views designed in SolidWorks. ....  | 14 |
| Figure 3.5: Top 40mm overall diameter, side 10mm thickness, and bottom views of the Steering Knuckle that transfers the movement from the servo to the motor mount, designed in SolidWorks. ....   | 15 |
| Figure 3.6: Top 40mm overall diameter and side views of the motor mount, designed in SolidWorks, that receives the transferred movement from the steering knuckle. ....  | 15 |
| Figure 3.7: The explicit steering mechanism of the rover, designed in SolidWorks. ....   | 15 |
| Figure 3.8: Top, side and back view of the servo mount that holds the servo motor and bogie link, designed in SolidWorks. ....   | 16 |
| Figure 3.9: Front view with dovetail 70° slit, side view with the holes for wires to enter with the bearing and rocker pivot housing, isometric view and top view with the tower at the back and standoffs for electrical components to power the rover. Designed in SolidWorks. ....                  | 17 |
| Figure 3.10: Various Links used in the design of the rocker-bogie rover. ....  | 17 |
| Figure 3.11: Top, side, bottom, and isometric views of the differential bar, designed in SolidWorks. ....  | 18 |
| Figure 3.12: The differential bar mechanism is connected to the ball joints, threaded rod, and rocker pivot. The shaft in the tower of the electrical box. Designed in SolidWorks. ....  | 18 |
| Figure 3.13: Steering angle and speed of the four steerable corner wheels. The inner wheels have angles a <sub>2</sub> , a <sub>3</sub> and speed v <sub>2</sub> . The outer wheels have angles a <sub>1</sub> , a <sub>4</sub> and speed v <sub>1</sub> . COR represents the center of rotation. .... | 19 |
| Figure 3.14: Diagram used to calculate the stall angle when the rover is climbing a ramp at an angle 'a'. ....   | 20 |
| Figure 3.15: The maximum servo angle achievable at the rear and front wheels. ....   | 21 |
| Figure 3.16: Final assembly of the rover in SolidWorks and in real life. The measurements (mm) show variations in track width and wheelbase length. ....   | 22 |
| Figure 3.17: Rocker pivot support bracket to prevent camber of the wheels when placed on the ground. ....  | 22 |
| Figure 4.1: The circuit diagram for the rover, designed in Fritzing. ....  | 27 |
| Figure 5.1: A Flow chart showing the software lifecycle of the rover, designed in Microsoft Powerpoint. ....   | 29 |
| Figure 5.2: A continuous-running void loop function that receives input from a PC keyboard through Bluetooth. Written in Arduino IDE. ....   | 30 |
| Figure 5.3: The motor speed function that calculates the speed of the wheels based on the heading angle 'alpha'. Written in Arduino IDE. ....  | 31 |
| Figure 5.4: Illustration of the maximum turning angles that can be achieved by preventing the geared motors hitting the rocker and bogie links. Designed in Microsoft Power Point. ....  | 32 |
| Figure 5.5: The servo angle function that calculates the angles at which the servos need to turn based on the heading angle. Written in Arduino IDE. ....  | 32 |
| Figure 5.6: Adjusting the servo angles takin into account for incorrect initial heading angles at the wheels. Written in Arduino IDE. ....   | 32 |

|   |    |
|---|----|
| <i>Figure 6.1: Completed motion simulation of the rover designed in SolidWorks for a straight line of 2m in length and the assembled rover set to be tested along a 2m long test track bounded by black tape.</i> ..... | 34 |
| <i>Figure 6.2: Box plot for the time taken to complete the 2m track at various speeds.</i> .....  | 34 |
| <i>Figure 6.3: Time series and scatter plot of RPM vs TIME.</i> .....   | 35 |
| <i>Figure 6.4: Motion simulation of the rover designed in SolidWorks to calculate the minimum radius of turning (mm) and the assembled rover set to be tested along a test track bounded by black tape.</i> .....       | 36 |
| <i>Figure 6.5: Shifting of the centre of rotation of the minimum radius circle by 125mm, 170mm and 305mm at S=30, S=50 and S=68, respectively.</i> .....  | 37 |
| <i>Figure 6.6: Box plot for the time taken to complete twenty rotations at various speeds.</i> .....  | 37 |
| <i>Figure 6.7: Time series plot of the inner and outer wheel RPMs.</i> .....  | 37 |
| <i>Figure 6.8: Scatter plot of RPM vs TIME showing the consistency of RPMs at various speeds.</i> .....   | 38 |
| <i>Figure 6.9: Motion simulation of the rover designed in SolidWorks for simulating on point turning and the assembled rover set to be tested along test track bounded by black tape.</i> .....                         | 39 |
| <i>Figure 6.10: Shifting of the centre of rotation of the minimum radius circle by 140mm, 135mm at S=30, S=50 respectively.</i> .....   | 39 |
| <i>Figure 6.11: Box plot for the time taken to complete twenty on point rotations at various speeds.</i> .....  | 40 |
| <i>Figure 6.12: Time series and scatter plot of RPM vs TIME.</i> .....  | 40 |
| <i>Figure 6.13: Completed motion simulation of the rover designed in SolidWorks for simulating driving over obstacles and the assembled rover tested along test track with obstacles bounded by black tape.</i> .....   | 41 |
| <i>Figure 6.14: Box plot of time taken to complete twenty laps over obstacles at speeds S2 and S3.</i> .....  | 42 |
| <i>Figure 6.15: Experimental setup for 30 degrees inclination for the rover to climb.</i> .....   | 43 |
| <i>Figure 6.16: Stall simulation to find the maximum angle of inclination at various rover mass.</i> .....  | 43 |
| <i>Figure 6.17: Distance the rover was able to climb over the ramp at the recorded IMU pitch angles.</i> .....  | 44 |
| <i>Figure 6.18: Experimental setup to calculate the tip over angle.</i> .....   | 45 |
| <i>Figure 7.1: SolidWorks drawing with physical dimensions of the rocker pivot.</i> .....   | 50 |
| <i>Figure 7.2: SolidWorks drawing with physical dimensions of the bogie pivot joint.</i> .....  | 50 |
| <i>Figure 7.3: SolidWorks drawing with physical dimensions of the rocker pivot.</i> .....   | 51 |
| <i>Figure 7.4: SolidWorks drawing with physical dimensions of the differential.</i> .....   | 51 |
| <i>Figure 7.5: SolidWorks drawing with physical dimensions of the electrical box.</i> .....   | 52 |
| <i>Figure 7.6: SolidWorks drawing with physical dimensions of the motor mount.</i> .....  | 52 |
| <i>Figure 7.7: SolidWorks drawing with physical dimensions of the servo mount.</i> .....  | 53 |
| <i>Figure 7.8: SolidWorks drawing with physical dimensions of the steering knuckle.</i> .....   | 53 |
| <i>Figure 7.9: SolidWorks drawing with physical dimensions of the rocker pivot support bracket.</i> .....   | 54 |
| <i>Figure 7.10: SolidWorks drawing with physical dimensions of the rocker pivot to bogie pivot bracket.</i> .....   | 54 |
| <i>Figure 7.11: SolidWorks drawing with physical dimensions of bogie pivot to mid motor.</i> .....  | 55 |
| <i>Figure 7.12: SolidWorks drawing with physical dimensions of the bogie pivot to back servo motor link.</i> .....  | 55 |
| <i>Figure 7.13: SolidWorks drawing with physical dimensions of the rocker to front motor mount link.</i> .....  | 56 |
| <i>Figure 7.14: Straight line performance.</i> .....  | 64 |
| <i>Figure 7.15: Minimum turning radius test.</i> .....  | 64 |
| <i>Figure 7.16: Turning on the spot test.</i> .....   | 64 |
| <i>Figure 7.17: Driving over obstacles test.</i> .....  | 65 |
| <i>Figure 7.18: Climbing a ramp to find maximum inclination angle.</i> .....  | 65 |
| <i>Figure 7.19: Finding the tilt angle.</i> .....   | 65 |

## List of tables

|  |    |
|--|----|
| <i>Table 4.1: Electrical specifications of the variable and fixed DC-DC buck converters[17] [18].</i> .....                                      | 23 |
| <i>Table 4.2: Electrical specifications of the L298N motor driver [19].</i> .....  | 24 |
| <i>Table 4.3: Electrical specifications of the geared DC motor [20] .</i> .....  | 24 |
| <i>Table 4.4: Electrical specifications of the SG-90 servo motor [21].</i> .....   | 24 |
| <i>Table 4.5: Electrical specifications of the Arduino Due [22].</i> .....   | 25 |
| <i>Table 4.6: Description of components used in the circuit diagram for the rover.</i> .....   | 27 |
| <i>Table 6.1: Standard deviations for the time taken to complete the course and the RPMs of the inner and outer wheels.</i> .....                | 34 |
| <i>Table 6.2: Standard deviations for the time taken to complete twenty rotations and the RPMs of the inner and outer wheels.</i> .....          | 36 |
| <i>Table 6.3: Standard deviations for the time taken to complete twenty rotations on point and the RPMs of the inner and outer wheels.</i> ..... | 39 |
| <i>Table 6.4: Standard deviations for the time taken to complete twenty laps over obstacles.</i> .....   | 41 |

## 1 Introduction

### 1.1 Background

The roots of exploring planetary surfaces began at NASA's JPL (Jet Propulsion Laboratory) in the early 1960s. NASA gave General Motors (GM) a contract to develop a Surveyor Lunar Rover Vehicle (SLRV) as part of an unmanned mission program. However, it was decommissioned due to a lack of mechanical stiffness to hold the lunar equipment and the inability of the state-of-the-art computers at that time to perform unmanned navigation. In November 1970, the Soviets landed the first remote-controlled lunar rover called 'Lunokhod'. The rover would be controlled using a 'move and wait command'. Later, in 1971, NASA's Apollo space program successfully deployed the Apollo Rover, which could carry around 500kg of passengers and payload. From the Apollo mission's experience and learnings, NASA began the Mars Science Microrover (MSM) program. This program led to the first Microrover to land and operate on Mars. 'Sojourner' landed on Mars on the 4<sup>th</sup> of July 1997. It weighed about 15.60kg and used the famous 'Rocker-Bogie' suspension system and the Computer Aided Remote Driving for navigation (CARD) [1].

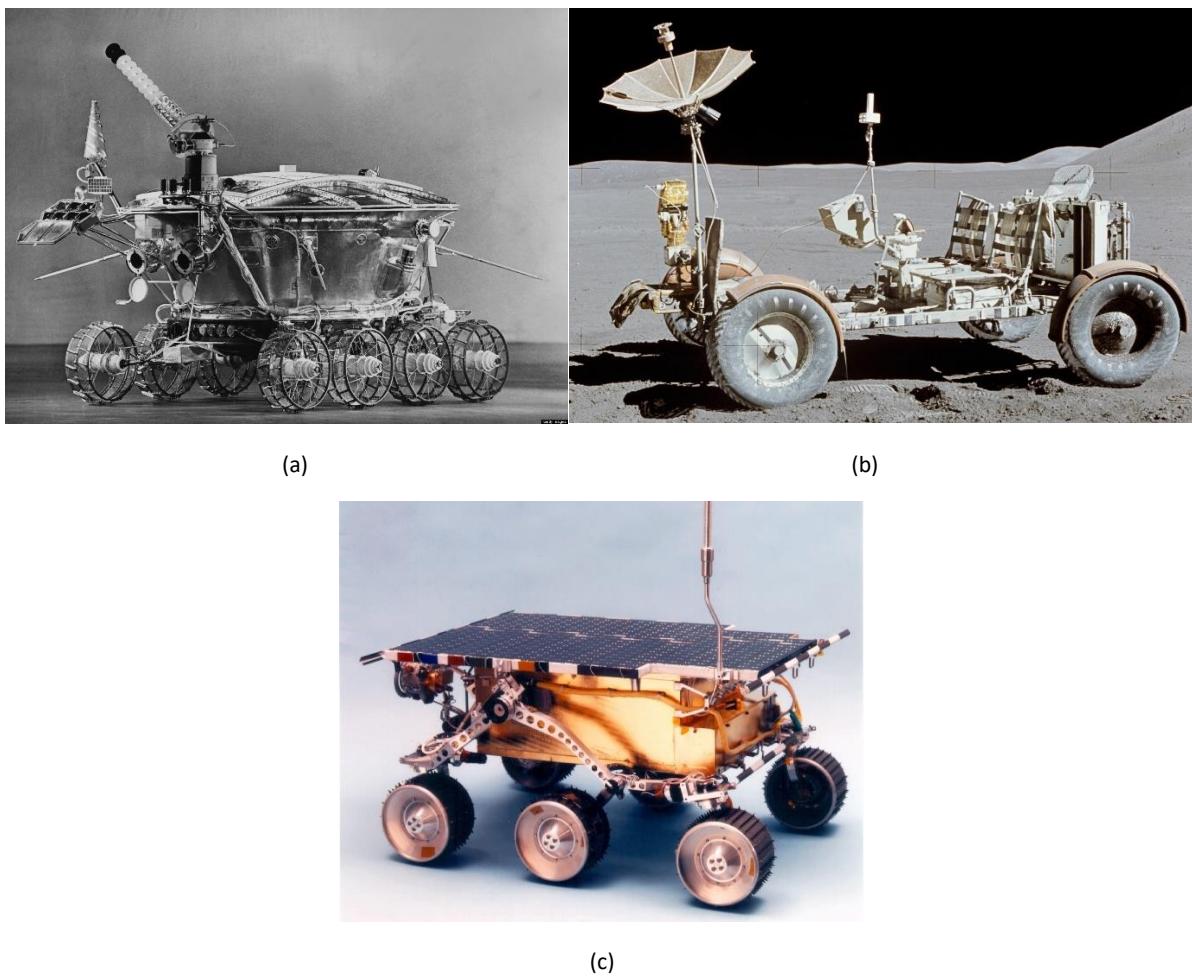


Figure 1.1: (a) Lunokhod, (b) Apollo 15 rover, (c) Sojourner, Source-NASA.

Various configurations of locomotion and subsystems are used on a planetary exploration vehicle. It ranges from the chassis, type of mobility, steering configuration, and suspension systems. The chassis can be articulated either actively or passively. It can also be fixed like the one used in the Apollo rover. The type of mobility can be continuous, like wheeled, tracked crawling, tumbling or discrete. Various steering configurations are available, such as skid, articulated, Ackerman, and crab steering. The suspension system can either be active or passive. Active suspension systems use air or coils and electronics in off-road vehicles. A more controlled and riskier environment requires a passive suspension system. Don Bickler developed the rocker-bogie suspension system during the MSM program. It falls under the passive kinematic suspension system [2].

The Sojourner rover's mobility architecture was distinguished by the implementation of explicit steering capabilities on its corner wheels, a groundbreaking feature for planetary exploration rovers at the time. This technological advancement significantly augmented the rover's navigational agility and precision across the Martian landscape, allowing it to manoeuvre around obstacles and accurately reach designated scientific targets adeptly. Integrating motorised steering mechanisms with the rover's sophisticated rocker-bogie suspension system was pivotal in ensuring optimal weight distribution and stability over uneven terrain. This was crucial for the rover's ability to traverse obstacles effectively, as the system allowed for consistent contact with the ground, even in challenging conditions. The explicit steering capability on the corner wheels and the rover's autonomous navigation algorithms allowed Sojourner to perform intricate manoeuvres, such as aligning itself with scientific points of interest and executing precise turns. This level of precision in movement and steering was instrumental in the successful execution of the rover's mission objectives, including the analysis of Martian rocks and soil. The design principles and technological innovations demonstrated by Sojourner's steering and mobility systems have impacted the design of subsequent planetary rovers. By providing valuable rover mobility and autonomous navigation lessons, Sojourner has paved the way for more advanced exploration vehicles capable of even greater scientific discovery and exploration on other planetary bodies [3]. Following Sojourner's pioneering design, subsequent Mars rovers such as Spirit, Opportunity, Curiosity, and Perseverance have incorporated and advanced the explicit steering capability on their corner wheels. Each of these rovers has built upon the initial design, enhancing mobility, autonomy, and the ability to navigate the Martian terrain with increased efficiency and precision.

The main aim of this project is to develop a rover with explicit directional steering compared to the Bogie Runts skid steering. In this process, the rover's structure and chassis will be redesigned to better suit the new rocker-bogie suspension system. Additional quality-of-life improvements will be incorporated for better usability.

## 1.2 Objectives

- Redesign the Rocker-Bogie suspension system with directional steering using servo motors while retaining the original Bogie Runt drive motors.
- Improve manoeuvrability using explicit steering.
- Improve the turning radius of the rover.
- Improve the Static Stability Factor (SSF) compared to the original Bogie Runt rover.
- Improve the climbing capabilities and the tilt angle.
- Implement a robust electrical system to enable the smooth operation of the rover while logging sensor data from the Inertial Measurement Unit (IMU) for further validation of the rover's performance.

## 2 Literature survey

### 2.1 Introduction to the rocker-bogie suspension

The rocker-bogie suspension system is tailored for low-speed traversal. It has a simple drive train with six wheels powered by six motors and a differential. The word rocker originates from the rocking feature of the most significant link. The pair of rockers are connected on each side with the help of a differential. When one of the rockers goes up, the other goes down relative to the chassis. The drive wheels are placed at the end of the rocker links. The other end is connected to the bogie. The bogies are attached through a free-rotating pivot at the other end of the rocker links. They contain the other two drive wheels at each end. The main objective of the bogies is to keep the entire vehicle stable and bear most of the vehicle's load. To climb, the front wheel is forced against the obstacle by the centre and rear wheels. The front wheels lift the rover over the obstacle as the middle wheels meet the obstacle. This process continues, and the rover quickly climbs over obstacles while always maintaining contact with the ground. This is a slow and tedious process [4].

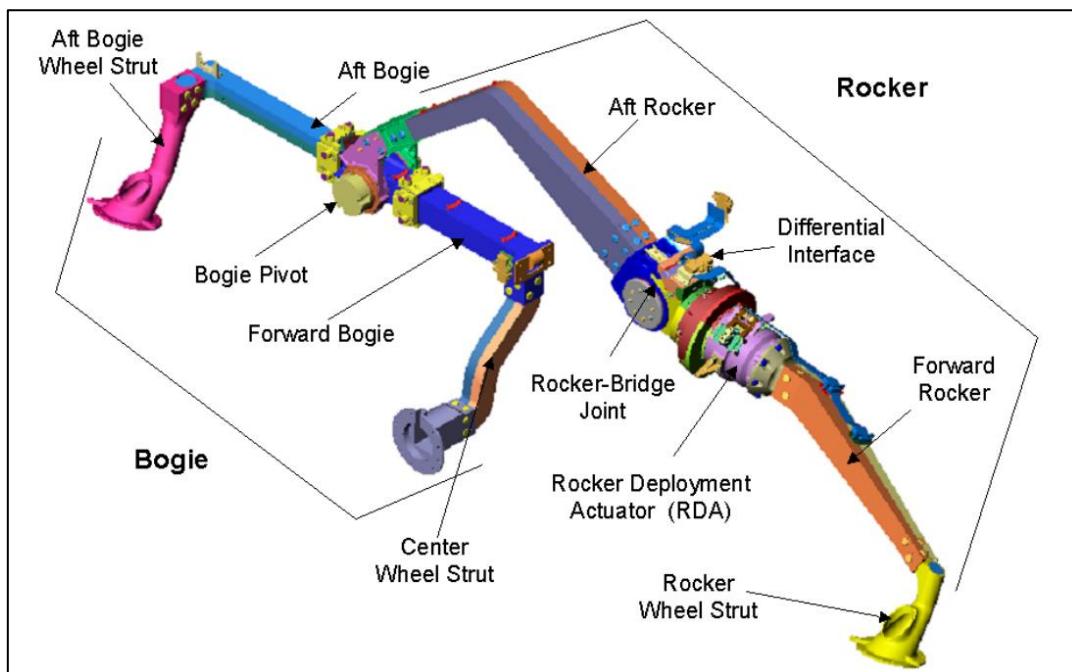


Figure 2.1: Rocker-Bogie nomenclature, Source-NASA,JPL[4].

There are several advantages to this method of climbing, such as using a rocker-bogie as the suspension system. This mechanism allows the rover to climb over obstacles while all six wheels maintain contact with the ground. The rover can traverse challenging terrain and obstacles as the two sides of the suspension move independently. Due to a differential, the chassis maintains an average pitch angle. All the wheels have independent drive motors and do not use springs or axles. This reduces the points of failure and makes the design simpler. Any jerks experienced by the wheels are transferred as a rotational force due to the use of the differential rather than being a translational force like in other suspension systems. The force of the rover is always in equilibrium with respect to the ground. This force distribution is essential as it plays a vital role in the rover's movement in soft terrain. Any excessive

pressure can result in the rover wheel sinking and getting stuck. Therefore, the suspension system must be soft to limit any excessive accelerations experienced by the rover during traversal. The challenge now is to determine how soft the suspension must be. A very soft suspension will create large deflections, and the links might come in contact with the ground, hampering the drivability of the rover. NASA proposed a suspension system between a truck and a luxury vehicle for its Mars rovers [4].

The disadvantages of the rocker-bogie suspension system are the slow traversal speed and stowing of the suspension system until deployment. Keeping the payload volume for launch is vital as it reduces the cost and decreases the complexity of the landing system. As a result, NASA decided to use a yoke and clevis design to stow the rockers and bogies. A total of six joints were incorporated into the rocker and bogie links. This system uses a latch with a pawl that clips into place when opened. A microswitch is used within the latch to tell the onboard computer that the suspension system has successfully been deployed [4].

## 2.2 Engineering aspects of the rover and rocker-bogie design

### 2.2.1 Rocker-Bogie design approach

The general method to come to a set value of the lengths of the rocker and bogie begins with certain assumptions. The obstacle's height must be comparable to the wheel's diameter. This allows the rover to climb over easily, and only one pair of wheels will be in the rising position at a time. Then, the rover's wheelbase (distance between the centre of the front and rear wheel) is decided based on the requirements and the payload it is meant to carry. The spacing between the wheels should be the same to enable proper climbing over obstacles. Then, using any CAD software, the general diagram of the links is drawn. The bogie links form a right-angled isosceles triangle, as do the links between the rocker and the rear bogie. Using Pythagoras theorem, the lengths of the unknown links can be found. This method shows that the lengths of the rocker links are the same, and the lengths of the bogie links are the same [5].

Properly designing the wheels on the rover is vital to prevent any slippage or issues with manoeuvrability on rough terrain. If the wheels are inappropriately designed, it will lead to more power consumption and the inability to climb over obstacles. Proper grip occurs when the force of traction at the wheel is lesser than the product of the normal force and the coefficient of friction of the surface in contact. An educated estimation of the wheel diameter can be made. First, an average velocity of the rover is assumed and using the relation between velocity and RPM, a suitable wheel diameter is selected [6]. An illustration of these two methods will be discussed later in this report.

### 2.2.2 Differential Mechanism

The two rocker and bogie configurations are connected through a differential mechanism. This differential mechanism can be of two types. It can be either geared or pivotal. A differential transmits the torque and rotation from one coordinate plane to another but is always normal to each other. The geared configuration was used in NASA's MER rover, but later iterations of rovers used the pivotal 'Differential bar'.

- Differential Gearbox

The geared differential is housed in the rover's main structure, the Warm Electronics Box (WEB). The differential has two epicyclic gears, as shown in Figure 2.2. Planetary and star gear configurations are used on the left and right sides. The arrangement is such that the gear ratio is 4:1 with opposite rotations. Together with the rover body, they create a three-linked differential motion. Therefore, when one side of the suspension goes up, the other side goes down. This results in the equilibrium of loads from the chassis to all the wheels and helps to reduce the ground pressure, preventing the rover from sinking in soft terrain. Titanium torque tubes connect these to assemblies as they are strong materials and can withstand the loads produced during locomotion. High-strength stainless steel was used to make the gears and bearings lubricated with perfluoropolyether (PFPE) capable of withstanding temperatures as low as  $-70^{\circ}\text{C}$  [7].

The differential gearbox assembly is severely affected by backlash between the gears. This occurs due to the small gaps between the teeth of the gear, as it is impossible to get a perfect mate between them due to the current manufacturing capabilities. Further, the gearbox must be fixed rigidly to the chassis and structurally robust to avoid any change in the axis of rotation of the gears [8].

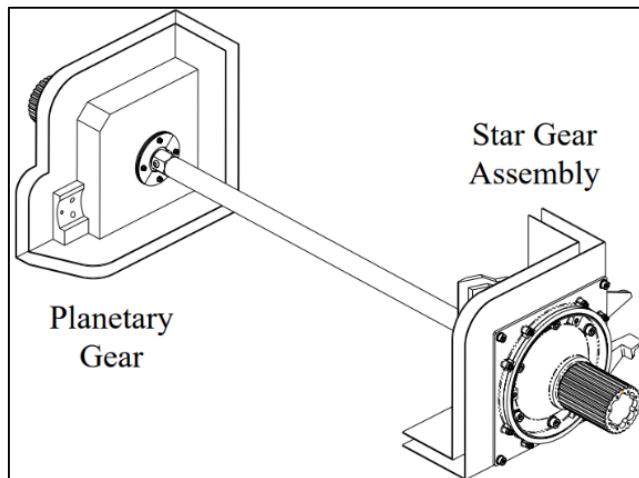


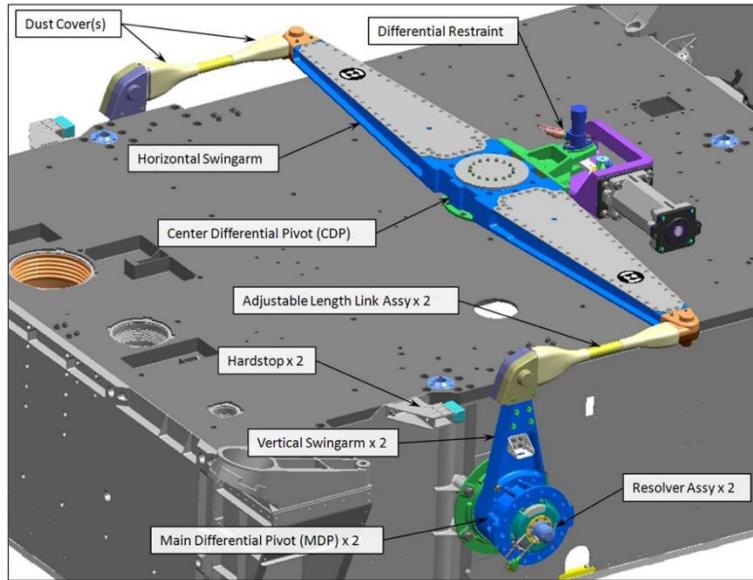
Figure 2.2: Differential gearbox assembly in the Mars Exploration Rover (MER), Source-NASA, JPL[7]

- Differential Bar

The differential bar uses linkages to connect to the rockers-bogie configuration on both sides through a pivot perpendicular to the drive shaft. The linkages drive the pivotal shaft, converting the translational motion to rotational motion. Figure 2.3 shows the differential bar and the links that connect to the rocker pivot. The differential system consists of a horizontal swing arm, linkages, and a vertical swing arm. Two ball socket joints on each side form the linkages. The pivot resides at the center of the chassis. The differential bar approach uses passive pivots and requires no electrical components. The restraint was designed to ensure the differential bar does not rotate during the MER rover's launch, entry, descent and landing phases. It is also a passive mechanism that uses a set number of Belleville springs. These springs are compressed during the differential movement and ensure they do not exceed a

specified limit. During landing, the restraint mechanism is disengaged using a pyro-pin plug, and the differential operates as usual [9].

The differential bar allows the rocker to pivot up to plus or minus 60° from the normal. It has significantly more secure joints between the linkages than the differential gearbox. The lack of gears removed the rotating axis's backlash and misalignment [8].



*Figure 2.3: The components of the differential bar on the Mars Science Laboratory (MSL) rover, Source-NASA, JPL [9].*

### 2.2.3 Centre of Gravity (CoG)

The CoG is an imaginary point at which the entire weight of the rigid body is thought to be concentrated. It affects how well a vehicle handles and operates. A lower CoG generally means better stability and handling. In vehicles with an asymmetrical design, the CoG is not in the exact centre and may be located a few centimetres below the top chassis. In a constant gravitational field, the Centre of Mass (CoM) and Centre of Gravity lie at the same point. Therefore, placed on the earth's surface, the CoG and CoM are the same [10].

### 2.2.4 Static Stability Factor (SSF)

The Static Stability Factor (SSF) given by equation 1, is an essential metric used to assess the rollover risk of a rover. It is calculated as the ratio of the rover's track width (TW) to twice the height of its centre of gravity from the ground (h). The track width refers to the lateral distance between the centres of the two front wheels of the rover. A rover with a lower SSF is generally more prone to rolling over. Conversely, a higher SSF is considered more stable and less likely to be top-heavy. The SSF is also used to determine the tilt angle at which a rover might tip over. This is achieved by calculating the arctangent of the SSF ( $\tan^{-1}(\text{SSF})$ ). Evaluation of the SSF for rovers is similar to that of a four-wheeled vehicle [11].

$$\text{SSF} = \frac{\text{TW}}{2h} \quad (1)$$

To ensure accuracy, the track width measurement is conducted meticulously by taking eight individual measurements—four at the front and four at the rear wheels, all with the steering set to a neutral angle of 0 degrees. This precision is crucial to avoid discrepancies affecting

the SSF calculation. The New Car Assessment Program (NCAP) documented SSF values ranging from a minimum of 0.92 to a maximum of 1.59. Until 2004, NCAP calculated the rollover angle based on theoretical SSF values. However, a dynamic rollover test was later introduced to better understand a vehicle's rollover threshold. In this test, a vehicle is driven on a flat surface and subjected to a sharp steering manoeuvre on an inclined slope. This process is repeated across various slope angles to identify the precise inclination at which the vehicle tips over. This dynamic approach offers a more realistic assessment of the vehicle's stability and propensity to roll over under actual driving conditions [12].

### 2.2.5 Lateral and longitudinal stability

The rover is laterally stable when both sides of the suspension are symmetric or quasi-static, and the rover does not tip over. A common way of finding the lateral stability of a rover is from the stability factor. Through the SSF, the minimum slope angle can be found before the rover tips over. The longitudinal stability of the rover depends on the condition that the normal force at the wheels is always greater than zero. This makes sure that the rover is more steerable. It also does not necessarily mean that the rover will tip over if one of the wheels loses contact with the ground [12].

### 2.2.6 Traction

A well-designed rover wheel must strike a balance between ensuring adequate traction and conserving power. High traction is essential to prevent slippage during navigation, especially on inclined planes. However, if traction is excessively high, it can lead to increased energy consumption by the motors, which, in turn, places additional strain on the motor driver components. Conversely, insufficient traction can result in the rover slipping on slopes. The incidence of slipping is governed by the relationship between the traction force at the wheels and the product of the coefficient of friction and the normal force at the wheel-soil interface. Thus, the engineering challenge lies in developing a wheel that possesses the optimal level of traction suitable for the variety of terrains the rover will encounter. The surface texture, material properties, and the wheel design all significantly influence the level of traction a rover wheel can maintain [7].

## 2.3 Steering Mechanisms

Steering helps to control the rover through rugged terrains. It aids mobility and flexibility by making sharper turns and increases energy efficiency by optimising the rover's path. There are several types of steering mechanisms.

### 2.3.1 Skid steering

Skid steering requires fewer parts and can have agility for point turning or line following using only components that drive straight. The wheels are allowed to rotate only along one axis. Therefore, a centralised drive is enough to power all the wheels [13]. To turn right, the left set of wheels rotates in the forward direction while the right side turns in the backward direction, or the left set of wheels rotates faster than the right set of wheels. This is illustrated in Figure 2.5. The speeds can be varied to control how far the turn is done. Though skid steering is easy to achieve, it comes with disadvantages. The amount of power consumed during steering can be unpredictable. It fails to achieve aggressive steering, and skid steering

on slopes is unpredictable. Since the lateral forces at the wheels are significantly high, they must be well-supported and structurally more robust [14].

### 2.3.2 Explicit steering

Explicit steering allows the wheels to be individually turned to any angle and move along two axes. It consumes lower power than skid steering but requires more actuators and components. If a centralised drive is used, then the transfer of torque becomes problematic as it has to pass through the universal joints and then the drive shaft. This results in more losses, reduces the efficiency, and increases the complexity of the drive train. This can be avoided by using individual motors to power the wheels [13]. By pointing the outer wheels at an acute angle and the inner wheels at an obtuse angle, the rover can be steered towards the right and the opposite configuration to move left. The lateral forces observed in explicit steering are less when compared to skid steering. Therefore, the structural support to the wheels need not be as robust as the ones using skid steering [14].

### 2.3.3 Crab steering

Crab steering allows the vehicle to manoeuvre in confined spaces. This type of steering is popular in four and six-wheeled rovers where the front and rear wheels are aligned at the same angle from the longitudinal axis towards the left or right. In the case of a six-wheeled rover, all the wheels have the same steering angle, as shown in Figure 2.4 (a). Therefore, the direction of the heading is the same but in a sideways diagonal translation [15].

### 2.3.4 Ackerman steering or Side pivot steering

Ackerman steering is the most popular steering mechanism used in vehicles where the front wheels are turned about a side pivot on each side. An axle connects the two pivots, helping to turn the wheels simultaneously. A set of sub-axels is attached to the steering wheel and the main axle to aid in turning, as shown in Figure 2.4 (b). The inner front wheels must turn at a larger angle than the front outer wheel to avoid skidding. If an imaginary line is drawn at the angles of the front wheels and the drive shaft of the rear wheel, they meet at a point called the centre of the turning circle. The wheels take a curved path along various radii from the centre of turning. The rear wheels are the actuators that propel the vehicle forward and have no form of steering [13].

