Final Design Review

A Final Design Review submitted to The University of Manchester for
Robotic Systems Design Project
in the Faculty of Science and Engineering

Year of submission
2025

Student ID
11548302, 11557189, 14122713, 11530725

School of Engineering

Contents

Contents **2**

1 Introduction **4**
1.1 Summary of the problem statement and objectives 4
1.2 Summary of how the team addressed feedback from Preliminary Design Review . . . 4
1.3 Summary of any modifications made since Preliminary Design Review 5

2 Sustainability Checklist **6**
2.1 Materials 6
2.2 Software 6
2.3 Energy 7
2.4 Waste 7
2.5 Emissions 8
2.6 Communications 8
2.7 Modularity 8
2.8 Location/Placement 9
2.9 Maintenance 9
2.10 Repurposing 9

3 Cyber Security Considerations **10**

4 System **10**

5 Mechanical Design **14**
5.1 Discussion of Design Trade-offs and Challenges with Component Placement 14
5.2 Design Files 22
5.3 Structural Strength 22
5.4 Manufacturability 24

6 Electrical Design **25**
6.1 Power connection diagram 25
6.2 Power Budget Analysis 27

7 Software Design **28**
7.1 Object Detection 29
7.2 Manipulator 29
7.3 Situational awareness and mobility 35
7.4 Github repository 39

8 Analysis **39**
8.1 Functional Environment Requirements 40
8.2 Functional Object Requirements 40
8.3 Functional Basic Mobility 40
8.4 Functional Rover Path Planning Requirements 41
8.5 Functional Object Manipulation and Retrieval Requirements 42

8.6 Functional Degree of Autonomy Requirements . . . 44
8.7 Product Design Size, Shape, Mass, and Style Requirements . . . 44
8.8 Product Design Cost Requirements . . . 44

9 Project Plan . . . 44

Appendices . . . 46

A Design Requirement Analysis . . . 46

B Team Workplace Charter . . . 54

1 Introduction

1.1 Summary of the problem statement and objectives

This document serves as the final design review of team 11, outlining the objectives and considerations for the development of a Leo Rover-based robot capable of autonomously retrieving coloured blocks and sorting them into designated storage bins based on their colour. This review aims at examining and verifying our initial concepts of design. Through this holistic review, the robot design will be compared and adjusted against the criteria established in the Design Requirements Analysis.

The central challenge addressed by this project is as follows:

”To develop an autonomous robot capable of identifying and retrieving coloured objects.”

This endeavour integrates a mobile robotic platform with various subsystems, including environmental perception sensors (Intel Realsense depth camera D435i, RPLIDAR A2-M12 360 degree laser scanner, Fisheye lens camera), mobility systems, and a manipulator (PincherX 150) capable of precise grasping and release. The project identifies four primary technical challenges:

1 :Achieve reliable environmental perception.

2 :Ensure capacity of navigation in an unmapped test field.

3 :Implement dependable object detection algorithms to identify coloured blocks accurately.

4 :Design an algorithm to control the manipulator for precise grasping, holding and releasing motions.

This review will delve into the design principles, component selection, and integration strategies employed to transform the Leo Rover into an autonomous robotic system tailored for the outlined objectives.

1.2 Summary of how the team addressed feedback from Preliminary Design Review

- **System Architecture:**
  The revised system block diagram effectively addresses the provided feedback by explicitly illustrating the interconnections between components rather than solely categorizing them. The updated diagram clearly depicts the interactions between computing units, peripherals, sensors, and power sources, ensuring a comprehensive representation of data flow and power distribution. Furthermore, each component is systematically described and justified, enhancing the clarity and structural coherence of the system architecture.

- **Electrical System:**
  The revisions to the electrical design, relative to the preliminary report, primarily pertain to the selection of supplementary power supplies. Our investigations revealed that the originally proposed power bank would be required to deliver power exceeding its rated output due to the higher-than-anticipated consumption of NUC Pro, thereby compromising system stability. Given that the Leo Rover must operate with considerable spatial freedom, a battery-powered configuration remains the most desirable option; consequently, this design has been provisionally retained. However, to address the worst-case scenario—namely, the unavailability of an onboard power supply device that meets the robot's requirements—we have incorporated an alternative solution involving a wired power supply.

- **Mechanical Design:**
  The 3D CAD model layers have been labeled to enhance the presentation effect. The component names have been modified and calibrated, and the alignment between images and descriptions has been optimized. The reason for mounting the LiDAR sensor beneath the chassis has been discussed, along with how the issue of blind spots in the sensor's field of view is addressed. Additionally, a load analysis of the key load-bearing structures of the model has been added.

- **Software systems:**
  Added support for Git repository documentation.

  Based on the feedback received, the documentation for mapping and navigation has been expanded to include detailed descriptions that outline the fundamental simulation process.

- **Design Requirements analysis:**
  The analysis of design requirements was redistributed among the team members responsible for the various sub-modules, in accordance with our established division of labor, to ensure that specialized expertise was applied to each component. We conducted a comprehensive re-evaluation of the current Leo Rover designs against the initial design requirements and subsequently refined and adjusted the system to enhance compliance with those specifications.

1.3 Summary of any modifications made since Preliminary Design Review

Object detection is implemented by subscribing to the topics published by the depth camera. By subscribing to the color topic, the system detects target objects using color information and determines their distance using depth data. By incorporating the camera's intrinsic parameters, the system converts the coordinates to the robotic arm's base coordinate frame, accurately determining the target object's position. Once the position is acquired, it is published as a topic, allowing other components to subscribe to it. Additionally, the system performs shape detection on the target object, enabling the robotic arm to adjust its posture based on the object's shape for more effective grasping.

The robot structure has been modified, with optimizations made to the suspension plate and radar baseplate structure. The protrusions on the suspension plate, which were used to secure the baseplate, have been removed, and the slotting position on the baseplate has been adjusted. This modification ensures secure fixation while facilitating easier laser cutting. Additionally, the thickness of all plates has been standardized and adjusted to ensure structural strength while minimizing material waste.

