Final Design Review

AERO62520 - Robotic Systems Design Project

Year of submission
2024

Student ID
Team 9

School of Engineering

Contents

Word count: 7340

1 Introduction

1.1 Problem Statement and Objectives

The Robotic Systems Design Project addresses the challenge of retrieving a randomly located object from an unknown environment, and safely returning it to a designated location. The proposed solution was to design and develop an autonomous mobile robot, equipped with a robotic manipulator, capable of performing this task. The Leo Rover was chosen for this task due to its stability, customisability, and its ability to withstand any environment. A set of objectives were set to address the challenges identified by the customer:

- Upgrade the mechanical design of the Leo Rover, such that it can fit the PincherX-150 robotic arm, the RPLiDAR A2 unit, and the Intel Realsense D435i depth camera
- Integrate the LiDAR and depth camera units with the system, allowing the Leo Rover to navigate through an unknown environment, avoid obstacles, and locate an object autonomously
- Integrate the robotic manipulator with the system, allowing the Leo Rover to pick up the object, and perform manipulation tasks
- Develop the software of the Leo Rover to operate at the highest possible level of autonomy, ensuring limited to no human intervention during the mission

This final design review covers the fulfilment of the project objectives mentioned above, building upon the preliminary design review and incorporates improvements based on feedback. It discusses the proposed mechanical, electrical, and software design of the system to address the identified problem. Additionally, an analysis of the proposed design was conducted using the Requirements Verification Matrix to validate its alignment with the customer's requirements. Finally, this report examines how sustainability and cyber security factors were considered during the design process.

1.2 Addressing feedback from Preliminary Design Review

After the customer's feedback on the preliminary design review, several aspects of the design were reviewed, and were summarised below.

Concerning the preliminary system overview, components were briefly discussed and lacked details such as component functionality, necessity, and connection types. Therefore, this report includes more details on what each component generally does, different connection types, and why each component is chosen in the design.

Regarding the electrical design, while voltage requirements were taken into account, there was no discussion on current specifications or whether the battery could adequately power the system. Thus, the amount of current that each component requires was added in this report. Moreover, the total electrical load was calculated and compared with the power capacity of the battery, evaluating the suitability of the chosen battery pack.

For the software design, the utilised ROS2 packages to complete the mission were not mentioned, and the mission software plan was not discussed. Consequently, a comprehensive task planning strategy was added, including decision trees, state management, and error recovery mechanisms. This ensures effective management and control in complex task scenarios and improves overall system performance.

As for the mechanical Design, dimensions in the CAD drawings of the system were missing, manufacturability, structural strength, sensor's positioning, and payload container were either briefly or not discussed. Hence, The design has been improved to eliminate potential sensor blocking and ensure a clear field of view between the sensor and the surroundings. Furthermore, the discussion now includes more detailed content such as material selection, sustainability, and manufacturability, which factors in production cost, time efficiency and technical feasibility.

1.3 Modifications since Preliminary Design Review

After testing the preliminary design of the system in the lab, two major problems were identified in the mechanical design, and modifications have been made accordingly. The following problems were caused by the initial positioning of the lidar and depth camera (Intel Realsense) in front of the camera of the Leo Rover as shown in Appendix C.

Close-range object detection was not possible due to the range specifications of the depth camera. Although the depth camera could have still detected objects from a farther distance and calculated the pose, the robot still needed to get closer to the object for the manipulator to grasp it, meaning that grasping would have been done blindly as the object would have been too close to the camera.

Moreover, the depth camera and lidar sled were blocking the camera of the Leo Rover. Although there was no intention in using the latter initially, the design was aimed to be kept modular, and the option of using this camera was intended to be preserved.

Therefore, a new tall tripod has been designed, 3D printed, and positioned on top of the payload sled to hold the depth camera, above and behind the robotic manipulator, giving it a wider field of view, as well as the capacity of detecting objects in the working area of the manipulator.

Additionally, the lidar sled has been redesigned to lower the lidar, in addition to opening the back of the lidar sled, giving access to the camera of the Leo Rover. Further discussion on all design limitations and analysis was undertaken in the revised mechanical design section.

Finally, to better manage system complexity, the system has been segmented into two parts, with each part focusing on a specific function, enhancing modularity and maintainability. The first part consists of the Leo Rover and Lidar, which are used to control the robot's movement and navigation, while the second part consists of the manipulator and depth camera, used for object detection and grasping.

2 Sustainability Checklist

2.1 Materials

The materials chosen for the construction of the LeoRover and its peripherals, such as the robot arm and depth camera, were beyond our control, and therefore consideration of their material was not available. For parts like the payload platform and support materials, Acrylic and PLA were used. However, our options were limited. ABS could have been used instead of PLA, but ABS is petroleum-based, making it less eco-friendly than PLA, which comes from renewable sources like corn starch or sugarcane. As for the acrylic, wood is a more sustainable option. However, acrylic was chosen for its strength. In the future, different materials could be explored that are both sustainable and strong.

2.2 Software

The software aims for minimal overhead, keeping operations straightforward. It achieves this by reducing power usage by limiting LeoRover activity through use of simulations, cutting back on the need to recharge the rover. Communication is streamlined, with direct links between components (NUC and LeoRover) to cut unnecessary chatter. High-level training techniques like machine learning and neural networks for object detection and navigation were deemed excessive and thus not utilised. The use of data storage is basic with data being processed in real time and the only data storage being maps and few images. This streamlined approach to the software helps keep the overhead as small as possible, keeping processing power low and therefore more sustainable. Coding however can be optimised in the future to make the project even more sustainable.

2.3 Energy

As control over the different components provided was lacking, managing their energy efficiency and charging methods fell outside our purview. Consequently, exploring cleaner energy alternatives wasn't possible.

2.4 Waste

The robot generates no waste during its operation, except for energy consumption. However, materials used for the payload sled and supports, as well as prototypes, may become waste due to frequent alterations. To mitigate this, more sustainable materials like PLA were chosen and exercise caution in each print to minimise unnecessary construction, reducing both material and energy waste.

2.5 Emissions

The robot doesn't emit any pollutants during its lifespan, except for the source of the energy it consumes, which was addressed in the previous 'Energy' section. Given the absence of emission sources, there's no possibility of reducing emissions.

2.6 Communications

To minimise data transmission between the robot and the NUC, ROS topics are utilised. The controller subscribes and publishes only to necessary topics. However, a drawback arises when the need to get/send data is no longer needed, for example when the robot is stationary, as the controller continues to publish zero wheel velocities, potentially leading to unnecessary data transmission.

2.7 Modularity

The robot boasts a highly modular design, featuring numerous components that are 3D printed. CAD drawings of 3D parts utilised by the LeoRover are publicly available, facilitating easy replacement. Similarly, the PincherX-150 robot arm is predominantly 3D printed and therefore easier to repair/replace. In terms of software, individual components of the robot behaviour are coded separately and then integrated, enhancing modularity and simplifying debugging and troubleshooting processes.

2.8 Location/Placement

The robot's deployment is highly sustainable, as it doesn't require any vehicles. Transportation for deployment and testing occurs within the same building, allowing the robot to be manually transported. This makes it one of the most sustainable options available.

2.9 Maintenance

The robot's mission duration is brief enough that diagnostic checks are deemed unnecessary. It undergoes regular weekly testing, allowing any performance decline to be manually monitored and assessed. This is evident in the rover's speed, robot arm functionality, and camera object detection capabilities, all of which are consistently observed by human operators. Performance of these components remains under constant scrutiny. While devices like the depth camera include diagnostic checks for monitoring their health, they are employed only if necessary.

2.10 Repurposing

The robot's modular construction greatly enhances its potential for repurposing, as each of its five main components (LiDAR, Depth Camera, Robot Arm, LeoRover, NUC) can be detached and utilised for other tasks. This flexibility significantly expands the scope of component reuse, contrasting with scenarios where components are fixed and limiting repurposing to missions closely resembling the current one. By enabling versatile adaptation, the robot's modular design ensures long-term utility and extends its applicability to various future endeavours.