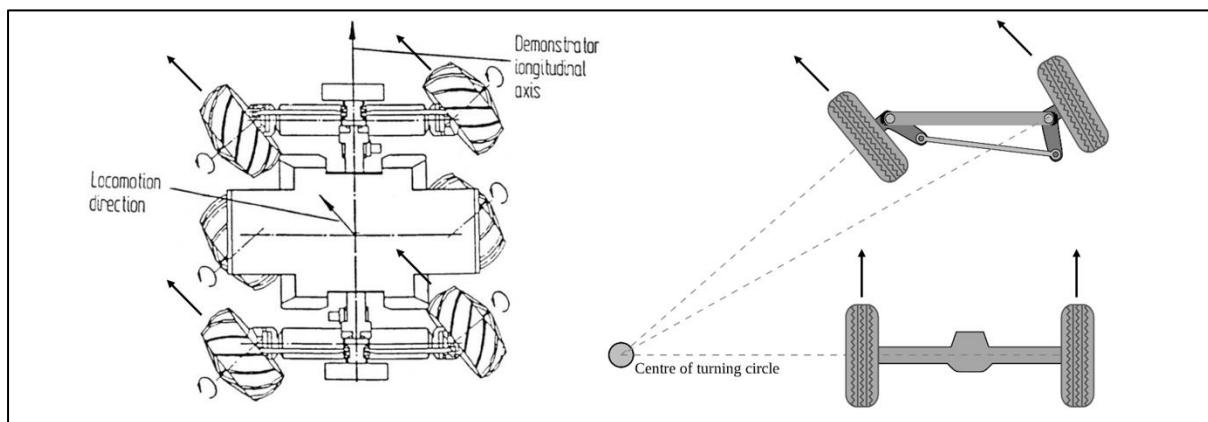


Figure 2.4: (a) Crab steering showing the six-wheeled rover locomotion direction with respect to the heading Source -[15],  
 (b) Ackerman steering mechanism showing the steering pivots and the main axel, Source-Andy Dingley.

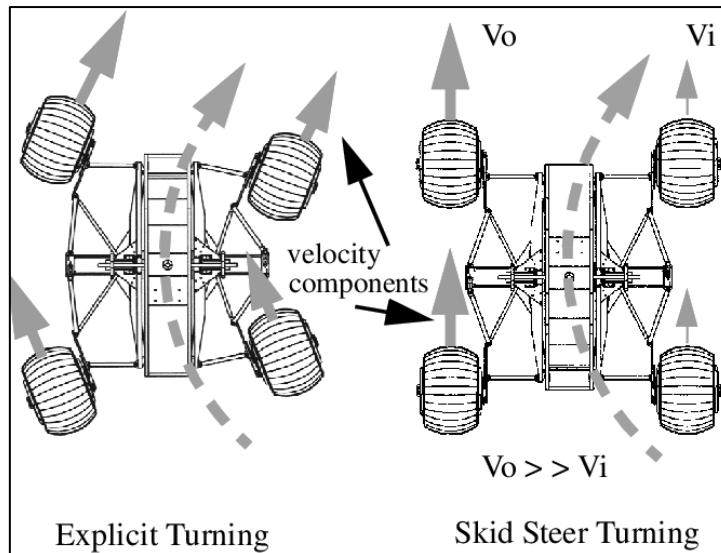


Figure 2.5: Example of explicit and skid steer towards the right, Source-[14]

## 2.4 Analysis of the Bogie-Runt Rover

The Bogie Runt rover uses the rocker-bogie suspension system with an overall dimension of 271mm x 251.50mm x 150 mm (L x W x H) made from Acrylonitrile Butadiene Styrene (ABS). It has a similar rocker-bogie configuration as NASA's Sojourner, a pair on each side connected via a differential bar that pivots only horizontally but lacks explicit steering capabilities. The Bogie Runt uses skid steering to manoeuvre around obstacles. This steering approach is not very accurate, and it cannot steer around tight corners. It adjusts the direction and speed of the wheels on either side rather than having explicitly steerable wheels. Section 2.3.1 goes into a detailed explanation of this steering mechanism.

The CoG of the Bogie runt lies at -34.23mm from point E and -6.99mm in the X-axis direction from point E in the Figure 2.7 [16]. Examining the physical dimensions of the Bogie-Runt that was designed using SolidWorks, the ground clearance is about 146mm. The wheelbase measures at a mere 207.36mm and has a trackwidth of 220mm. It consists of six press-fit wheels measuring a diameter of 65mm, driven by six DC-gear motors operating at a voltage range of 4.5 to 6V. Each of these motors has a stall torque of 0.08kgm consuming 0.55A of current at 6V. The wheels have an outer rubber layer that provides grip. As described in section 2.2.1 and using Figure 2.7, the trackwidth is 207.36mm, and the wheel centres are placed such that they are at an equal distance from each other within the trackwidth. From the isosceles triangle ABC, we know that AC is 103.68mm. Using Pythagoras' theorem,  $AC^2 = AB^2 + BC^2$ , where  $AB = BC = X$  (rocker link lengths). Evaluating for X, we get 73.31mm. Similarly, the bogie link lengths are calculated using triangle CEF. We know the trackwidth to be 207.36mm, using Pythagoras theorem  $EF = EC = 146.62\text{mm}$ . Since this length is too high for the upper platform, leading to a poor SSF, it is lowered along the same size to 98.93mm on both sides.

However, this project uses an explicit steering mechanism where each corner wheel can be individually steered. For this to happen, the entire rocker-bogie will be redesigned with radial bearings to support the rocker, bogie, and differential bar pivots. Instead of having flat panel

rocker-bogies, the new design will imitate NASA's Mars Perseverance Rover with tubular links for better structural stability. The differential bar mechanism will be used similarly to the one on the Bogie-Runt but will be updated to improve performance. The wheels will be revised to a larger diameter to increase the rover's capability to move over larger obstacles than the Bogie-Runt. The same DC-gearred motors will be used as they provide sufficient torque to move the rover. These design changes will lead to a better Center of Gravity, Static Stability Factor and stability than the Bogie-Runt.

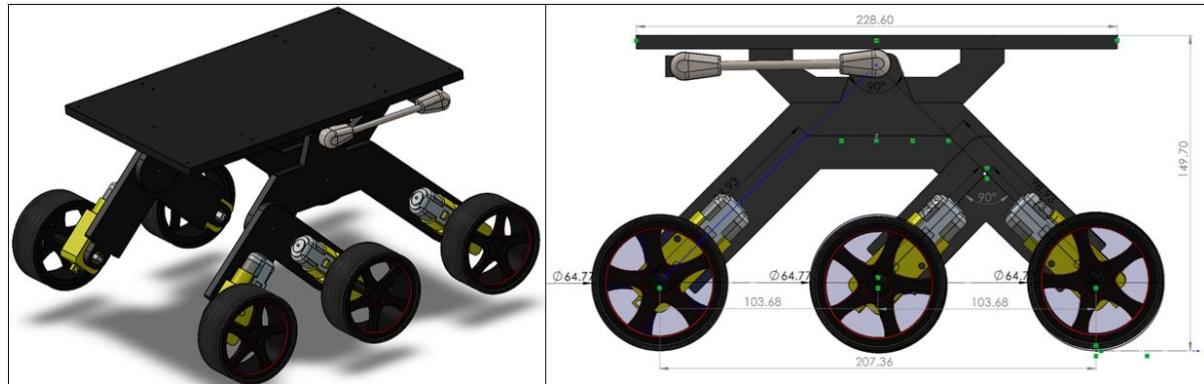


Figure 2.6: Bogie-Runt designed in SolidWorks with physical dimensions in mm.

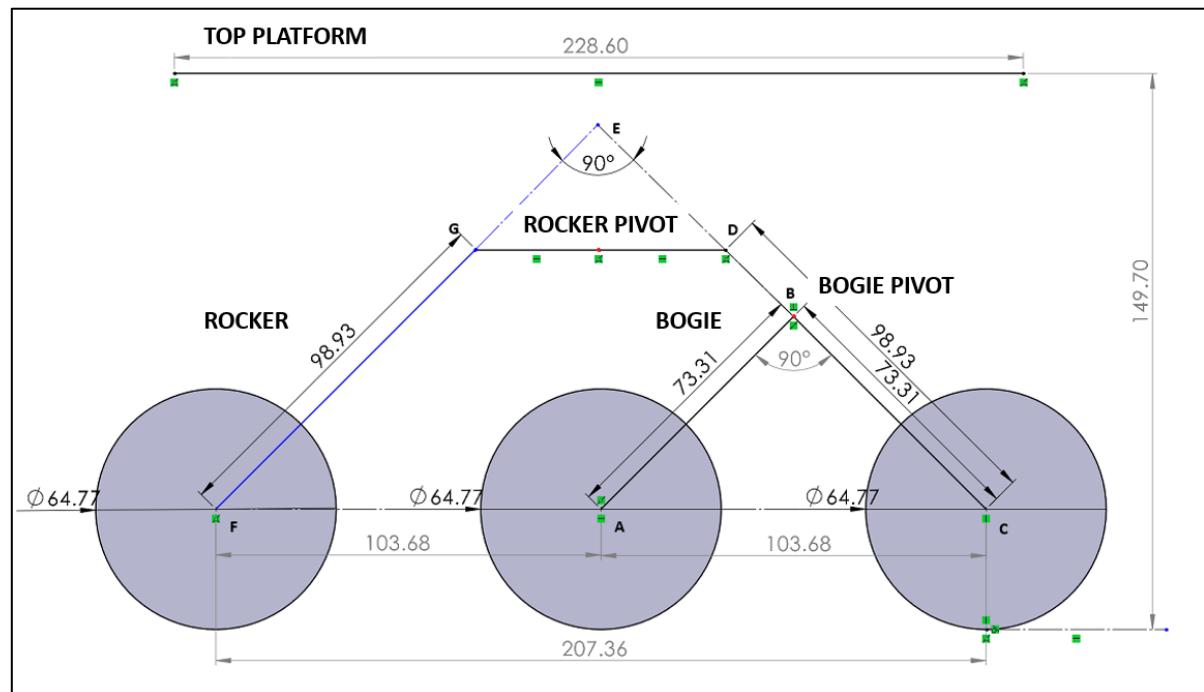


Figure 2.7: Rocker-Bogie link lengths of the Bogie-Runt designed in SolidWorks (Units-mm).

### 3 Systematic design of the directional steering rover

The method used to traverse the various slopes, terrains, and obstacles plays an essential role in the rover's operation. As discussed previously, multiple considerations must be taken to design the rover's chassis, suspension, and differential design. The wheel design is directly sourced from NASA [17], used on the perseverance rover just resized to a diameter of 70mm in the CAD software. They provide sufficient grip and are designed to traverse on sandy and rugged terrain. Further sections of this report will deal with the mechanical design considerations and development of the mobility system for the rover.

#### 3.1 Design of the rocker-bogie link lengths

Let us assume the wheelbase to be 450mm. The same geared 200RPM DC motor used in the Boge-Runt is employed as it is sufficient for the new design. Let the maximum speed required be about  $0.5\text{ms}^{-1}$ . Using the formula in equation 2, we can find the diameter (D) of the wheel required to achieve the speed in meters per second, where W is the angular speed in revolutions per minute (RPM). The diameter is calculated to be 73.45mm; let us consider it to be 70mm. The wheels are placed at equal distances from each other.

$$v = \frac{2\pi DW}{120} \quad (2)$$

One of the objectives of the new design is to increase the SSF. This can be achieved by reducing the ground clearance to 100mm and increasing the trackwidth to 430mm. Since a steering mechanism is required, a change in the design methodology mentioned in section 2.2.1 is required. The angle at the rocker and bogie pivots is  $150^\circ$  rather than  $90^\circ$ , allowing the wheelbase to be 450mm and keeping the ground clearance to 100mm. This helps to reduce the COG height and increase the SSF. The rocker pivot is placed at the height centre of the electrical box and horizontally right behind the middle wheel, which allows for a balanced weight distribution once all the electrical components are placed inside. The links are now drawn to the front and back wheels to a height just above the wheel's surface. This accommodates the steering mechanism at the end of the links. The CAD drawing is done based on the above constraints, resulting in 263.72mm and 201.86mm for the rocker link lengths. Similarly, the bogie pivot is placed halfway along the rocker link, resulting in 131.27mm and 97mm in link lengths.

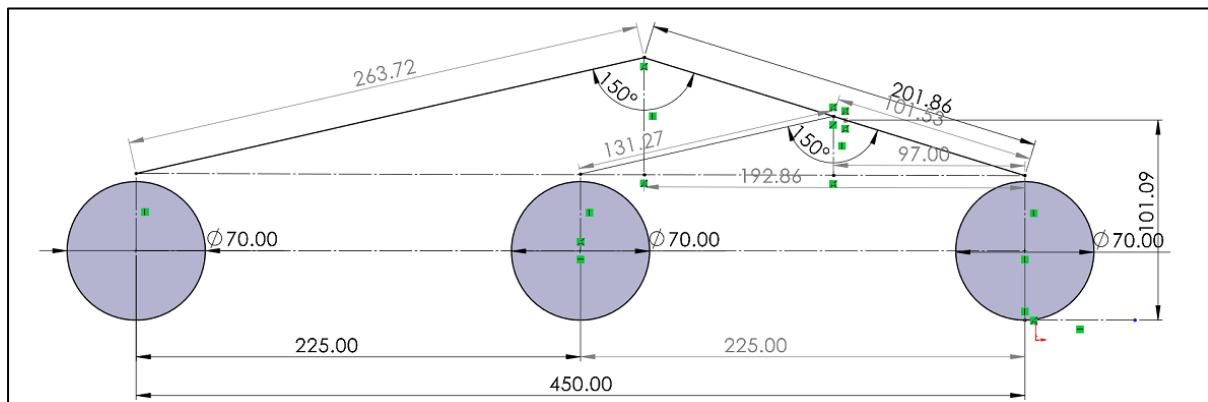


Figure 3.1: Rocker-Bogie link lengths of the explicit steering rover designed in SolidWorks (Units-mm).

### 3.2 Design of the rocker-bogie pivots

The rocker pivot bears a significant load of the suspension. A radial ball bearing is used to distribute this load at the pivot. The rocker pivot also holds the ball socket joint that connects to the differential through a threaded rod. The shaft that enters the bearing at the pivot has a diameter of 9.50mm and a length of 12mm that snugly fits into the radial bearing. The entire pivot's outer diameter is 26mm and 'boss-extruded' to a width of 30mm. A vertical joint is also extruded with a thickness of 5mm to a distance of 35.27mm to allow the connection of the ball-socket joint for the differential bar. A plane at an angle of  $15^0$  with joints of diameter 25mm is made towards the front along the  $150^0$  line designed in Figure 3.1 so that the link rod can join at the front wheel side, as seen in Figure 3.2. Another plane at an angle  $0^0$  is made towards the back for the link that joints the bogie pivot. These drawings are then 'boss-extruded' with a hole and notch for the link shafts. All the sharp external edges are smoothed using the 'fillet' tool. All the dimensions of the rocker pivot are mentioned in the appendix Figure 7.1.

The bogie pivot connects to the rear two wheels and is designed similarly to the rocker pivot. The shaft diameter is 9.50mm at 37mm and sits inside a radial bearing. The overall diameter of the pivot is 26mm 'boss-extruded' at  $0^0$  planes along the  $150^0$  line designed in Figure 3.2 with joints of 25mm diameter towards the front and back, as seen in Figure 3.3. It again has a hole and notch for the link shafts of the same diameter of 15mm. The sharp external edges are smoothed using the 'fillet' tool. All the dimensions of the bogie pivot are mentioned in the appendix Figure 7.3. Their pivots are 'mirrored' for the other side of the suspension.



Figure 3.2: Front  $15^0$  offset plane with 25mm diameter joint for the front link rod, side and back  $0^0$  plane with 25mm diameter joint for rear link rod to the bogie pivot, views of the rocker pivot designed in SolidWorks.

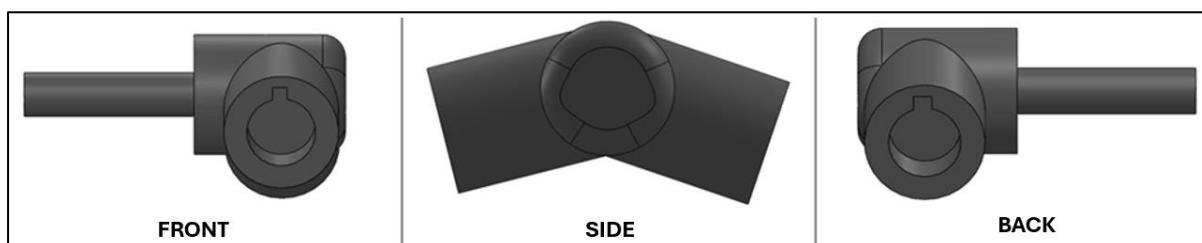


Figure 3.3: The bogie pivot's front  $0^0$  plane with 25mm diameter joint for the mid link rod, side, and back  $0^0$  plane with 25mm diameter joint for the rear link rod, views designed in SolidWorks.

### 3.3 Design of the bogie pivot to the rocker pivot bracket

This bracket holds the bogie pivot with the ball bearing and links it to the back of the rocker pivot. The diameter of the bracket is slightly larger than the bogie pivot's overall diameter as it sits inside to rotate freely. The inner diameter that holds the shaft is 11mm, and the outer diameter that holds the bogie pivot is 27mm, with an overall width of 45mm. The side view in Figure 3.4 shows the extension joint that houses the link connecting to the rocker pivot at the  $150^{\circ}$  link line. It has the exact dimensions of the rocker pivot where the link connects to it. All the bracket dimensions are mentioned in the appendix Figure 7.2.

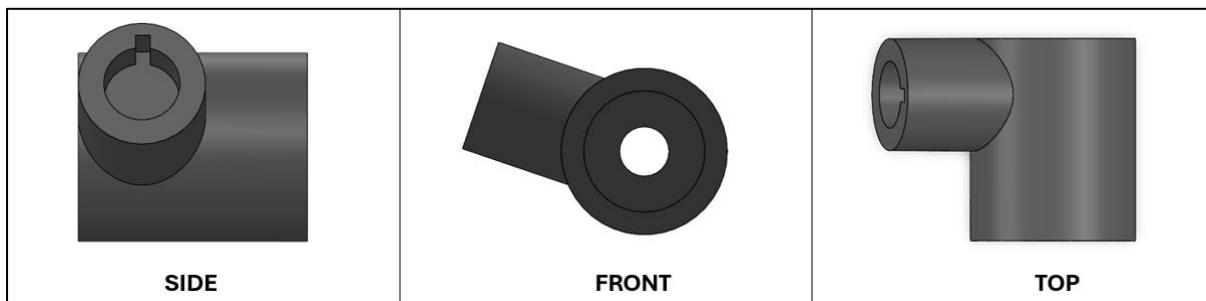


Figure 3.4: The bogie pivot to rocker pivot bracket, side, front and top views designed in SolidWorks.

### 3.4 Design of the steering mechanism

The explicit steering mechanism consists of a servo motor that rotates the steering knuckle. The steering knuckle transfers the servo's movements to the motor mount that drives the wheels. This configuration is repeated to the four corner wheels of the rover for explicit steering. As shown in Figure 3.5, a circle with a diameter of 40mm is 'boss-extruded' to a height of 10mm. The servo connector is placed at the circle's centre and then subtracted from the top side using 'subtract bodies' in the 'combine' tool. The bottom side is then boss extruded to the shape as shown in bottom view of Figure 3.5 by 5mm. Holes are then 'extruded-cut' where necessary for screws. All the dimensions of the steering knuckle are mentioned in the appendix Figure 7.8.

Similarly, the motor mount is drawn from a 40mm circle 'boss-extruded' to a height of 15mm. As before, the steering knuckle bottom is placed flush into the motor mount, and using 'subtract bodies' in the 'combine' tool, the 5mm counter sink is made. The motor mount is parallel to the knuckle tangent on the outer surface towards the axis of the force motor. Holes are then 'extruded-cut' where necessary for screws, and a notch is made for the nut where it joins the drive motor screws. Finally, the steering knuckle and the motor mount are joined, forming the steering mechanism, as shown in Figure 3.7. All the dimensions of the motor mount are mentioned in the appendix Figure 7.6.

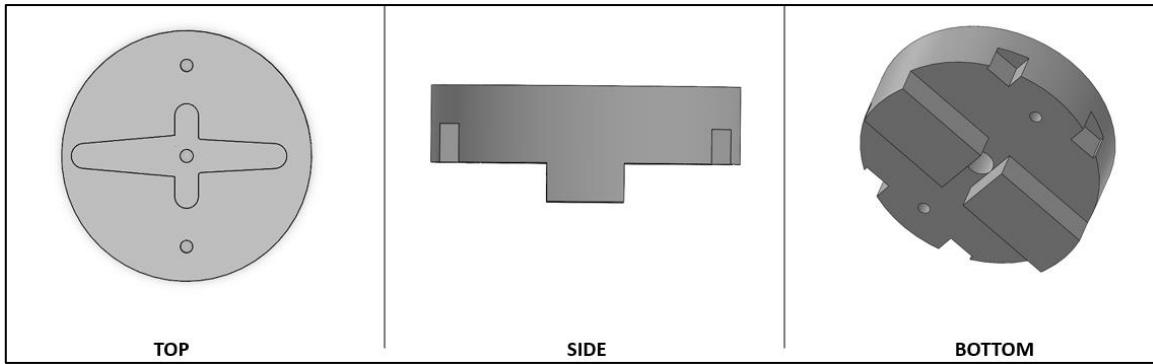


Figure 3.5: Top 40mm overall diameter, side 10mm thickness, and bottom views of the Steering Knuckle that transfers the movement from the servo to the motor mount, designed in SolidWorks.

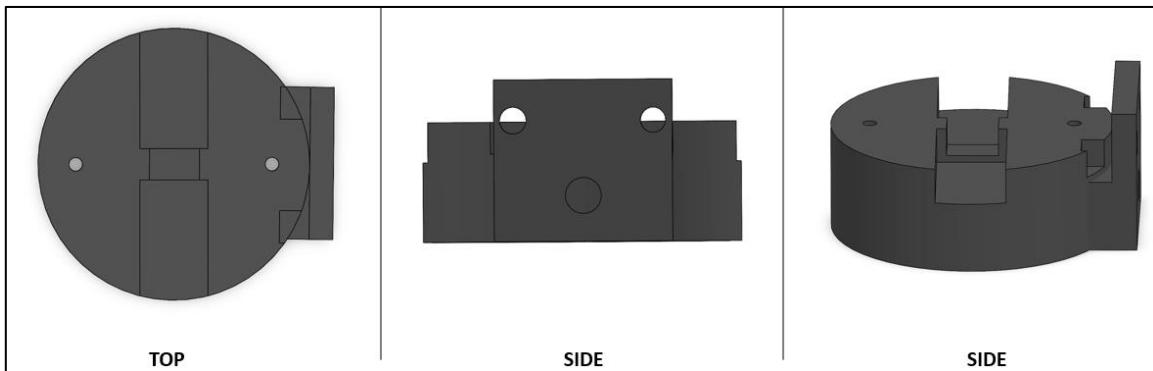


Figure 3.6: Top 40mm overall diameter and side views of the motor mount, designed in SolidWorks, that receives the transferred movement from the steering knuckle.

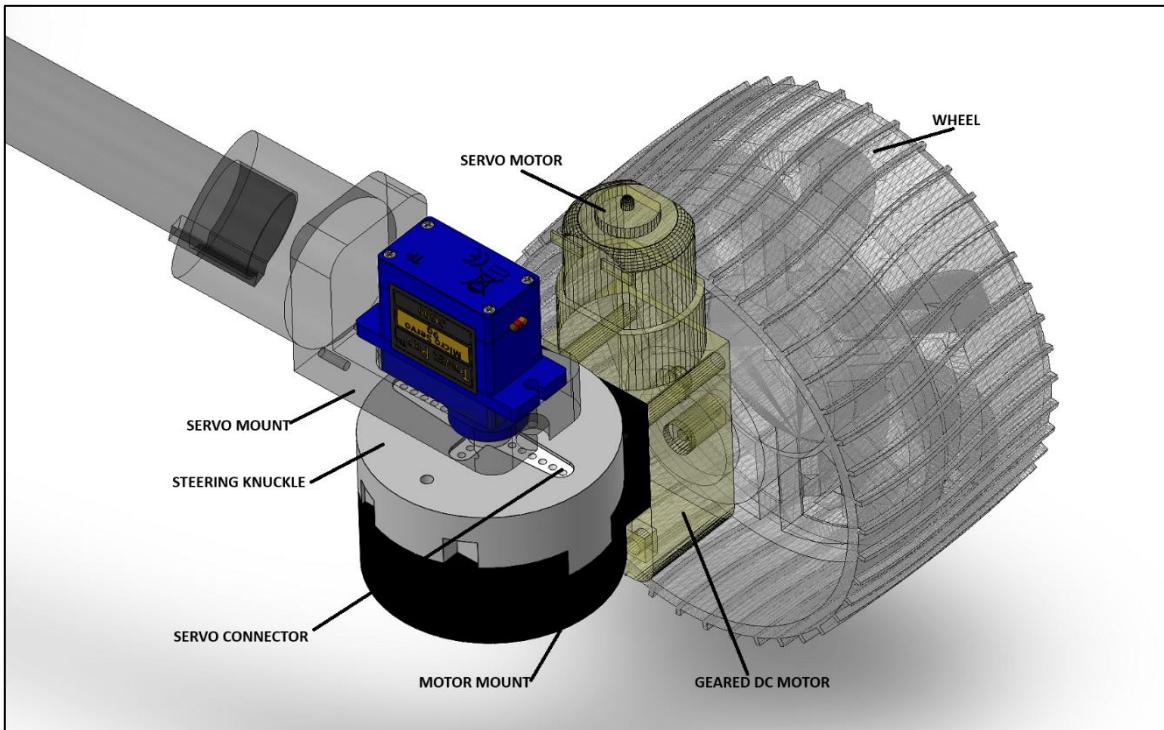


Figure 3.7: The explicit steering mechanism of the rover, designed in SolidWorks.

### 3.5 Design of the servo mount

The servo mount is a simple L-shaped bracket that connects to the rocker or bogie links at the ends of the four corners of the rover suspension that hold the servo motors. The connection to the rocker or bogie links is made at the appropriate angle with the same 15mm diameter holes with the notch. The servo motor (MG-90D) is counter-sunk into the bottom of the bracket using the ‘subtract bodies’ in the ‘combine’ tool to fit into the mount, avoiding any movement while driving snugly. 1.5mm diameter holes are made on either side for the screws. All the dimensions of the servo mount are mentioned in the appendix Figure 7.7.

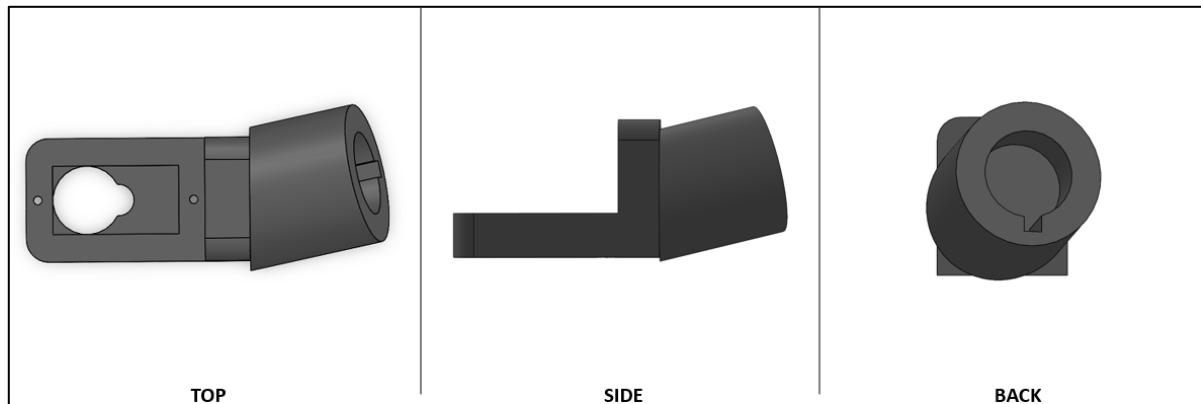
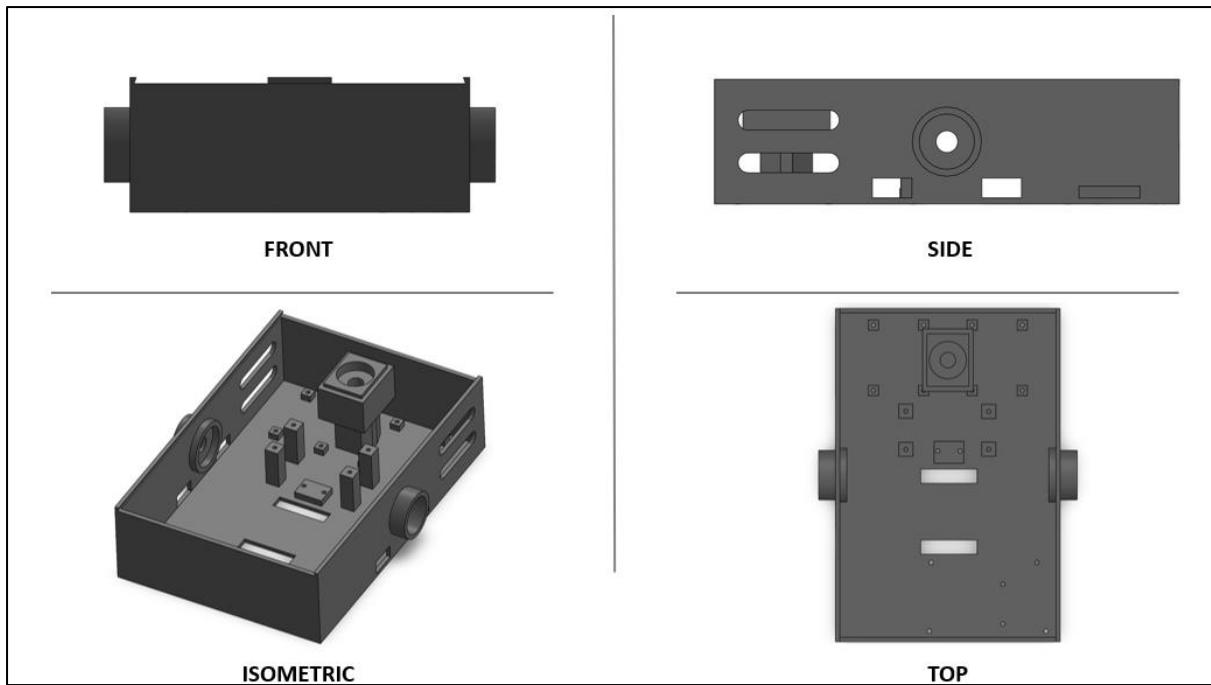


Figure 3.8: Top, side and back view of the servo mount that holds the servo motor and bogie link, designed in SolidWorks.

### 3.6 Design of the electrical box

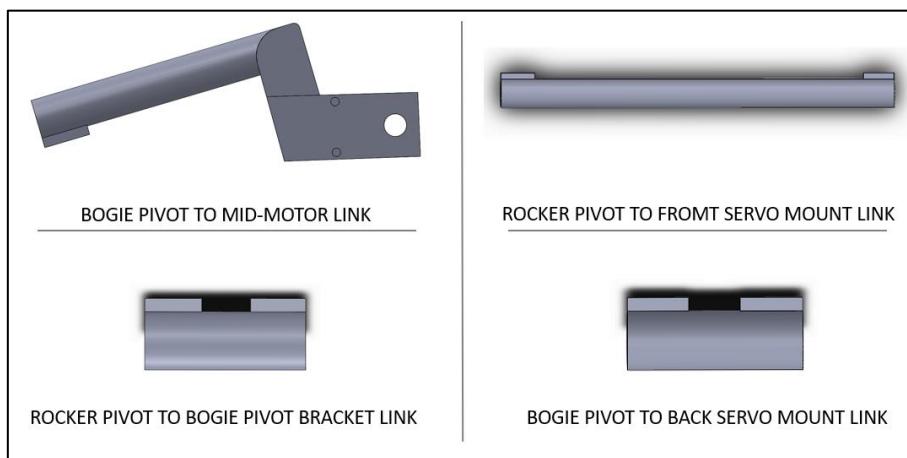
The electrical box contains the electrical components that power the rover. It is 235mm X 159mm X 63mm X 3mm (L x W x H x T). It holds the 20A and a 5A DC-DC buck converter, three motor drivers, a LiPo battery, an Arduino Due and an MPU-6050 Inertial Measurement Unit (IMU). Standoffs 10mm X 10mm X 5mm are ‘boss-extruded’ with 3mm diameter holes for mounting the motor drivers. Standoffs for the IMU (21.11 X 15.40mm X 5mm) and the DC-DC buck converters (10mm X 10mm X 30mm) with screw holes of 3mm diameter are also extruded. The differential bar shaft of 11mm diameter is housed through the tower in the electrical box at the back. The left and right walls contain the rocker pivot connections, which also house the radial bearings with an overall diameter of 35mm, an 11mm diameter hole for the rocker shaft, and a 7mm thick wall from the outer wall to hold the rocker pivot securely. Rectangular holes are made on either side to route the wires from the wheel and servo motors. A dovetail lid is designed to cover the top of the electrical box by making a slit at an angle of 70° from the outer walls on the front and back faces. Figure 3.9 shows the electrical box with all the extrusions. All the electrical box dimensions are mentioned in the Appendix Figure A.4.



*Figure 3.9: Front view with dovetail 70° slit, side view with the holes for wires to enter with the bearing and rocker pivot housing, isometric view and top view with the tower at the back and standoffs for electrical components to power the rover.*  
*Designed in SolidWorks.*

### 3.7 Rocker and bogie links

Once the above components are assembled and the relative positions are known, the links are drawn from inside the joint holes with a diameter of 7mm with the extruded notch so that they do not rotate and stay in position. This helps the servo mount parallel with the ground where the wheels and the steering mechanism join. There are four separate links: one from the rocker pivot to the front servo connector, the second is from the rocker pivot to the bogie pivot bracket, the third is from the bogie pivot to the servo mount on the back, and the fourth is from the bogie pivot to the middle wheel which also holds the drive motor. These links are just mirrored for the other side of the suspension. Figure 3.10 shows all the links used for the suspension. All the physical dimensions of the links are mentioned in the appendix Figure 7.10 to 7.14



*Figure 3.10: Various Links used in the design of the rocker-bogie rover.*

### 3.8 Design of the differential bar

As described in section 2.2.2, the differential bar is used for its advantages and simplicity. The differential bar dimensions are decided once the entire suspension is assembled in SolidWorks with the electrical box. The locus of the point at which the differential bar rotates freely makes a circle. This diameter of this circle measures 236.12mm. A 5mm diameter hole is made for the 5mm ball joint at either end on the locus 180° apart. A 5mm threaded rod connects this ball joint to another one on the rocker pivots. The shaft that enters the tower in the electrical box is 47mm long. Figure 3.10 shows the differential bar. A screw and washer on the bottom, holds it in place but allows it to pivot. The ends of the bar are rounded to avoid sharp corners. All the dimensions of the differential bar are mentioned in the appendix Figure 7.5.