For the electrical system, the power consumption of each component was recalculated using the updated parameters. Additionally, an alternative design incorporating a wired power supply based on the original NUC Pro power adapter was integrated.

In the software design section, new content on the simulation and real-world implementation of the PincherX-150 robotic arm is added. It details the simulation of the manipulator using MoveIt, and the process of controlling the arm's movements in real-world implementation. Additionally, a discussion on the RQT graph of the manipulator is included, which explains the ROS communication structure, highlighting key nodes managing joint states, controllers, and real-time feedback.

In the preliminary report, mapping and navigation within the software design were only briefly introduced. Currently, we have completed the SLAM simulation for the Leo Rover, segmenting various sub-functions into distinct packages. The simulation outcomes demonstrate that the existing Leo Rover system is capable of performing SLAM in a static environment.

2 Sustainability Checklist

The following 10 points are guidelines for sustainable development of robots.

2.1 Materials

The current material selection prioritizes structural integrity and functionality. Structural components primarily use aluminium, steel, and ABS plastics, ensuring durability and ease of fabrication. Electronics and sensors incorporate PCB-based modules, lithium-ion batteries, and protective housings made from plastics or metal alloys. The robotic manipulator utilizes aluminium alloys for strength-to-weight efficiency, while the wheels are composed of rubber for optimal traction.

2.2 Software

To enhance power efficiency, the robotic system employs strategies such as algorithm optimization, intelligent sensor utilization, and hardware resource management. This includes refining SLAM and navigation algorithms, adopting lightweight object detection models, and implementing adaptive sensor scheduling to minimize energy consumption. Computational tasks are distributed between Intel NUC and Raspberry Pi 4, optimizing resource allocation and preventing redundant processing.

For communication efficiency, the system integrates data compression, lightweight protocols, and edge computing to reduce bandwidth usage and reliance on external servers. Effective data management is achieved through real-time filtering, selective storage, and data lifecycle policies, ensuring efficient use of storage resources.

In terms of sustainable AI model training, the system leverages pre-trained models, transfer learning, and cloud-based training to minimize computational demands. Additionally, modular software design, optimized coding techniques, and dynamic resource management contribute to improved performance, lower energy consumption, and enhanced maintainability, supporting long-term system sustainability.

2.3 Energy

Energy efficiency has been a key focus in the system design, with a comprehensive power budget analysis estimating a total power consumption of 72.71 W and a battery runtime of approximately 53 minutes. To address high power demands, a Lenovo Go USB-C Laptop Power Bank is proposed as a supplementary energy source for extended operation. Further energy reduction strategies include optimizing SLAM algorithms, adjusting sensor data collection frequency, implementing operation scheduling, and refining mechanical design to reduce weight and enhance mobility efficiency.

The robot currently relies on a 3S lithium-ion battery (11.1V, 5800mAh, 64.38Wh), with potential environmentally sustainable alternatives such as Lithium Iron Phosphate ($LiFePO_4$) and Sodium-Ion batteries, known for longer lifespans and reduced ecological impact. Future advancements could explore solar-assisted charging, renewable energy integration, or hydrogen fuel cells to enhance sustainability and operational efficiency.

2.4 Waste

The robotic system primarily consumes electrical energy, generating waste throughout its lifecycle, including electronic waste (e.g., lithium-ion batteries, electronic modules, sensors, and wiring), mechanical waste (e.g., structural components and worn rubber materials), and operational waste (e.g., packaging from component replacements). To minimize waste, preventative maintenance is recommended through regular system checks and software-driven monitoring to prevent premature failures. Software optimization can further reduce battery degradation, extending its lifespan.

For waste management, repair and upgrades should be prioritized over direct replacements, and recyclable materials such as metals and plastics should be clearly labeled for streamlined disposal. Certified recycling services should be engaged for end-of-life components, while packaging waste can be minimized through reuse and sustainable procurement practices. These strategies collectively support waste reduction and responsible disposal, enhancing the system's sustainability.

2.5 Emissions

The robotic system's potential pollution sources include chemical pollutants from lithium-ion batteries, material degradation from wheel wear and plastic components, and electronic waste from discarded sensors and modules. Improper disposal of batteries can lead to chemical leakage and fire hazards, while wheel and structural degradation may contribute to microplastic pollution. Additionally, e-waste poses risks of heavy metal contamination if not responsibly managed.

To mitigate these environmental impacts, safer battery alternatives with advanced management systems should be adopted, in conjunction with proper battery disposal protocols and collaboration with certified recycling centers. Sustainable materials such as bioplastics, bio-composites, and recycled rubber can reduce long-term degradation effects. Preventative maintenance can minimize wear-related pollution, while prioritizing repair and upgrades over replacements helps reduce e-waste generation. Implementing these strategies supports a more sustainable and environmentally responsible robotic system.

2.6 Communications

The robotic system processes and transmits extensive sensor data, including LiDAR scans, depth camera feeds, fisheye images, IMU readings, and encoder data, supporting navigation and object manipulation. Processed outputs such as SLAM-generated maps, navigation commands, and telemetry updates further contribute to data flow. While real-time sensor data and processed navigation outputs are essential, continuous high-resolution video streaming and frequent telemetry updates can be optimized to reduce transmission load.

To improve efficiency, selective data transmission should be implemented by limiting high-resolution sensor output to event-triggered scenarios. Adaptive transmission rates can further minimize unnecessary data flow during low-demand periods. Storage management strategies should focus on retaining only critical sensor data while automating periodic cleanup of outdated logs and redundant files, ensuring optimal data utilization without excessive storage consumption.

2.7 Modularity

The robotic system follows a modular hardware design, with distinct structural layers including a base computing platform, an upper manipulator section, and a lower sensor mounting area. Subsystems are integrated through dedicated mounting points, precision-manufactured components, and easily accessible fasteners, enabling efficient maintenance and component interchangeability. However, structural reinforcements and compact arrangements may slightly impact repair accessibility. To enhance modularity, adopting standardized mechanical and electrical interfaces, along with comprehensive assembly documentation, is recommended.