3 Cyber Security Considerations

During the design process of the system, potential cybersecurity risks have been considered and mitigated if possible.The following table summarises the identified risks and the existing measures to mitigate them. Within the scope of this project, cybersecurity threats were identified during three main activities: Mission/testing, software development, and robot storage.

Activity	**Threat**	**Existing measures to mitigate risk**	**Risk Rating**
Data transmission between NUC and Leo Rover	Transmitted data may be intercepted by attackers (malicious control of the robot).	Using safe Ethernet connection. The attacker will need direct hardware access to either the NUC or Raspberry Pi Perform proper network configuration for the Raspberry Pi and NUC.	Low
Code Development on issued laptops	Unauthorised individuals gaining access to code and data by exploiting system vulnerabilities	Encryption and Password Protection - Secure Development Practices - Software Updates.	Medium
Code Sharing	Sensitive code or data could be leaked and exploit	Using trusted platforms like Google Drive and Github - Access Control - Utilise two-factor authentication (2FA) for accessing Google Drive.	Low
Using open source software (ROS2)	Open source software could have security vulnerabilities or be maliciously altered	Regularly check and update the ROS2 packages and their dependencies to ensure they are using the latest versions with known security vulnerabilities patched. Obtain packages only from the official repository or other trusted sources to minimise the risk.	Low
Hardware Storage and Access	Unauthorised access to hardware, data theft, alteration, or malicious control of the robot.	Physical Security: The hardware components required for the robot should be stored in a secure, access-controlled environment. Time-Limited Access: Ensure that access to the hardware components is restricted to designated time periods to minimise the risk of unauthorised use or removal.	Low

Table 1. Cybersecurity Risk Assesment

4 Updated Design Review

4.1 System

4.1.1 System Overview

This section provides a top-down perspective of the entire system, illustrating the connections among its different components. The following figure shows the system block-diagram of all the components in the robot.

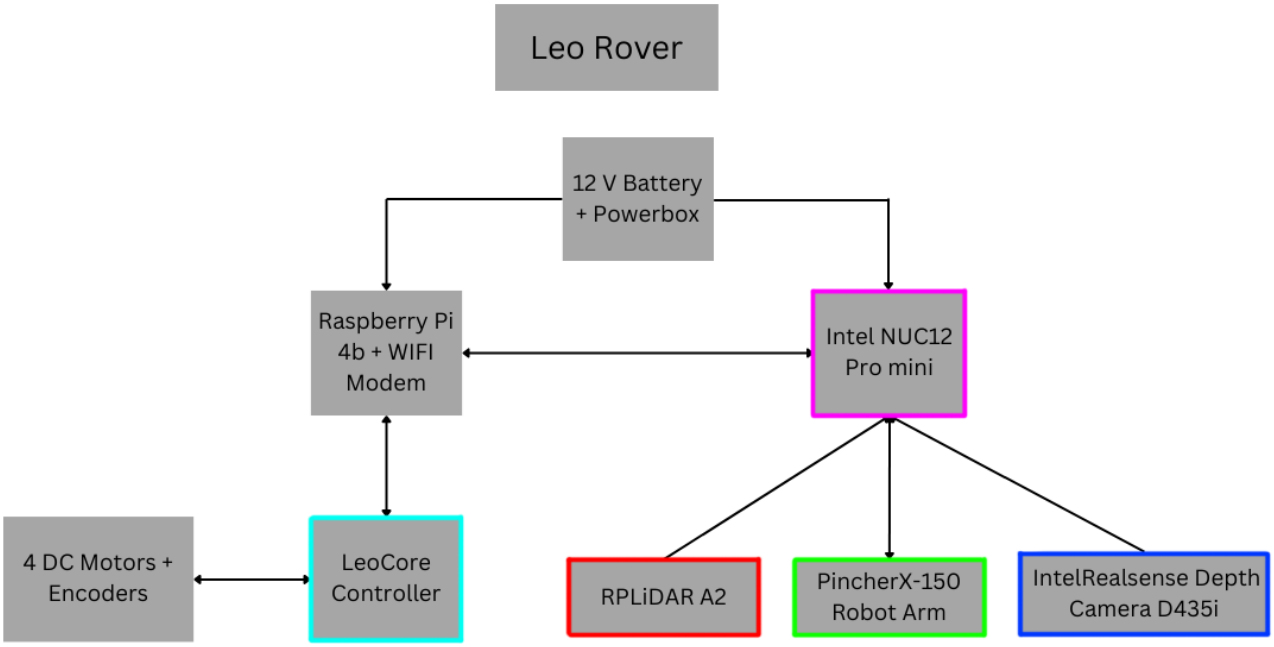

Fig. 1. System block-diagram of all the components

4.1.2 Component Description

The system consists of several essential components, each designed to fulfil a particular role that aligns with the project objectives, as outlined in the table below.

Component	Description/Specs	Necessity	Power
Leo Rover Kit: LeoCore Controller + Raspberry Pi 4B	Control unit responsible for managing the motor functions of the Leo Rover. It interfaces with various sensors and actuators.	Ensures precise movement and navigation of the Leo Rover. Essential for motor control and sensor data processing.	12 V DC 2 A
Intel NUC12 Pro mini	Small computing unit: Intel Core i5 processor, 4 GB RAM, 500 GB SSD storage	Provides high computational power required for complex perception, autonomous navigation algorithms, and manipulator control.	12 V DC 3 - 5 A
PincherX-150	5-DOF Robotic Manipulator Payload capacity: 50g Reach: 0.45 m open-source code	Executes object grasping and manipulation tasks.	12 V DC 5 A
RPLiDAR A2	2D Lidar - 360 degrees scanning Range: 0.15 - 12m	Provides distance measurements for environmental mapping (SLAM), and obstacle detection.	5 V 0.3 - 0.5 A
Intel RealSense Depth Camera	Stereoscopic Depth camera, Resolution: 1280 x 720 Range: 0.3 - 3 m	Essential for object detection and provides distance to object.	5 V 0.6 - 0.8 A
Battery	Li-Ion with internal PCM Waterproof case (IP66 compliant)	Powers the whole system.	11.1 V DC 5000 mAh Max: 8 A

Table 2. Component Description

Power and Compute

The computing and control equipment, which provides intelligence to the robot, consists of LeoCore controller, Raspberry PI 4 and Intel NUC12. Using these three devices together, a hierarchical control system can be achieved, from real-time control at the bottom (microcontrollers), to more complex algorithms at the top (Intel NUC). The Raspberry Pi serves as an interface between the LeoCore Controller and the NUC. Finally, these devices are connected through ethernet, and powered by the 12 volts battery.

Sensor Fusion

The system uses two main sensors to perform SLAM, and object detection: RPLiDAR A2 unit and Intel RealSense Depth Camera D435i. Both sensors are connected to the Intel NUC12 through USB, and send sensor data to the NUC through ROS2 nodes.

Actuation and Interaction

The Leo Rover contains four DC motors which it uses to move and traverse the environment, while the PincherX-150 robot arm is used to pick up objects. The latter is connected to the Intel NUC12 through USB, and is controlled through ROS2 nodes.

4.1.3 Electrical Design

Figure 2 shows a more detailed perspective of the system, which includes how the components of the system are electrically connected.