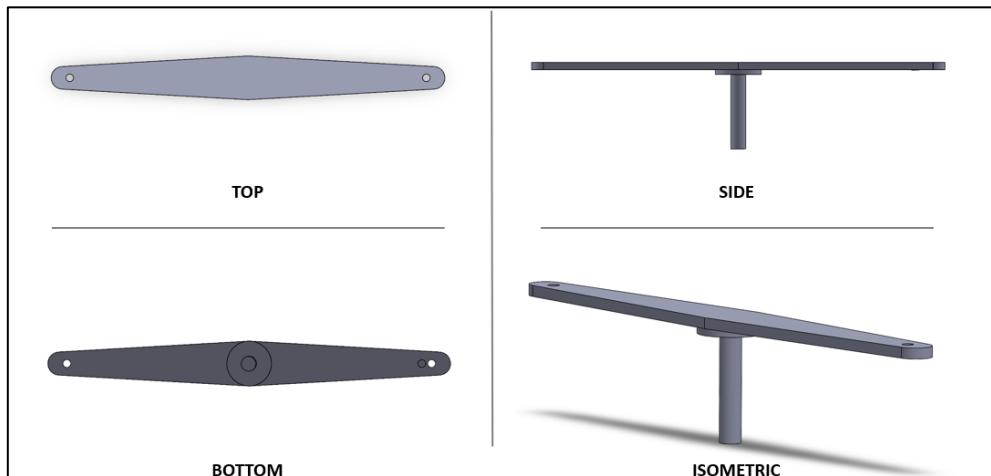


Figure 3.11: Top, side, bottom, and isometric views of the differential bar, designed in SolidWorks.

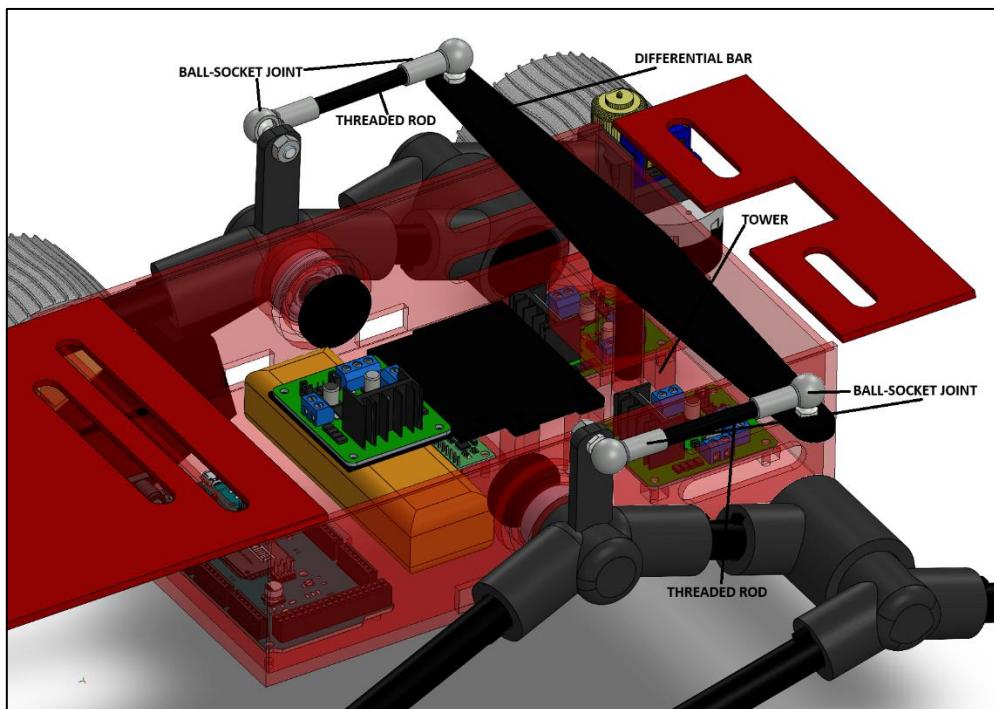


Figure 3.12: The differential bar mechanism is connected to the ball joints, threaded rod, and rocker pivot. The shaft in the tower of the electrical box. Designed in SolidWorks.

### 3.8 Steering configuration

The rover has four individually steerable wheels at the outer corners. This allows the rover to achieve a smaller curve radius. The curve radius is the distance from the rover's centre to the centre of rotation (COR). The outer wheels have a larger curve radius compared to the inner wheels. Since the inner and outer wheels are on the same curve radius, respectively, only two-wheel speeds  $v_1$  and  $v_2$  need to be calculated. Using Figure 3.11 ,the angle  $a_1, a_2, a_3$  and  $a_4$  of the wheels can be calculated based on the required curve radius using the following equations:

$$a_1 = \tan^{-1} \left( \frac{0.5 \text{ wheelbase}}{\text{curve radius} + 0.5 \text{ track width}} \right) \text{ From } \Delta AOB \quad (3)$$

$$a_2 = \tan^{-1} \left( \frac{0.5 \text{ wheelbase}}{\text{curve radius} - 0.5 \text{ track width}} \right) \text{ From } \Delta COD \quad (4)$$

$$a_4 = a_1 \text{ and } a_3 = a_2 \quad (5)$$

The speeds of the inner and outer wheels can be calculated for a set curve radius and required speed  $v_r$  using the following equations:

$$v_1 = v_r \cdot \left( \frac{\sqrt{(0.5 \text{ wheelbase})^2 + (0.5 \text{ trackwidth} - \text{curve radius})^2}}{\text{curve radius}} \right) \quad (6)$$

$$v_2 = v_r \cdot \left( \frac{\sqrt{(0.5 \text{ wheelbase})^2 + (0.5 \text{ trackwidth} + \text{curve radius})^2}}{\text{curve radius}} \right) \quad (7)$$

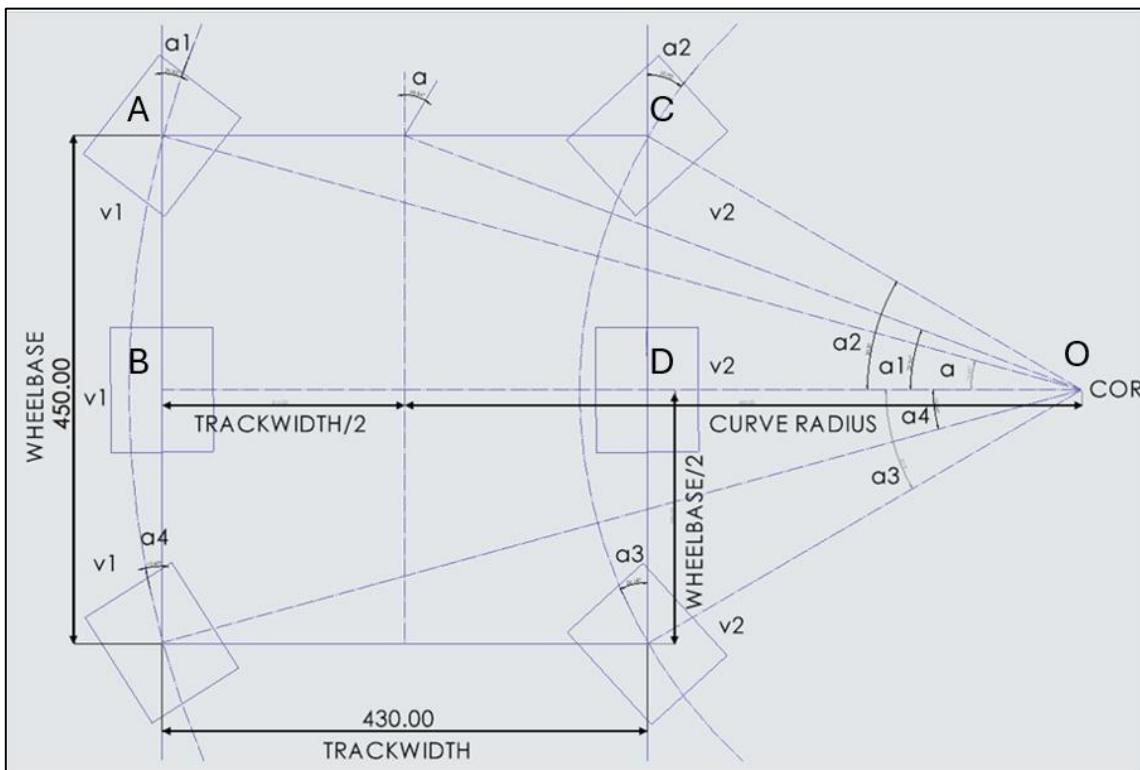


Figure 3.13: Steering angle and speed of the four steerable corner wheels. The inner wheels have angles  $a_2, a_3$  and speed  $v_2$ . The outer wheels have angles  $a_1, a_4$  and speed  $v_1$ . COR represents the center of rotation.

The minimum and maximum heading angles can be calculated using equation 3 and 4 and the code mentioned in appendix B.1. The minimum is 0° (heading straight), and the maximum is 22°. The minimum curve radius of 556.89mm can be calculated using the following equation derived from Figure 3.11 at the maximum heading angle,  $a = 22^\circ$ :

$$\text{curve radius} = \frac{\text{wheelbase}}{2 \cdot \tan(a)} \quad (8)$$

The angle required for the servo motors at the four corner wheels to perform on point rotation is calculated using the following equation derived from Figure 3.11:

$$\text{On point turning angle} = \text{atan}\left(\frac{\text{wheelbase}}{\text{track width}}\right) \quad (9)$$

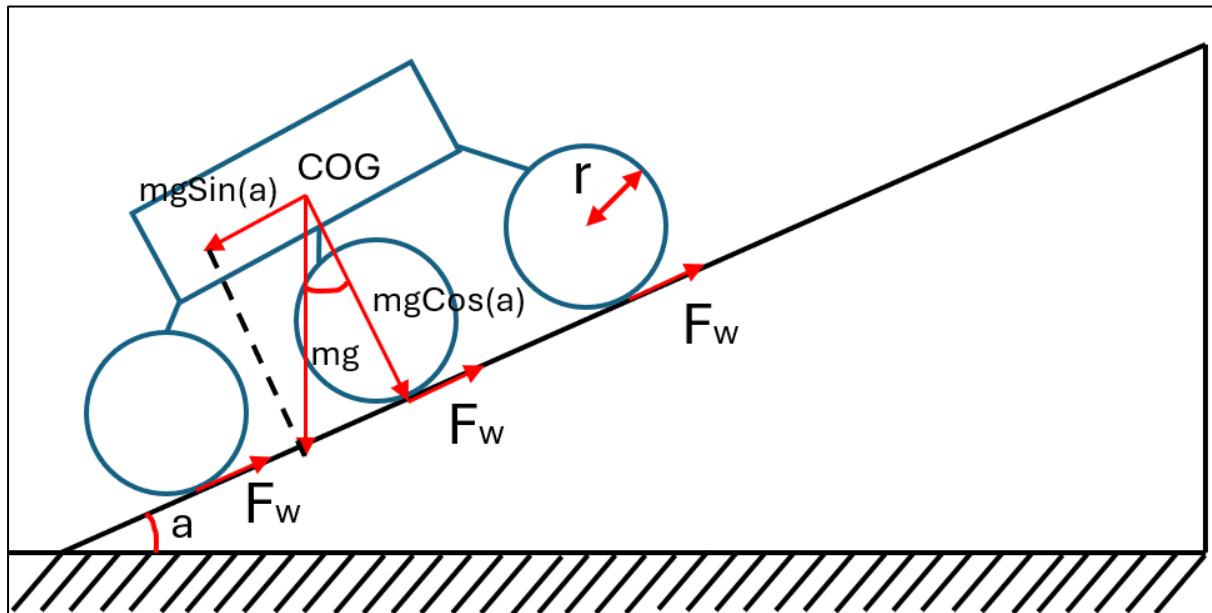


Figure 3.14: Diagram used to calculate the stall angle when the rover is climbing a ramp at an angle 'a'.

The stall angle is the angle at which the rover fails to climb further on an inclined slope. It can be derived using Figure 3.12 where  $r$  is the radius of the wheel,  $a$  is the angle of inclination or stall angle,  $m$  is the mass of the rover,  $g$  is the earth's gravity, COG is the Centre Of Gravity height from the floor,  $F_w$  is the force applied at the wheels by the drive motor ( $F_w = \text{Motor Torque}/r$ ) and the six represents the six drive motors :

$$a = \arcsin\left(\frac{6 \cdot \text{Motor Torque}}{m \cdot g \cdot r}\right) \quad (10)$$

### 3.9 Final assembly and preliminary testing

The designed parts in SolidWorks is 3D printed using Acrylonitrile Butadiene Styrene (ABS) on its excellent material properties like rigidity, temperature resistance and resistance to abrasion. The wheelbase and trackwidth changed from 450mm and 430mm to 440mm and 460mm, respectively, as shown in Figure 3.15. The increase in the trackwidth is due to the rocker pivots sitting slightly away from the initial position as the bearings are glued in using epoxy in the electrical box. The reduction in the wheelbase is due to the rocker and bogie links being glued in a fraction further into the pivots using super glue. Once all the parts were assembled and placed on the ground, the wheels had a slight camber due to the weight and increased trackwidth. This was countered by designing a bracket to hold the rocker pivots in place perfectly parallel to each other, as shown in Figure 3.16. The gap in the centre allows the rocker pivot differential connection to move when engaged while keeping it straight easily. The exact dimension of the bracket is mentioned in appendix Figure 7.9. The electrical components were mounted and screwed in place for further connection. The differential bar is connected to the rocker pivots using a threaded 5mm rod and ball-socket joints on both sides. The length of the threaded rod is 6.5mm. All the screw threads were coated with Loctite so that they don't come loose due to vibrations or motion when the rover is being operated.

#### Preliminary testing results:

The rover showed good structural stability and rigidity. The bearings epoxied in the electrical box, and the bogie pivot mount for the differential bar and rocker and bogie pivots held the load of the rover when lifted away from the ground comfortably without bending. All the joints showed good motion along their degrees of freedom. The height of the COG was measured at 105mm from the ground. The trackwidth and wheelbase are 460mm and 440mm, respectively. The servo motors at the four corner wheels cannot be actuated beyond  $33^{\circ}$  from the normal as they collide with the servo motor mount. The maximum heading angle for turning the servo motors at the wheel to  $33^{\circ}$  is  $22^{\circ}$ , which is calculated using the code mentioned in appendix B.1. This does not impede the functionality of the rover to make turns and is explained in detail in the 'Calculating the servo angles' in section 5.4 later in the report. The total mass of the rover measured at 2.45kg.

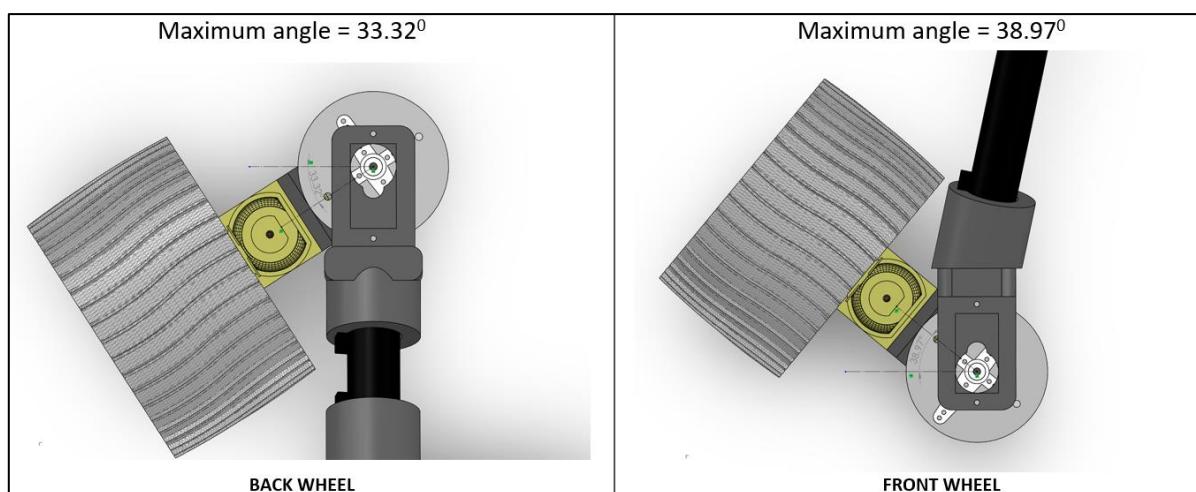


Figure 3.15: The maximum servo angle achievable at the rear and front wheels.

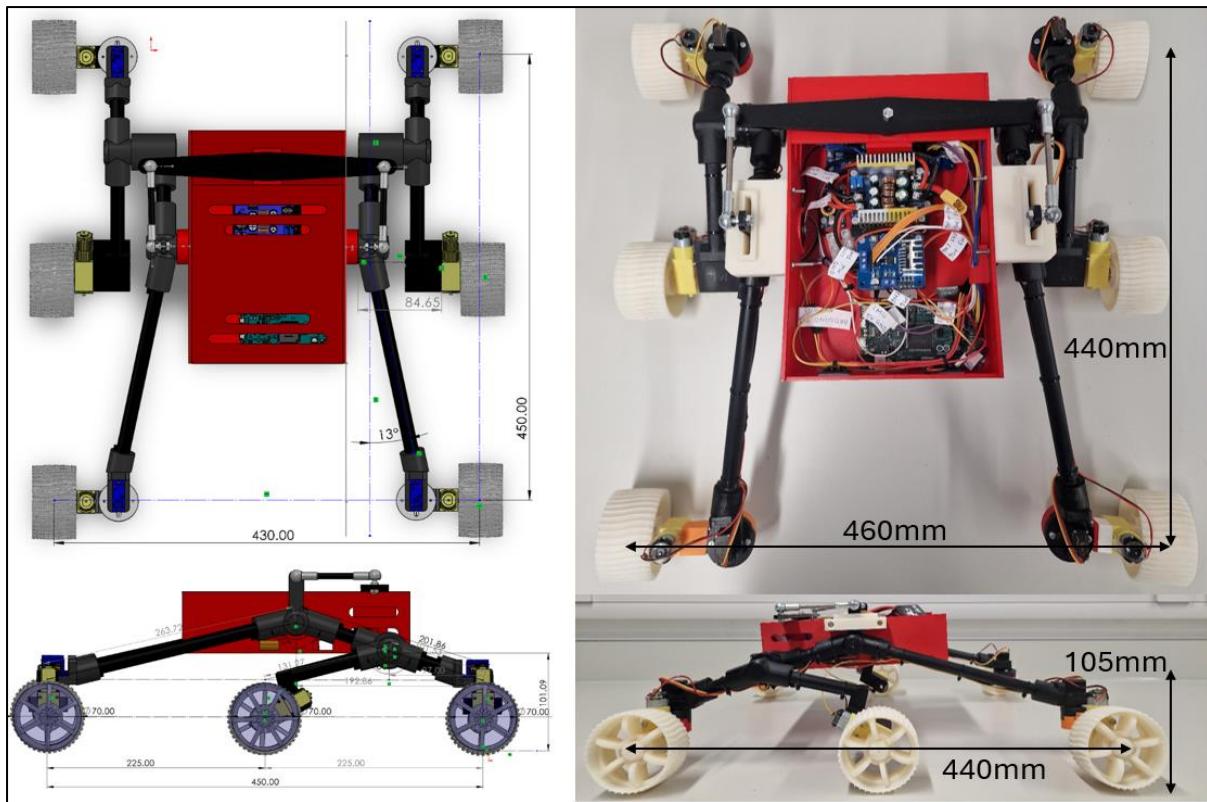


Figure 3.16: Final assembly of the rover in SolidWorks and in real life. The measurements (mm) show variations in track width and wheelbase length.

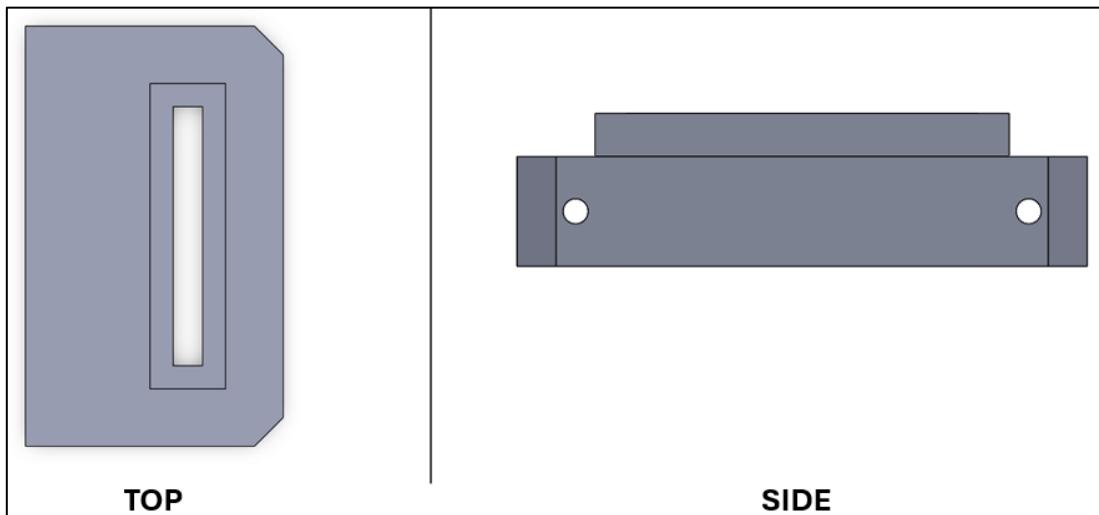


Figure 3.17: Rocker pivot support bracket to prevent camber of the wheels when placed on the ground.

## 4 Electrical and sensor design

The new rover's electrical design is modified from the existing Bogie Runt rover. The existing Bogie-Runt has six drive motors, two motor drivers, and an Arduino because it runs off a LiPo (Lithium Polymer) battery. The new steering design requires the addition of servo motors. The addition of these motors increased the complexity of the electrical design as more current was needed to keep all the actuators in working condition. The new electrical design is divided into four stages—power regulation, power distribution, actuators, telemetry transmission and the programmable microcontroller.

### 4.1 Power regulation

The power regulation circuit consists of two DC-DC step-down variable buck converters. These DC-DC bucks regulate the high voltage from the LiPo battery to a stable required output. These buck converters can withstand a wide range of input voltages with high current capabilities. Large heat sinks ensure that the Power Metal Oxide Semiconductor Field Effect Transistors (PMOSFETs) do not overheat and fail during high current loads. The significant advantage of using a variable buck converter is that the output voltage and output current can be varied or limited using the on-board potentiometer.

#### 4.1.1 Preliminary test results

The 20A DC-DC buck converter was subjected to an input voltage of 15V from a four-cell (4S) LiPo battery. Using the onboard variable potentiometer, the output voltage was limited to 7.5V at 20A. This is done as the motor drivers have a voltage drop of about 1.8V at the output. Therefore, to get an output of about 6V at the motors, the input voltage has to be 7.5V. All the drive motors and servo motors were activated at the same time to test the buck converter at maximum load. The output voltage measured at the motors was 5.7V. Similarly, the 5A DC-DC buck converter was subjected to an input voltage of 15V from the same LiPo battery, and the output was set to 6.5V. The four servos were powered under full load, and the buck converter output was stable.

| SPECIFICATION            | VARIABLE BUCK 1 | VARIABLE BUCK 2 |
|--------------------------|-----------------|-----------------|
| Input voltage range (V)  | 6 to 40         | 5 to 32         |
| Output voltage range (A) | 1.2 - 36        | 0.8 - 30        |
| Max output current (A)   | 20              | 5               |

Table 4.1: Electrical specifications of the variable and fixed DC-DC buck converters[17] [18].

### 4.2 Power distribution

Once the 7.5V and 6.5V are available at the buck converters' output, they are routed to the three motor drivers and servos, respectively. The motor driver uses the LN298N IC (Integrated Circuit) to supply a maximum of 2A continuous current per channel. The motor driver uses a dual h-bridge configuration. This configuration allows the user to control the output motor speed and direction of rotation with the help of a microcontroller using the digital input pins on the motor driver. Each motor driver is connected to three motors to distribute the output load among the six drive motors. The heat sinks to make sure that the IC does not overheat.

An additional 5V output, which powers the microcontroller board, is available on the motor driver board.

#### 4.2.1 Preliminary test results

A voltage drop of 1.8V was measured when an input of 7.5V was applied to the motor driver. This voltage was set so the output would be close to 6V after the voltage drop, as the drive motors are rated to run at maximum torque at 6V. The final voltage measured at full load at the output was 5.7V.

| SPECIFICATION                     | L298 MOTOR DRIVER |
|-----------------------------------|-------------------|
| Input voltage range $V_{in}$ (V)  | 5 to 12           |
| Output voltage (V)                | $V_{in} - 1.8$    |
| Max output current per channel(A) | 2                 |

Table 4.2: Electrical specifications of the L298N motor driver [19].

### 4.3 Actuators

This project uses two actuators: an angled DC geared motor and a servo motor.

#### 4.3.1 Geared DC motor

The gearbox motor used has a gear ratio of 1:48, operating at 6V at around 200 RPM. The gearbox allows the motor shaft to be positioned right-angled with respect to the original shaft. The maximum stall torque available at the shaft is 0.8Kgcm. These are the same motors used in the original Bogie-Runt. It is also a cheap and reliable option in the current design.

| SPECIFICATION       | GEARED DC MOTOR |
|---------------------|-----------------|
| Input voltage (V)   | 6               |
| No load current (A) | 180m            |
| Stall current (A)   | 0.85            |
| Torque (kg-cm)      | 0.8             |
| Speed (RPM)         | 200             |

Table 4.3: Electrical specifications of the geared DC motor [20].

#### 4.3.2 MG90D Servo motor

The original Bogie-Runt had no explicit steering capability and used skid steering to rotate. This makes it difficult to manoeuvre around tight corners. Therefore, a servo motor is employed in the new design to enable explicit steering at the four corner wheels. The servo motor can rotate 180 degrees. It operates with a maximum torque of 6.6V. A PWM (Pulse Width Modulation) signal can control the servo motor's position using the microcontroller board and the digital pin on the servo motor.

| SPECIFICATION           | MG-90D SERVO MOTOR |
|-------------------------|--------------------|
| Input voltage range (V) | 4.8 to 6.6         |
| No load current (A)     | 6m                 |
| Stall current (A)       | 700m               |
| Speed (s/60°)           | 0.08 at 6.6V       |
| Stall torque (kg-cm)    | 2.4                |

Table 4.4: Electrical specifications of the SG-90 servo motor [21].

### 4.3.3 Preliminary test results

The geared DC drive motors were connected to the motor driver and tested at total capacity with the test code written in the Arduino IDE. The motors were rotated in clockwise and anticlockwise directions at different speeds. The servo motors were connected to the variable 5A DC-DC buck converter and tested at full load sweeping from  $0^{\circ}$  to  $180^{\circ}$ , the test code is available in the appendix B.3.

## 4.4 Arduino due and telemetry

### 4.4.1 Arduino Due

The microcontroller board used in the project is the Arduino Due, which helps the user communicate with the various sensors and actuators connected to it and control them. The Arduino Due is based on the 32-bit ARM architecture. It consists of two DACs (digital to analogue converters), twelve analogue inputs, and fifty-four digital input/output pins; the two CAN (Controller Area Network) buses can be used for large-scale projects. At the heart, the microcontroller used is the Atmel SAM3X8E ARM cortex-M3. It operates at 84MHz with 96kb of SRAM (Static Random Access Memory) [22]. This allows the operation of tasks written in code to be performed much faster than the previous generation Arduino boards. The board is powered using one of the 5V outputs of the motor driver. The four servo motor data cables and the digital inputs to the motor drivers are connected to the Arduino Due's digital I/O pins. The Arduino Due can be programmed using the micro-USB (Universal Serial Bus) port. The Arduino IDE is used to program the board as required.

| SPECIFICATION           | ARDUINO DUE |
|-------------------------|-------------|
| Input voltage range (V) | 7 - 12      |
| I/O voltage             | 3.3         |
| Group 1 pin current (A) | 9m          |
| Group 2 pin current (A) | 3m          |
| Max I/O current (A)     | 130m        |

Table 4.5: Electrical specifications of the Arduino Due [22].

### 4.4.2 Bluetooth module

The HC-06 Bluetooth module is a versatile, easy-to-use device that facilitates wireless communication with Arduino microcontrollers. This module operates solely as a slave device, which means it can accept connections from other Bluetooth-enabled devices but cannot initiate them. Its primary mode of communication with the Arduino is through the Universal Asynchronous Receiver/Transmitter (UART) interface, a crucial standard in serial communication. UART allows asynchronous serial communication, transmitting and receiving data on separate pins, labelled RX (Receiver) and TX (Transmitter). The HC-06 module seamlessly integrates with the Arduino environment by connecting its RX and TX pins to the corresponding TX and RX pins on the Arduino, facilitating a two-way serial communication link. It operates at 5V and can communicate with the master device at a maximum distance of 10m with a direct line of sight [23].

#### 4.4.3 MPU-6050 IMU

The IMU (Inertial Measurement Unit) is an additional data point for robot navigation. Navigation is a complex process; therefore, IMU data is used for path planning and mapping. The MPU-6050 is an IMU with 6-axis motion tracking. It consists of a 3-axis accelerometer and a 3-axis gyroscope. It can also be used to report ambient temperature readings, but it is inaccurate. The MPU-6050 is based on MEMS (Micro Electromechanical System) technology operating at 5V. The SDA (Serial Data) and SCL (Serial Clock) pins are connected to the Arduino DUE's SDA and SCL ports, while the 5V and ground pins are attached to the Due's 5V and GND. The IMU uses the I2C protocol to communicate between the device and the host [24].

The positioning of the IMU is essential to avoid any errors and tedious code edits. The COG (Centre of Gravity) is the point where the entire weight of the assembly is evenly distributed in all directions. At this point, the effective forces can be easily calculated with little to no error. Therefore, the IMU must be close to the robot's COG. In this case, the COG is the same as the COM (Centre of Mass), as the gravitational field is constant ( $9.8\text{m/s}^2$ ). As a result, the IMU is placed at the COM of the rover, which is conveniently calculated and displayed by the CAD software. The 3-axis X, Y and Z corresponds to the following:

- X: It is the length forward of the IMU/COG (Positive values are towards the front, and negative values are towards the back)
- Y: It is the length right of the IMU/COG (Positive values are towards the right, and negative values are towards the left)
- Z: It is the length below the IMU/COG (Positive values are towards the top, and negative values are towards the floor)

In this design, the Z-axis height must be corrected in code as the COG is a bit lower than the point of placement of the IMU on the rover.

#### 4.4.4 Optical motor encoder

The optical encoder is used to measure the speed of the wheels in RPM. The disk that has the slits is mounted to the motor shaft in between the IR LED and the photo receiver. Once the wheel rotates, an electrical pulse is generated at the photo receiver due to continuous transmission and blocking of the window slits. This pulse takes the shape of a PWM signal. The Arduino reads the PWM signal at its digital pin, later converted to RPM in code. The encoder operates at 5V logic. It is mounted on the middle wheels on the inner side of the rover.

### 4.5 Circuit diagram

The circuit diagram consists of the components as described in the table below. The LiPo battery is rated at 1500mAh at 100C, so it can supply a maximum continuous current of 150A ( $1.5 \times 100$ ). A LiPo battery is used because it can provide large amounts of current as the estimated current draw can be up to 15A at maximum load. The variable DC-DC buck converters are set at 7.5V and 6.5V at maximum current using the on-board potentiometer. The outputs are connected to the L298N motor drivers to the 12V port. Each motor driver has six digital input pins connected to the Arduino digital output pins. Motors M1 and M2 are controlled using the digital pins D30, D32, D34, and D36, with D2 and D3 pins controlling the

PWM output for speed control. Motors M3 and M4 are controlled using the digital pins D22, D24, D26, and D28, with D4 and D5 pins controlling the PWM output for speed control. Motor M5 and M6 are controlled using the digital pins D23, D25, D27 and D29, with D6 and D7 pins controlling the PWM output for speed control. The other variable regulator set at 6V and 5A gets its input directly from the LiPo battery, and the outputs are connected to the power inputs of the servo motors. The servo motors S1, S2, S3 and S4 are controlled using the PWM pins 8, 9, 10 and 11, respectively. The IMU has its 5V input connected to the 5V output of the Arduino Due, with the data SDA and clock SCL connected to pins D20 and D21, respectively. The HC-06 Bluetooth module is powered by the 5V output and Gnd pins on the Arduino Due. The Tx and Rx pins on the BT module are connected to the Tx(16) and Rx(17) pins on the Arduino Due, respectively. The optical encoders are connected to the digital pins D38 and D40 with the 5V and GND pins shared between the Arduino Due and C1 motor driver.

| COMPONENT                      | DESCRIPTION                        |
|--------------------------------|------------------------------------|
| M1 to M6                       | Geared DC motors                   |
| S1 to S4                       | MG-90D servo motors                |
| B1 and B2                      | Variable DC-DC buck converters     |
| C1 to C3                       | L298N motor drivers                |
| E1 and E2                      | Optical motor encoders             |
| IMU                            | MPU-6050 inertial measurement unit |
| BT                             | HC-06 Bluetooth module             |
| Lithium polymer battery (LiPo) | 4S, 14.8V, 1500mAh, 100C           |

Table 4.6: Description of components used in the circuit diagram for the rover.