The modularity of the software is achieved through a ROS2-based architecture, where independent nodes manage SLAM, navigation, and sensor processing, allowing for easier debugging, updates, and system upgrades. However, some modules have tight dependencies, which may affect flexibility. Introducing additional abstraction layers for hardware interfaces and maintaining well-documented, structured code can further improve system adaptability and long-term maintainability.

2.8 Location/Placement

The robotic system is primarily deployed in controlled indoor environments, such as university laboratories, where direct environmental impacts are minimal. However, indirect effects may arise from transportation between storage and demonstration areas. Given the robot's compact size (9.25 kg), sustainable transport strategies are recommended. Manual carrying is feasible for short distances, while wheeled carts can minimize emissions for intra-campus movement. For longer distances, electric vehicles offer a lower-impact alternative. Centralized storage near demonstration areas can further reduce transport needs, and reusable, durable packaging should be utilized to minimize waste.

2.9 Maintenance

The robotic system follows a structured maintenance approach, including mechanical inspections of structural components, wheels, and sensors, alongside electrical system checks for battery performance and electronic integrity. To improve reliability, formalized maintenance schedules should be implemented, covering routine inspections of batteries, sensors, actuators, and computing units. Pre-operational checks can ensure system readiness, while predictive maintenance using real-time data analytics can help anticipate potential failures.

On the software side, the ROS2 architecture supports robust diagnostics through real-time monitoring of sensor outputs, battery health, and system status. The use of ROS2 diagnostic tools enables continuous performance tracking, allowing early detection of problems and improved system longevity.

2.10 Repurposing

The robot's modular and flexible design allows for diverse applications beyond its initial deployment. In education, it serves as a practical platform for student projects involving autonomy, SLAM, navigation, manipulation, and vision-based tasks. For research, it provides a versatile testing environment for developing and evaluating advanced localisation, mapping, and object detection algorithms. In addition, with the integration of additional sensors, it can support laboratory mapping and environmental monitoring applications, such as air quality assessments, further extending its usability.

3 Cyber Security Considerations

In the development and deployment of this project, it is critical to address potential cyber security risks to ensure the integrity and safety of the system. Given that the robot operates autonomously, is connected to wireless networks, and interacts with external sensors and devices, it is essential to secure all communication channels and data processing systems. Below are the primary cyber security considerations:

- **Data Encryption**:
  All data transmitted between the robot's components, including the sensors, controllers, and external devices, should be encrypted to prevent unauthorized access. The use of secure communication protocols such as TLS (Transport Layer Security) ensures that sensitive data, including navigation and sensor data, is protected during transmission.

- **Authentication and Authorization**:
  Access to the robot's control systems should be restricted to authorized users only. Implementing strong authentication methods (e.g., two-factor authentication) and role-based access control ensures that only trusted personnel can modify the system's settings or commands.

- **Network Security**:
  The robot's wireless connections should be secured using encryption methods like WPA3 for Wi-Fi or VPN for remote access. Additionally, firewall configurations should be applied to prevent unauthorized access from external sources.

- **Software Integrity**:
  The robot's software and firmware should be regularly updated to patch vulnerabilities. Code signing should be employed to verify the integrity of the software being executed on the robot, preventing malicious code from being uploaded and executed.

- **Incident Response Plan**:
  A protocol for detecting, responding to, and recovering from cyber-attacks should be established. This includes monitoring the robot's network activity for suspicious behavior and ensuring that logs are securely stored for later analysis.

By implementing these security measures, the system will be better protected against cyber threats, ensuring both the safety and functionality of the robot.

4 System

This robotic system combines cutting-edge sensing, computing, and mobility technologies to achieve autonomous operation with remarkable precision and efficiency. At its heart, a Battery powers the system, ensuring a reliable energy supply to all components through the LEO Powerbox, a centralized unit that manages and distributes power to essential modules, such as the LEO Core Board and Intel NUC.

The LEO Core Board functions as the motion controller, driving DC Motors with Encoders for accurate movement and leveraging an integrated IMU to provide real-time data on orientation, acceleration, and angular velocity, ensuring smooth navigation. The Raspberry Pi 4 Model B acts as a communication bridge, processing sensor data from the Camera and IMU while coordinating interactions between the LEO Core Board and Intel NUC.

For perception, the system utilizes advanced sensors like the Intel Realsense Depth Camera for 3D environmental mapping and object detection, the RPLIDAR A2M12 for creating 2D maps through 360-degree laser scanning, and a Camera for visual processing. These sensors enable the robot to perform Simultaneous localisation and Mapping (SLAM) and obstacle avoidance, with the Intel NUC serving as the primary processor for complex algorithms and path planning.

The system also features a 2.4GHz WiFi Modem for wireless connectivity, facilitating remote control and data exchange. The Trossen PincherX 150 Manipulator adds an extra layer of functionality, offering advanced grasping and manipulation capabilities to interact with objects.

This sophisticated integration of power, mobility, sensing, and computing enables the robot to excel in autonomous navigation, environmental awareness, and interaction, making it ideal for a variety of advanced robotics applications.

- **Battery**:
  Provides the entire robotic system with the power it needs to operate autonomously.
- **LEO Powerbox**:
  Acts as a power management unit, regulating and distributing power to various components such as the LEO Core Board and Intel NUC.
- **LEO Core Board**:
  Responsible for driving DC motors and reading encoder feedback for precise movement and navigation of the robot chassis.
  Connects to the built-in IMU to provide orientation and motion information.
- **DC Motors with Encoders**:
  Provides robot mobility and precise motion control through encoder feedback.
- **Raspberry Pi 4 Model B**:
  Used to read camera and IMU data and pass sensor data between the Intel NUC and the LEO Core Board.
- **Camera**:
  Provides visual information for target detection and environment perception.
- **LEO Core built-in IMU**:
  Transmits real-time direction, acceleration, and angular velocity information to aid navigation and motion control.
- **2.4GHz WiFi Modem (WiFi Modem + Antenna)**:
  Provides wireless communication capability for remote control and data transfer.