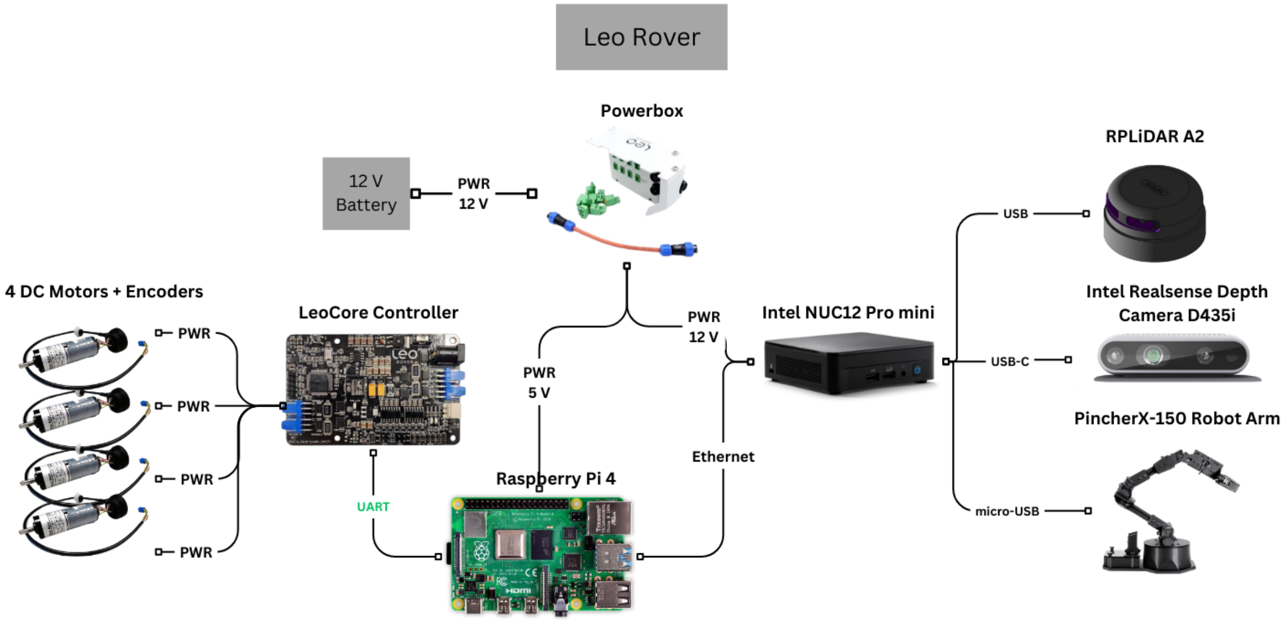

Fig. 2. Electrical Diagram

After testing and examining the power capacity (voltage and current), and discharge characteristics of the battery mentioned in table 2, it is currently not possible to power the whole system using the proposed battery pack. The proposed battery can produce a maximum of 8 amps, while the total load requires a minimum of 12 amps. Therefore, the battery pack will only be used to power the Leo Rover, while other components such as the NUC and manipulator will use an external power source (temporarily).

4.2 Software Design

In this section, we present an overview of the software design for our system, employing visualisation tools such as RQT graphs and decision trees to illustrate its structural organisation and decision-making processes. The first topic of interest will be showing the relationship between hardware and software components.

4.2.1 Hardware to Software

The following table shows the role of the different electrical components of the robot, and the ROS2 topics allocated to that specific hardware. The connection between the different components was analysed to verify that the proposed system design aligns with the functional requirements, and the overall task.

ROS 2 serves as the foundation of the robot software architecture, facilitating the communication between the different components of the system. We will examine various topics and nodes to ensure the software design is in compliance with customer requirements, as outlined in the verification matrix.

System	**Actuation**	**Sensing and Perception**	**Compute and Control**	**Power**	**Interaction**
Hardware	4 DC motors + wheels	RPLiDAR A2, Intel RealSense Depth Camera D435i, encoders	LeoCore Controller, Raspberry Pi, Intel NUC12	12 V battery, powerbox	PincherX-150 robot arm
Software	/firmware, /cmd_vel	/scan, /camera/color/image_raw, /camera/depth/image_rect_raw	/leo_rover_controller	N/A	/px150/joint_states

Table 3. Software and Electrical Design Summary of Leo Rover System

To facilitate the integration of various system components, the '/leo_rover_controller_node' is employed. This node serves to interconnect different peripherals within the system and can publish diverse messages to modify the robot's behaviour based on sensor readings. You can explore the GitHub repository for this node and potential future nodes by clicking here.

The 'leo_rover_controller' node serves as the central hub for collecting and distributing data, and is computed on the NUC. It receives information from the LiDAR and Camera, and sends control signals to both the manipulator and wheels. This versatile node performs tasks based on the sensor readings, such as determining when to pick up objects and avoiding obstacles. It also tracks the robot's position, enabling it to return to the original position, fulfilling requirement 9.

4.2.2 RQT Graph

This section presents the system's initial RQT graph (see figure 3). The various colours in the graph can be connected to the system's block diagram (see figure 1), illustrating the interactions among different components. It also depicts the topics utilised for communication and includes information about the rates at which messages are published to these topics.

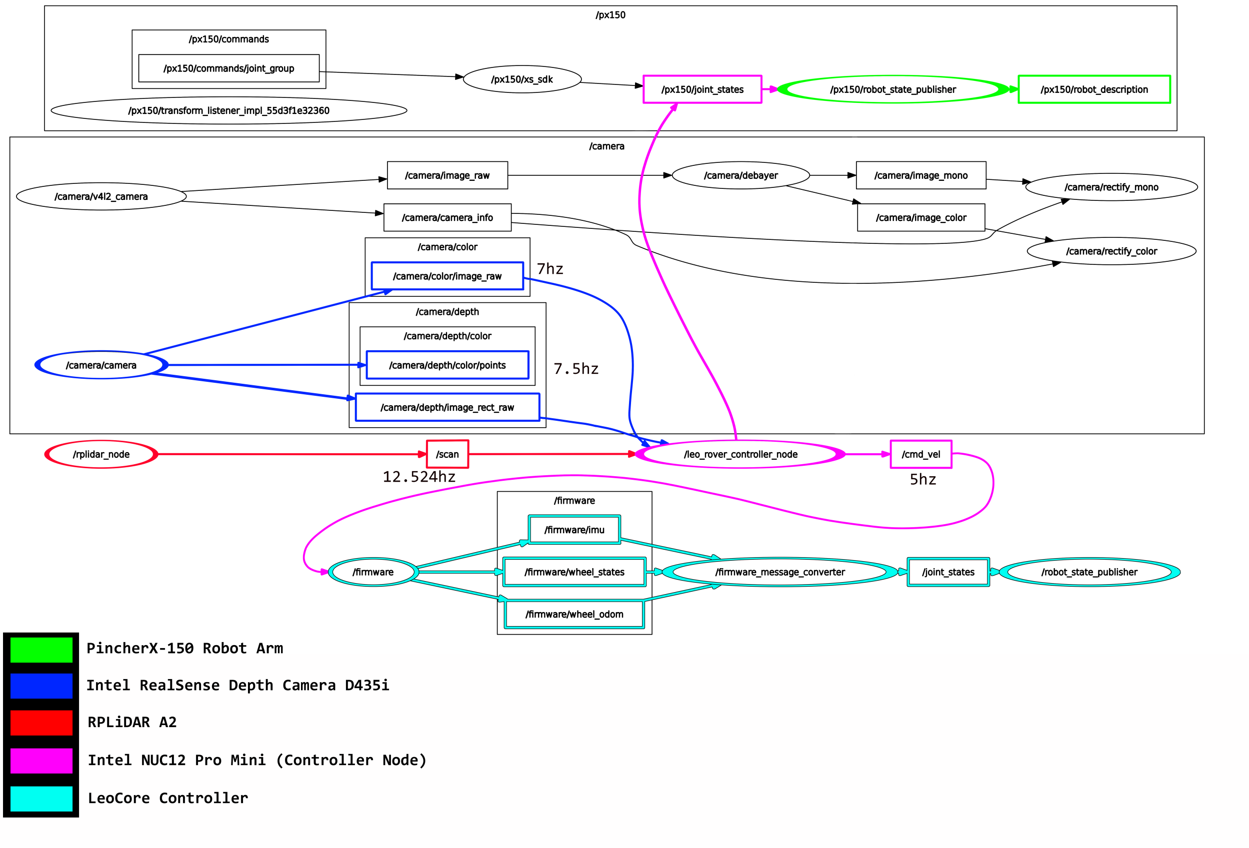

Fig. 3. RQT Graph of the system

The blue section outlines the image topics utilised, namely '/camera/color/image_raw' for object detection and '/camera/depth/image_rect_raw' for object depth determination. The accompanying RQT graph illustrates the flow of information between these topics and the subscribed and publishing nodes, notably including the '/leo_rover_controller_node' for data processing. For image processing, OpenCV's cv2 interface will be used, offering essential functions like masking, contour creation, and visualisation aids. Additionally, packages like pyrealsense2 (python wrapper for the Intel RealSense SDK 2.0) assist in calculating block positions relative to the camera.