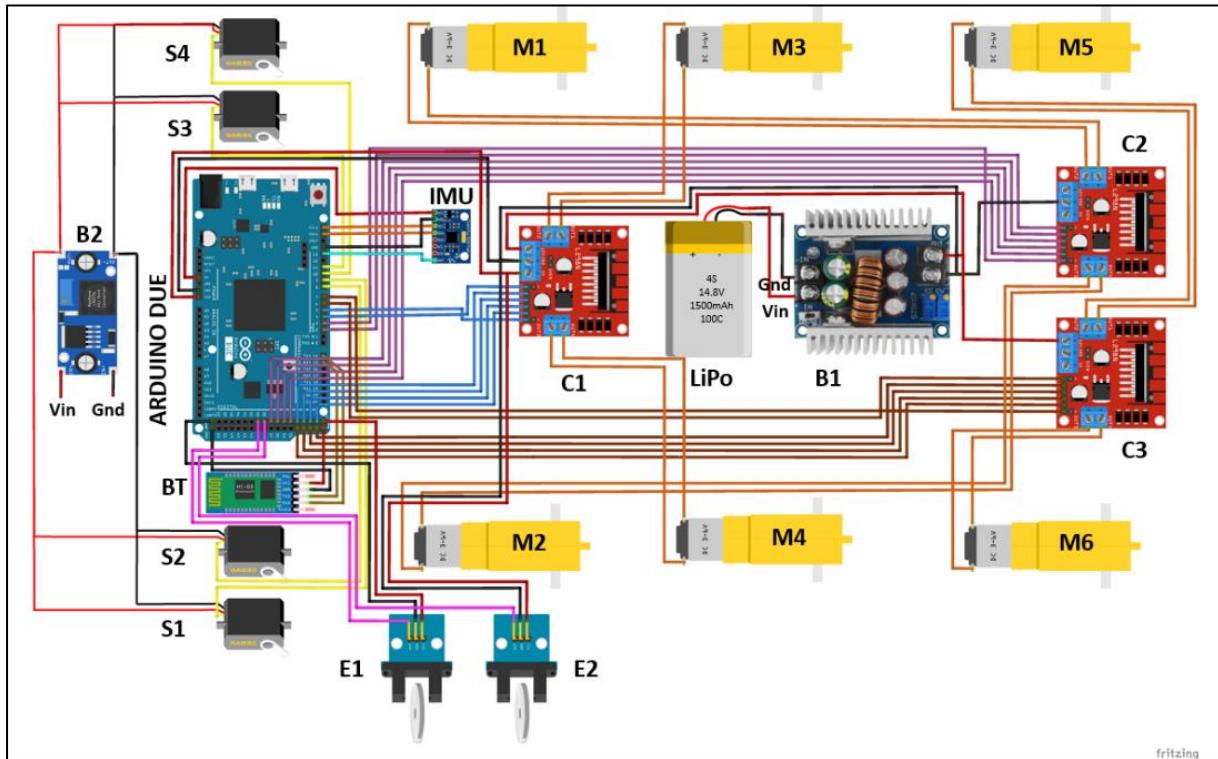


Figure 4.1: The circuit diagram for the rover, designed in Fritzing.

## 5 Software and Programming

### 5.1 Programming flowchart

The flowchart shows the control system on an Arduino Due to manage interactions between the user and rover. The initial phase contains the definition of global constants and variables, including the Bluetooth module's configuration for communication, the motor drivers for driving the rover's wheels, servo motors for steering, IMU for logging pitch, roll and yaw angles and motor encoders to find the speed of the drive motors in RPM. Ensuring these components are correctly initialised is critical for the subsequent control and feedback mechanisms to function as intended.

The software's control logic facilitates a continuous operation and feedback loop, maintaining a real-time connection between the user's commands and the rover's movements. It performs continuous calculations of the appropriate steering angles and motor speeds required to achieve the desired directional changes and velocity. Once the initialisation is completed, the control enters the input handling phase, which is pivotal for the interactive control of the rover. User inputs from a laptop keyboard through the Putty terminal, specifically from the 'A', 'D', 'W', 'S', 'R' and 'X' keys, are processed to adjust the rover's trajectory and speed. Since the Arduino IDE is unable to take continuous inputs without pressing the enter key, [Putty](#) terminal is used. It is configured at the required baud rate and specific COM port connected through the Bluetooth module, the configuration image is mentioned at the end of appendix B. The program translates these inputs into corresponding adjustments in the rover's steering and motor speed. The 'A' and 'D' keys are mapped to the heading angle 'alpha', controlling the servos to alter the turning radius as the heading angle increases or decreases, while the 'W' and 'S' keys manage the forward and backward motion. The 'R' key performs on point rotation and the 'X' key stops the rover at any point of time. The system is designed to respond swiftly to these inputs, ensuring that the rover's movements are smooth and precise, which is essential for navigating through varied terrains and obstacles the rover may encounter.

Additionally, the system logs data from the Inertial Measurement Unit (IMU) and the motor encoder to the serial monitor, which provides immediate feedback on the rover's orientation and speed of motion for the operator's awareness and serves as a critical component for potential future enhancements such as autonomous navigation. Through this feedback mechanism, the software can facilitate adjustments to the control signals sent to the motors and servos, thus ensuring precise navigation. This comprehensive approach to software design ensures that the rover's responsiveness is always maintained, reflecting a dynamic and robust system capable of complex manoeuvres and real-time adaptability to user input. The entire code for operating the rover is mentioned in the appendix B.4.

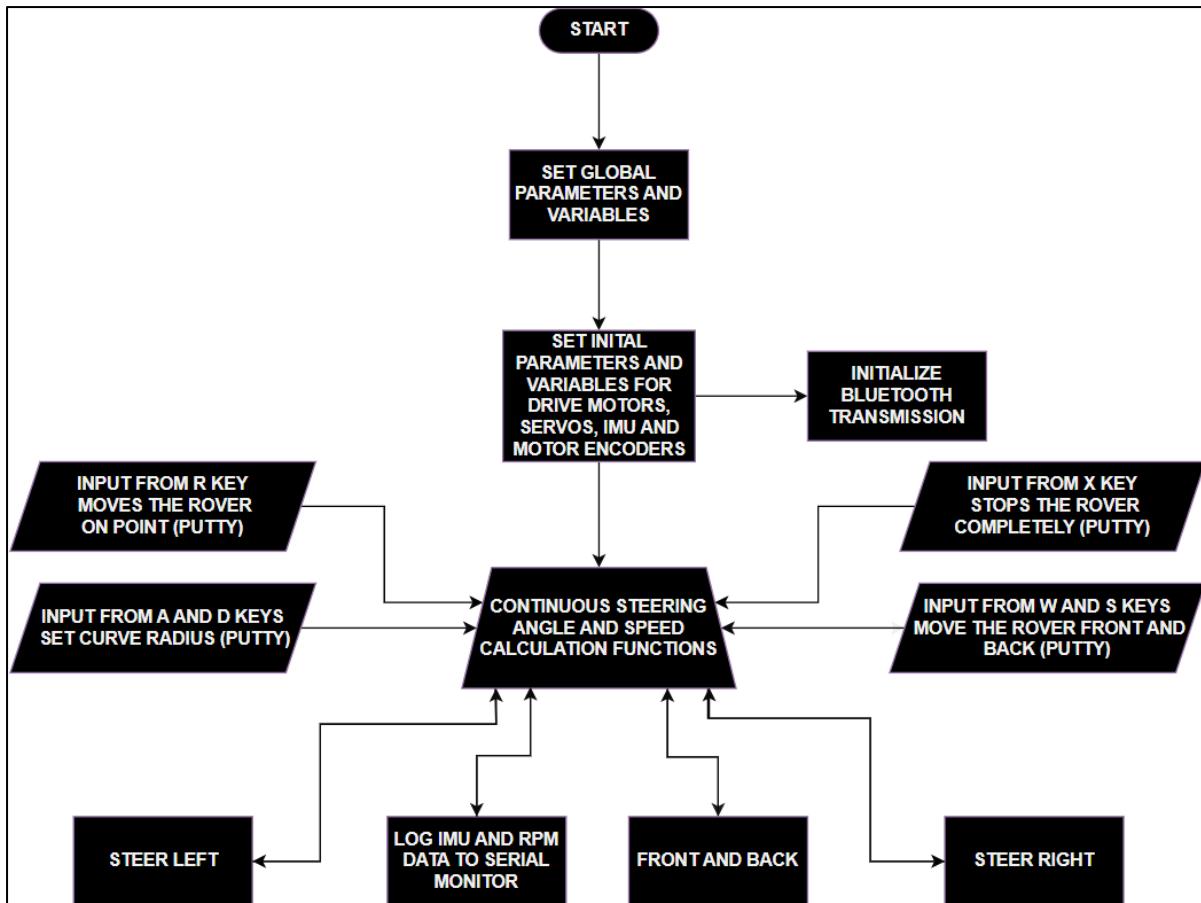


Figure 5.1: A Flow chart showing the software lifecycle of the rover, designed in Microsoft Powerpoint.

## 5.2 Input function

The void loop function houses the main functionality of the rover's speed and direction of locomotion. It receives the inputs for a PC keyboard connected to the Arduino Due through Bluetooth. A serial monitor like Putty is used to send commands. If the connection is successful, which is checked by the `BT.available()` function, the transmitted inputs are stored in the 'inp' variable using the `BT.read()` function. To make the commands received instantaneous rather than pressing the enter key every time, the `mills()` function calculates the timeout to stop the rover if no command is received within a specified time limit. The switch branching logic the rover responds to 'w' or 'W' to move forward, 's' or 'S' to move backwards, 'a' or 'A' to turn left, 'd' or 'D' to move right, 'r' or 'R' to rotate on point and 'x' or 'X' to stop all movement. While turning right, alpha is increased and limited to twenty-two degrees to avoid the geared motors hitting the links. Similarly, alpha is decreased and limited to minus twenty-two degrees while turning left, for each of the inputs the speed and angle is calculated using the `motorSpeed()` and `servoAngle()` function.

```
// Control logic for moving the rover
if (BT.available()) {
  inp = BT.read();
  Serial.print(inp);
  lastCommandTime = millis();
  switch (inp) {
    case 'w': // Move forward
    case 'W':
      motorSpeed();
      moveForward(speed1PWM, speed2PWM);
      break;
    case 's': // Move back
    case 'S':
      motorSpeed();
      moveBackward(speed1PWM, speed2PWM);
      break;
    case 'd': // Turn right
    case 'D':
      alpha += 1; // Increase alpha for right turn
      if (alpha > 22) alpha = 22; // Limit max turn angle
      moveLeft();
      break;
    case 'a': // Turn left
    case 'A':
      alpha -= 1; // Decrease alpha for left turn
      if (alpha < -22) alpha = -22; // Limit max turn angle
      moveRight();
      break;
    case 'r': // Rotate on point
    case 'R':
      motorSpeed();
      rotateOnPoint(speed1PWM, speed2PWM);
      break;
    case 'x': // Stop all movement
    case 'X':
      stopMovement();
      break;
  }
} // Check for command timeout to stop the rover if no command is received
if (millis() - lastCommandTime > commandTimeout) {
  stopMovement();
}
```

Figure 5.2: A continuous-running void loop function that receives input from a PC keyboard through Bluetooth. Written in Arduino IDE.

### 5.3 Calculating motor speed

The speed of the rovers six wheels depends on the heading angle ‘alpha’. ‘S’ represents the speed of the rover in percentage. To turn left and right it is multiplied by a factor to account for the variable speed. Using [equation 6](#), [equation 7](#), and [equation 8](#), the speeds of the wheels can be calculated. Since we know that the maximum heading angle is  $22^{\circ}$ , the maximum speed at which the rover can be turned left and right is at S=68, as any increase in the speed of the outer most wheels will lead to incorrect turning velocity. To turn right, the ‘D’ key is pressed. Holding the ‘D’ key increases alpha. Similarly, to move left, the ‘A’ key is held. While turning, the inner wheels rotate at a speed lower than the outer wheels as they need to cover a smaller distance than the outer. Once speed1 and speed2 are calculated, whose values lie

between 0 and 100 percent is, then converted to PWM signals that need to be between 0 and 255 using the `map(speed_in_percent, min_in_percent, max_in_percent, min_in_PWM, max_in_PWM)` function. It is then stored in `speed1PWM` and `speed2PWM` which is passed on to the wheels. If the heading angle is zero, then the wheels move at 50% of the maximum speed to move forward and backwards.

```
void motorSpeed() {
    s = 50;           // speed set to 50%
    if (alpha > 0.0) { // Turning right
        turning_radius = (wheelbase / (2 * tan(abs(alpha) * PI / 180)));
        speed1 = s * sqrt(pow((0.5 * wheelbase), 2) + pow((0.5 * trackwidth - turning_radius), 2)) / turning_radius;
        speed2 = s * sqrt(pow((0.5 * wheelbase), 2) + pow((0.5 * trackwidth + turning_radius), 2)) / turning_radius;

        speed1PWM = map(round(speed1), 0, 100, 0, 255);
        speed2PWM = map(round(speed2), 0, 100, 0, 255);
    }

    else if (alpha < 0.0) { // Turning left
        turning_radius = (wheelbase / (2 * tan(abs(alpha) * PI / 180)));
        speed1 = s * sqrt(pow((0.5 * wheelbase), 2) + pow((0.5 * trackwidth - turning_radius), 2)) / turning_radius;
        speed2 = s * sqrt(pow((0.5 * wheelbase), 2) + pow((0.5 * trackwidth + turning_radius), 2)) / turning_radius;

        speed1PWM = map(round(speed2), 0, 100, 0, 255);
        speed2PWM = map(round(speed1), 0, 100, 0, 255);
    }

    else { // Moving forward and back
        speed1 = speed2 = s;
        speed1PWM = map(round(s), 0, 100, 0, 255);
        speed2PWM = map(round(s), 0, 100, 0, 255);
    }
}
```

Figure 5.3: The motor speed function that calculates the speed of the wheels based on the heading angle 'alpha'. Written in Arduino IDE.

## 5.4 Calculating the servo angles

Using [equation 3 and equation 4](#), the servo angles are calculated. The `servoAngle()` function is called when the 'A' and 'D' keys are pressed, which calculates the angles required to make the turn. Initially, the heading angle 'alpha' is converted to radians to be used inside the `atan()` function, with positive angles for turning right and negative angles for turning left. Once the angles are calculated, they are rounded to the nearest one using the `round()` function and stored in 'alpha1' and 'alpha2'. Once alpha1 and alpha2 are calculated, they are passed to the `moveRight()` or `moveLeft()` function, depending on the input keys. Since the connectors at the servo motor do not connect perfectly at ninety degrees, an error angle needs to be accounted for using the `setXAngle` variable where X is from one to four for the four servos. The alpha1 and alpha2 angles correspond to the front two wheels which is added to the `setXAngle`, for the rear wheels it is just subtracted from the `setXAngle`. Using [equation 9](#), we can calculate the angles for the servos to perform on-point rotation. The `rotateOnPoint()` function sets the servos at  $43.73^{\circ}$  outwards from the normal and rotates the left three wheels clockwise and the right three wheels anticlockwise to rotate on point towards the right side.

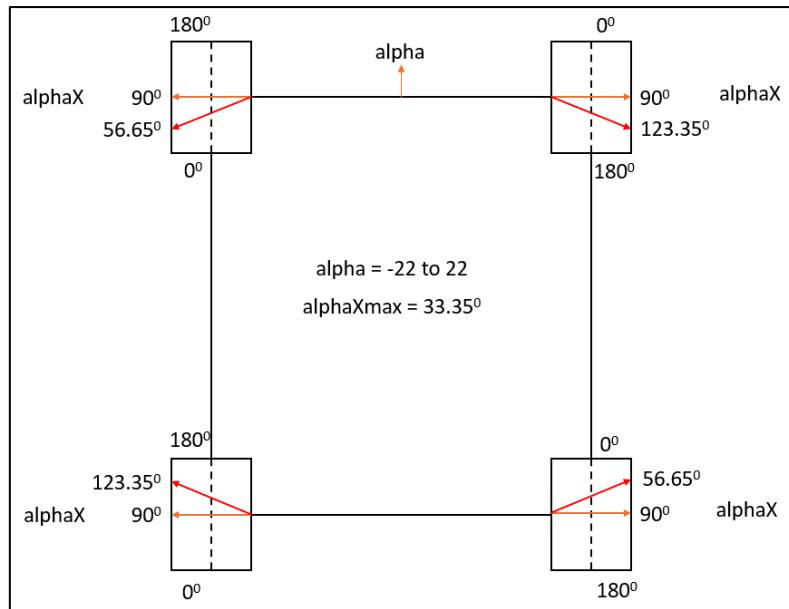


Figure 5.4: Illustration of the maximum turning angles that can be achieved by preventing the geared motors hitting the rocker and bogie links. Designed in Microsoft Power Point.

```
void servoAngle() {
    alpha_rad = alpha * PI / 180; // Convert degrees to radians
    //Calculate the servo angles in degrees
    alpha1 = round(atan(wheelbase / ((wheelbase / tan(alpha_rad)) + trackwidth)) * 180 / PI);
    alpha2 = round(atan(wheelbase / ((wheelbase / tan(alpha_rad)) - trackwidth)) * 180 / PI);
}
```

Figure 5.5: The servo angle function that calculates the angles at which the servos need to turn based on the heading angle. Written in Arduino IDE.

```
void moveRight() {
    servoAngle(); // Update alpha1 and alpha2 based on current alpha

    // Adjust servo angles for turning right
    servo1.write(se1Angle + alpha1);
    servo2.write(se2Angle + alpha2);
    servo3.write(se3Angle - alpha1);
    servo4.write(se4Angle - alpha2);
}

void moveLeft() {
    servoAngle(); // Update alpha1 and alpha2 based on current alpha

    // Adjust servo angles for turning left
    servo1.write(se1Angle + alpha1);
    servo2.write(se2Angle + alpha2);
    servo3.write(se3Angle - alpha1);
    servo4.write(se4Angle - alpha2);
}
void rotateOnPoint(float speedPWM1, float speedPWM2) {
    // Adjust servo angles for turning on point
    servo1.write(se1Angle + alpha_on_point);
    servo2.write(se2Angle - alpha_on_point);
    servo3.write(se3Angle - alpha_on_point);
    servo4.write(se4Angle + alpha_on_point);
```

Figure 5.6: Adjusting the servo angles taking into account for incorrect initial heading angles at the wheels. Written in Arduino IDE.

## 6 Experimental testing and validation

In the Experimental Testing and Validation section, a series of functional tests designed to evaluate the performance and reliability of the rover. These tests are crafted to simulate real-world conditions under controlled environments, ensuring the collection of accurate and reproducible data. Initially, baseline tests to establish the fundamental characteristics and initial performance metrics of the rover framework will be done. Following this, functionality tests will be conducted to assess the rover framework's ability to perform its intended tasks efficiently and effectively in the real world.

### 6.1 Straight line test

#### Objective:

To ensure the rover can travel in a straight line at various speeds.

#### Methodology:

- Test Track: Use a flat, even surface marked with a straight path.
- Procedure: Start the rover at the beginning of the path. Incrementally increase the motor RPM from a minimum to a maximum value in predefined steps. At each RPM setting, allow the rover to travel the length of the track.
- Measurements: Record the deviation from the straight path using a line marked on the test track or a laser line tool.
- Repetitions: Repeat the test 20 times for each RPM setting to ensure consistency.

#### Results:

The motion simulation of the assembled rover in SolidWorks showed no deviation in the path taken. The simulation achieved similar completion times compared to the rover in the real-world test at S=30, S=50 and S=68 (Final speeds in percentages of maximum RPM). The small variation in times from the simulation and the actual word test are a result of unknown surface friction at the wheels. In the real world, the rover achieved average times of 11.19s, 5.23s and 4.14s at S=30, S=50 and S=68, respectively. Since the rover is moving straight, the RPM of the wheels are all the same. The average RPMs of the wheels are 58.45, 131.90 and 185.15 at S=30, S=50 and S=68 respectively. RPM1 represents the inner wheels, while RPM2 represents the outer wheels.

The standard deviation for the time series and RPM series is mentioned in the table below. The times achieved for the three speed values have very low standard deviations, showing remarkable consistency in the data collected. The RPM values also show reasonable standard deviations, though this greatly depends on the motor encoders' resolution. This consistency of the time data can further be described using the box plot shown in Figure 6.2. For S1, the time data is tightly grouped, with a median close to the lower quartile, suggesting a skew towards faster times. The presence of an outlier indicates at least one trial took slightly longer than the rest. S2 has the lowest median time compared to S1, indicating it generally had the fastest trials. There's less spread in the data than in S1, suggesting more consistent performance. S3 also has narrow spread times, indicated by the short box, which suggests consistent performance across trials. S2 and S3 have outliers indicating at least one trial that took slightly longer than the rest. Similarly, the RPMs also showed very low deviations for the

various speeds during testing. The time series plot and scatter plot shown in Figure 6.3 show consistent data over the period of time that the test was run.

| SPEED IN PERCENTAGE | TIME (S) | RPM1 | RPM2 |
|---------------------|----------|------|------|
| S=30                | 0.098    | 0.74 | 0.74 |
| S=50                | 0.068    | 0.94 | 0.94 |
| S=68                | 0.081    | 0.48 | 0.48 |

Table 6.1: Standard deviations for the time taken to complete the course and the RPMs of the inner and outer wheels.

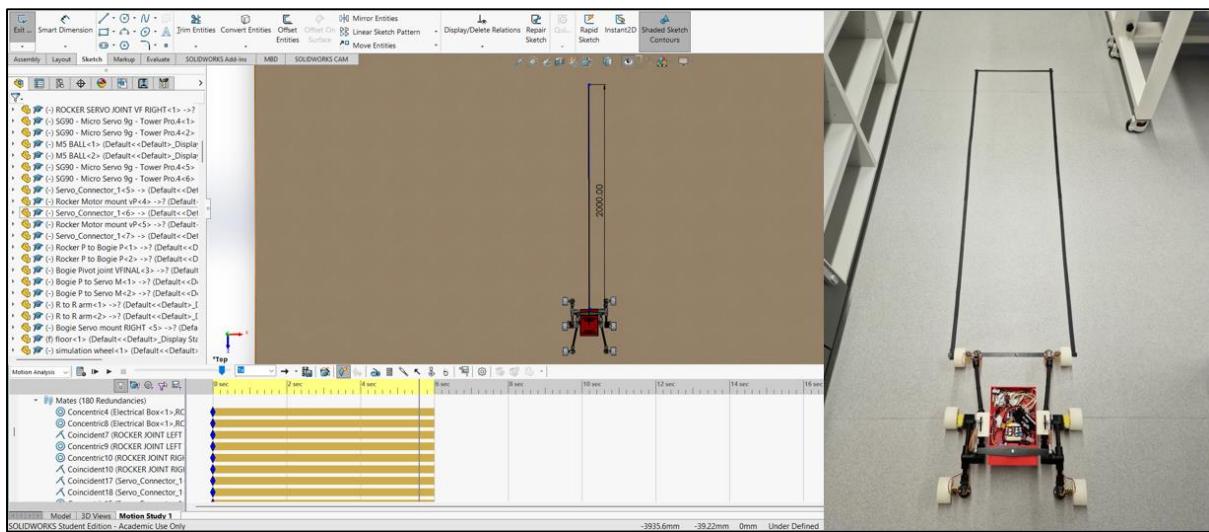


Figure 6.1: Completed motion simulation of the rover designed in SolidWorks for a straight line of 2m in length and the assembled rover set to be tested along a 2m long test track bounded by black tape.

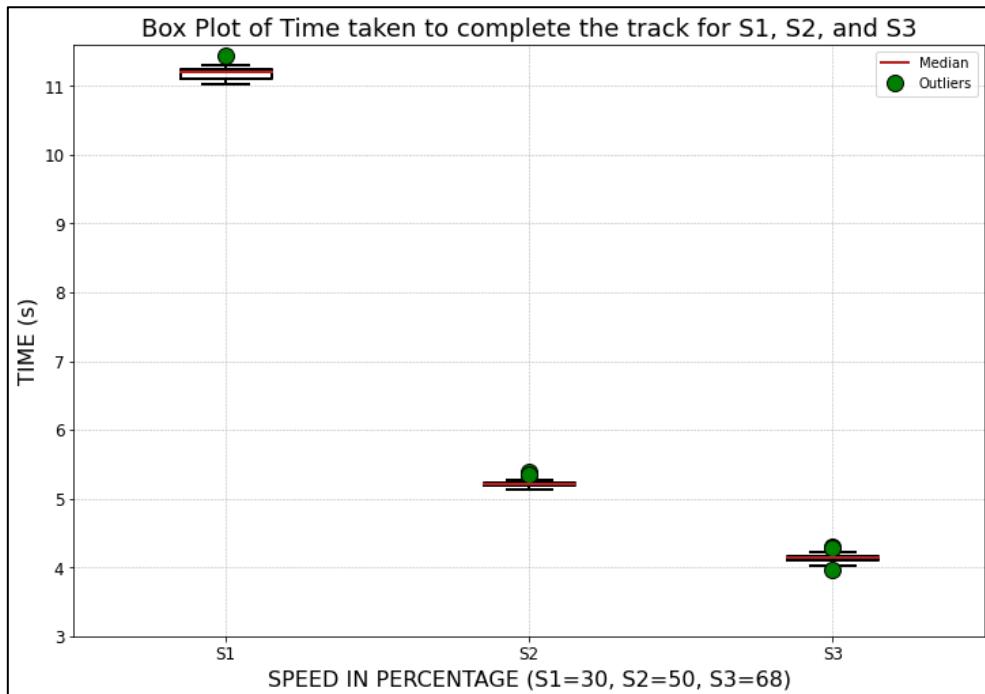


Figure 6.2: Box plot for the time taken to complete the 2m track at various speeds.

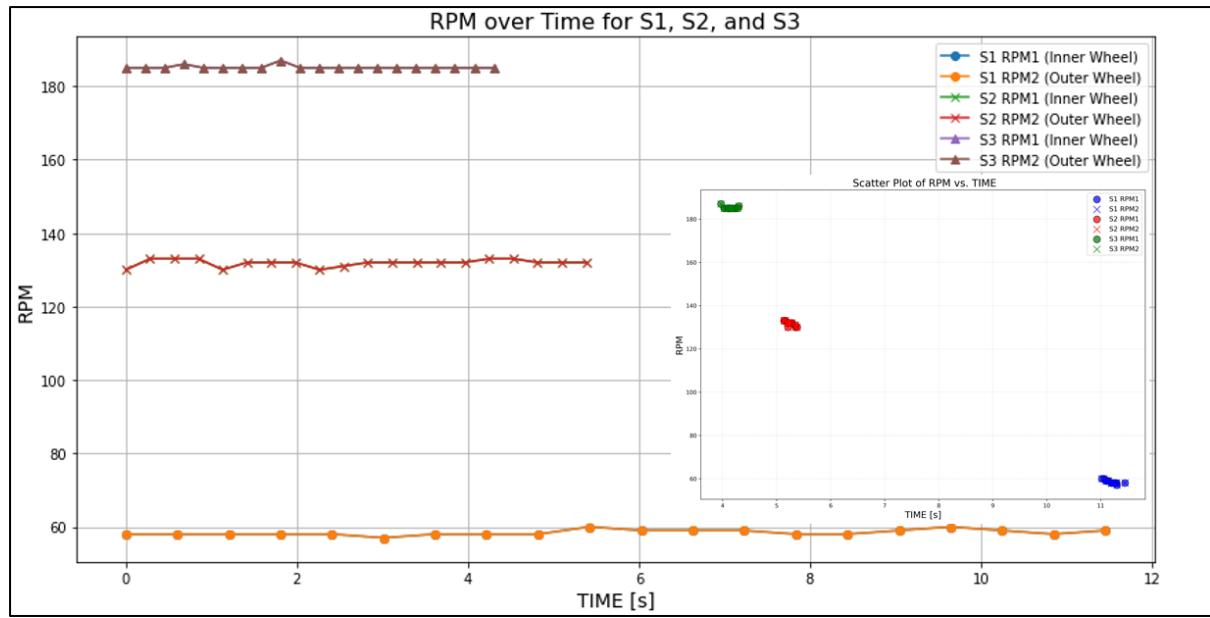


Figure 6.3: Time series and scatter plot of RPM vs TIME.

## 6.2 Minimum turning radius test

### Objective:

To find the smallest radius of the circle, the rover can turn within.

### Methodology:

- Test Area: Use a flat, open area with ample space.
- Procedure: Start with the rover at a fixed RPM setting. Gradually command the rover to turn at its tightest possible radius without reversing the wheels.
- Measurements: Mark the starting and ending points of the front and rear wheels. Measure the radius of the circle formed by these points.
- Repetitions: Conduct 20 trials for each RPM setting, noting any variations in the turning radius.

### Results:

The motion simulation of the assembled rover in SolidWorks showed a slight deviation in the path taken. The simulation achieved a minimum radius of 597.50mm (1195mm diameter), as shown in Figure 6.4. The ideal minimum radius calculated using [equation 8](#) with  $a=22^{\circ}$  is 556.90mm. This increase in the radius is due to the joints in SolidWorks being extremely rigid and due to the coefficient of friction at the wheels being low, resulting in some slipping. In the real-world test, the wheel base reduced by 10mm and trackwidth increased by 30mm, resulting in the theoretical minimum radius being 544.52mm. The rover comfortably achieved 554mm of minimum radius for all twenty trials for all speeds, but the centre of rotation drifted towards the left, as shown in Figure 6.5. The centre of rotation shifted by 125mm, 170mm and 305mm at S=30, S=50 and S=68, respectively, which is a 6.25mm, 8.5mm and 15.25mm shift every rotation. The average times of 11.67s, 6.69s and 5.58s at S=30, S=50 and S=68, respectively, were achieved. The average RPMs of the inner and outer wheels are 21.05, 42.30 for S1, 35.55, 77.55 for S=50 and 48.4, 105.4 for S=68.

The standard deviation for the time series and RPM series is mentioned in the table below. The times achieved for the three speed values have very low standard deviations, showing great consistency in the data collected. The RPM values also show good standard deviations. From Figure 6.6, we can conclude that S1 has the highest median time, suggesting it's the slowest among the three. The range of times is relatively small, as indicated by the box's size, but it's larger than the other two settings. There are two outliers, showing two rotations that took significantly longer than the rest at this speed setting. S2 has a lower median time than S1, indicating faster completion on average. The interquartile range (IQR) is quite tight, showing that the times were relatively consistent. There are no outliers, which suggests that all the rotations fell within a predictable range of times. S3 has the lowest median time, substantially faster than S1 and slightly faster than S2. The IQR is exceptionally narrow, even more so than S2, indicating consistent times. There are no outliers, similar to S2, which again suggests consistent performance. In summary, as the speed percentage increases from S1 to S3, the time taken for rotation decreases, indicating that rotations are completed more quickly at higher speeds. Additionally, S2 and S3 show more consistency in rotation times than S1, as reflected by the tighter interquartile ranges and the absence of outliers. Similarly, the RPMs also showed very low deviations for the various speeds during testing. The time series plot and scatter plot shown in Figure 6.7 and Figure 6.8 show consistent data over the period of time that the test was run.

| SPEED IN PERCENTAGE | TIME (S) | RPM1 | RPM2 |
|---------------------|----------|------|------|
| S=30                | 0.081    | 0.22 | 0.56 |
| S=50                | 0.16     | 0.67 | 0.67 |
| S=68                | 0.11     | 0.49 | 0.49 |

Table 6.2: Standard deviations for the time taken to complete twenty rotations and the RPMs of the inner and outer wheels.

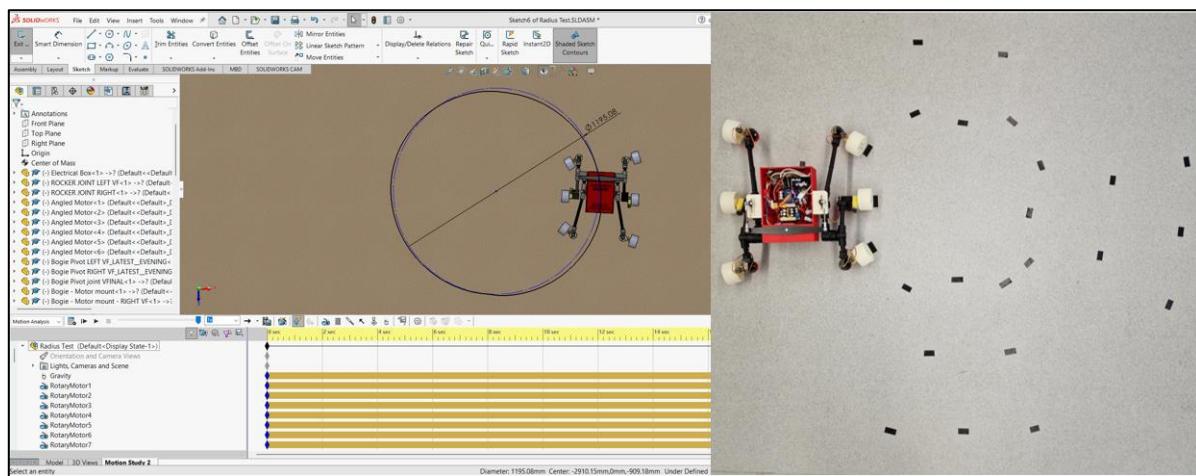


Figure 6.4: Motion simulation of the rover designed in SolidWorks to calculate the minimum radius of turning (mm) and the assembled rover set to be tested along a test track bounded by black tape.

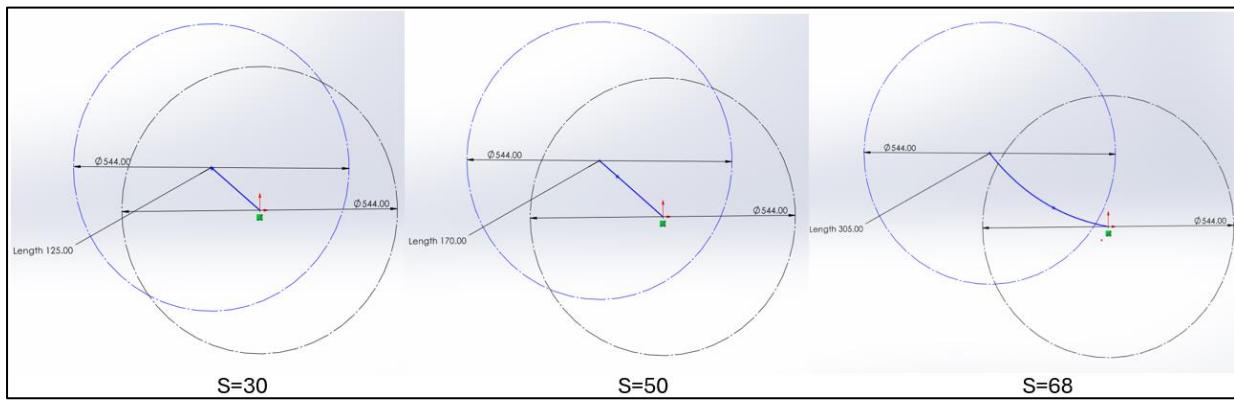


Figure 6.5: Shifting of the centre of rotation of the minimum radius circle by 125mm, 170mm and 305mm at S=30, S=50 and S=68, respectively.