- **Intel NUC**:
  As the main computing platform, it runs complex SLAM (Simultaneous Localisation and Map Building) algorithms and path planning.
- **RPLIDAR A2M12**:
  Generates a 2D map of the robot's surroundings via 360-degree laser scanning for navigation and obstacle avoidance.
- **RPLIDAR USB Adapter Board**:
  Responsible for communication and data transfer between RPLIDAR and Intel NUC.
- **Intel Realsense Depth Camera**:
  Provides 3D perception for environment modelling and target detection.
- **Trossen PincherX 150 Manipulator**:
  Responsible for performing grasping and manipulation tasks for interacting or manipulating objects.

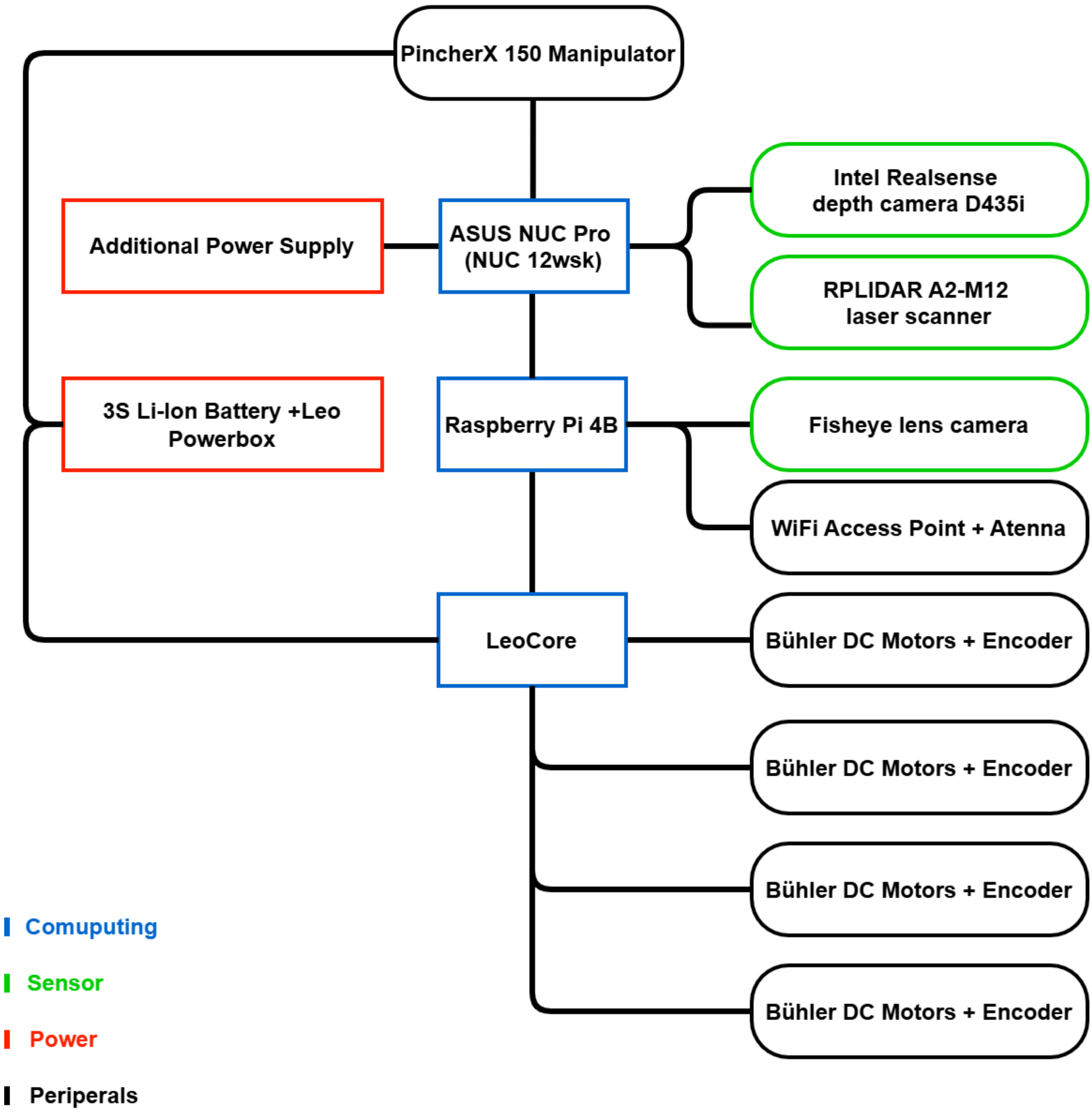


Fig. 1. System block diagram

5 Mechanical Design

5.1 Discussion of Design Trade-offs and Challenges with Component Placement

Fig. 2. 3D CAD model design

- **Base Computing Level (L0)**: Fixes the entire structure to the robotic vehicle.
- **Upper Manipulation Level (L1)**: Secures the robotic arm.
- **Lower Scanning Level (L-1)**: Holds the LiDAR.

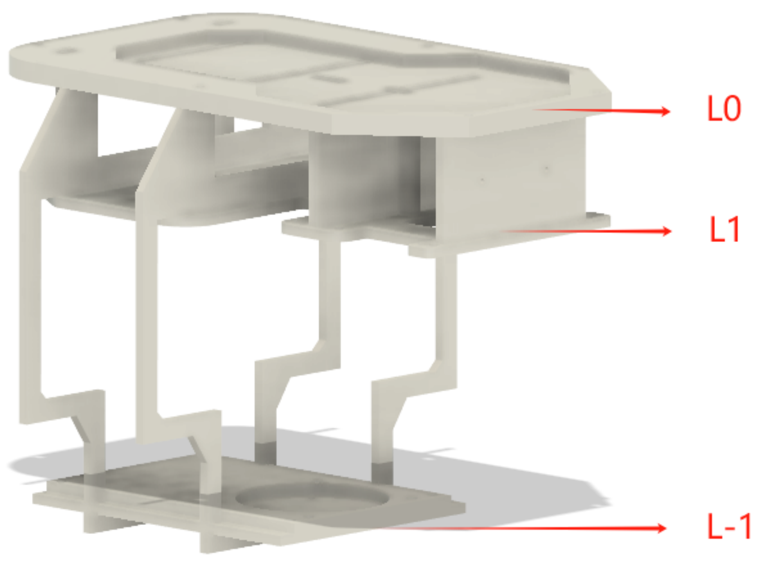


Fig. 3. Hierarchical annotation

5.1.1 Base Computing Level (L0)

The robotic vehicle has mounting holes only on its top surface, necessitating the creation of a baseplate, referred to as the Base Computing Level (L0), to serve as the foundation for the entire structure. This base plate is secured to the vehicle using screws and acts as a crucial connecting point for both the Upper Manipulation Level (L1) and Lower Scanning Level (L-1).