In the red section, attention shifts to the RPLiDAR A2 topic ('/scan') and its interaction with the controller node. LiDAR scan processing is facilitated by libraries such as "rplidar_ros," which will be used in conjunction with other navigation packages such as:

- slam_toolbox
- nav2_bt_navigator

- nav2_behaviors
- nav2_planner
- nav2_controller
- nav2_lifecycle_manager
- explore_lite
- leo_description

Providing all the tools necessary to complete navigation for an autonomous robot. Enabling the robot to generate real-time maps and identify obstacles in the test site, thereby fulfilling the requirements (Appendix B.1, B.2).

To help simulate the robot and fine tune its behaviour, software such as Gazebo Ignition/Classic and Rviz will be utilised with the following packages for ros integration into these simulators and visualisation of components:

- ros_gz_sim
- robot_stat_publisher
- ros_gz_bridge
- ros_gz_image
- rviz2

Moving on to the cyan section, it highlights the LeoCore Controller's topics, demonstrating velocity transmission via '/cmd_vel' to the '/firmware' node, subsequently converting it into wheel and joint states for the robot. Thus, the velocity of the robot is regulated and the robot could attain speeds up to 0.35 m/s, as needed by the requirements (Appendix B.2a).

The purple/pink section shows the created '/leo_rover_controller_node' and which topics it subscribes to and which it publishes to, indicated by the arrows showing the flow of data.

Finally, the green section delineates the topics and nodes associated with the PincherX-150 robot arm, showcasing message publication via the '/px150/joint_states' topic through the leo rover controller. Trossen Robotics provides libraries enabling seamless operation of the arm, ensuring user-friendly functionality such as:

- interbotix_xsarm_descriptions - contains the meshes and URDFs (including accurate inertial models for the links) for all arm platforms
- interbotix_xsarm_control - contains the motor configuration files and the 'root' launch file that is responsible for launching the robot arm

- interbotix_xsarm_sim - contains the config files necessary to simulate an arm in Gazebo classic
- interbotix_xsarm_ros_control - contains configuration files necessary to set up ROS 2 controllers between MoveIt and the physical robot arm
- interbotix_xsarm_moveit - contains the config files necessary to launch an arm using MoveIt

4.2.3 Decision Tree

Understanding and deciding the next course of action for the robot in various states can be complex and daunting. Visual aids provide clarity and facilitate the integration of robust behaviours and user-friendly comprehension. Below is a decision tree illustrating this concept.

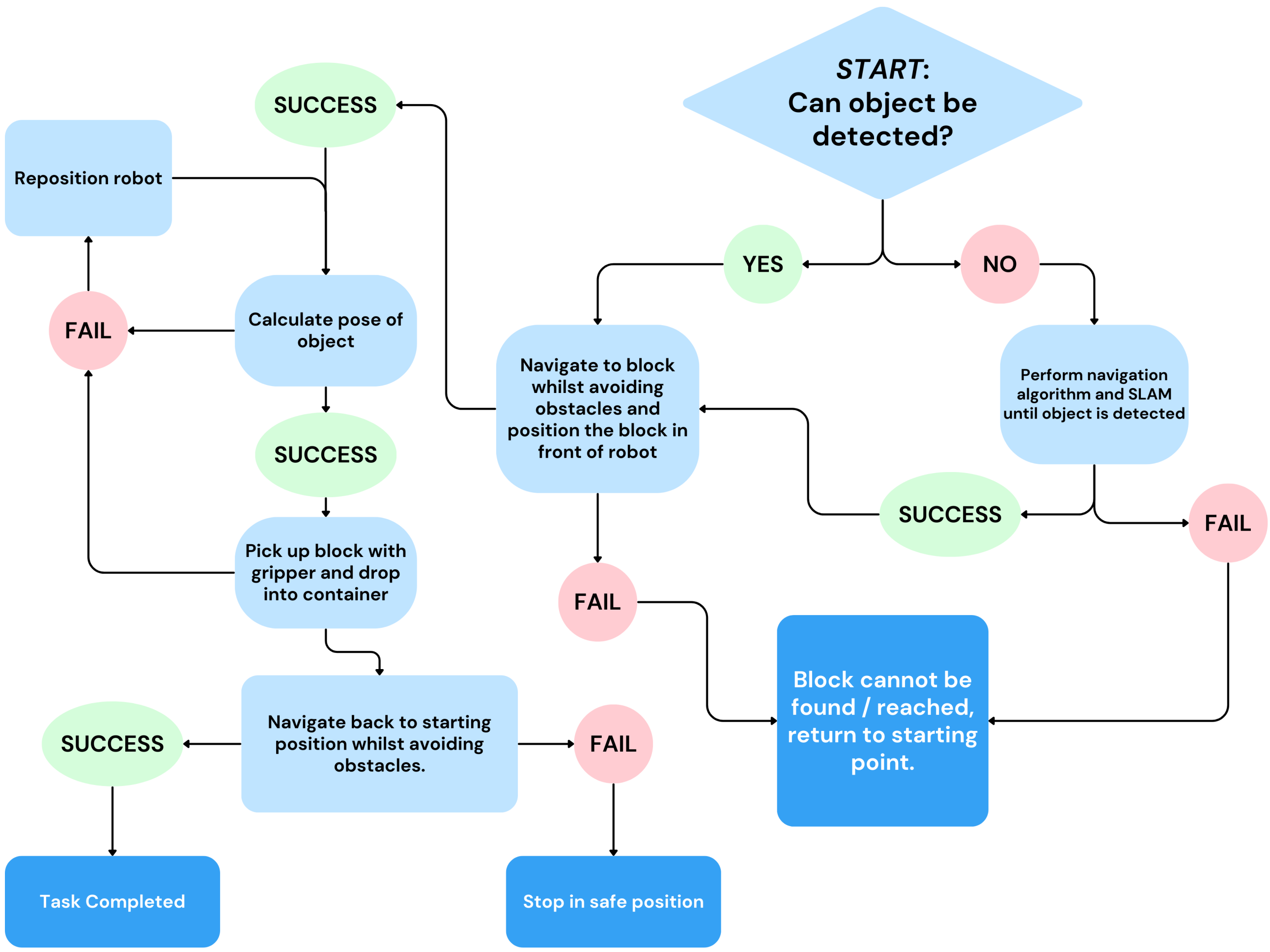

Fig. 4. Decision tree of robot behaviour

This decision tree illustrates the robot's behaviour in various situations. It begins at the top diamond and progresses through branches marked by green circles for success and red circles for failure, depending on the outcome described in the light blue boxes. The dark blue boxes denote the end of the behaviour, signalling where the robot will cease its actions and no longer proceed.

4.3 Mechanical Design

This section focuses on the design of the payload sled mounted on top of the Leo Rover, as well as the different parts designed to attach the depth camera and lidar.

The following two figures present the final design of the robot, including the PincherX-150 robotic manipulator, Intel RealSense D435i Depth Camera, RPlidar A2, and Intel NUC 12 Pro mini.

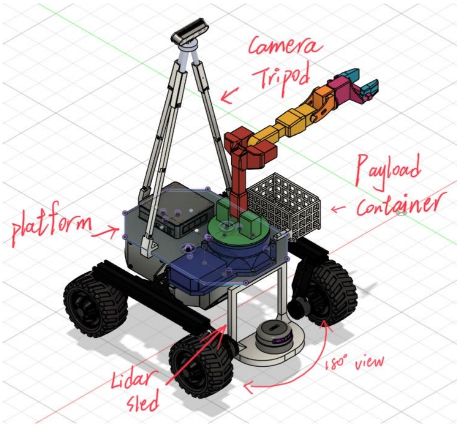

Fig. 5. Isometric View

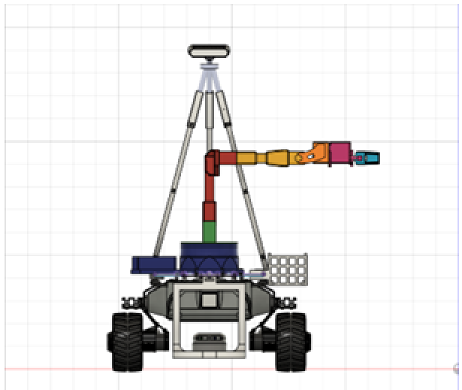

Fig. 6. Front View

For detailed design file, please check the following GitHub link.