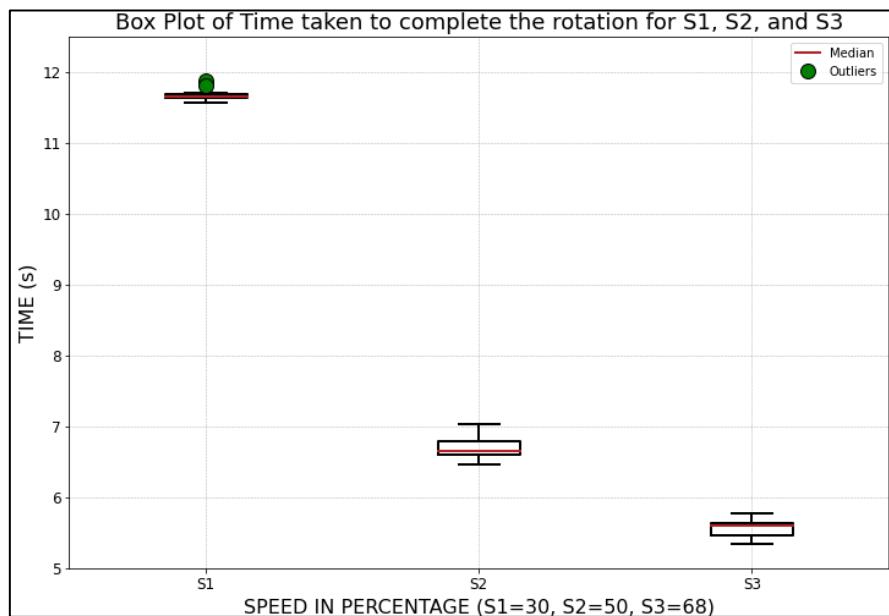


Figure 6.6: Box plot for the time taken to complete twenty rotations at various speeds.

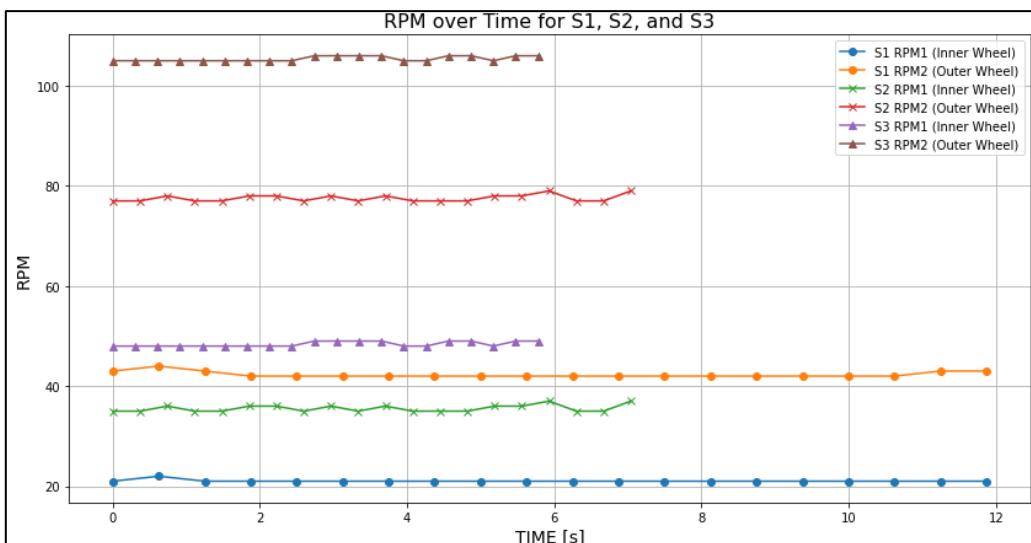


Figure 6.7: Time series plot of the inner and outer wheel RPMs.

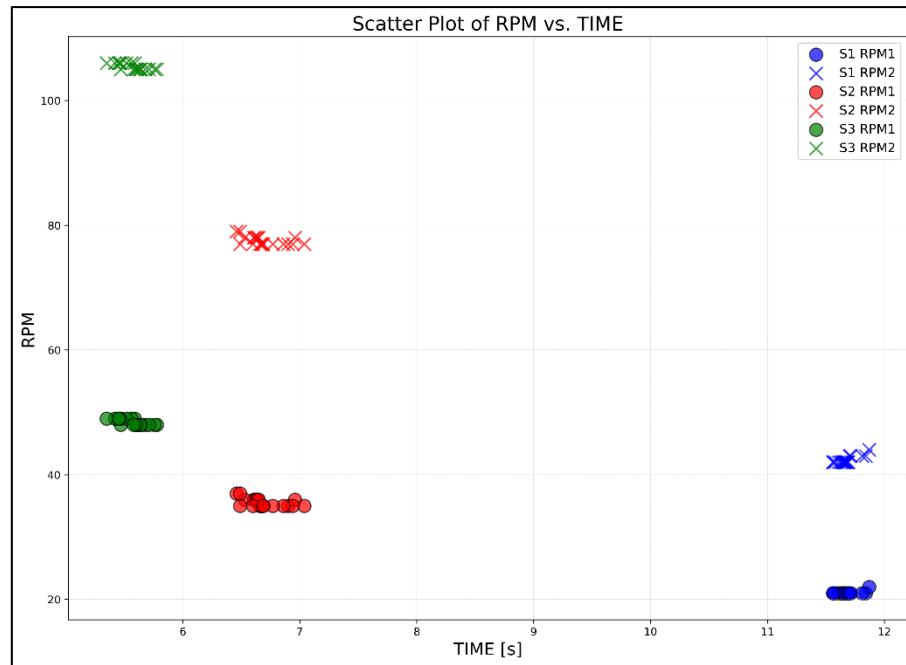


Figure 6.8: Scatter plot of RPM vs TIME showing the consistency of RPMs at various speeds.

### 6.3 On point turning test

#### Objective:

Evaluate the rover's ability to rotate 360 degrees on its vertical axis.

#### Methodology:

- Test Area: A flat, even surface.
- Procedure: Position the rover to turn on the spot by having equal and opposite angles for the front and rear wheels. The mid-wheels turn in opposite directions of each other. Start at the lowest RPM and gradually increase, observing the rover's rotation.
- Measurements: Time each complete 360-degree rotation. Note any irregular movements or inability to maintain position.
- Repetitions: Complete 20 trials at each RPM setting.

#### Results:

The angle at which the four corner wheels need to be for on point rotation is calculated using [equation 9](#), which amounts to  $43.73^{\circ}$ . The motion simulation of the assembled rover in SolidWorks showed no deviation in the path taken, as seen in Figure 6.9. In the real world, the rover turns on point for twenty continuous rotations, but the centre of rotation drifts towards the left by 140mm and 135mm at S=30 and S=50, respectively, as shown in Figure 6.10. This means there is an average of 7mm and 6.5mm of drift per rotation for the two S values, respectively. The average RPM for the inner and outer wheels remains the same as all the outer wheels at the same angle as the normal. The average RPMs are 23.25 and 88.75 for S=30 and S=50, respectively.

The standard deviation for the time series and RPM series for the entire twenty rotations is mentioned in the table below. The times and RPMs achieved for the two speed values have very low standard deviations, showing great consistency in the data collected. Figure 6.11 shows that the median time for S1 is higher than that of S2, indicating slower completion on

average. The interquartile range is wider, suggesting more variability in the times taken for these rotations compared to S2. There are no outliers in S1, which means all times are within the typical range. S2 has a much lower median time, implying faster completion of the rotations. The interquartile range is also narrower, showing that the times were more consistent than S1. No outliers are present in S2 either, reinforcing the idea of consistent performance. Overall, the box plot suggests that increasing the speed from S1 to S2 significantly decreases the time it takes to complete the rotations. It does so with greater consistency in the times recorded. Figure 6.12 shows the time series plot and scatter plot of RPMs with great consistency clustered close to each other.

| SPEED IN PERCENTAGE | TIME (S) | RPM1 | RPM2 |
|---------------------|----------|------|------|
| S=30                | 0.18     | 0.43 | 0.43 |
| S=50                | 0.16     | 0.43 | 0.43 |

Table 6.3: Standard deviations for the time taken to complete twenty rotations on point and the RPMs of the inner and outer wheels.

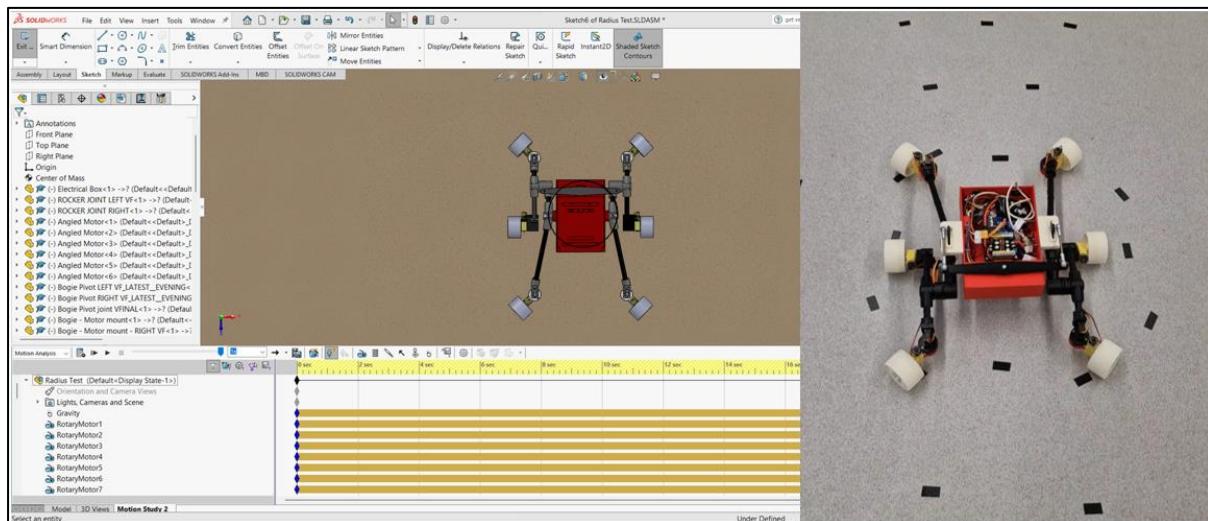


Figure 6.9: Motion simulation of the rover designed in SolidWorks for simulating on point turning and the assembled rover set to be tested along test track bounded by black tape.

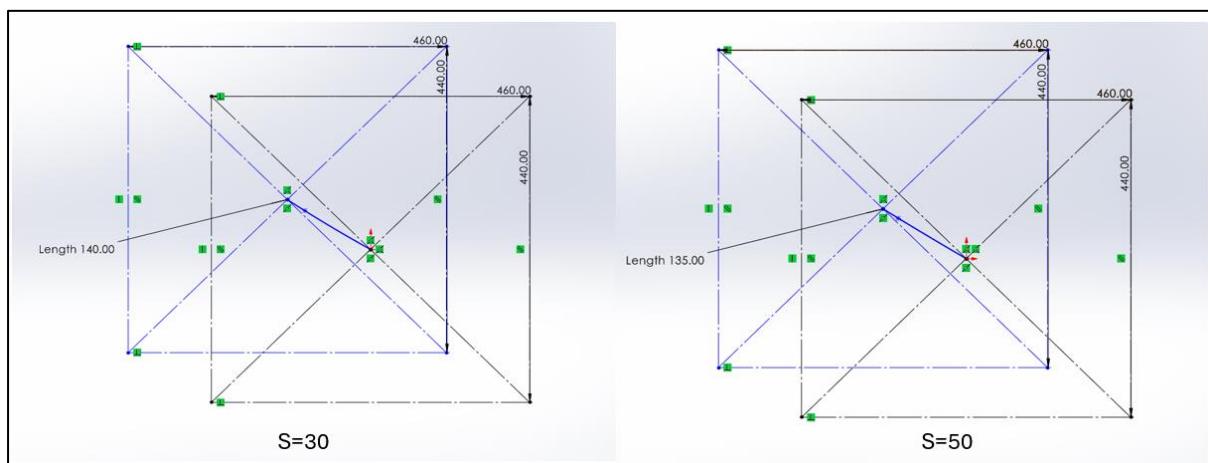


Figure 6.10: Shifting of the centre of rotation of the minimum radius circle by 140mm, 135mm at S=30, S=50 respectively.

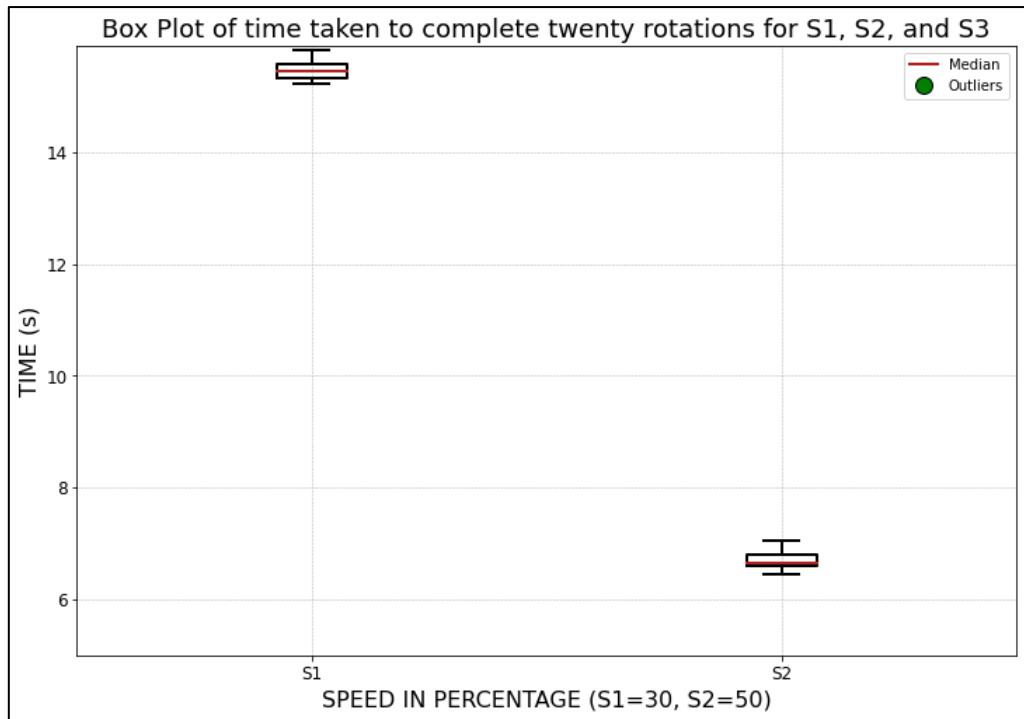


Figure 6.11: Box plot for the time taken to complete twenty on point rotations at various speeds.

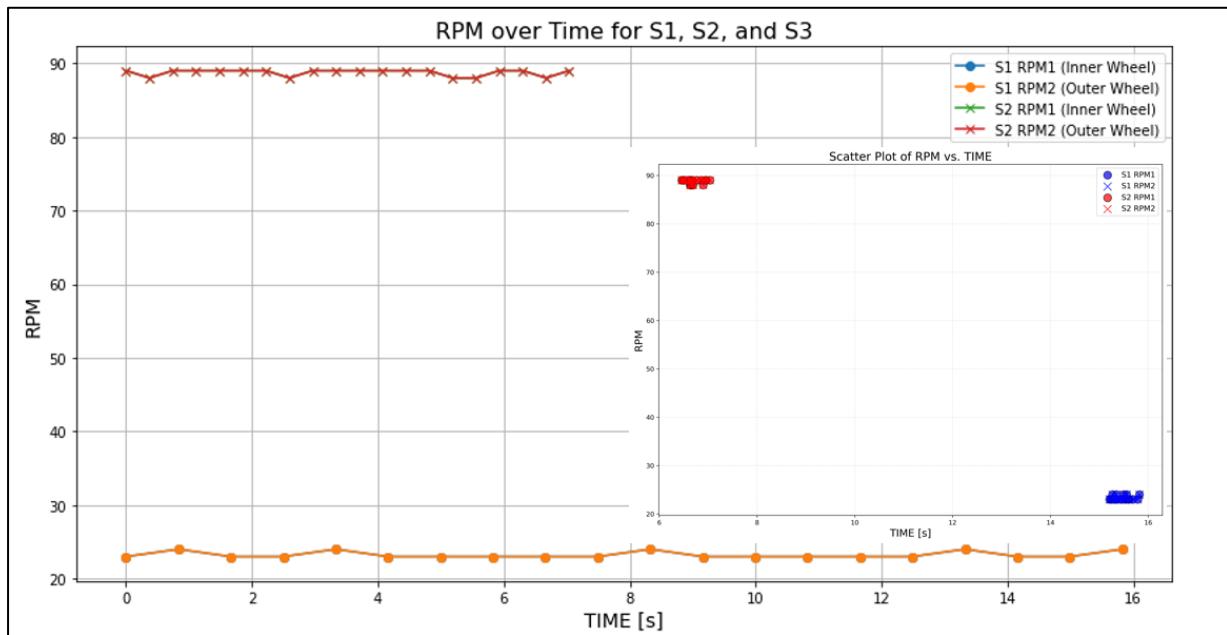


Figure 6.12: Time series and scatter plot of RPM vs TIME.

## 6.4 Driving over obstacles test

### Objective:

To assess the rover's manoeuvrability in navigating around predefined obstacles.

### Methodology:

- Test Course: Set up a course with various obstacles spaced at regular intervals.
- Procedure: Navigate the rover through the course, starting at the lowest RPM. Incrementally increase the RPM with each trial, observing the rover's ability to manoeuvre and check if all the wheels are in contact with the ground at all times.
- Measurements: Course completion time.
- Repetitions: Perform 20 runs at each RPM setting.

### Results:

The SolidWorks motion simulation showed no deviation while climbing, maintaining a straight line path for the 2m test track length. The Rocker-Bogie suspension with the differential bar maintained all six wheels grounded to the floor while climbing the obstacles of 40mm in height. In the real world the rover failed to climb at a speed S=30 due to lack of torque at the drive motors. At S=50 and S=68, the rover easily climbed and completed the test track course, actuating both sides of the rocker-bogie suspension. The average completion times are 6.39s and 4.33s for S=50 and S=68, respectively.

From the box plot for the second speed setting (S2), the median value is above the halfway point of the time scale. The range of most lap times extends from the middle to the upper half of the time scale, with no outliers. For the third speed setting (S3), the median is located towards the lower half of the time scale, indicating a quicker completion of laps on average. The data points are more tightly grouped, showing less variability in times and, like S2, there are no outliers. Overall, the increase from S2 to S3 speed corresponds to a shift towards faster laps and a decrease in the spread of times, indicating more consistency in performance at the higher speed.

| SPEED IN PERCENTAGE | TIME (S) |
|---------------------|----------|
| S=30                | 0.24     |
| S=50                | 0.25     |

Table 6.4: Standard deviations for the time taken to complete twenty laps over obstacles.

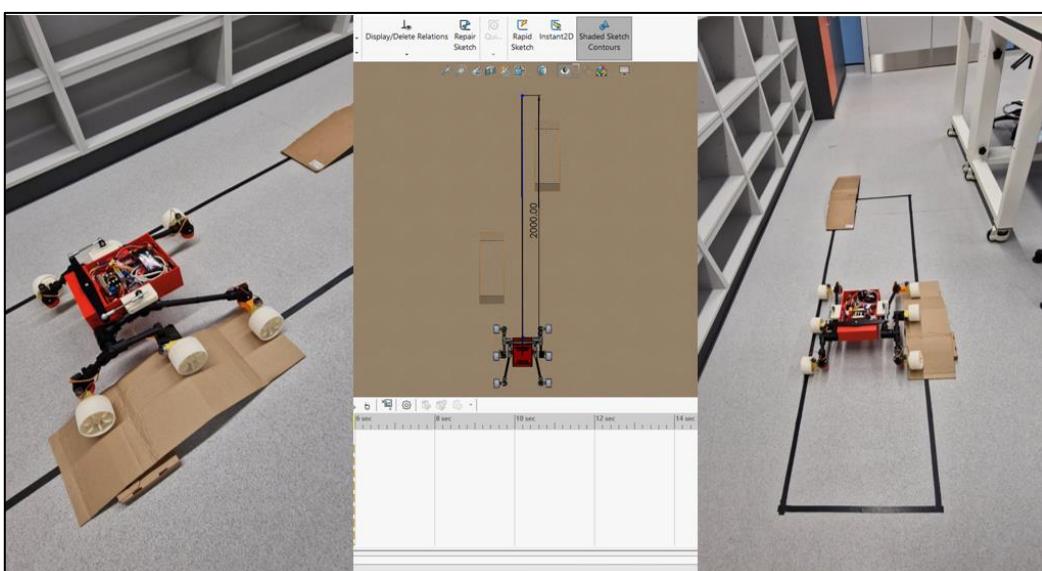


Figure 6.13: Completed motion simulation of the rover designed in SolidWorks for simulating driving over obstacles and the assembled rover tested along test track with obstacles bounded by black tape.

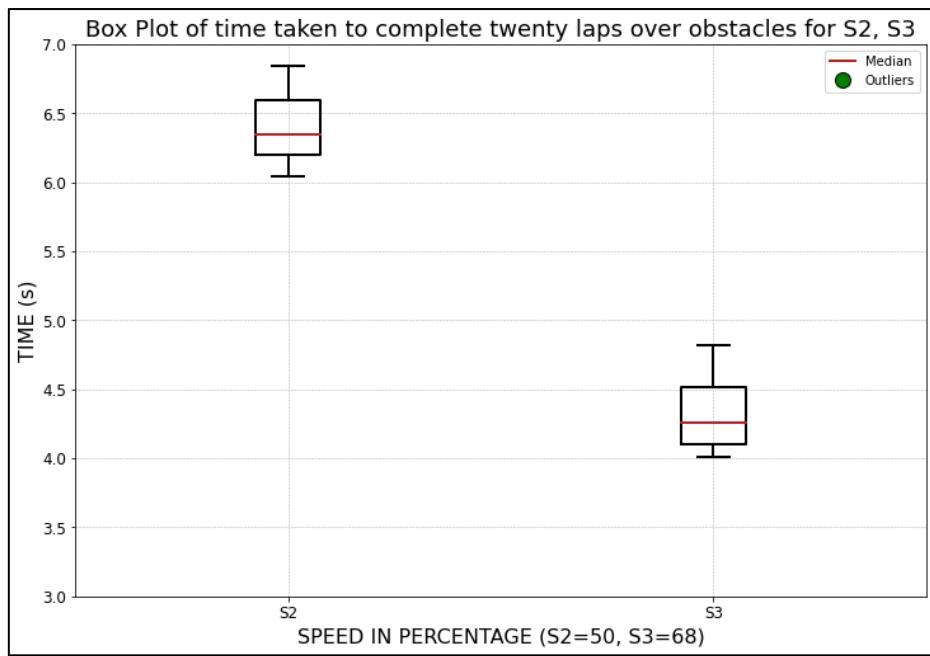


Figure 6.14: Box plot of time taken to complete twenty laps over obstacles at speeds S2 and S3.

## 6.5 Maximum climbing angle test

### Objective:

To determine the maximum inclination angle the rover can climb.

### Methodology:

- Test Ramp: Use a ramp adjustable from 0 to 40 degrees.
- Procedure: Start at 0 degrees and increase the ramp's angle in small increments until the rover fails to climb. Begin at the lowest RPM and increase the steps for each angle.
- Measurements: Note the maximum angle climbed at each RPM setting before losing traction or power.
- Repetitions: Attempt 20 climbs for each combination of angle and RPM setting.

### Results:

The stall simulation is based on [equation 10](#). It gives an estimate of how steep the rover can climb based on the rover's mass in kg, the torque of the drive motors in Nm, earth's gravity in  $\text{ms}^{-2}$  and the radius of the wheels. Figure 6.16 shows the achievable inclinations at various masses of the rover. The measured mass was 2.45kg for which the expected maximum inclinations are  $24^\circ$ ,  $32^\circ$ , and  $42^\circ$  for S=50, S=68 and S=100. Using the wooden ply sheet of length 1.47m and a spirit leveller as shown in the Figure 6.15 the rover was tested at  $10^\circ$ ,  $20^\circ$ ,  $30^\circ$ , and  $40^\circ$ .

At S=50, the rover fails to climb at an inclination greater than  $30^\circ$  and stalls at 0.9m at  $20^\circ$  and 0.52m at  $30^\circ$ . At S=68, the rover fails traction at a distance of 0.65m at an angle of  $30^\circ$  and 0.5m at an angle of  $40^\circ$ . The IMU recorded pitch angles of  $9^\circ$ ,  $18.8^\circ$ ,  $28^\circ$  and  $39^\circ$  due to the inconsistencies in the plywood surface when propped up. The rover completes the climb over the entire plywood sheet at all other lower inclinations. At S=100, the rover fails to climb at an inclination of  $40^\circ$  due to lack of grip. There is little to no friction between the ABS wheels and the plywood sheet. The rover completes the climb over the entire plywood sheet at all

other lower inclinations. This data is shown in Figure 6.17. In conclusion, the rover performed well and achieved climbing inclinations close to the expected values from the simulation.



Figure 6.15: Experimental setup for 30 degrees inclination for the rover to climb.

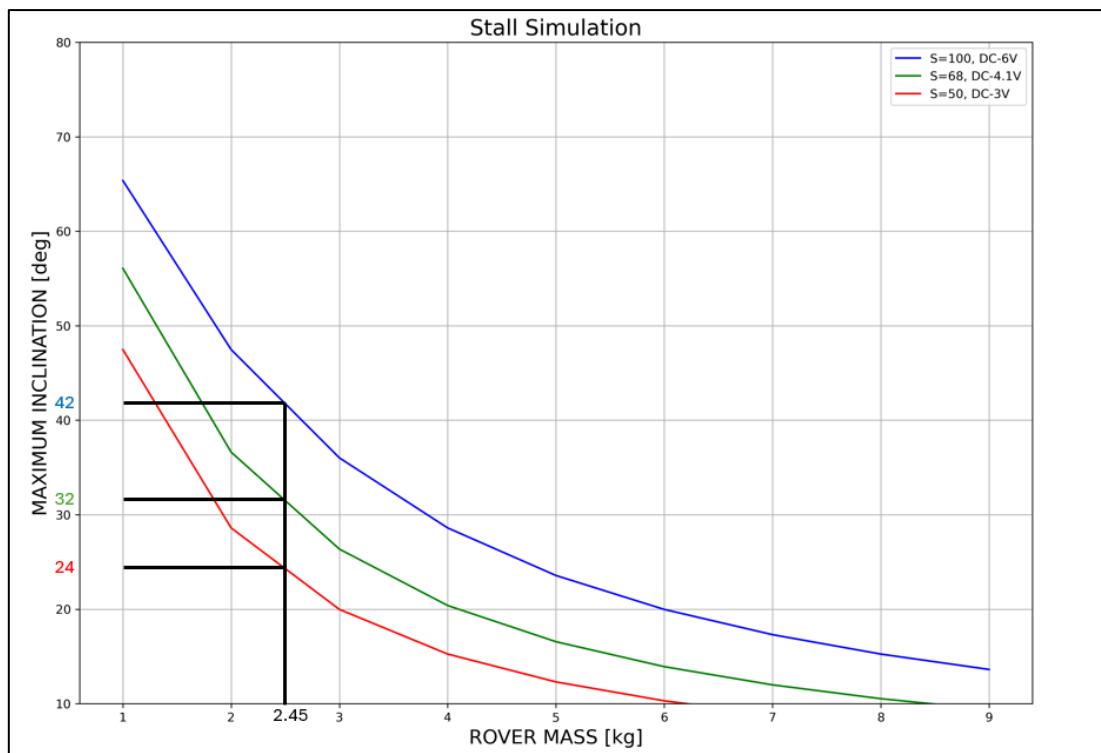


Figure 6.16: Stall simulation to find the maximum angle of inclination at various rover mass.

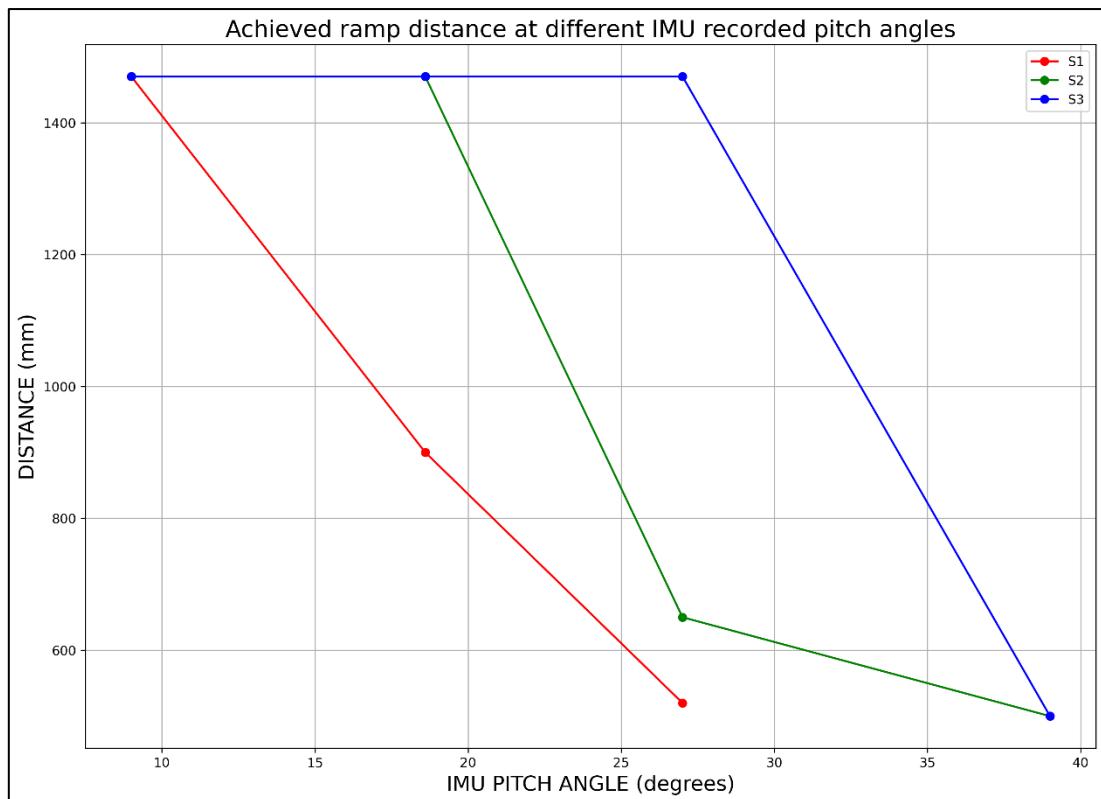


Figure 6.17: Distance the rover was able to climb over the ramp at the recorded IMU pitch angles.

## 6.6 Tip over angle test

### Objective:

To find the angle at which the rover tips over sideways.

### Methodology:

- Test Area: Use a tiltable platform that can be angled precisely.
- Procedure: Place the rover on the platform, gradually increasing the tilt angle until the rover tips over. Start at the lowest RPM setting and increase.
- Measurements: Record the angle at which the rover tips over for each RPM setting.
- Repetitions: Conduct 20 trials for each RPM level to ensure accuracy.

### Results:

The tip-over test is based on the Static Stability Factor (SSF). Using the arctan of [equation 1](#), we can estimate the maximum angle at which the rover will tip over. It results in an angle of  $65.46^0$ . Figure 6.19 shows the angles achievable at various rover COG heights. The experimental setup consists of the same plywood sheet and the spirit level. The plywood sheet is propped using support boxes below and set at the required angles, as shown in Figure 6.18.

The rover successfully traversed the entire plywood board at an inclination of  $10^0$ ,  $20^0$  and  $25^0$  and showed no sign of tipping over for  $S=30$ . For inclinations greater than  $25^0$ , the rover loses grip over the plywood board and slides down even before it moves. A surface with more friction is required to test at greater inclination angles. Figure 6.20 shows the recorded IMU roll angles achieved compared to the ideal angles. The IMU recorded  $9.6^0$ ,  $18.37^0$  and  $23^0$ .