Fig. 4. Baseplate

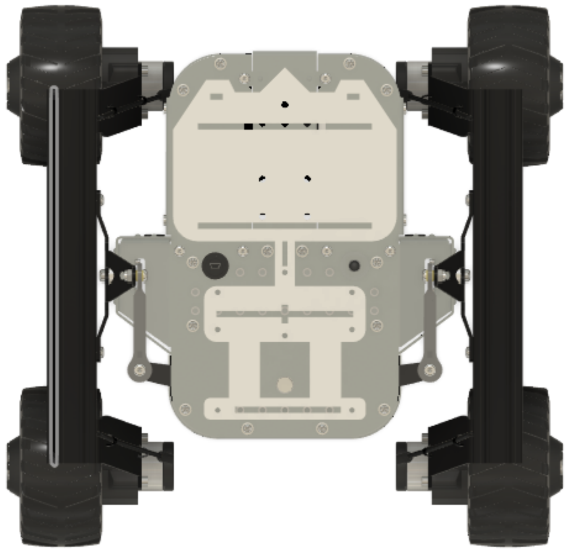


Fig. 5. Rendering of the installed baseplate

The base plate is perforated in appropriate locations for secure mounting and slotted to accommodate the support plates. It is designed to maximize surface area while avoiding existing protruding screws on the top of the robot, thereby enhancing overall stability.

5.1.2 Upper Manipulation Level (L1)

To secure the robotic arm, an upper plate is installed above the Base Computing Level (L0). The Upper Manipulation Level (L1) consists of two sub-plates:

- **Lower Top Plate**: Supports and secures the robotic arm, with slots for inserting support plates, as shown in Fig. 6.
- **Upper Top Plate**: Features a groove in the center, enabling the arm's base to nestle securely. This design minimizes vibrations during vehicle movement. The two top plates are fastened together with screws to ensure robustness, as shown in Fig. 7.

Fig. 6. Lower Top Plate

Fig. 7. Upper Top Plate

After installing the top and bottom plates, the structure appears as shown in Fig. 8, with the robotic arm base fitting into the groove.

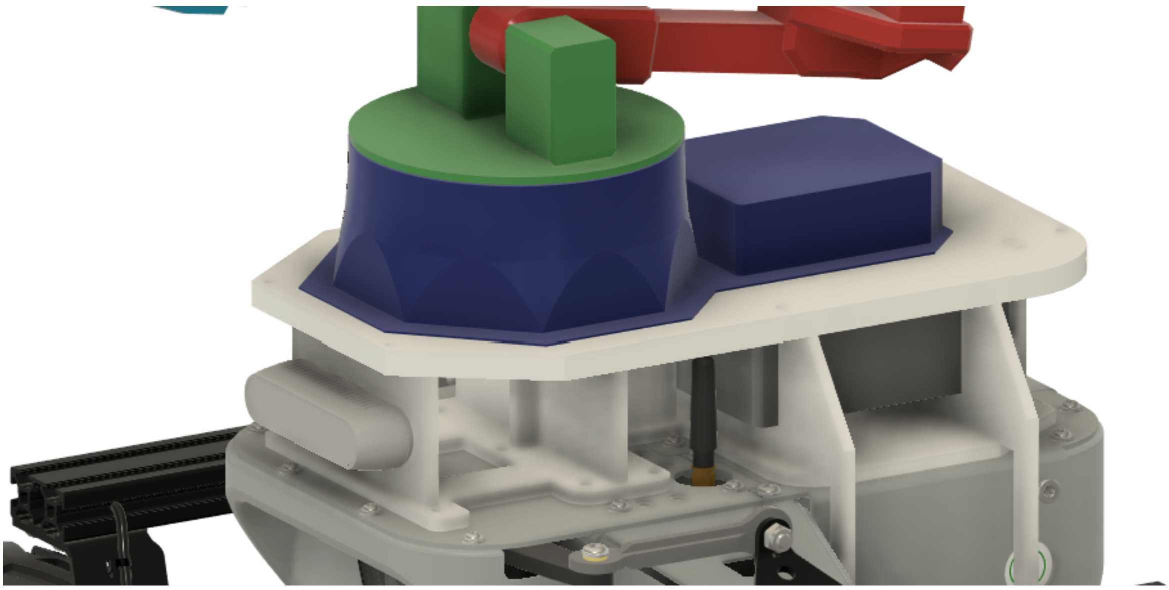


Fig. 8. Upper Manipulation Level (L1)

5.1.3 Lower Scanning Level (L-1)

The LiDAR, with its limited pitch angle, is positioned beneath the robotic vehicle to improve its ability to detect environmental obstacles. This placement requires a support structure extending downward from the Base Computing Level (L0) to hold a platform. The platform consists of two plates:

- **Lower Bottom Plate**: The lower layer of the LiDAR base plate features slots on both sides for inserting suspension plates, providing support while preventing planar misalignment.

Fig. 9. Lower Bottom Plate

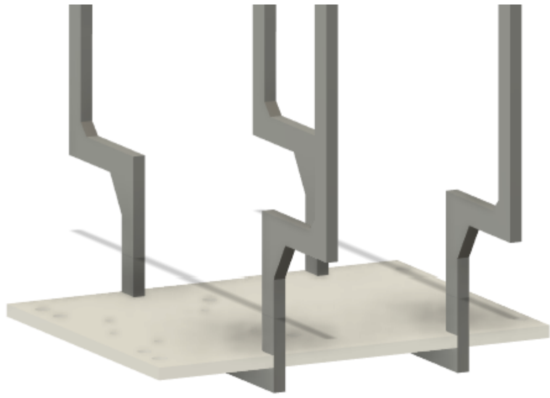


Fig. 10. Lower Bottom Plate inserted into the suspension plate.