4.3.1 Payload Platform

The platform is fabricated from a 6 mm thick acrylic board, measuring 350 mm in length and 240 mm in width. This construction provides sufficient area and strength to support the manipulator, NUC, and additional payloads. The platform weighs approximately 530g.

Consideration of the robot's size is crucial in the design, impacting handling, workspace accessibility, and task performance. The dimensions of the chassis of the robot (including the payload sled) are 410 x 450 x 330 mm. According to these dimensions, the width of the payload matches the width of the robot, which aligns with the environment requirements

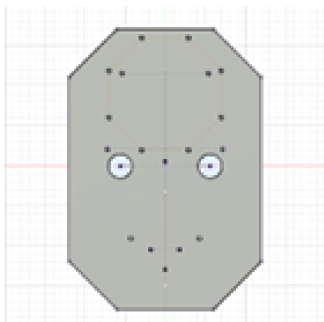

Fig. 7. Top View of the Platform

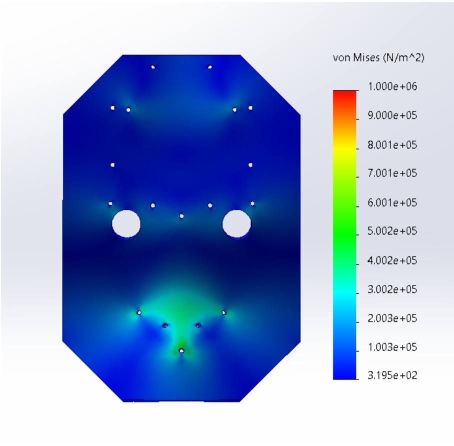

Fig. 8. Stress distribution result

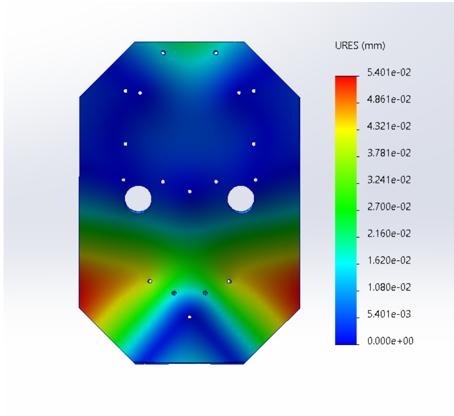

Fig. 9. Displacement result

As depicted in Figures 8 and 9, areas shaded in blue indicate low stress levels, whereas those shaded in red signify the highest stress levels.

Based on the outcomes of the simulation, it can be substantiated that the structural integrity of the platform is sufficient to sustain the load imposed by the manipulator, NUC, and additional payloads, as evidenced by the maximum displacement being a mere 0.054 mm. Nonetheless, the findings also suggest a potential enhancement, namely, the augmentation of support structures at the platform's rear to mitigate stress concentration and reduce displacement.

4.3.2 Lidar Sled

The Sled for Lidar is designed to have dimensions of 160 mm in length and 160 mm in width. It is 3D printed using PLA, with a mass of approximately 100g, and is attached to the Leo Rover using screws.

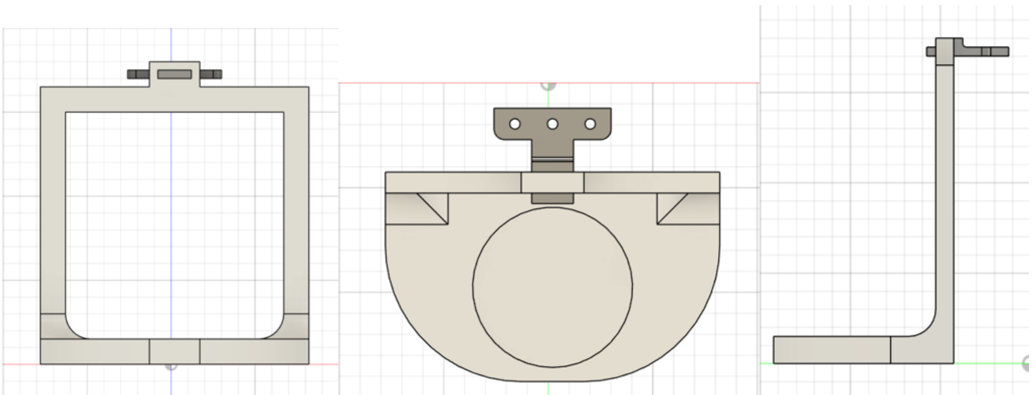

Fig. 10. Lidar Sled

A shallow circular groove on the base part is intended for placing the Lidar. In comparison to the preliminary design, the new frame is wider, the base is lower, and the shelf for the depth camera is omitted, allowing the original camera of the Leo Rover to have a wider field of view.

The Lidar is positioned at the lower front of the entire robot, providing two advantages. Firstly, it allows the manipulator to be installed closer to the ground, compared to installing it on a second payload cell above the Lidar. This setup increases the reach of the manipulator and lowers the overall centre of gravity, improving stability. Secondly, the lower position of the Lidar enables it to detect obstacles shorter than the robot.

Nevertheless, additional noise could potentially be introduced into the readings by the wheels. Additionally, the narrow space between the base and the ground reduces the Leo Rover's passability. While this disadvantage is acceptable for missions on flat ground, the design may need to be revised for more complex scenarios, such as uneven terrain, in the future

4.3.3 Depth Camera Tripod

The tripod legs are 3D printed using PLA, with a total weight of about 100g. Each leg measures 47 cm in length and forms a 75-degree angle with the platform. They are secured to the platform using screws on the 'foot'. The three legs together form an isosceles triangle, supporting the depth camera at a height of 45 cm above the platform, or approximately 67 cm above the ground.

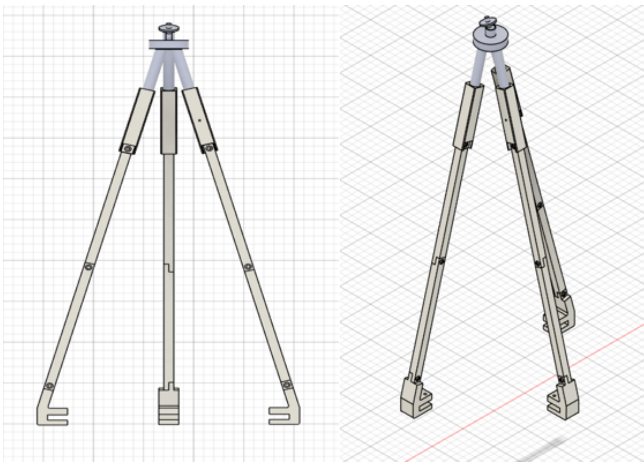

Fig. 11. Depth Camera Tripod

Testing of the preliminary design revealed that the depth camera could not detect objects in the workspace of the manipulator as it was too close to the camera. By positioning the depth camera to 'look down' from the upper rear of the robot, it can effectively detect objects close to the robot as it will be in the range of the camera.

4.3.4 Payload Container

The container is manufactured through 3D printing using PLA, with an estimated weight of approximately 80 g. Its dimensions are specifically crafted to measure 125 mm in length, 75 mm in width, and 60 mm in height. Square holes, each measuring 15 x 15 mm, are incorporated to reduce weight. Given that the dimensions of the smallest object to be retrieved are 20 x 20 x 20 mm, the container's size adequately meets the requirements for multi-object missions.