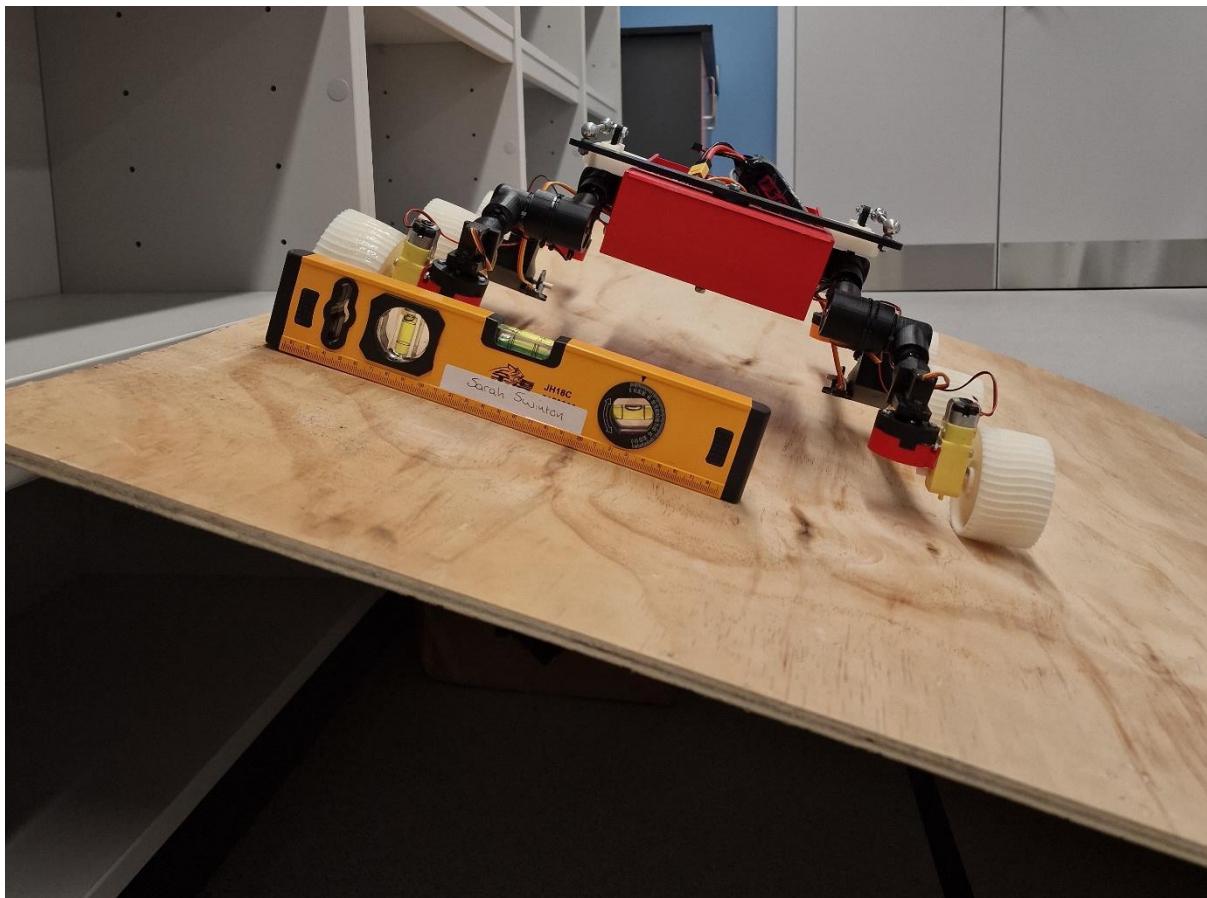
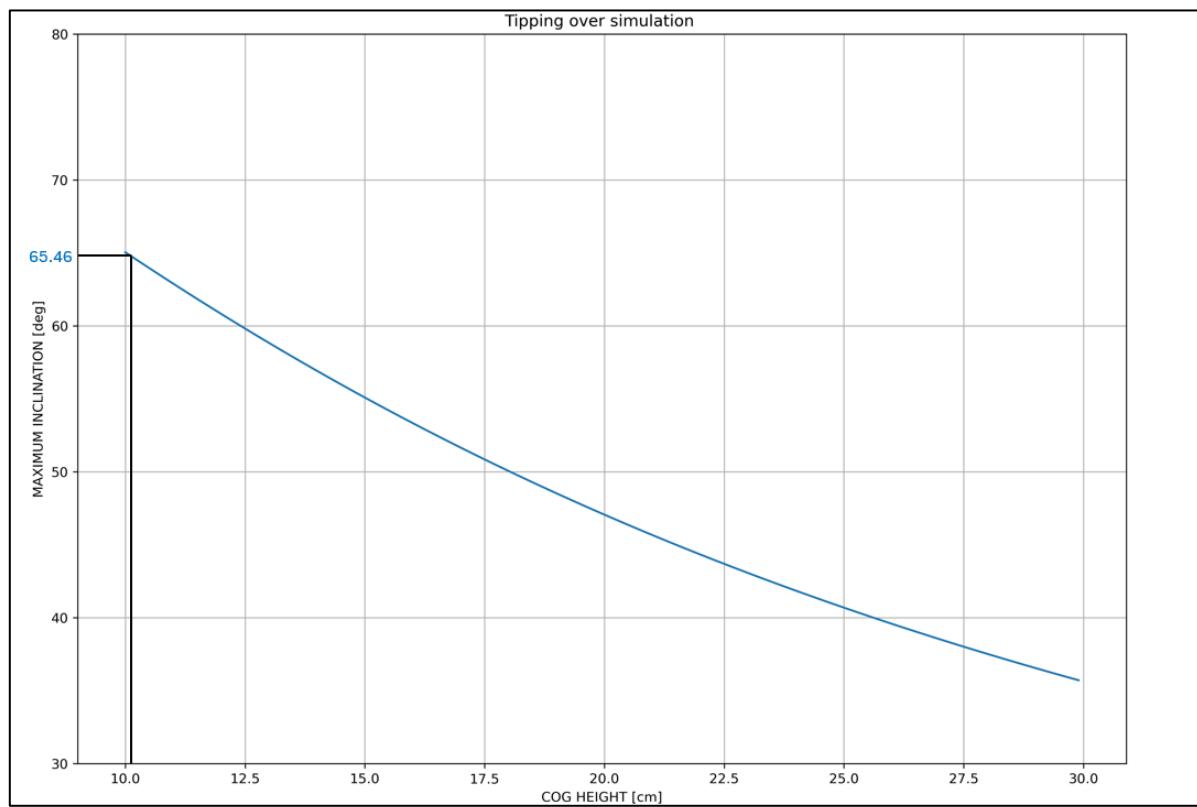
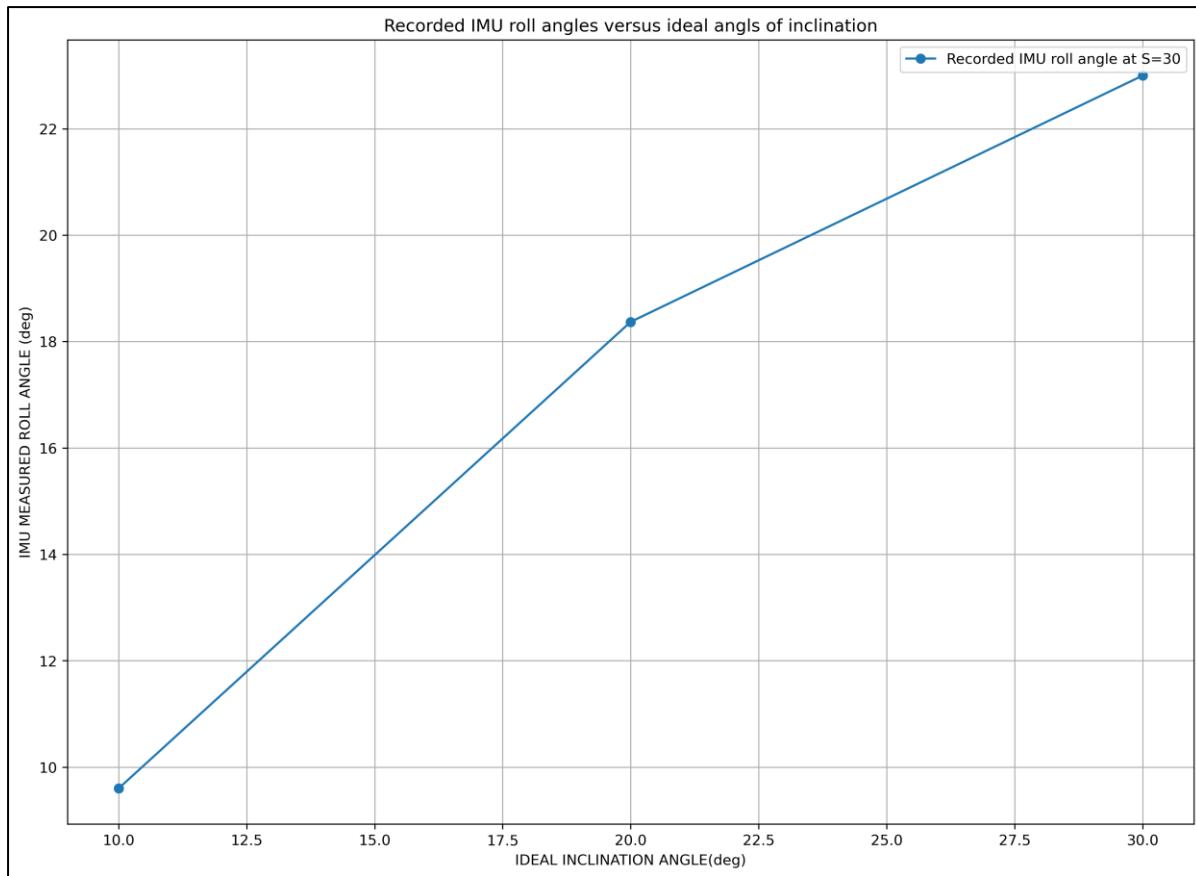


Figure 6.18: Experimental setup to calculate the tip over angle.





## 7 Conclusions and future scope

This dissertation has discussed the design, development, and validation of a sophisticated rover employing the rocker-bogie mechanism. It is renowned for its robust terrain adaptability, particularly in extraterrestrial exploration, as NASA's Mars rovers demonstrated. The rocker-bogie design, central to our rover's mobility, was engineered to ensure optimal balance and stability, catering to the demanding requirements of navigating uneven surfaces. The inclusion of a differential mechanism further enhanced the rover's ability to maintain stability while traversing slopes, thereby mitigating the risk of overturning. Additional hardware was designed to account for final assembly errors, mitigating wheel camber and wheel alignment. Electrical and sensor integration constituted a pivotal aspect of this project, focusing on efficient power regulation, distribution, and the integration of actuators and sensors. The Arduino Due, serving as the rover's brain, performed the operations of various sensors and actuators, facilitating real-time data acquisition and control. The software architecture, designed with modularity and scalability in mind, enabled precise control over the rover's movements. The development of algorithms for calculating motor speeds and servo angles was instrumental in achieving precise steering and manoeuvrability. The programming framework not only facilitated efficient command execution but also allowed for easy integration of additional functionalities.

Experimental testing and validation of the design objectives were crucial to putting the rover to its paces in the real world. The series of tests conducted, ranging from straight-line traversals to turning with minimum radius, validated the rover's operational capabilities and provided valuable insights into areas of potential enhancement. The rover achieved a minimum turning radius of 544mm with an improved Static Stability Factor of 2.19. This is better than the Bogie-Runt. Although the climbing and tilt angle was limited due to the lack of grip on the surface, the rover achieved a commendable maximum climbing angle of  $28^{\circ}$  at S=100 and a tilt angle of  $23^{\circ}$  at S=30. The rover could travel over an obstacle of 400mm in height. In conclusion, the rover achieved the objectives set out at the beginning of this project, but there are a few areas where it can be improved.

The future scope of the rover includes designing a lighter frame of the rover using carbon fibre 3D filament. This will reduce the weight and make it incredibly strong against forces experienced while climbing obstacles. Implementing servo motors at the middle wheels would unlock more steering configurations, allowing for more complex manoeuvres. Higher torque drive motors can be used to improve the climbing capabilities of the rover at very slow speeds. The grip at the wheels can be improved by using a rubber coating or 3D printing a rubber glove for the wheel using TPU filament. This will increase the grip irrespective of the surface on which the rover is being tested. A closed loop system where the IMU can constantly correct any variations in the rovers direction while turning. A PID controller would play a significant role in this implementation. Finally, for simultaneous and efficient data transmission and collection, LoRa transceivers can be used instead of Bluetooth as they provide more bandwidth for the data that can be received or transmitted.

## References

- [1] Muirhead BK. Mars rovers, past and future. IEEE Aerospace Conference Proceedings 2004;1:128–34. <https://doi.org/10.1109/AERO.2004.1367598>.
- [2] Flessa T, McGookin EW, Thomson DG. Taxonomy, Systems Review and Performance Metrics of Planetary Exploration Rovers 2014:1554–9.
- [3] Wilcox B, Nguyen T. Sojourner on mars and lessons learned for future planetary rovers. SAE Technical Papers 1998. <https://doi.org/10.4271/981695>.
- [4] Harrington BD, Voorhees C. The Challenges of Designing the Rocker-Bogie Suspension for the Mars Exploration Rover n.d.
- [5] Babu B, Dhayanidhi N, Dhamotharan S. DESIGN AND FABRICATION OF ROCKER BOGIE MECHANISM GEOSURVEY ROVER 2018.
- [6] Verma A, Yadav C, Singh B, Gupta A, Mishra J, Saxena A. Design of Rocker-Bogie Mechanism Chapter 1 INTRODUCTION OF PROJECT 1.1 BASIC DESCRIPTION. Int J Innov Sci Res Technol 2017;2.
- [7] Lindemann RA, Voorhees CJ. Mars exploration rover mobility assembly design, test and performance. Conf Proc IEEE Int Conf Syst Man Cybern 2005;1:450–5. <https://doi.org/10.1109/ICSMC.2005.1571187>.
- [8] Woodall M, Baechler G, Salehinia I, Pohlman NA. Mars Rover Rocker-Bogie Comparative Differential Study. Int J Innov Res Sci Eng Technol 2020.
- [9] Elizabeth Jordan. Mars Science Laboratory differential restraint : the devil is in the details. Pasadena, CA : Jet Propulsion Laboratory, National Aeronautics and Space Administration 2012. <https://trs.jpl.nasa.gov/handle/2014/42431>.
- [10] Gross D, Hauger W, Schröder J, Wall WA, Rajapakse N. Engineering Mechanics 1. Engineering Mechanics 1 2013. <https://doi.org/10.1007/978-3-642-30319-7/COVER>.
- [11] Edgar T V., Urynowicz MA, Hamann JC. The static stability factor - A dynamic introduction to engineering. ASEE Annual Conference and Exposition, Conference Proceedings 2005:12987–95. <https://doi.org/10.18260/1-2--14846>.
- [12] Trends and Rollover-Reduction Effectiveness of Static Stability Factor in Passenger Vehicles n.d. <https://trid.trb.org/view/1485061>.
- [13] Shamah B. Experimental Comparison of Skid Steering Vs. Explicit Steering for a Wheeled Mobile Robot 1999.
- [14] (PDF) Effect of Tire Design and Steering Mode on Robotic Mobility in Barren Terrain n.d.  
[https://www.researchgate.net/publication/237768170\\_Effect\\_of\\_Tire\\_Design\\_and\\_Steering\\_Mode\\_on\\_Robotic\\_Mobility\\_in\\_Barren\\_Terrain](https://www.researchgate.net/publication/237768170_Effect_of_Tire_Design_and_Steering_Mode_on_Robotic_Mobility_in_Barren_Terrain).

- [15] (PDF) Motion Control for a Planetary Exploration Rover with Six Steerable Wheels n.d. [https://www.researchgate.net/publication/251772420\\_Motion\\_Control\\_for\\_a\\_Planetary\\_Exploration\\_Rover\\_with\\_Six\\_Steerable\\_Wheels](https://www.researchgate.net/publication/251772420_Motion_Control_for_a_Planetary_Exploration_Rover_with_Six_Steerable_Wheels).
- [16] Solli ML, Mcgookin E, Watt J. Simulations and Control of a Six-Wheel Rover. 2019.
- [17] BGTXINGI 20A 300W High-power Buck Converter Module DC 6-40V to DC 1.2-36V: Amazon.co.uk.
- [18] Hailege 3pcs XL4005 5A Adjustable DC-DC Buck Step Down Power Module 4-35V to 1.25-32V : Amazon.co.uk.
- [19] L298N Dual H-Bridge Motor Driver Module – 4tronix n.d. <https://shop.4tronix.co.uk/products/l298n-dual-h-bridge-motor-driver-module>.
- [20] 2547 | Right Angle Geared TT Hobby Motor | RS n.d. <https://uk.rs-online.com/>.
- [21] MG90D – Tower Pro n.d. <https://www.towerpro.com.tw/product/mg90d-2/> (accessed March 21, 2024).
- [22] Due | Arduino Documentation | Arduino Documentation n.d. <https://docs.arduino.cc/hardware/due> (accessed June 23, 2022).
- [23] HC 06 Bluetooth module pinout, features & datasheet n.d. <https://components101.com/wireless/hc-06-bluetooth-module-pinout-datasheet> (accessed March 21, 2024).
- [24] MPU6050 Module Pinout, Configuration, Features, Arduino Interfacing & Datasheet n.d. <https://components101.com/sensors/mpu6050-module>.

## Appendices

### Appendix A – Design drawings of the parts with dimensions in mm

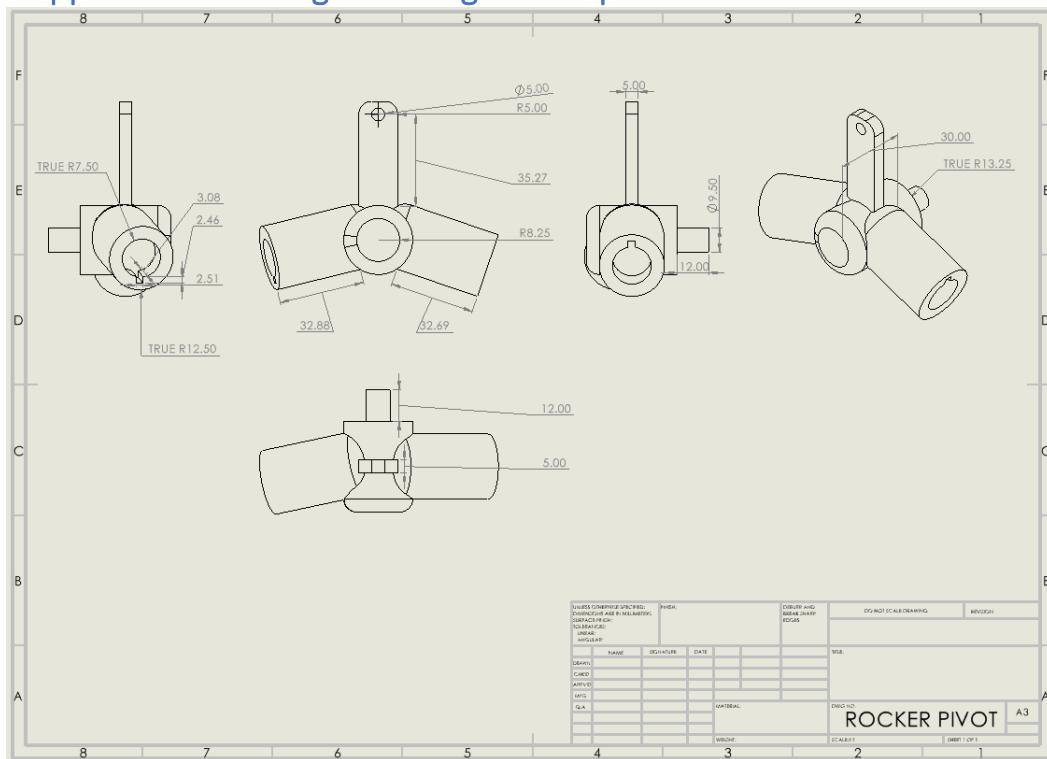


Figure 7.1: SolidWorks drawing with physical dimensions of the rocker pivot.

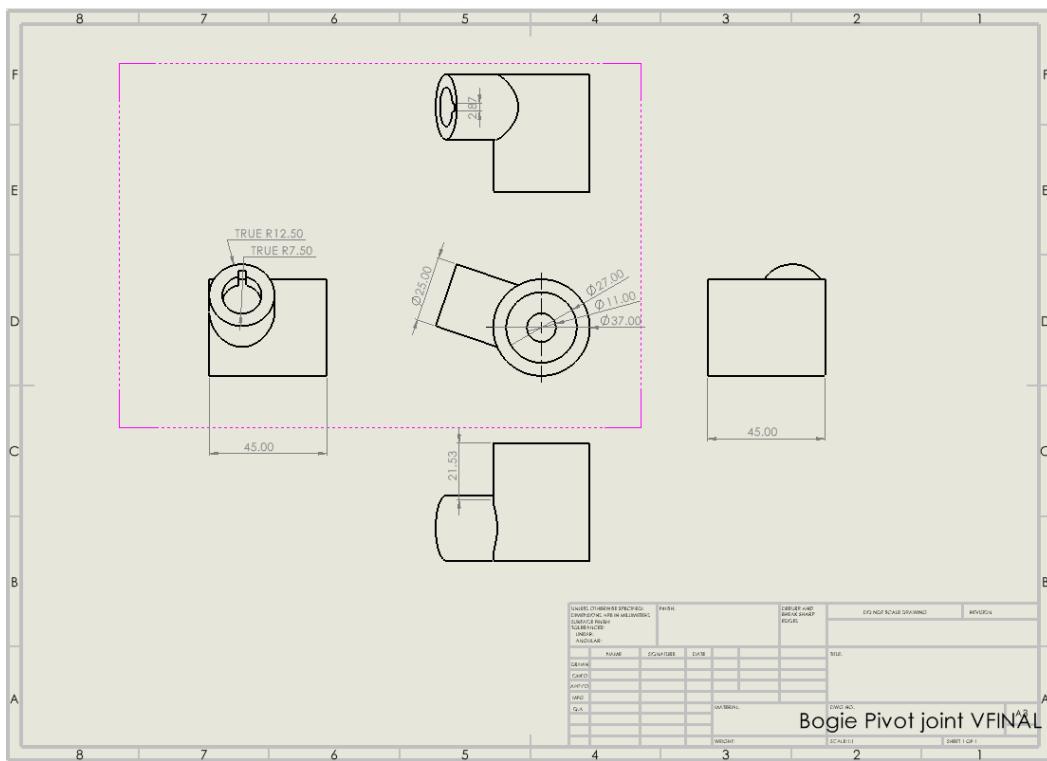


Figure 7.2: SolidWorks drawing with physical dimensions of the bogie pivot joint.

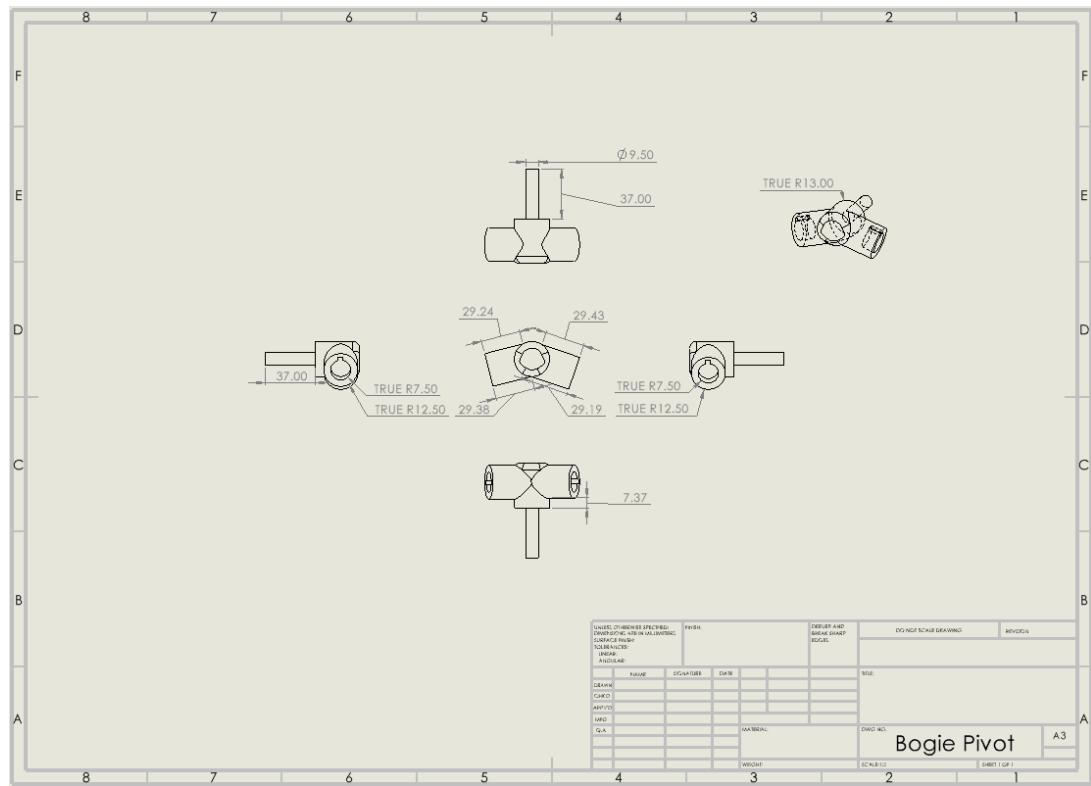


Figure 7.3: SolidWorks drawing with physical dimensions of the rocker pivot.

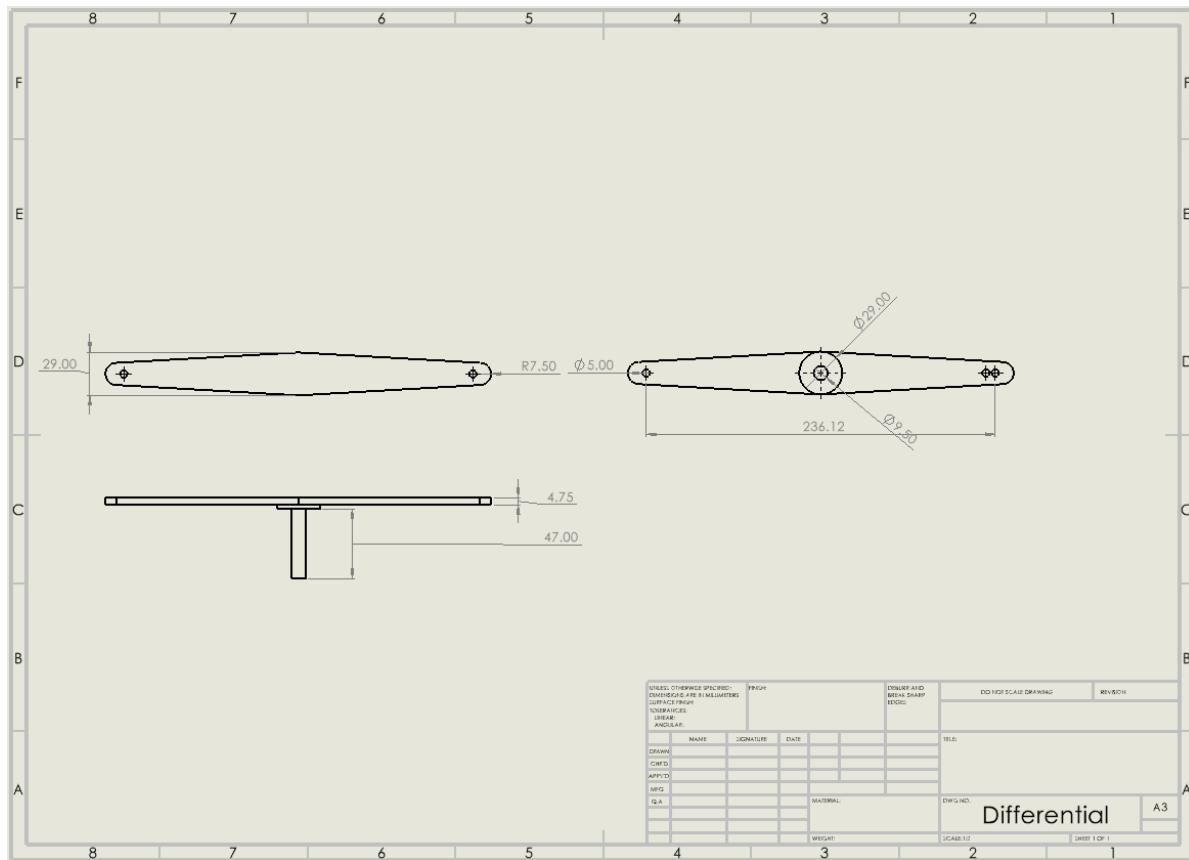
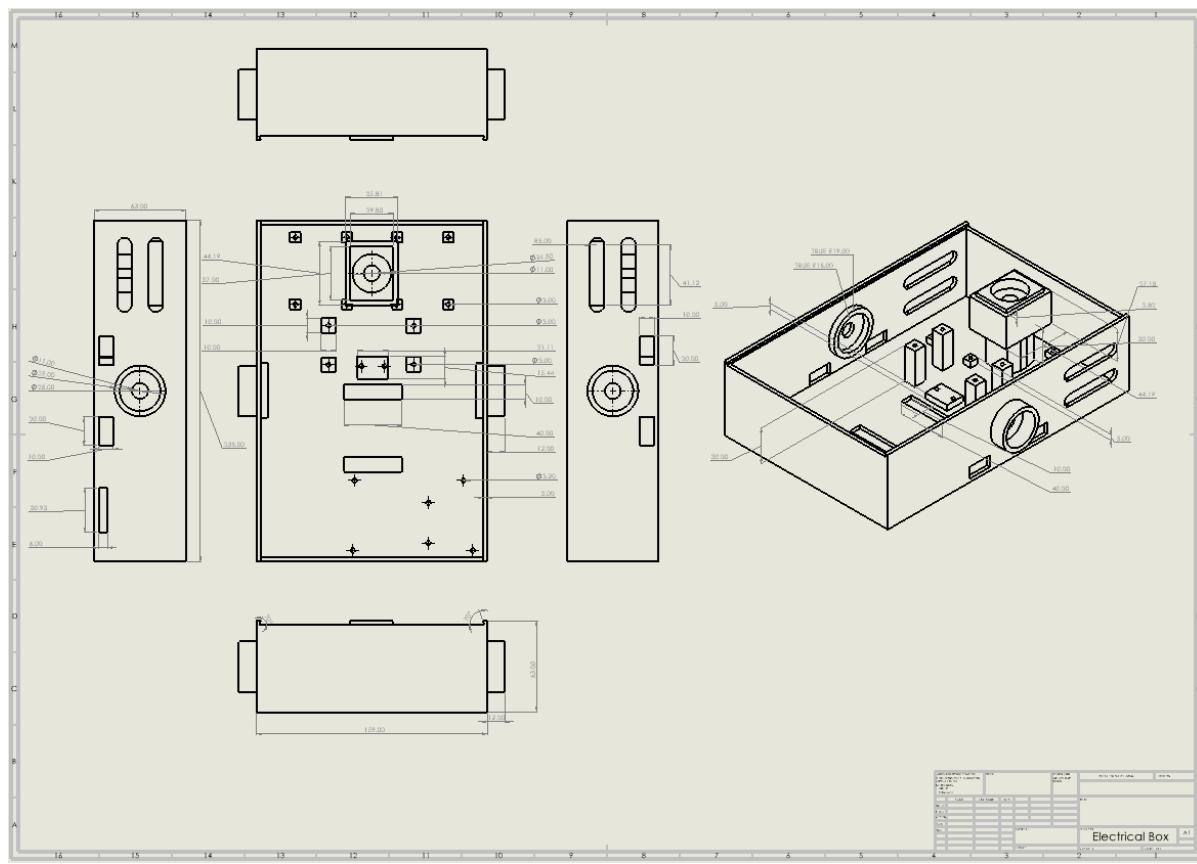
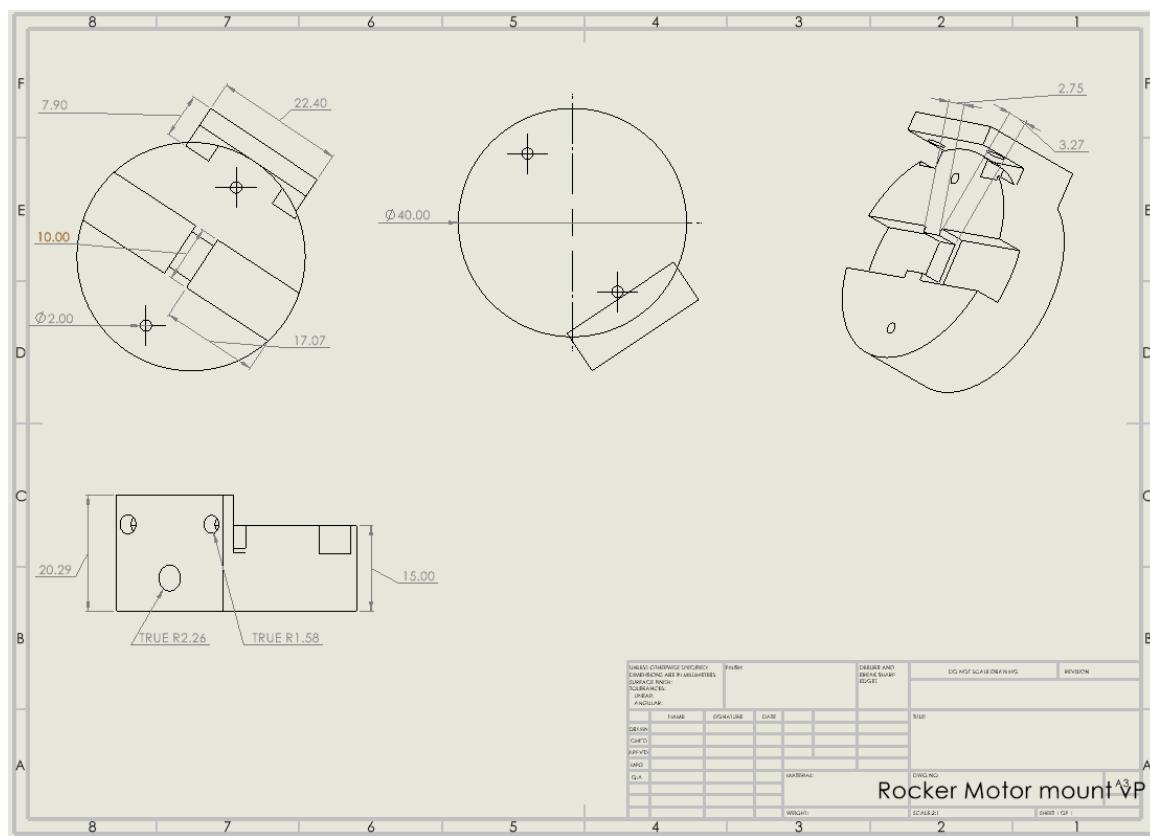


Figure 7.4: SolidWorks drawing with physical dimensions of the differential.



*Figure 7.5: SolidWorks drawing with physical dimensions of the electrical box.*



*Figure 7.6: SolidWorks drawing with physical dimensions of the motor mount.*

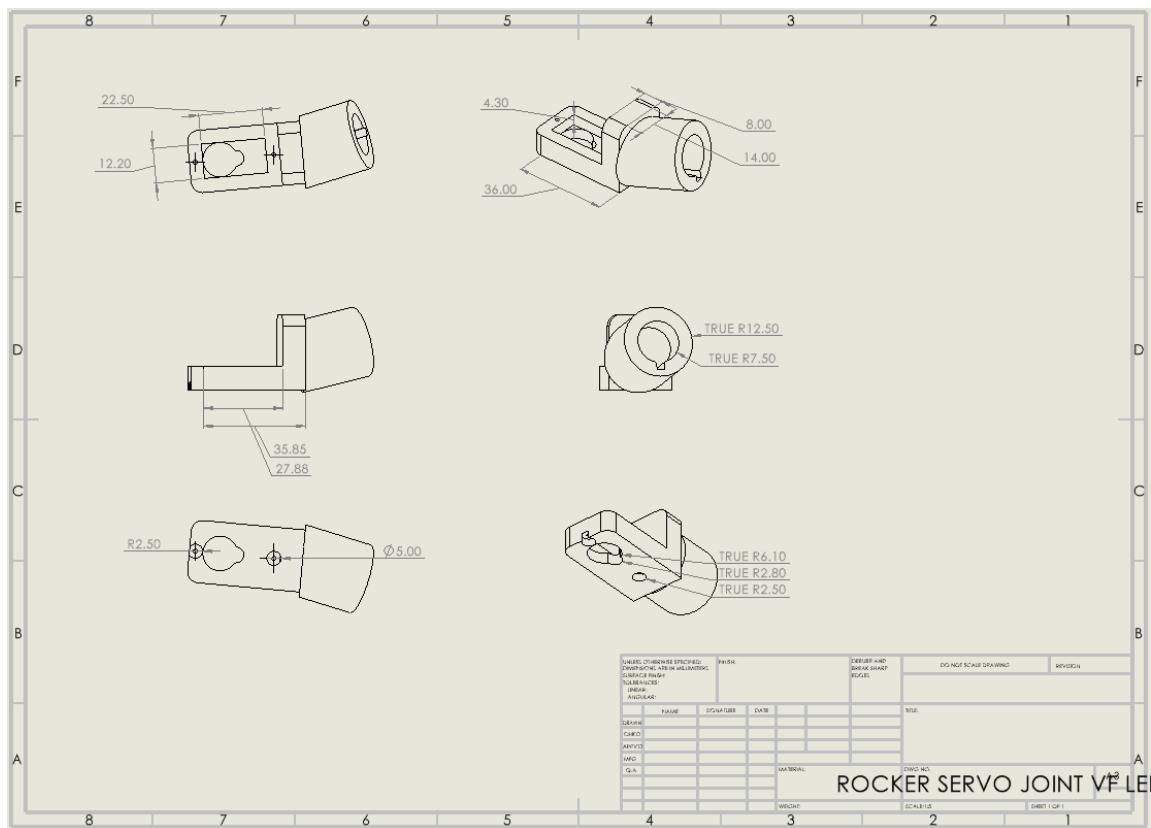


Figure 7.7: SolidWorks drawing with physical dimensions of the servo mount.