- **Upper bottom Plate**: Positioned above the lower plate, the upper plate is fastened to the lower plate using screws. This double-layer design simplifies the assembly process and enhances the platform's stability, ensuring reliable operation of the LiDAR.Grooves are also designed, allowing the LiDAR to be mounted either using its own base or directly onto the bottom plate.

Fig. 11. Upper bottom Plate

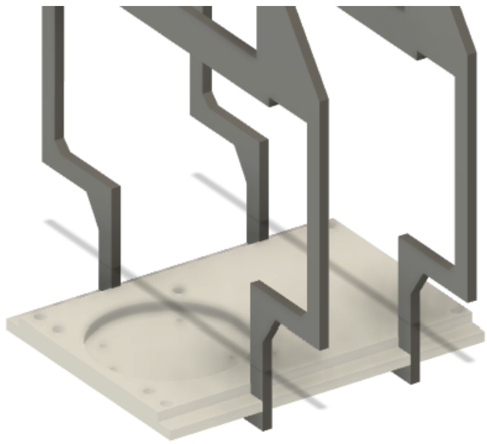


Fig. 12. Upper and lower layers of the bottom plate assembly.

Through testing, it was found that the LiDAR has a elevation angle of approximately 1.3°. Due to this angle, obstacles outside this range cannot be detected, which affects navigation functionality. The higher the LiDAR is mounted, the less likely it is to detect nearby obstacles. Therefore, the LiDAR is placed at the bottom of the robot, ensuring detection of both distant high obstacles and nearby small obstacles.

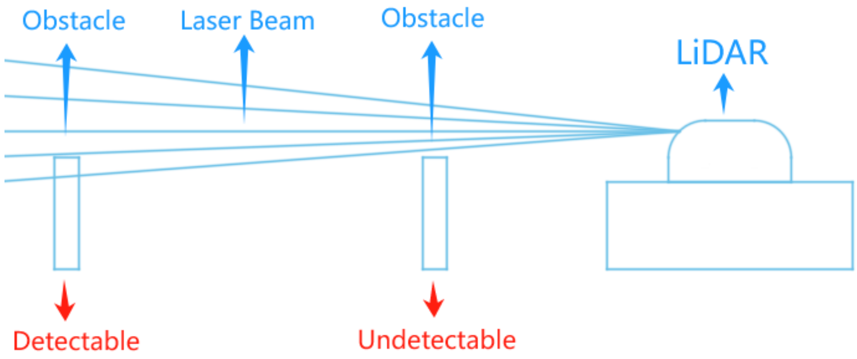


Fig. 13. Elevation angle diagram

Additionally, the bottom has a relatively clear field of view, with only the four wheels causing minor obstruction, whereas multiple sensors and support plates at the L0 layer, as well as the robotic arm at the L1 layer, cause greater obstruction of the field of view.

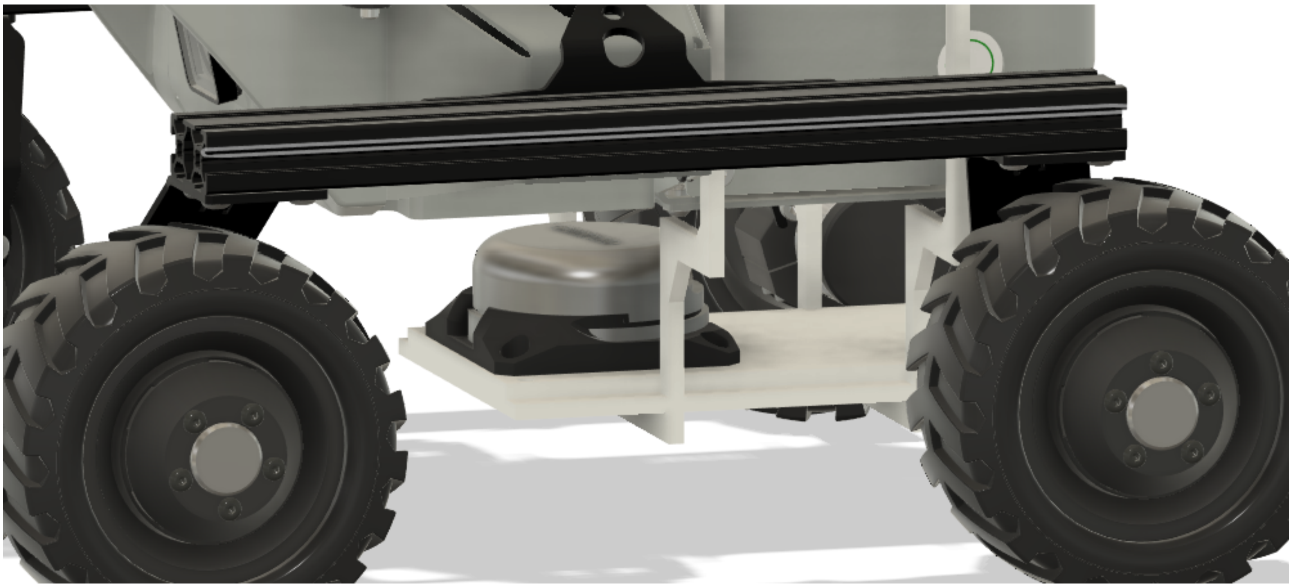


Fig. 14. L-1

After discussing the rationale for mounting the lidar sensor beneath the chassis of the Leo Rover, the following section addresses the challenges associated with this design. The primary distinction between this under-chassis configuration and traditional setups is that the sensor's scanning may result in a larger blind area. In conventional configurations, blind spots are typically confined to the sides or rear of the vehicle due to the installation of brackets and other components, while also maximizing the working space for the manipulator. In contrast, mounting the lidar underneath causes the front wheels to obstruct a broader range of directions, potentially leading to collisions with obstacles in these blind regions and thereby affecting the efficiency of mapping and localisation.