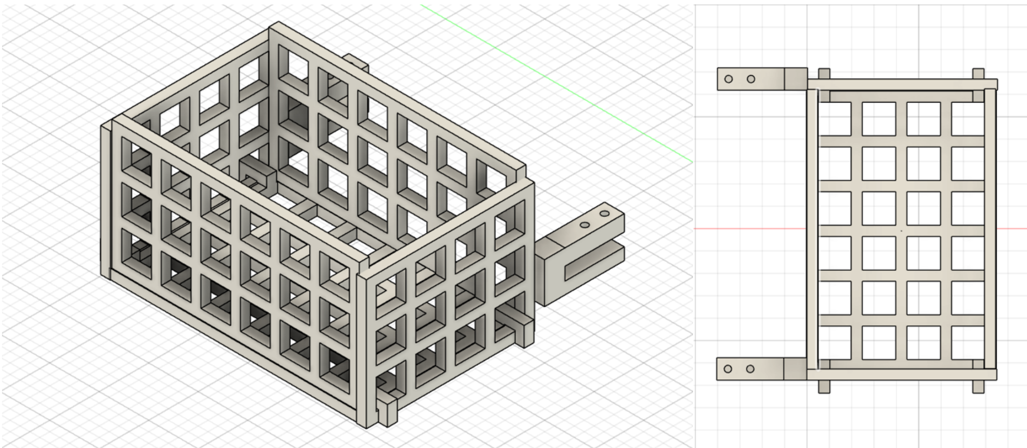

Fig. 12. Payload Container

4.4 Analysis

The system can be broken down into five main components:

- Intel NUC 12
- PincherX-150 robot arm
- Intel RealSense D435i Depth Camera
- LeoRover
- RPLiDAR A2

Each of these components plays a crucial role in completing the task. Therefore, we will provide an explanation of each component, detailing how it contributes to fulfilling specific requirements specified in Appendix B 'Functional and Product Design Requirements'.

Intel NUC 12:
One could argue that the NUC plays a significant role in fulfilling the majority, if not all, of the requirements. This is because the NUC facilitates communication between different robot components, aids in the movement and data processing of every other component. As a result, it is involved in almost every aspect of meeting the requirements. Primarily, its role lies in ensuring the autonomy of the robot by monitoring its state and determining the next steps to complete the task. This makes it a major contributor to fulfilling requirements 'B.1 Autonomy Requirements', needed for data processing in 'B.4 Object Identification Requirements', and keeping track of its location to complete 'B.5 Task Requirements'.

PincherX-150 robot arm:
The robot arm plays a crucial role in completing the task by collecting the object, making it an essential component for meeting the requirements under 'B.3 Object Retrieval.' It has the capability to retrieve objects of various shapes and sizes, thus fulfilling the specified requirements effectively.

Intel RealSense D435i Depth Camera:
The depth camera plays a crucial role in identifying objects and determining their pose, contributing to fulfilling multiple requirements such as 'B.3 Object Retrieval' by utilising the depth feature to calculate the depth and therefore the pose of the object ready for retrieval by the robot arm and 'B.4 Object Identification Requirements' by using the RGB camera to identify the object by its colour. While a standard camera might suffice for requirement B.3, a depth camera is essential for task B.4.

LeoRover:
The Leo Rover was selected for its versatility, capable of meeting several traversal requirements and providing sufficient space to carry other components. Its primary role is in fulfilling traversal requirements, specifically 'B.1 Autonomy Requirements,' 'B.2 Environment Requirements,' 'B.5 Task Requirements,' and 'B.6 Robot Safety and Debugging Requirements.' It achieves this by traversing with its four wheels and adjusting its behaviour based on inputs from the NUC.

RPLiDAR A2:
The LiDAR enhances the rover's environmental perception beyond what the depth camera alone provides. This expanded perception enables efficient object avoidance, essential for meeting requirements 'B.1 Autonomy Requirements' and 'B.2 Environment Requirements'.

The following requirements verification matrix illustrates the methods for confirming whether the design aligns with the specified requirements.

Requirement No	Paragraph	Requirement Statement	Verification Success Criteria	Verification Method
1	B.1 Autonomy Requirements	The robot shall autonomously avoid obstacles observable by the robot.	1. The LiDAR will be tested with various objects of different colours and sizes placed within the specified range (0.15 m - 5 m) to see if the objects are detectable. 2. The LiDAR will then be placed on the robot ensuring the obstacles are detected then traversed around. This will be tested in various environments and against various obstacles. The success rate criterion is set to exceed 95%.	Test
2a	B.2 Environment Requirements	The robot shall be able to move at a maximum speed of 0.35 m/s, to be able to complete the task within the robot's battery life.	The robot's speed will be tested to see if the robot can maintain a speed of 0.35 m/s and not exceed this, further testing the controller limits of the software. This is checked by having measurements on the floor of the lab and timing how long it takes to traverse over a set distance. The success rate criterion is set to exceed 95%.	Test
2b	B.2 Environment Requirements	The designed payload will not exceed the width of the robot, which is 450 mm, such that the robot can navigate between obstacles (assuming that the minimum gap between obstacles can only fit the Leo Rover chassis).	The payload will be measured manually, and mounted on the robot. A 20 mm tolerance will be accepted. The robot's capability to navigate through the narrowest spaces between obstacles will undergo testing. This is also under the assumption of the arm being in it's rest position as it should be during transit.	Inspection
3	B.2 Environment Requirements	The robot should navigate the environment, perform obstacle avoidance, and pick up the object autonomously, with no human intervention.	The ability of the robot to navigate the environment and perform obstacle avoidance will be simulated then tested in the lab. The success rate criterion is set to exceed 90 %.	Test
4a	B.3 Object Retrieval Requirements	Robot arm shall be able to pick up cuboid wooden objects of sizes up to 0.04 m x 0.04 m x 0.04 m, with a maximum weight of 50 g.	A wooden cube of various sizes ranging from 0.02 m to 0.04 m and various weights up to 50 g will be attempted to be picked up by the robot arm. The success rate criterion is set to exceed 90%.	Test
4b	B.3 Object Retrieval Requirements	Robot arm should be able to pick up spherical wooden objects of sizes up to a diameter of 0.04m, with a maximum weight of 50 g.	A wooden sphere of various sizes ranging from 0.02 m to 0.04 m in diameter and various weights up to 50 g will be attempted to be picked up by the robot arm. The success rate criterion is set to exceed 90%.	Test
5	B.4 Object Identification Requirements	The robot shall be able to identify the target object by its colour.	The robot will return a message of the correct colour of the certain area from the image information. The success rate criterion is set to exceed 90%.	Test
6	B.4 Object Identification Requirements	The robot should identify the target object with limited human intervention.	The robot will return a message after identification. The message will provide information regarding the recognized color and shape of the object. The success rate criterion is set to exceed 90%.	Test