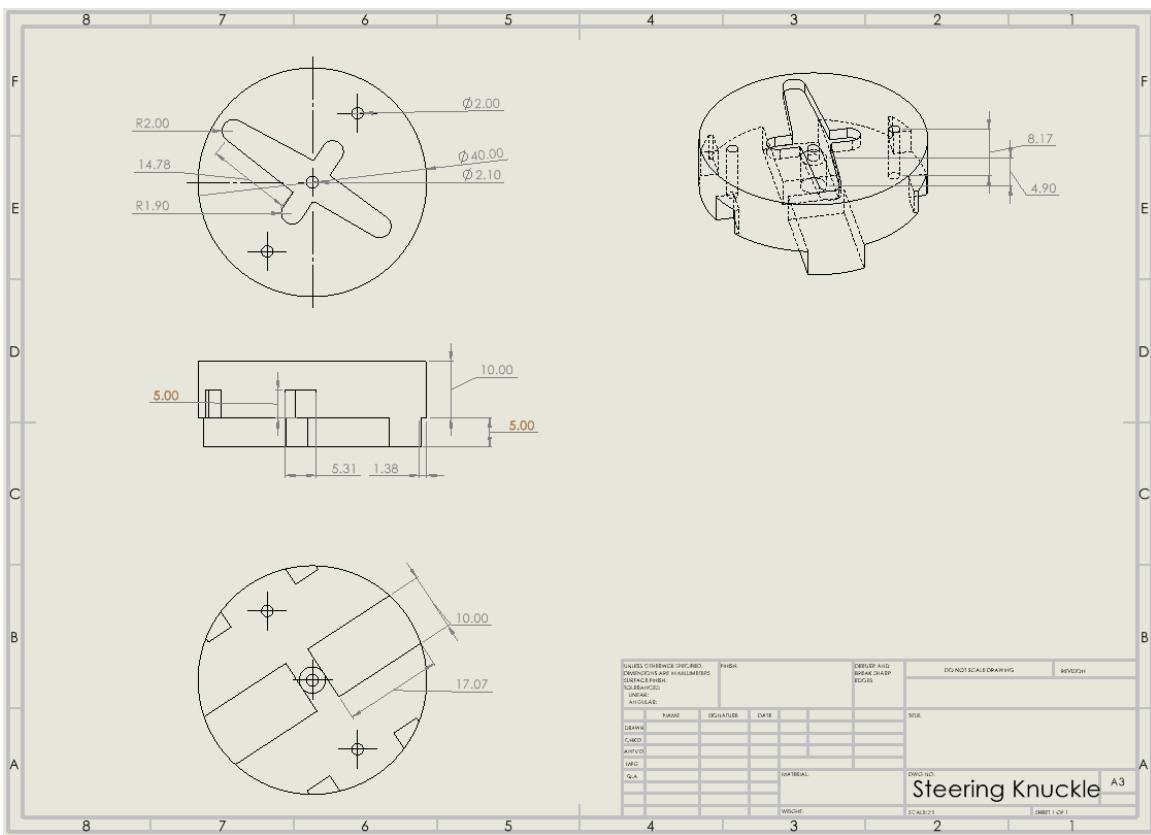


Figure 7.8: SolidWorks drawing with physical dimensions of the steering knuckle.

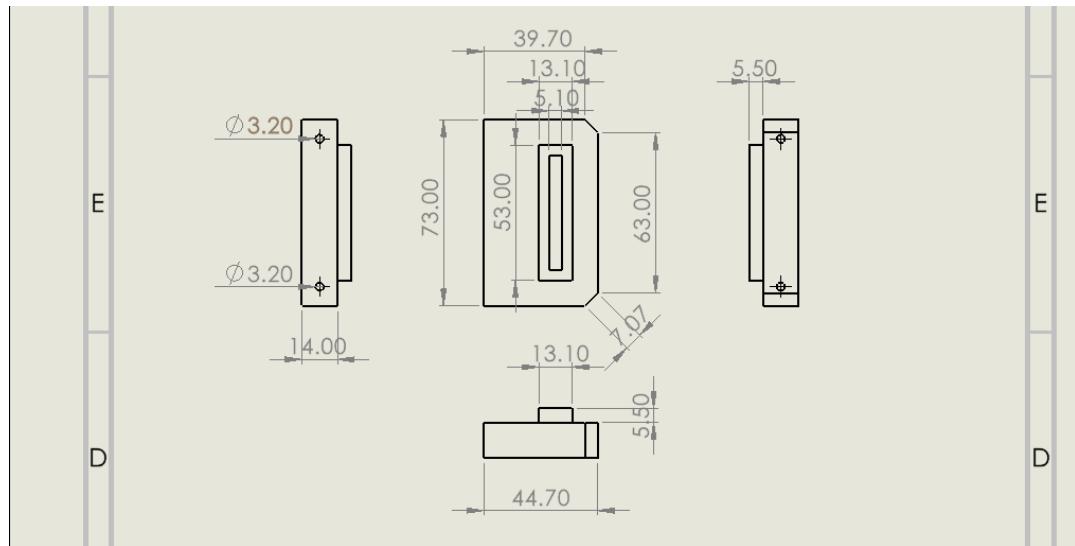


Figure 7.9: SolidWorks drawing with physical dimensions of the rocker pivot support bracket.

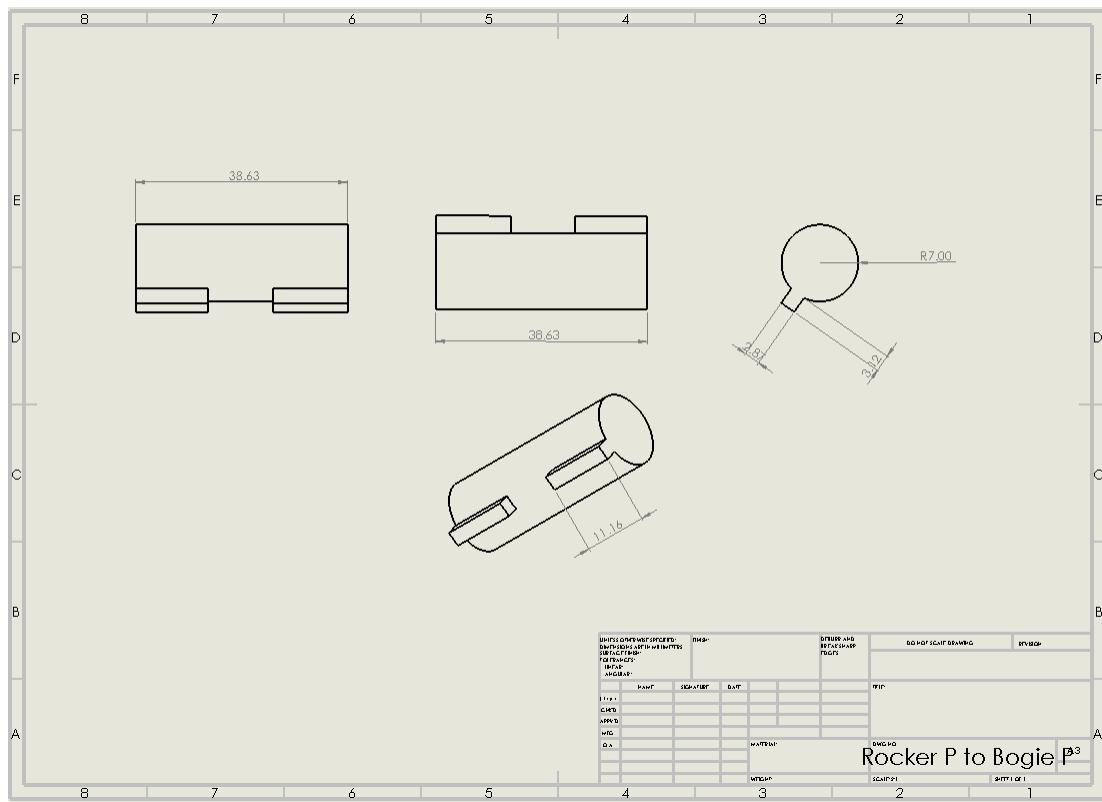


Figure 7.10: SolidWorks drawing with physical dimensions of the rocker pivot to bogie pivot bracket.

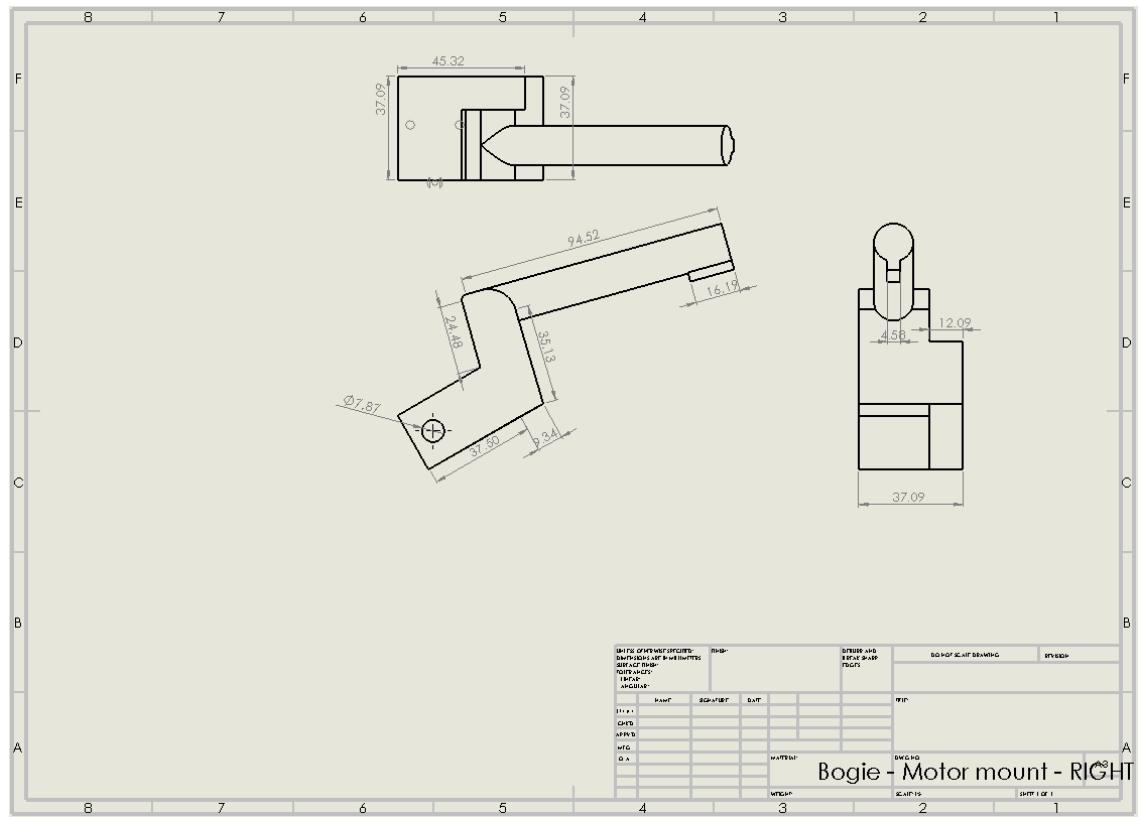


Figure 7.11: SolidWorks drawing with physical dimensions of bogie pivot to mid motor.

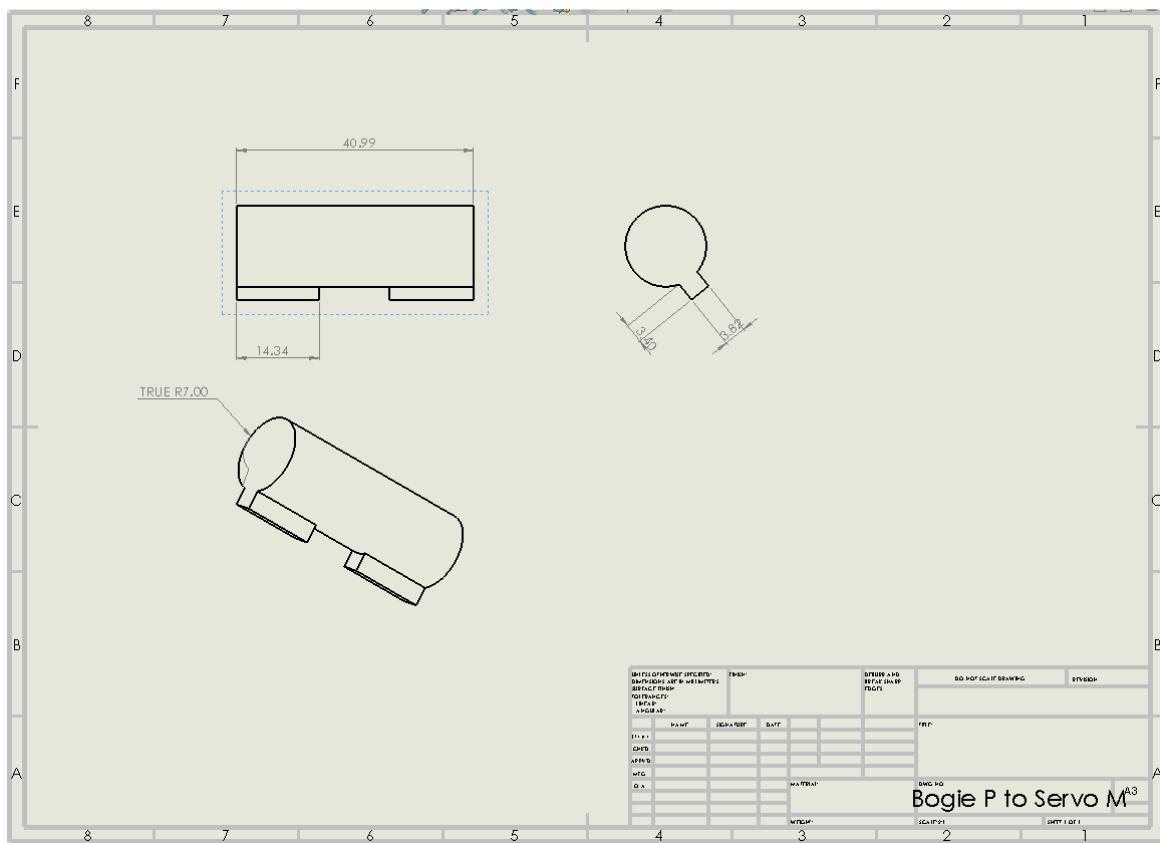


Figure 7.12: SolidWorks drawing with physical dimensions of the bogie pivot to back servo motor link.

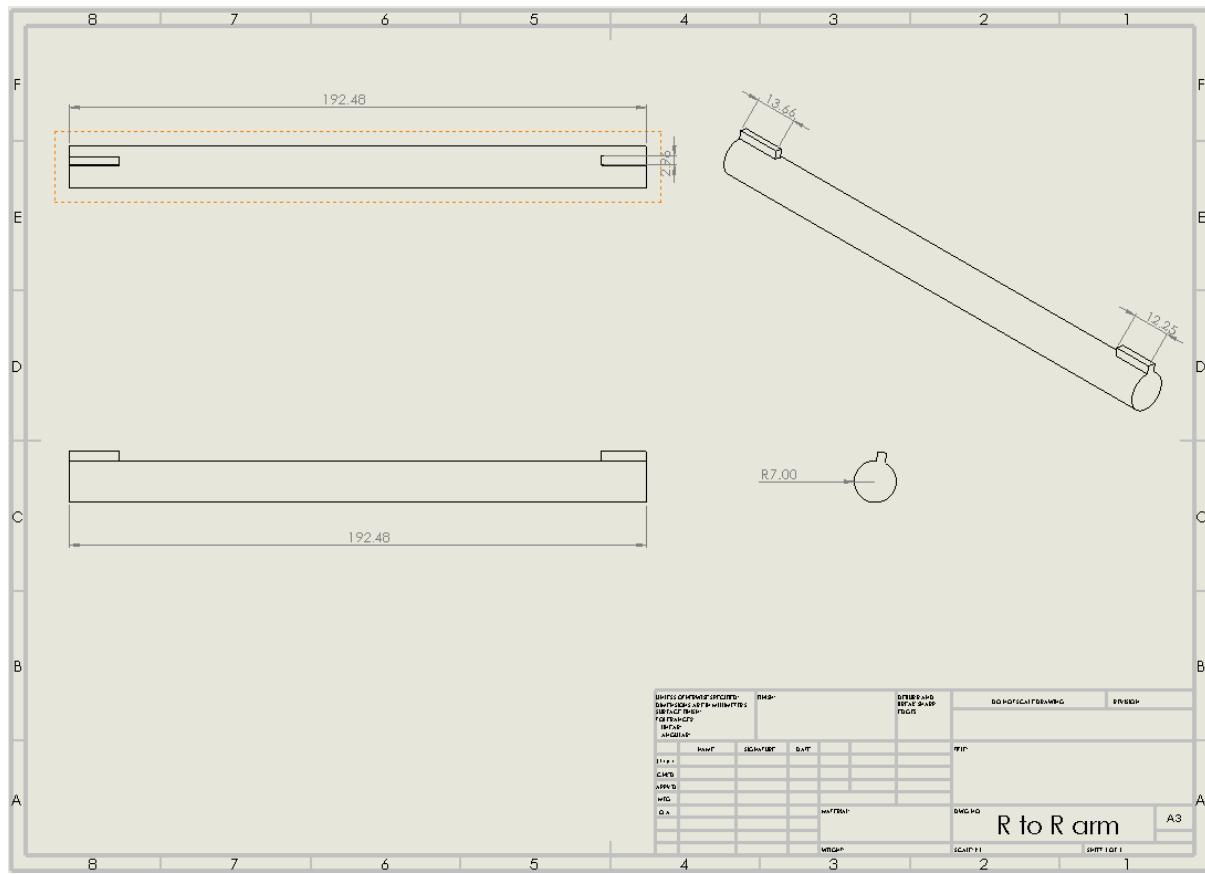


Figure 7.13: SolidWorks drawing with physical dimensions of the rocker to front motor mount link.

## Appendix B -- Codebase

### 1 Code to find the minimum and maximum heading angle ‘a’ for steering.

```
#include <iostream>
using namespace std;
#include <math.h>
#include <cmath>

int main() {

    int wheelbase = 450;
    int trackwidth = 430;
    // Defining the turning angles alpha1(a1) and alpha2(a2) and alpha_rad.
    float alpha1, alpha2, alpha_rad;
    // Defining the heading angle alpha(a).
    int alpha;
    /* for loop goes through all the angles from 0 degrees to 89 degrees
    to find the range of heading angles that have a positive solution for alpha1 and alpha2. */
    for (alpha = 1; alpha < 89; alpha++) {
        // Converting degrees to radians.
        alpha_rad = alpha * M_PI / 180;

        alpha1 = atan(wheelbase / ((wheelbase / tan(alpha_rad)) + trackwidth));

        alpha2 = atan(wheelbase / ((wheelbase / tan(alpha_rad)) - trackwidth));
        // Converting radians to degrees and printing the heading angle, a1 and a2.
        cout << alpha << " A1: " << alpha1 * 180 / 3.1415 << " A2: " << alpha2 * 180 / 3.1415 << endl;
    }

    return 0;
}
```

### 2 Code for testing geared DC motors.

```
// Define the pins used on the Arduino Due.
// Enable pins for controlling the speed of the motor(enX).
#define en1 2
#define en2 3
#define en3 4
#define en4 5
#define en5 6
#define en6 7
// Input pins of the motor(inX and inY).
// Motor 1
#define in1a 30
#define in1b 32
// Motor 2
#define in2a 34
#define in2b 36
// Motor 3
#define in3a 22
#define in3b 24
// Motor 4
#define in4a 26
#define in4b 28
// Motor 5
#define in5a 23
#define in5b 25
// Motor 6
#define in6a 27
#define in6b 29
void setup() {
    // Initialize the pins as either INPUT or OUTPUT.
    pinMode(en1, OUTPUT);
    pinMode(en2, OUTPUT);
    pinMode(en3, OUTPUT);
    pinMode(en4, OUTPUT);
    pinMode(en5, OUTPUT);
    pinMode(en6, OUTPUT);

    pinMode(in1a, OUTPUT);
    pinMode(in1b, OUTPUT);
    pinMode(in2a, OUTPUT);
    pinMode(in2b, OUTPUT);
    pinMode(in3a, OUTPUT);
    pinMode(in3b, OUTPUT);
    pinMode(in4a, OUTPUT);
    pinMode(in4b, OUTPUT);
    pinMode(in5a, OUTPUT);
    pinMode(in5b, OUTPUT);
    pinMode(in6a, OUTPUT);
    pinMode(in6b, OUTPUT);
    // Setting the speed of the motors using PWM signal 0-off to 255-max.
    analogWrite(en1, 255);
    analogWrite(en2, 255);
    analogWrite(en3, 255);
    analogWrite(en4, 255);
    digitalWrite(in2a, LOW);
    digitalWrite(in2b, LOW);

    digitalWrite(in1a, LOW);
    digitalWrite(in3b, LOW);

    digitalWrite(in4a, LOW);
    digitalWrite(in4b, LOW);

    digitalWrite(in5a, LOW);
    digitalWrite(in5b, LOW);

    digitalWrite(in6a, LOW);
    digitalWrite(in6b, LOW);
}
```

### 3 Code for testing servo motors.

```
#include <Servo.h>

// Create a Servo object for each servo
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;

// Define the servo PWM pin numbers
int servoPin1 = 2;
int servoPin2 = 3;
int servoPin3 = 4;
int servoPin4 = 5;

void setup() {
    // Attach each servo to its corresponding pin
    servo1.attach(servoPin1);
    servo2.attach(servoPin2);
    servo3.attach(servoPin3);
    servo4.attach(servoPin4);

    // Initially set each servo to 0 degrees steering angle
    servo1.write(90);
    servo2.write(90);
    servo3.write(90);
    servo4.write(90);
    delay(5000); // Wait for 5 seconds

    // Move each servo to 90 degrees
    servo1.write(45);
    servo2.write(45);
    servo3.write(45);
    servo4.write(45);
    delay(5000); // Wait for 5 seconds

    // Move each servo to 180 degrees
    servo1.write(135);
    servo2.write(135);
    servo3.write(135);
    servo4.write(135);
    delay(5000); // Wait for 5 seconds

    // Return each servo to 0 degrees
    servo1.write(90);
    servo2.write(90);
    servo3.write(90);
    servo4.write(90);

    delay(5000); // Wait for 5 seconds
}

void loop() {
    // Your loop code here (if needed)
}
```

### 4 Code for the entire rover.

```
// Include required libraries for peripherals
#include <Servo.h>
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#include "Wire.h"
#define BT Serial3
// Define the motor encoder pins
#define ENCODER1_INTERRUPT_PIN 40
#define ENCODER2_INTERRUPT_PIN 42
// The number of slits in the encoder disk
#define ENCODER_N 20
// Timing variables for the motor encoder interrupt functions
unsigned long T1 = 0, T2 = 0, Ta;
unsigned long T3 = 0, T4 = 0, Tb;
bool MeasDone1 = 0;
bool MeasDone2 = 0;
// Motor encoder RPM variables
int Motor1_RPM = 0;
int Motor2_RPM = 0;
// Create mpu object for the MPU6050 IMU
MPU6050 mpu;
// yaw/pitch/roll angles (in degrees) calculated from the quaternions coming from the FIFO
#define OUTPUT_READABLE_YAWPITCHROLL
#define LED_PIN 13
#define IMU_INTERRUPT_PIN 38
bool blinkState = false;
// MPU control/status vars
bool dmpReady = false; // set true if DMP init was successful
uint8_t mpuIntStatus; // holds actual interrupt status byte from MPU
uint8_t devStatus; // return status after each device operation (0 = success, !0 = error)
uint16_t packetSize; // expected DMP packet size (default is 42 bytes)
uint16_t fifoCount; // count of all bytes currently in FIFO
```

```


// orientation/motion vars
Quaternion q;           // [w, x, y, z]      quaternion container
VectorInt16 aa;          // [x, y, z]        accel sensor measurements
VectorInt16 aaReal;       // [x, y, z]        gravity-free accel sensor measurements
VectorInt16 aaWorld;      // [x, y, z]        world-frame accel sensor measurements
VectorFloat gravity;     // [x, y, z]        gravity vector
float euler[3];          // [psi, theta, phi] Euler angle container
float ypr[3];            // [yaw, pitch, roll] yaw/pitch/roll container and gravity vector

volatile bool mpuInterrupt = false; // indicates whether MPU interrupt pin has gone high
void dmpDataReady() {
| mpuInterrupt = true;
}
// Enable pins for controlling the speed of the motor(enX).
#define en1 2
#define en2 3
#define en3 4
#define en4 5
#define en5 6
#define en6 7
// Input pins of the motor(inX and inY).
// Motor 1
#define in1a 30
#define in1b 32
// Motor 2
#define in2a 34
#define in2b 36
// Motor 3
#define in3a 22
#define in3b 24
// Motor 4
#define in4a 26
#define in4b 28
// Motor 5
#define in5a 23
#define in5b 25
// Motor 6
#define in6a 27
#define in6b 29
// Create a Servo object for each servo
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;
// Define the servo PWM pin numbers
int servoPin1 = 8;
int servoPin2 = 9;
int servoPin3 = 10;
int servoPin4 = 11;
//Initial servo motor angle in degrees
int se1Angle = 80;
int se2Angle = 88;
int se3Angle = 97;
int se4Angle = 87;
// s is the required speed in percentage 0 to 100%
// s=68 is the maximum while turning
int s = 0;
float speed1, speed2 = 0;
float speed1PWM, speed2PWM = 0;
// alpha is the heading angle, alphai and alphaz represent the inner and outer wheel servos
float alpha, alphai, alphaz = 0;
float alpha_rad;
// On point turning angle calculated from equation 9

float alpha_on_point = 43.73;
// Inputs received from the keyboard
char Inp;
float wheelbase = 460.0;
float trackwidth = 440.0;
float turning_radius;

unsigned long lastCommandTime = 0;           // Stores the last time a command was received
const unsigned long commandTimeout = 500; // Timeout in milliseconds

void setup() {
| // join I2C bus (I2Cdev library doesn't do this automatically)
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
| Wire.begin();
| Wire.setClock(400000); // 400kHz I2C clock.
#elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
| Fastwire::setup(400, true);
#endif

| // put your setup code here, to run once:
Serial.begin(9600);
BT.begin(9600);

| // initialize device
Serial.println(F("Initializing I2C devices..."));
mpu.initialize();
pinMode(IMU_INTERRUPT_PIN, INPUT);

| // verify connection
Serial.println(F("Testing device connections..."));
Serial.println(mpu.testConnection() ? F("MPU6050 connection successful") : F("MPU6050 connection failed"));

| // load and configure the DMP
}

```

```

Serial.println(F("Initializing DMP..."));
devStatus = mpu.dmpInitialize();

// gyro offsets
mpu.setGyroOffset(500);
mpu.setGyroOffset(0.27);
mpu.setGyroOffset(3.18);
mpu.setAccelOffset(1788);

if (devStatus == 0) {
    // Calibration time
    mpu.CalibrateAccel(6);
    mpu.CalibrateGyro(6);
    mpu.PrintActiveOffsets();
    Serial.println(F("Enabling DMP..."));
    mpu.setDMPEnabled(true);

    // Enable Arduino DUE interrupt detection
    Serial.print(F("Enabling interrupt detection (Arduino external interrupt "));
    Serial.print(digitalPinToInterrupt(IMU_INTERRUPT_PIN));
    Serial.println(F(")..."));

    attachInterrupt(digitalPinToInterrupt(IMU_INTERRUPT_PIN), dmpDataReady, RISING);
    mpuIntStatus = mpu.getIntStatus();
    Serial.println(F("DMP ready! Waiting for first interrupt..."));

    dmpReady = true;
    packetSize = mpu.dmpGetFIFOPacketSize();
} else {
    Serial.print(F("DMP Initialization failed (code "));
    Serial.print(devStatus);
    Serial.println(F(")"));
}

// Initialize the motor encoder
pinMode(ENCODER1_INTERRUPT_PIN, INPUT);
pinMode(ENCODER2_INTERRUPT_PIN, INPUT);

pinMode(ENCODER2_INTERRUPT_PIN, INPUT);
attachInterrupt(digitalPinToInterrupt(ENCODER1_INTERRUPT_PIN), ISR_1, RISING);
attachInterrupt(digitalPinToInterrupt(ENCODER2_INTERRUPT_PIN), ISR_2, RISING);

// Initialize the pins as either INPUT or OUTPUT.
pinMode(en1, OUTPUT);
pinMode(en2, OUTPUT);
pinMode(en3, OUTPUT);
pinMode(en4, OUTPUT);
pinMode(en5, OUTPUT);
pinMode(en6, OUTPUT);

pinMode(in1a, OUTPUT);
pinMode(in1b, OUTPUT);
pinMode(in2a, OUTPUT);
pinMode(in2b, OUTPUT);
pinMode(in3a, OUTPUT);
pinMode(in3b, OUTPUT);
pinMode(in4a, OUTPUT);
pinMode(in4b, OUTPUT);
pinMode(in5a, OUTPUT);
pinMode(in5b, OUTPUT);
pinMode(in6a, OUTPUT);
pinMode(in6b, OUTPUT);

digitalWrite(in1a, LOW);
digitalWrite(in1b, LOW);
digitalWrite(in2a, LOW);
digitalWrite(in2b, LOW);
digitalWrite(in3a, LOW);
digitalWrite(in3b, LOW);
digitalWrite(in4a, LOW);

digitalWrite(in4b, LOW);
digitalWrite(in5a, LOW);
digitalWrite(in5b, LOW);
digitalWrite(in6a, LOW);
digitalWrite(in6b, LOW);

// Attach each servo to its corresponding pin
servo1.attach(servoPin1);
servo2.attach(servoPin2);
servo3.attach(servoPin3);
servo4.attach(servoPin4);

// Set the initial angles to move straight
servo1.write(se1Angle);
servo2.write(se2Angle);
servo3.write(se3Angle);
servo4.write(se4Angle);

// configure LED for output
pinMode(LED_PIN, OUTPUT);
}

void loop() {

    Motor1_RPM = (60000000) / (Ta * ENCODER_N);
    Motor2_RPM = (60000000) / (Tb * ENCODER_N);

    // read a packet from FIFO
    if (mpu.dmpGetCurrentFIFOPacket(fifoBuffer)) { // Get the latest packet

        // display Euler angles in degrees
        mpu.dmpGetQuaternion(&q, fifoBuffer);
        mpu.dmpGetGravity(&gravity, &q);
        mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
        Serial.print(" YAW:\t");
    }
}

```

```

Serial.print(ypr[0] * 180 / M_PI);
Serial.print(" ROL:\t");
Serial.print(ypr[2] * 180 / M_PI);
Serial.print(" PTH:\t");
Serial.println(ypr[1] * 180 / M_PI);
// blink LED to indicate activity
blinkState = !blinkState;
digitalWrite(LED_PIN, blinkState);
}

// Print the RPM data and move the cursor to the begining of the serial monitor
Serial.print(" RPM1:\t");
Serial.print(Motor1_RPM);
Serial.print(" RPM2:\t");
Serial.println(Motor2_RPM);

// Control logic for moving the rover
if (BT.available()) {
    inp = BT.read();
    lastCommandTime = millis();
    switch (inp) {
        case 'w': // Move forward
        case 'W':
            motorSpeed();
            moveForward(speed1PWM, speed2PWM);
            break;
        case 's': // Move back
        case 'S':
            motorSpeed();
            moveBackward(speed1PWM, speed2PWM);
            break;
        case 'd': // Turn right
        case 'D':
            alpha += 1; // Increase alpha for right turn
            if (alpha > 22) alpha = 22; // Limit max turn angle
            moveLeft();
            break;
        case 'a': // Turn left
        case 'A':
            alpha -= 1; // Decrease alpha for left turn
            if (alpha < -22) alpha = -22; // Limit max turn angle
            moveRight();
            break;
        case 'r': // Rotate on point
        case 'R':
            motorSpeed();
            rotateOnPoint(speed1PWM, speed2PWM);
            break;
        case 'x': // Stop all movement
        case 'X':
            stopMovement();
            break;
    }
} // Check for command timeout to stop the rover if no command is received
if (millis() - lastCommandTime > commandTimeout) {
    stopMovement();
}
}

void moveForward(float speedPWM1, float speedPWM2) // Forward
{
analogWrite(en1, speedPWM1);
digitalWrite(in1a, HIGH);
digitalWrite(in1b, LOW);
analogWrite(en2, speedPWM2);

digitalWrite(in2b, LOW);
analogWrite(en3, speedPWM1);
digitalWrite(in3a, LOW);
digitalWrite(in3b, HIGH);
analogWrite(en4, speedPWM2);
digitalWrite(in4a, LOW);
digitalWrite(in4b, HIGH);
analogWrite(en5, speedPWM1);
digitalWrite(in5a, HIGH);
digitalWrite(in5b, LOW);
analogWrite(en6, speedPWM2);
digitalWrite(in6a, HIGH);
digitalWrite(in6b, LOW);
}

void moveBackward(float speedPWM1, float speedPWM2) // Backward
{
analogWrite(en1, speedPWM1);
digitalWrite(in1a, LOW);
digitalWrite(in1b, HIGH);
analogWrite(en2, speedPWM2);
digitalWrite(in2a, LOW);
digitalWrite(in2b, HIGH);
analogWrite(en3, speedPWM1);
digitalWrite(in3a, HIGH);
digitalWrite(in3b, LOW);
analogWrite(en4, speedPWM2);
digitalWrite(in4a, HIGH);
digitalWrite(in4b, LOW);
analogWrite(en5, speedPWM1);
digitalWrite(in5a, LOW);
}