However, an analysis of the Leo Rover's steering performance and the orientation of the blind areas suggests that the rover's turning radius is sufficient to detect and avoid obstacles before they become critical. Furthermore, although the increased blind area may initially reduce the volume of mapping data, the impact on overall mapping efficiency diminishes as the rover begins to execute autonomous navigation maneuvers.

5.1.4 Structural Connections and Reinforcements

To connect the upper plates to the Base Computing Level (L0), two horizontal plates are introduced:

- **Front Horizontal Plate**: Features holes to mount a depth camera.
- **Middle Horizontal Plate**: Includes grooves for joinery connections (mortise and tenon) to reinforce the structure.

Fig. 15. Front Horizontal Plate

Fig. 16. Middle Horizontal Plate

At the rear, the horizontal plate is replaced by the extended LiDAR support structure, which raises its height to support the Upper Manipulation Level (L1). The extended supports also include grooves for accommodating the main control unit.

Fig. 17. Extended Lidar support structure

5.1.5 Addressing Stability and Vibration Challenges

Given the potential instability of the horizontal plates when subjected to front-to-back vibrations, additional reinforcements were introduced:

- A vertical plate is added to the middle horizontal plate using a mortise and tenon connection, enhancing structural rigidity.

Fig. 18. Middle vertical plate

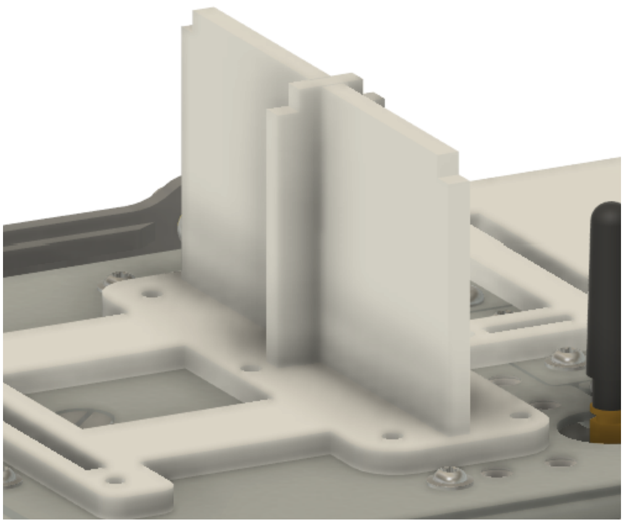


Fig. 19. Assembly of the middle vertical plate

- To secure the main control unit,a vertical plate is added in front of the front suspension plate, preventing the unit from sliding forward. This plate also stabilizes the Upper Manipulation

Level (L1) against front-to-back vibrations.

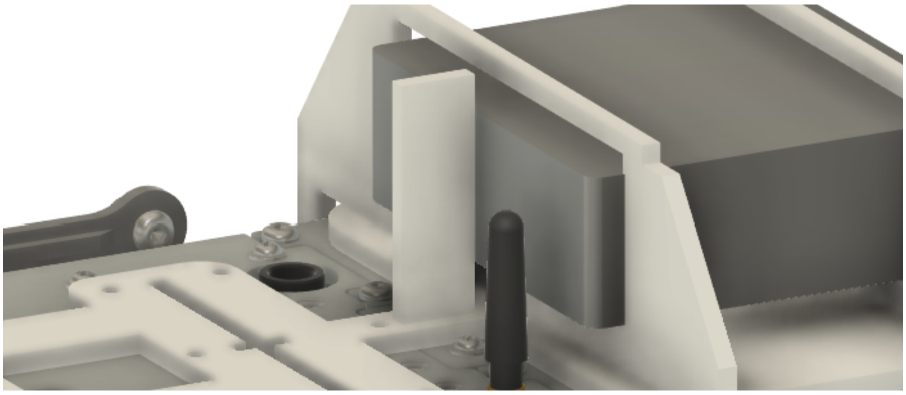


Fig. 20. Vertical columns

- Two vertical columns are added behind the rear suspension plate, preventing the main control unit from sliding backward and reducing vibrations. These columns extend through the Upper Manipulation Level (L1) and Base Computing Level (L0), ensuring that the main control unit remains secure while allowing easy cable management.

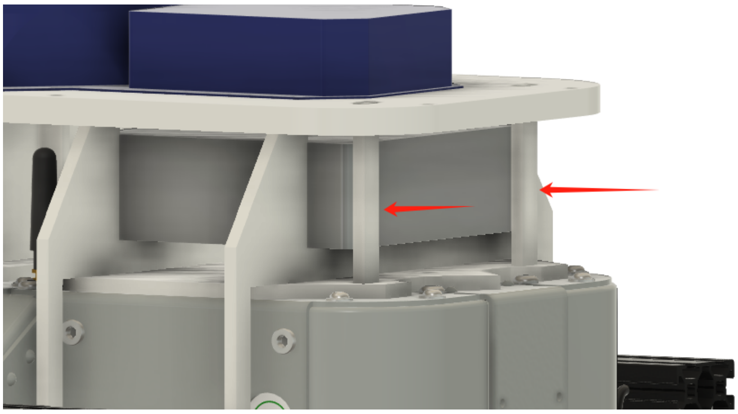


Fig. 21. Vertical columns

5.1.6 Design Trade-offs

- **Stability vs. Assembly Complexity**: The addition of multiple layers and reinforcement structures significantly improves stability and vibration resistance but increases the complexity of assembly and manufacturing.
- **Weight vs. Functionality**: The use of double-layer platforms and reinforced connections adds weight but ensures the reliability and durability of the structure during operation.
- **Space Constraints**: The tight integration of components such as the robotic arm, LiDAR, depth camera, and main control unit requires precise spatial planning and compromises in component placement.
- **Modularity vs. Flexibility**: The design does not incorporate 3D printing of the entire platform as a single piece, but rather consists of multiple individual components. The goal is to

enhance the portability and flexibility of the platform, allowing for future component replacements as needed.

5.2 Design Files

The design files show in Figure 21 are the overall dimensions and clearances of the frame for the rover.