Table 4. Requirements Verification Matrix

Requirement No	Paragraph	Requirement Statement	Verification Success Criteria	Verifica-tion Method
7	B.4 Object Identification Requirements	The robot will be equipped with a vision system that can identify objects by colour, distinguishing between a predefined range of colours under standard laboratory lighting conditions.	The vision system will be capable of identifying objects based on color and return a message of the correct color of the object. The success rate criterion is set to exceed 95%.	Test
8a	B.4 Object Identification Requirements	Depth camera should be able to detect objects as small as 0.02 m x 0.02 m x 0.02 m at a distance in the range of 0.28 m - 5 m	A block of a specified colour will be placed 5m away at the size of 0.02 m x 0.02 m x 0.02 m and see if the object is detected then, placed closer to the robot at regular intervals until reach the minimum 0.28 m mark. The success rate criterion is set to 90%.	Test
9	B.5 Task Requirements	The robot shall be able to bring picked up objects back to the robot's original starting location.	1. Starting from the foundation of the requirement 3 test. 2. The robot will then traverse the lab at a speed of around 0.35 m/s to verify that the manipulator can maintain control of the block over the course of the objective. 3. The robot and wooden block will then be placed in random locations in the environment, the robot will retrieve the block, then return to its starting location. The success rate criterion is set to exceed 90%.	Test
10	B.5 Task Requirements	The payload will be no more than 0.25 m above the ground level to allow the robot to pick up objects from the ground comfortably, as the working area of the manipulator is a sphere with diameter 0.63 m.	When the payload is attached to the robot, the height will be measured to ensure it does not exceed the threshold. The success rate should be 100% as this should always be the case.	Inspection
11	B.6 Robot Safety and Debugging Requirements	The combined weight of the designed payload, LiDAR, NUC, and depth camera will not exceed 5 kg.	All the components will be weighed to ensure they do not exceed the threshold. The success rate should be 100% as this should always be the case.	Analysis
12	B.6 Robot Safety and Debugging Requirements	The robot should include an emergency stop feature that can be activated by the operator to immediately cease all movement and operations.	The stop feature will be sent to the robot during during multiple runs at different sections of its task, ensuring the stop feature works at any stage of its operation. The robot should be able to stop all operations within 2 seconds with a success criterion set to exceed 90%.	Test
13	B.6 Robot Safety and Debugging Requirements	The robot should provide a log of its activities, including objects picked up, paths taken, and any errors or obstacles encountered.	The logs will be manually checked and crossed off against what was actually accomplished by the robot in a series of tests of a full completion, one where the block is not present and another what the block is retrieved but progress to the starting point is made impossible, this is to ensure it records correctly its state. The success criterion is set to exceed 90%.	Test

Table 5. Requirements Verification Matrix 2

Appendices

A Equality, Diversity, Inclusion and Accessibility Policy

Our Policy's Purpose

Within Team 9, we aspire to create an environment for all team members, free of bullying, harassment, victimisation and discrimination, towards each other and other people. Our goal is accomplished by having all members committed to treating everyone with equality, respect and dignity. In accordance with our established policies and regulations, all voices are valued, and all team members can express themselves openly without facing unjust criticism or the worry of retaliation.

Team 9 is committed to provide an inclusive, equitable and respectful workplace that promotes diversity and accessibility for all. Equal opportunities, fair treatment, and an inclusive decision-making process is ensured for all team members regardless of age, disability, gender, marriage, race, religion or belief, and sexual orientation.

Team rules

1. Freedom of speech is protected and encouraged to help give the ability for everyone's views to be heard, viewed and considered equally.
2. Provide constructive feedback and be open to receiving it.
3. Always give the benefit of the doubt, assuming good intentions.
4. Team members are responsible to complete their task on time, based on the deadline and the task loads. If someone needs assistance, they should inform other people as early as possible to avoid delay.
5. Try the best to speak English during meetings, labs, and online. Be patient if anyone encounters difficulty in expressing their thoughts.
6. Actively seek input from all team members during the decision-making process. Major decisions to be made by majority vote. If in the event of a tie, it will be discussed until a compromise is made or until the majority agrees.
7. Maintain confidentiality: Keep sensitive information confidential to foster trust.
8. Weekly updates for personal and meeting logs are required, ensuring accurate records of each team member's contributions and achievements.
9. Consider accessibility requirements when organising shared folders, ensuring that naming rules and structures are clear for all team members.
10. Ensure that code documentation is inclusive and understandable to all team members, taking into account the diversity level of programming experience.

11. Working Hours: Tuesday meeting (11 am - 12 pm), Wednesday lab (2 - 6 pm) , Friday meeting (11 am - 1 pm) - No work on weekends (unless there is a submission)

Our Approach to Grievances

In the case of any conflict, misunderstanding, harassment, or a member was offended by a discussion or treatment by another team member, a plan of action has been established to address the problem rapidly and resolve it:

1. If possible, the grievance is solved between the members that feel offended and the offender, calmly talking to each other, and ensuring all parties involved understand the problem, avoiding misunderstandings. Otherwise, the following steps will take place.
2. Considering all members' timetables, a meeting is organised at the earliest convenience. The member in question should discuss his experience and feelings with the others, brainstorming potential solutions, until a consensus or meeting ground is reached.
3. A vote is then made after the discussion, objectively considering the best approach to be taken. Subsequently, the team will adhere to the best solution.
4. In case the discussion becomes unproductive, there will be a pause to make sure all members communicate effectively. A follow-up meeting will then be organised.
5. In the event that the issue is beyond the responsibility or capability of the team, all team members have the option to consider escalation. Staff can step in and mediate the situation to ensure the safety of all members involved.

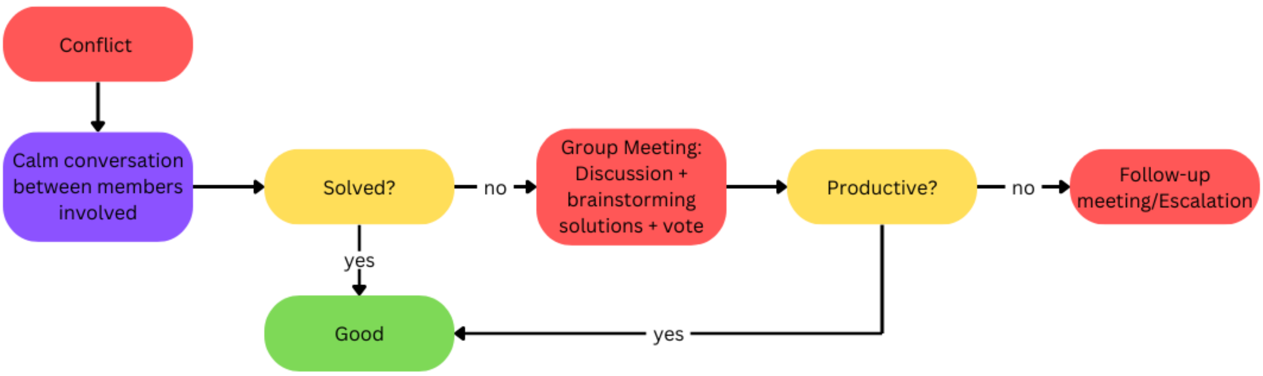

Fig. 13. Conflict Resolution Flow Chart

Preferably, any signs of conflict should be recognised early on, and an open communication within the team should be encouraged, where members feel safe expressing their concerns.

Accountability and Monitoring

Failure to comply with the policy can result in the matter being escalated. This is to help adhere to the policy and have all members be held accountable for their actions. The team leader could monitor the behaviour of the team and ensure that the policy and team rules are respected.

B Functional and Product Design Requirements

This section highlights the functional and design requirements of the robot, allowing it to achieve the objectives set out in the problem statement (Section 2 of this document).

B.1 Autonomy Requirements

1. The robot shall autonomously avoid obstacles observable by the robot.
   (a) The LiDAR should be able to detect obstacles at a distance in the range of 0.15 m - 5 m, to be able to perform obstacle avoidance.

B.2 Environment Requirements

2. The robot shall be able to traverse indoor environments
   (a) The robot shall be able to move at a maximum speed of 0.35 m/s, to be able to complete the task within the robot's battery life.
   (b) The designed payload will not exceed the width of the robot, which is 450 mm, such that the robot can navigate between obstacles (assuming that the minimum gap between obstacles can only fit the Leo Rover chassis).
3. The robot should navigate the environment, perform obstacle avoidance, and pick up the object autonomously, with no human intervention.

B.3 Object Retrieval Requirements

4. The robot should be able to pick-up lightweight objects that are within the size and torque capability of the grippers
   (a) Robot arm shall be able to pick up cuboid wooden objects of sizes up to 0.04 m x 0.04 m x 0.04 m, with a maximum weight of 50 g.
   (b) Robot arm should be able to pick up spherical wooden objects of sizes up to a diameter of 0.04m, with a maximum weight of 50 g

B.4 Object Identification Requirements

5. The robot shall be able to identify the target object by colour.
6. The robot should identify the target object with limited human intervention (e.g. inputting object colour and shape).
7. The robot will be equipped with a vision system that can identify objects by colour, distinguishing between a predefined range of colours under standard laboratory lighting conditions.

8. The robot will integrate a depth camera to identify the object.
   (a) Depth camera should be able to detect objects as small as 0.02 m x 0.02 m x 0.02 m at a distance in the range of 0.28 m - 5 m.