```

```

digitalWrite(in5b, HIGH);
analogWrite(en6, speedPWM2);
digitalWrite(in6a, LOW);
digitalWrite(in6b, HIGH);
}

void moveRight() {
    servoAngle(); // Update alpha1 and alpha2 based on current alpha

    // Adjust servo angles for turning right
    servo1.write(se1Angle + alpha1);
    servo2.write(se2Angle + alpha2);
    servo3.write(se3Angle - alpha1);
    servo4.write(se4Angle - alpha2);
}

void moveLeft() {
    servoAngle(); // Update alpha1 and alpha2 based on current alpha

    // Adjust servo angles for turning left
    servo1.write(se1Angle + alpha1);
    servo2.write(se2Angle + alpha2);
    servo3.write(se3Angle - alpha1);
    servo4.write(se4Angle - alpha2);
}

void rotateOnPoint(float speedPWM1, float speedPWM2) {
    // Adjust servo angles for turning on point
    servo1.write(se1Angle + alpha_on_point);
    servo2.write(se2Angle - alpha_on_point);
    servo3.write(se3Angle - alpha_on_point);
    servo4.write(se4Angle + alpha_on_point);

    analogWrite(en1, speedPWM1);
    digitalWrite(in1a, HIGH);
    digitalWrite(in1b, LOW);
    analogWrite(en2, speedPWM2);
    digitalWrite(in2a, LOW);
    digitalWrite(in2b, HIGH);
    analogWrite(en3, speedPWM1);
    digitalWrite(in3a, LOW);
    digitalWrite(in3b, HIGH);
    analogWrite(en4, speedPWM2);
    digitalWrite(in4a, HIGH);
    digitalWrite(in4b, LOW);
    analogWrite(en5, speedPWM1);
    digitalWrite(in5a, HIGH);
    digitalWrite(in5b, LOW);
    analogWrite(en6, speedPWM2);
    digitalWrite(in6a, LOW);
    digitalWrite(in6b, HIGH);
}

void stopMovement() {
    analogWrite(en1, 0);
    digitalWrite(in1a, LOW);
    digitalWrite(in1b, LOW);
    analogWrite(en2, 0);
    digitalWrite(in2a, LOW);
    digitalWrite(in2b, LOW);
    analogWrite(en3, 0);
    digitalWrite(in3a, LOW);
    digitalWrite(in3b, LOW);
    analogWrite(en4, 0);
    digitalWrite(in4a, LOW);

    digitalWrite(in4b, LOW);
    analogWrite(en5, 0);
    digitalWrite(in5a, LOW);
    digitalWrite(in5b, LOW);
    analogWrite(en6, 0);
    digitalWrite(in6a, LOW);
    digitalWrite(in6b, LOW);
}

void motorSpeed() {

    if (alpha > 0.0) { // Turning right
        s = 30; // speed set to 30%
        turning_radius = (wheelbase / (2 * tan(abs(alpha) * PI / 180)));

        speed1 = s * sqrt(pow((0.5 * wheelbase), 2) + pow((0.5 * trackwidth - turning_radius), 2)) / turning_radius;
        speed2 = s * sqrt(pow((0.5 * wheelbase), 2) + pow((0.5 * trackwidth + turning_radius), 2)) / turning_radius;

        speed1PWM = map(round(speed1), 0, 100, 0, 255);
        speed2PWM = map(round(speed2), 0, 100, 0, 255);
    }

    else if (alpha < 0.0) { // Turning left
        s = 30; // speed set to 30%
        turning_radius = (wheelbase / (2 * tan(abs(alpha) * PI / 180)));

        speed1 = s * sqrt(pow((0.5 * wheelbase), 2) + pow((0.5 * trackwidth - turning_radius), 2)) / turning_radius;
        speed2 = s * sqrt(pow((0.5 * wheelbase), 2) + pow((0.5 * trackwidth + turning_radius), 2)) / turning_radius;
    }
}

```

```

speed1PWM = map(round(speed2), 0, 100, 0, 255);
speed2PWM = map(round(speed1), 0, 100, 0, 255);
}

else { // Moving forward and back
s = 30; // speed set to 30%
speed1 = speed2 = s;
speed1PWM = map(round(s), 0, 100, 0, 255);
speed2PWM = map(round(s), 0, 100, 0, 255);
}

void servoAngle() {

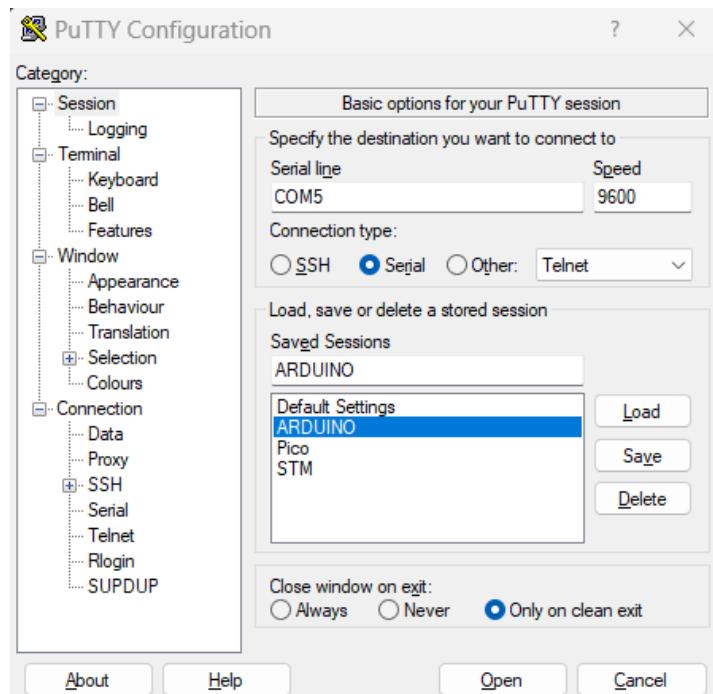
alpha_rad = alpha * PI / 180; // Convert degrees to radians
//calculate the servo angles in degrees
alpha1 = round(atan(wheelbase / ((wheelbase / tan(alpha_rad)) + trackwidth) * 180 / PI);
alpha2 = round(atan(wheelbase / ((wheelbase / tan(alpha_rad)) - trackwidth) * 180 / PI);

}

// Motor encoder interrupt functions
void ISR_1(void) {
if (MeasDone1) {
T2 = micros();
Ta = T2 - T1;
MeasDone1 = 0;
} else {
T1 = micros();
MeasDone1 = 1;
}
}

void ISR_2(void) {
if (MeasDone2) {
T4 = micros();
Tb = T4 - T3;
MeasDone2 = 0;
} else {
T3 = micros();
MeasDone2 = 1;
}
}
}

```



Putty terminal configuration to connect the HC-06 through Bluetooth.

## Appendix C – Test data and general information on parts used to build the rover.

### TEST 1:

| TEST 1 - Straight line test |    |       |       |          |                  |    |     |       |         |          |                  |     |        |       | WHEELBASE (mm)  | 440      |                  |                 |     |
|-----------------------------|----|-------|-------|----------|------------------|----|-----|-------|---------|----------|------------------|-----|--------|-------|-----------------|----------|------------------|-----------------|-----|
| NO                          | S1 | RPM 1 | RPM 2 | TIME (s) | PATH LENGTH (mm) | NO | S2  | RPM 1 | RPM 2   | TIME (s) | PATH LENGTH (mm) | NO  | S3     | RPM 1 | RPM 2           | TIME (s) | PATH LENGTH (mm) | TRACKWIDTH (mm) | 460 |
| 1                           | 30 | 58    | 58    | 11.21    | 2000 1           | 50 | 130 | 5.21  | 2000 1  | 68       | 185              | 185 | 4.23   | 2000  | COG Z-axis (mm) | 105      |                  |                 |     |
| 2                           | 30 | 58    | 58    | 11.21    | 2000 2           | 50 | 133 | 5.16  | 2000 2  | 68       | 185              | 185 | 4.23   | 2000  | WEIGHT (kg)     | 2.45     |                  |                 |     |
| 3                           | 30 | 58    | 58    | 11.25    | 2000 3           | 50 | 133 | 5.14  | 2000 3  | 68       | 185              | 185 | 4.1    | 2000  | MIN RPM         | 58       |                  |                 |     |
| 4                           | 30 | 58    | 58    | 11.23    | 2000 4           | 50 | 133 | 5.14  | 2000 4  | 68       | 186              | 186 | 4.3    | 2000  | MAX RPM         | 186      |                  |                 |     |
| 5                           | 30 | 58    | 58    | 11.29    | 2000 5           | 50 | 130 | 5.35  | 2000 5  | 68       | 185              | 185 | 4.28   | 2000  | S1/S2/S3        |          |                  |                 |     |
| 6                           | 30 | 57    | 57    | 11.3     | 2000 6           | 50 | 132 | 5.28  | 2000 6  | 68       | 185              | 185 | 4.09   | 2000  | Speed in %      |          |                  |                 |     |
| 7                           | 30 | 58    | 58    | 11.23    | 2000 7           | 50 | 132 | 5.21  | 2000 7  | 68       | 185              | 185 | 4.09   | 2000  | Inner wheels    |          |                  |                 |     |
| 8                           | 30 | 58    | 58    | 11.21    | 2000 8           | 50 | 132 | 5.22  | 2000 8  | 68       | 185              | 185 | 4.1    | 2000  | Outer wheels    |          |                  |                 |     |
| 9                           | 30 | 58    | 58    | 11.2     | 2000 9           | 50 | 130 | 5.38  | 2000 9  | 68       | 187              | 187 | 3.97   | 2000  | S max           | 68       |                  |                 |     |
| 10                          | 30 | 60    | 60    | 11.06    | 2000 10          | 50 | 131 | 5.35  | 2000 10 | 68       | 185              | 185 | 4.1    | 2000  | S min           | 30       |                  |                 |     |
| 11                          | 30 | 59    | 59    | 11.1     | 2000 11          | 50 | 132 | 5.22  | 2000 11 | 68       | 185              | 185 | 4.03   | 2000  |                 |          |                  |                 |     |
| 12                          | 30 | 59    | 59    | 11.14    | 2000 12          | 50 | 132 | 5.27  | 2000 12 | 68       | 185              | 185 | 4.15   | 2000  |                 |          |                  |                 |     |
| 13                          | 30 | 59    | 59    | 11.1     | 2000 13          | 50 | 132 | 5.22  | 2000 13 | 68       | 185              | 185 | 4.16   | 2000  |                 |          |                  |                 |     |
| 14                          | 30 | 58    | 58    | 11.28    | 2000 14          | 50 | 132 | 5.22  | 2000 14 | 68       | 185              | 185 | 4.15   | 2000  |                 |          |                  |                 |     |
| 15                          | 30 | 58    | 58    | 11.45    | 2000 15          | 50 | 132 | 5.21  | 2000 15 | 68       | 185              | 185 | 4.03   | 2000  |                 |          |                  |                 |     |
| 16                          | 30 | 59    | 59    | 11.13    | 2000 16          | 50 | 133 | 5.15  | 2000 16 | 68       | 185              | 185 | 4.1    | 2000  |                 |          |                  |                 |     |
| 17                          | 30 | 60    | 60    | 11.02    | 2000 17          | 50 | 133 | 5.14  | 2000 17 | 68       | 185              | 185 | 4.12   | 2000  |                 |          |                  |                 |     |
| 18                          | 30 | 59    | 59    | 11.1     | 2000 18          | 50 | 132 | 5.2   | 2000 18 | 68       | 185              | 185 | 4.2    | 2000  |                 |          |                  |                 |     |
| 19                          | 30 | 58    | 58    | 11.2     | 2000 19          | 50 | 132 | 5.22  | 2000 19 | 68       | 185              | 185 | 4.16   | 2000  |                 |          |                  |                 |     |
| 20                          | 30 | 59    | 59    | 11.1     | 2000 20          | 50 | 132 | 5.22  | 2000 20 | 68       | 185              | 185 | 4.15   | 2000  |                 |          |                  |                 |     |
|                             |    |       |       | 58.45    | 11.1905          |    |     | 131.9 | 5.2255  |          |                  |     | 185.15 |       |                 | 4.137    |                  |                 |     |

Figure 7.14: Straight line performance.

### TEST 2:

| TEST 2 - Minimum turning radius |    |       |       |          |                |         |    |       |       |          |                |    |     |       | WHEELBASE (mm) | 440                  |                |                 |     |
|---------------------------------|----|-------|-------|----------|----------------|---------|----|-------|-------|----------|----------------|----|-----|-------|----------------|----------------------|----------------|-----------------|-----|
| NO                              | S1 | RPM 1 | RPM 2 | TIME (s) | MIN RADIUS(mm) | NO      | S2 | RPM 1 | RPM 2 | TIME (s) | MIN RADIUS(mm) | NO | S3  | RPM 1 | RPM 2          | TIME (s)             | MIN RADIUS(mm) | TRACKWIDTH (mm) | 460 |
| 1                               | 30 | 21    | 43    | 11.84    | 544 1          | 50      | 35 | 77    | 6.67  | 544 1    | 68             | 48 | 105 | 5.78  | 544            | COG Z-axis (mm)      | 105            |                 |     |
| 2                               | 30 | 22    | 44    | 11.87    | 544 2          | 50      | 35 | 77    | 6.9   | 544 2    | 68             | 48 | 105 | 5.76  | 544            | WEIGHT (kg)          | 2.45           |                 |     |
| 3                               | 30 | 21    | 43    | 11.81    | 544 3          | 50      | 36 | 78    | 6.96  | 544 3    | 68             | 48 | 105 | 5.68  | 544            | MIN RPM              | 21             |                 |     |
| 4                               | 30 | 21    | 42    | 11.68    | 544 4          | 50      | 35 | 77    | 7.04  | 544 4    | 68             | 48 | 105 | 5.71  | 544            | MAX RPM              | 105            |                 |     |
| 5                               | 30 | 21    | 42    | 11.66    | 544 5          | 50      | 35 | 77    | 6.94  | 544 5    | 68             | 48 | 105 | 5.61  | 544            | S1/S2/S3             |                |                 |     |
| 6                               | 30 | 21    | 42    | 11.61    | 544 6          | 50      | 36 | 78    | 6.61  | 544 6    | 68             | 48 | 105 | 5.64  | 544            | Speed in %           |                |                 |     |
| 7                               | 30 | 21    | 42    | 11.58    | 544 7          | 50      | 36 | 78    | 6.61  | 544 7    | 68             | 48 | 105 | 5.47  | 544            | Inner wheels         |                |                 |     |
| 8                               | 30 | 21    | 42    | 11.65    | 544 8          | 50      | 35 | 77    | 6.66  | 544 8    | 68             | 48 | 105 | 5.6   | 544            | Outer wheels         |                |                 |     |
| 9                               | 30 | 21    | 42    | 11.64    | 544 9          | 50      | 36 | 78    | 6.53  | 544 9    | 68             | 48 | 105 | 5.63  | 544            | S max                | 68             |                 |     |
| 10                              | 30 | 21    | 42    | 11.64    | 544 10         | 50      | 35 | 77    | 6.49  | 544 10   | 68             | 49 | 106 | 5.59  | 544            | S min                | 30             |                 |     |
| 11                              | 30 | 21    | 42    | 11.69    | 544 11         | 50      | 36 | 78    | 6.64  | 544 11   | 68             | 49 | 106 | 5.42  | 544            | Expected Radius (mm) | 544.52         |                 |     |
| 12                              | 30 | 21    | 42    | 11.67    | 544 12         | 50      | 35 | 77    | 6.86  | 544 12   | 68             | 49 | 106 | 5.56  | 544            |                      |                |                 |     |
| 13                              | 30 | 21    | 42    | 11.56    | 544 13         | 50      | 35 | 77    | 6.77  | 544 13   | 68             | 49 | 106 | 5.47  | 544            |                      |                |                 |     |
| 14                              | 30 | 21    | 42    | 11.63    | 544 14         | 50      | 35 | 77    | 6.68  | 544 14   | 68             | 48 | 105 | 5.64  | 544            |                      |                |                 |     |
| 15                              | 30 | 21    | 42    | 11.65    | 544 15         | 50      | 36 | 78    | 6.63  | 544 15   | 68             | 48 | 105 | 5.6   | 544            |                      |                |                 |     |
| 16                              | 30 | 21    | 42    | 11.66    | 544 16         | 50      | 36 | 78    | 6.65  | 544 16   | 68             | 49 | 106 | 5.35  | 544            |                      |                |                 |     |
| 17                              | 30 | 21    | 42    | 11.56    | 544 17         | 50      | 37 | 79    | 6.46  | 544 17   | 68             | 49 | 106 | 5.52  | 544            |                      |                |                 |     |
| 18                              | 30 | 21    | 42    | 11.68    | 544 18         | 50      | 35 | 77    | 6.6   | 544 18   | 68             | 48 | 105 | 5.58  | 544            |                      |                |                 |     |
| 19                              | 30 | 21    | 43    | 11.7     | 544 19         | 50      | 35 | 77    | 6.69  | 544 19   | 68             | 49 | 106 | 5.46  | 544            |                      |                |                 |     |
| 20                              | 30 | 21    | 43    | 11.71    | 544 20         | 50      | 37 | 79    | 6.49  | 544 20   | 68             | 49 | 106 | 5.45  | 544            |                      |                |                 |     |
|                                 |    |       |       | 21.05    | 42.3           | 11.6745 |    |       | 35.55 | 77.55    | 6.694          |    |     |       | 48.4           | 105.4                | 5.576          |                 |     |

Note: For S= 30, the Center Of Radius shifted by 125mm towards the left for turning right after the 20 trials continuously i.e. 6.25mm every 360 degrees of rotation. However the Min radius remained the same.

Note: For S = 50, the Center Of Radius shifted by 170mm towards the left for turning right after the 20 trials continuously i.e. 8.5mm every 360 degrees of rotation. However the Min radius remained the same.

Note: For S = 68, the Center Of Radius shifted by 305mm towards the left for turning right after the 20 trials continuously i.e. 15.25mm every 360 degrees of rotation. However the Min radius remained the same.

### TEST 3:

| TEST 3 - Turning on the spot |    |       |       |             |          |    |    |       |       |             |          |    |    |       | WHEELBASE (mm) | 440      |             |                 |     |
|------------------------------|----|-------|-------|-------------|----------|----|----|-------|-------|-------------|----------|----|----|-------|----------------|----------|-------------|-----------------|-----|
| NO                           | S1 | RPM 1 | RPM 2 | ANGLE (deg) | TIME (s) | NO | S2 | RPM 1 | RPM 2 | ANGLE (deg) | TIME (s) | NO | S3 | RPM 1 | RPM 2          | TIME (s) | ANGLE (deg) | TRACKWIDTH (mm) | 460 |
| 1                            | 30 | 23    | 23    | 43.73       | 15.24 1  | 50 | 89 | 89    | 89    | 43.73       | 7.41     |    |    |       |                |          |             |                 |     |
| 2                            | 30 | 24    | 24    | 43.73       | 15.27 2  | 50 | 88 | 88    | 88    | 43.73       | 7.32     |    |    |       |                |          |             |                 |     |
| 3                            | 30 | 23    | 23    | 43.73       | 15.6 3   | 50 | 89 | 89    | 89    | 43.73       | 7.43     |    |    |       |                |          |             |                 |     |
| 4                            | 30 | 23    | 23    | 43.73       | 15.68 4  | 50 | 89 | 89    | 89    | 43.73       | 7.6      |    |    |       |                |          |             |                 |     |
| 5                            | 30 | 24    | 24    | 43.73       | 15.82 5  | 50 | 89 | 89    | 89    | 43.73       | 7.46     |    |    |       |                |          |             |                 |     |
| 6                            | 30 | 23    | 23    | 43.73       | 15.21 6  | 50 | 89 | 89    | 89    | 43.73       | 7.52     |    |    |       |                |          |             |                 |     |
| 7                            | 30 | 23    | 23    | 43.73       | 15.53 7  | 50 | 89 | 89    | 89    | 43.73       | 7.44     |    |    |       |                |          |             |                 |     |
| 8                            | 30 | 23    | 23    | 43.73       | 15.79 8  | 50 | 88 | 88    | 88    | 43.73       | 7.42     |    |    |       |                |          |             |                 |     |
| 9                            | 30 | 23    | 23    | 43.73       | 15.35 9  | 50 | 89 | 89    | 89    | 43.73       | 7.43     |    |    |       |                |          |             |                 |     |
| 10                           | 30 | 23    | 23    | 43.73       | 15.24 10 | 50 | 89 | 89    | 89    | 43.73       | 7.61     |    |    |       |                |          |             |                 |     |
| 11                           | 30 | 24    | 24    | 43.73       | 15.5 11  | 50 | 89 | 89    | 89    | 43.73       | 7.41     |    |    |       |                |          |             |                 |     |
| 12                           | 30 | 23    | 23    | 43.73       | 15.54 12 | 50 | 89 | 89    | 89    | 43.73       | 7.43     |    |    |       |                |          |             |                 |     |
| 13                           | 30 | 23    | 23    | 43.73       | 15.3 13  | 50 | 89 | 89    | 89    | 43.73       | 7.47     |    |    |       |                |          |             |                 |     |
| 14                           | 30 | 23    | 23    | 43.73       | 15.35 14 | 50 | 89 | 89    | 89    | 43.73       | 7.48     |    |    |       |                |          |             |                 |     |
| 15                           | 30 | 23    | 23    | 43.73       | 15.33 15 | 50 | 88 | 88    | 88    | 43.73       | 7.31     |    |    |       |                |          |             |                 |     |
| 16                           | 30 | 23    | 23    | 43.73       | 15.46 16 | 50 | 88 | 88    | 88    | 43.73       | 7.43     |    |    |       |                |          |             |                 |     |
| 17                           | 30 | 24    | 24    | 43.73       | 15.35 17 | 50 | 89 | 89    | 89    | 43.73       | 7.44     |    |    |       |                |          |             |                 |     |
| 18                           | 30 | 23    | 23    | 43.73       | 15.61 18 | 50 | 89 | 89    | 89    | 43.73       | 7.45     |    |    |       |                |          |             |                 |     |
| 19                           | 30 | 23    | 23    |             |          |    |    |       |       |             |          |    |    |       |                |          |             |                 |     |

**TEST 4:**

| TEST 4 - Driving over obstacles |    |          |                  |    |    |          |                  |    |    | WHEELBASE (mm) | 440              |                 |            |
|---------------------------------|----|----------|------------------|----|----|----------|------------------|----|----|----------------|------------------|-----------------|------------|
| NO                              | S1 | TIME (s) | PATH LENGTH (mm) | NO | S2 | TIME (s) | PATH LENGTH (mm) | NO | S3 | TIME (s)       | PATH LENGTH (mm) | TRACKWIDTH (mm) | 460        |
| 1                               | 30 | -        | 2400             | 1  | 50 | 6.44     | 2400             | 1  | 68 | 4.39           | 2400             | COG Z-axis (mm) | 105        |
| 2                               | 30 | -        | 2400             | 2  | 50 | 6.56     | 2400             | 2  | 68 | 4.45           | 2400             | WEIGHT (kg)     | 2.45       |
| 3                               | 30 | -        | 2400             | 3  | 50 | 6.48     | 2400             | 3  | 68 | 4.36           | 2400             | S1              | Speed in % |
| 4                               | 30 | -        | 2400             | 4  | 50 | 6.51     | 2400             | 4  | 68 | 4.34           | 2400             |                 |            |
| 5                               | 30 | -        | 2400             | 5  | 50 | 6.65     | 2400             | 5  | 68 | 4.24           | 2400             |                 |            |
| 6                               | 30 | -        | 2400             | 6  | 50 | 6.94     | 2400             | 6  | 68 | 4.96           | 2400             |                 |            |
| 7                               | 30 | -        | 2400             | 7  | 50 | 6.47     | 2400             | 7  | 68 | 4.99           | 2400             |                 |            |
| 8                               | 30 | -        | 2400             | 8  | 50 | 6.27     | 2400             | 8  | 68 | 4.59           | 2400             |                 |            |
| 9                               | 30 | -        | 2400             | 9  | 50 | 6.69     | 2400             | 9  | 68 | 4.19           | 2400             |                 |            |
| 10                              | 30 | -        | 2400             | 10 | 50 | 6.25     | 2400             | 10 | 68 | 4.08           | 2400             |                 |            |
| 11                              | 30 | -        | 2400             | 11 | 50 | 6.84     | 2400             | 11 | 68 | 4.61           | 2400             |                 |            |
| 12                              | 30 | -        | 2400             | 12 | 50 | 6.48     | 2400             | 12 | 68 | 4.9            | 2400             |                 |            |
| 13                              | 30 | -        | 2400             | 13 | 50 | 6.41     | 2400             | 13 | 68 | 4.07           | 2400             |                 |            |
| 14                              | 30 | -        | 2400             | 14 | 50 | 6.6      | 2400             | 14 | 68 | 4.33           | 2400             |                 |            |
| 15                              | 30 | -        | 2400             | 15 | 50 | 6.42     | 2400             | 15 | 68 | 4.97           | 2400             |                 |            |
| 16                              | 30 | -        | 2400             | 16 | 50 | 6.21     | 2400             | 16 | 68 | 4.98           | 2400             |                 |            |
| 17                              | 30 | -        | 2400             | 17 | 50 | 6.97     | 2400             | 17 | 68 | 4.11           | 2400             |                 |            |
| 18                              | 30 | -        | 2400             | 18 | 50 | 6.07     | 2400             | 18 | 68 | 4.06           | 2400             |                 |            |
| 19                              | 30 | -        | 2400             | 19 | 50 | 6.88     | 2400             | 19 | 68 | 4.8            | 2400             |                 |            |
| 20                              | 30 | -        | 2400             | 20 | 50 | 6.73     | 2400             | 20 | 68 | 4.08           | 2400             |                 |            |
|                                 |    |          |                  |    |    | 6.5435   |                  |    |    | 4.475          |                  |                 |            |

Figure 7.17: Driving over obstacles test.

**TEST 5:**

| NO | INC 1 LENGTH [mm] | INC 2 LENGTH [mm] | INC 3 LENGTH [mm] | INC 1 SLIP LENGTH [mm] | INC 2 SLIP LENGTH [mm] | INC 3 SLIP LENGTH [mm] | NO   | INC 1 LENGTH [mm] | INC 2 LENGTH [mm] | INC 3 LENGTH [mm] | INC 4 LENGTH [mm] | INC 1 SLIP LENGTH [mm] |  |
|----|-------------------|-------------------|-------------------|------------------------|------------------------|------------------------|------|-------------------|-------------------|-------------------|-------------------|------------------------|--|
| 1  | 50                | 1470              | 900               | 520                    | 1                      | 68                     | 1470 | 500               | 1                 | 100               | 1470              | 500                    |  |
| 2  | 50                | 1470              | 900               | 520                    | 2                      | 68                     | 1470 | 500               | 2                 | 100               | 1470              | 500                    |  |
| 3  | 50                | 1470              | 900               | 520                    | 3                      | 68                     | 1470 | 500               | 3                 | 100               | 1470              | 500                    |  |
| 4  | 50                | 1470              | 900               | 520                    | 4                      | 68                     | 1470 | 500               | 4                 | 100               | 1470              | 500                    |  |
| 5  | 50                | 1470              | 900               | 520                    | 5                      | 68                     | 1470 | 500               | 5                 | 100               | 1470              | 500                    |  |
| 6  | 50                | 1470              | 900               | 520                    | 6                      | 68                     | 1470 | 500               | 6                 | 100               | 1470              | 500                    |  |
| 7  | 50                | 1470              | 900               | 520                    | 7                      | 68                     | 1470 | 500               | 7                 | 100               | 1470              | 500                    |  |
| 8  | 50                | 1470              | 900               | 520                    | 8                      | 68                     | 1470 | 500               | 8                 | 100               | 1470              | 500                    |  |
| 9  | 50                | 1470              | 900               | 520                    | 9                      | 68                     | 1470 | 500               | 9                 | 100               | 1470              | 500                    |  |
| 10 | 50                | 1470              | 900               | 520                    | 10                     | 68                     | 1470 | 500               | 10                | 100               | 1470              | 500                    |  |
| 11 | 50                | 1470              | 900               | 520                    | 11                     | 68                     | 1470 | 500               | 11                | 100               | 1470              | 500                    |  |
| 12 | 50                | 1470              | 900               | 520                    | 12                     | 68                     | 1470 | 500               | 12                | 100               | 1470              | 500                    |  |
| 13 | 50                | 1470              | 900               | 520                    | 13                     | 68                     | 1470 | 500               | 13                | 100               | 1470              | 500                    |  |
| 14 | 50                | 1470              | 900               | 520                    | 14                     | 68                     | 1470 | 500               | 14                | 100               | 1470              | 500                    |  |
| 15 | 50                | 1470              | 900               | 520                    | 15                     | 68                     | 1470 | 500               | 15                | 100               | 1470              | 500                    |  |
| 16 | 50                | 1470              | 900               | 520                    | 16                     | 68                     | 1470 | 500               | 16                | 100               | 1470              | 500                    |  |
| 17 | 50                | 1470              | 900               | 520                    | 17                     | 68                     | 1470 | 500               | 17                | 100               | 1470              | 500                    |  |
| 18 | 50                | 1470              | 900               | 520                    | 18                     | 68                     | 1470 | 500               | 18                | 100               | 1470              | 500                    |  |
| 19 | 50                | 1470              | 900               | 520                    | 19                     | 68                     | 1470 | 500               | 19                | 100               | 1470              | 500                    |  |
| 20 | 50                | 1470              | 900               | 520                    | 20                     | 68                     | 1470 | 500               | 20                | 100               | 1470              | 500                    |  |

Note: At 50°, the rover fails to climb at an inclination greater than 30deg and fails at 20deg, and 0.53m at 30deg.  
Note: At 50°, the rover fails to climb at an angle of 0.65m at an angle of 30deg and at 0.5m at an angle of 40deg. At all other lower inclinations, the rover completes the climb over the entire plywood sheet.  
Note: At 50°, the rover fails to climb at an angle of 0.65m at an angle of 30deg and at 0.5m at an angle of 40deg. At all other lower inclinations, the rover completes the climb over the entire plywood sheet.

Figure 7.18: Climbing a ramp to find maximum inclination angle.

**TEST 6:**

| NO | S1 | IMU ROL ANGLE (deg) | IMU ROL ANGLE (deg) | IMU ROL ANGLE (deg) | PATH LENGTH (mm) | INCLINATIONS (deg) | Ideal | Recorded |
|----|----|---------------------|---------------------|---------------------|------------------|--------------------|-------|----------|
| 1  | 30 | 9.6                 | 18.37               | 23                  | 1470             | INC 1              | 10    | 9.6      |
| 2  | 30 | 9.6                 | 18.37               | 23                  | 1470             | INC 2              | 20    | 18.37    |
| 3  | 30 | 9.6                 | 18.37               | 23                  | 1470             | INC 3              | 30    | 23       |
| 4  | 30 | 9.6                 | 18.37               | 23                  | 1470             | INC 4              | 40    |          |
| 5  | 30 | 9.6                 | 18.37               | 23                  | 1470             | INC 5              | 50    |          |
| 6  | 30 | 9.6                 | 18.37               | 23                  | 1470             | S max              | 68    |          |
| 7  | 30 | 9.6                 | 18.37               | 23                  | 1470             | S min              |       |          |
| 8  | 30 | 9.6                 | 18.37               | 23                  | 1470             | IMU AVG OPTS (deg) |       |          |
| 9  | 30 | 9.6                 | 18.37               | 23                  | 1470             | PTH                | 0.1   |          |
| 10 | 30 | 9.6                 | 18.37               | 23                  | 1470             | ROL                | 0.05  |          |
| 11 | 30 | 9.6                 | 18.37               | 23                  | 1470             | YAW                | -1    |          |
| 12 | 30 | 9.6                 | 18.37               | 23                  | 1470             |                    |       |          |
| 13 | 30 | 9.6                 | 18.37               | 23                  | 1470             |                    |       |          |
| 14 | 30 | 9.6                 | 18.37               | 23                  | 1470             |                    |       |          |
| 15 | 30 | 9.6                 | 18.37               | 23                  | 1470             |                    |       |          |
| 16 | 30 | 9.6                 | 18.37               | 23                  | 1470             |                    |       |          |
| 17 | 30 | 9.6                 | 18.37               | 23                  | 1470             |                    |       |          |
| 18 | 30 | 9.6                 | 18.37               | 23                  | 1470             |                    |       |          |
| 19 | 30 | 9.6                 | 18.37               | 23                  | 1470             |                    |       |          |
| 20 | 30 | 9.6                 | 18.37               | 23                  | 1470             |                    |       |          |

Note: The rover successfully traversed the entire plywood board at an inclination of 10, 20 and 25deg and showed no sign of tipping over. For inclinations greater than 25 deg, the rover loses grip over the plywood board.

Figure 7.19: Finding the tilt angle.

## 1 Bearings

Bearings provide a rolling contact between two machine parts. They help reduce the friction experienced between two surfaces while supporting the load and helping to transfer motion. Bearings are usually made up of steel or copper alloy, but it depends on the type of bearing and the field of application [15].

A bearing may fail due to various reasons. There is a 40% chance a bearing might fail if there is a violation in the lubrication conditions, such as the quantity and quality. There is a 30% chance of failure if the bearing is not assembled or installed correctly. There is a 20% chance of failure for using the bearing for the wrong type of load, like axial and radial. 20% chance of failure if the vibrations are very high and a 10% chance of failure due to operational wear or contamination of the lubricant. Fatigue in the bearing can lead to the formation of cracks in the rolling elements. The emergence of fatigue vastly depends on the transverse stress and the rotational frequency of the bearing [16].

### 1.1 Radial bearing

A radial bearing consists of a set of stainless-steel balls inside an inner and outer running surface. The shaft is held tightly by the inner running surface. This bearing is good at supporting structures that have forces perpendicular to a rotating shaft. It has a low starting torque with minimal noise. The balls are well shielded from the environment using a rubber or gap seal. Lubricant is used to keep the bearing operating smoothly. The housing must provide adequate support to the bearing outer ring so there is no shaft deflection. However, the major disadvantage is it has poor immunity to external vibrations [15].

### 1.2 Linear bearing

Linear bearings reduce the force of friction in machine parts that act along a straight or sometimes curved path. They contain rolling elements and fluid film. They can be categorised into two types. Slide-type and rolling-type bearings. Slide-type bearings rely on a thin lubrication film between the bearing and the shaft. Rolling-type bearings depend on a series of recirculating balls and rollers. They are prone to more failure than the slide type and must be shielded from the environment [17].