B.5 Task Requirements

9. The robot shall be able to bring picked up objects back to the robots original starting location.
10. The payload will be no more than 0.25 m above the ground level to allow the robot to pick up objects from the ground comfortably, as the working area of the manipulator is a sphere with diameter 0.63 m.

B.6 Robot Safety and Debugging Requirements

11. The combined weight of the designed payload, LiDAR, NUC, and depth camera will not exceed 5 kg.
12. The robot should include an emergency stop feature that can be activated by the operator to immediately cease all movement and operations.
13. The robot should provide a log of its activities, including objects picked up, paths taken, and any errors or obstacles encountered.

C Preliminary Mechanical Design

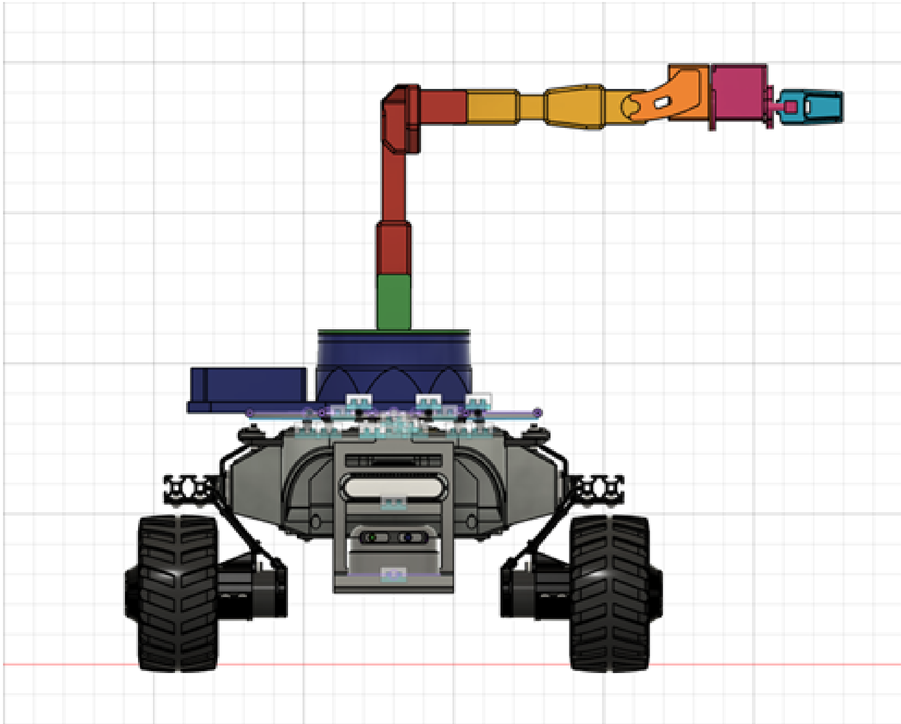

Fig. 14. Initial Design: Robot Front View

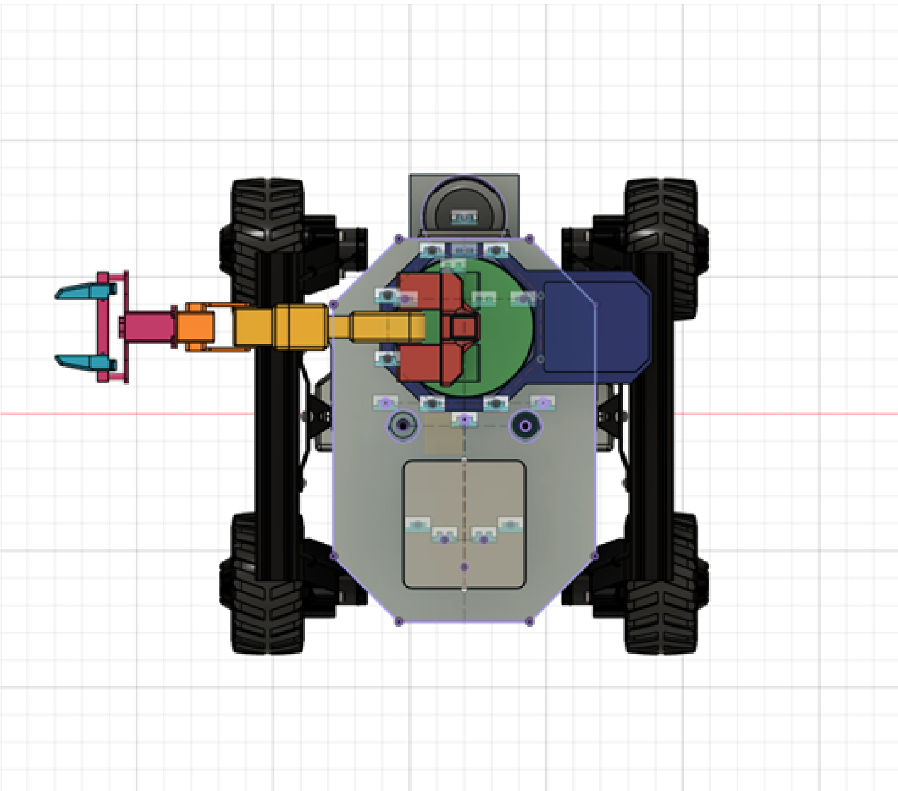

Fig. 15. Initial Design: Robot Top View

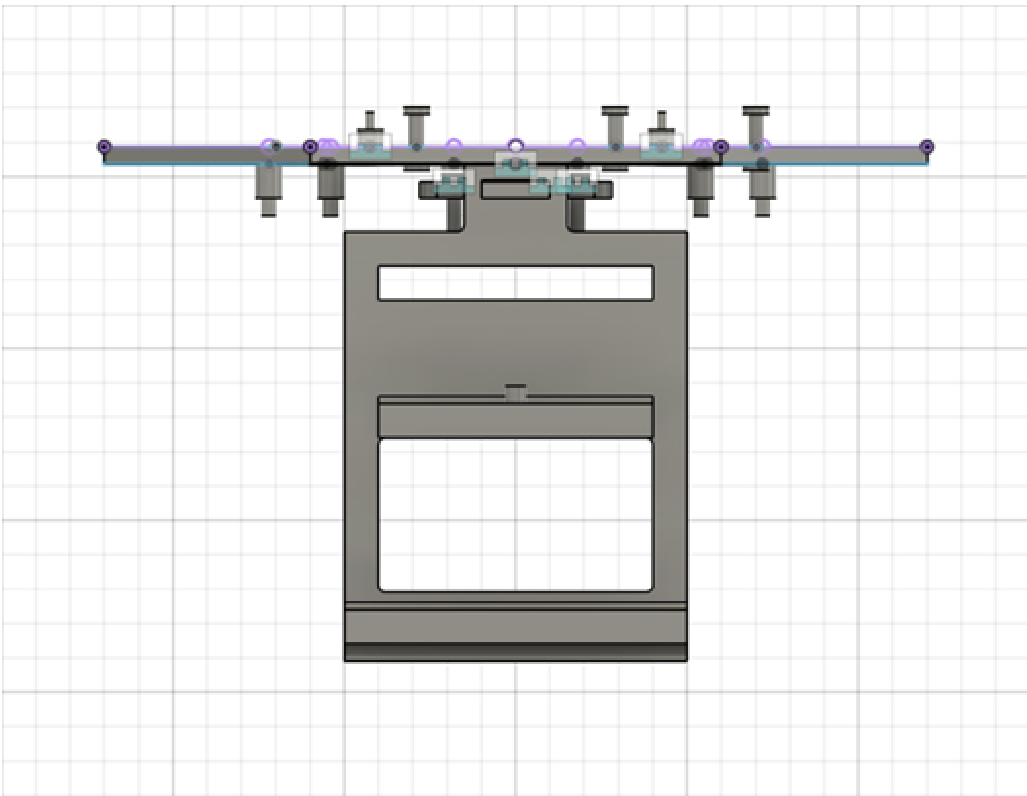

Fig. 16. Initial Design: Sled Front view