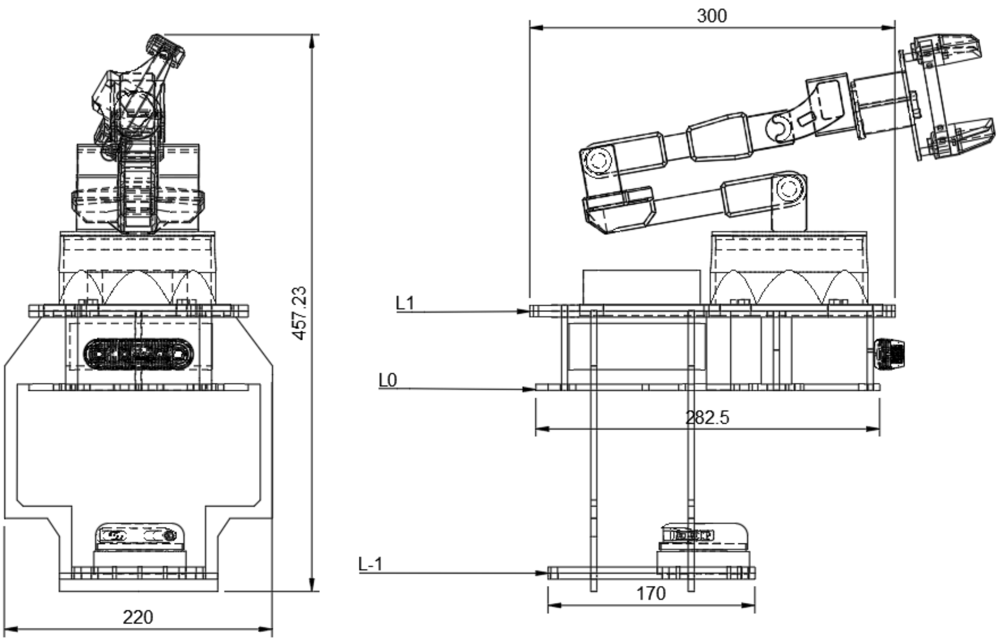


Fig. 22. Engineering drawing

5.3 Structural Strength

In the design of this robotic platform, the most critical components likely to bear complex loads are the two rear suspension plates. These suspension plates not only support the L1 layer to ensure the stable installation of the robotic arm but also feature a central slot to secure the main control unit while extending downward to support the L-1 platform. Due to their multifunctional load-bearing role, they are the most structurally complex and require the most detailed analysis.

From a loading perspective, the suspension plates may face two primary failure modes: bending deformation and shear failure. The upper section, which supports the L1 layer, mainly bears the robotic arm's load and the dynamic forces generated during its operation. The central slot region may lead to localized stress concentration, while the lower extension supporting the L-1 platform must withstand the load transferred from the bottom platform.

To further verify the structural stability of the suspension plates, we conducted static structural analysis to simulate their performance under real working conditions. During the analysis, we assumed a maximum platform load of 5 kg and accounted for the additional torque and vibration effects caused by the robotic arm's movements.

Additionally, to enhance structural strength and minimize deformation, we adopted a multi-point fixation approach, ensuring that the loads are distributed evenly. We also optimized the material thickness and the placement of supporting ribs, further improving the overall rigidity and reducing the risk of structural failure.

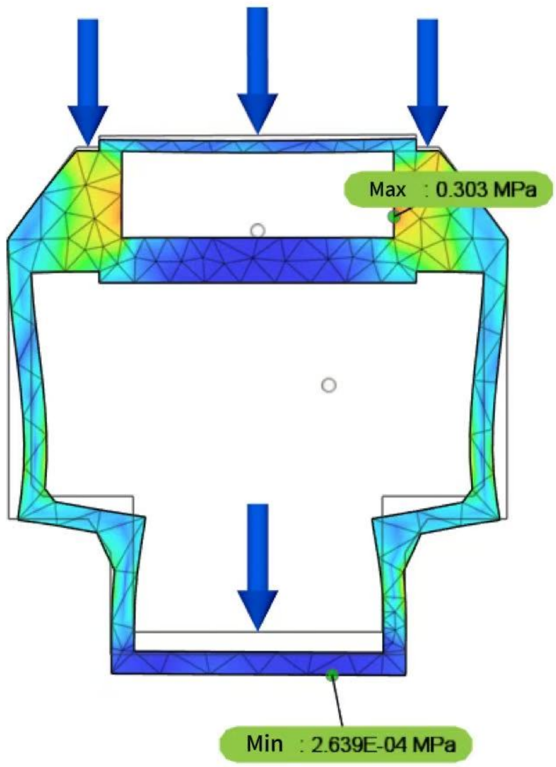


Fig. 23. Stress type: Von Mises, Material Compressive Yield Strength (ABS): 40–60 MPa

To enhance the strength and resistance to deformation at the right-angle corners of the suspension plates, we have incorporated reinforcing diagonal connections at all internal corners. These diagonal connections serve the following key purposes:

- **Reducing stress concentration:**:Right-angle corners are prone to stress concentration, which can lead to material cracking or deformation under load. The diagonal connections provide a smooth transition for load distribution, preventing excessive localized stress.
- **Enhancing overall rigidity:**:The diagonal connections form a triangular support structure, which is significantly more stable than a simple right-angle connection, thereby increasing the overall strength of the suspension plates.
- **Preventing deformation or failure:**When the suspension plates bear the robotic arm's weight,

the load from the L1 layer, and the support force from the L-1 platform, the diagonal connections effectively distribute the forces, preventing structural deformation or failure.

5.4 Manufacturability

To ensure efficient manufacturing of the bracket design, laser cutting technology is employed, offering several key advantages:

Laser cutting provides high precision and efficiency, making it ideal for mass production. Compared to 3D printing, laser cutting is faster, has higher material utilization, and is more cost-effective, especially for processing metals and plastics. Additionally, laser cutting requires no molds, offering greater flexibility and adaptability to different materials and thicknesses, thus reducing preparation and machining time. In comparison to CNC machining, laser cutting is quicker, simpler, and better suited for large-scale production. It also outperforms stamping in handling complex shapes and fine details.

By utilizing laser cutting, the design achieves precise cuts and quick assembly while minimizing post-processing and refinement, improving overall manufacturing efficiency and quality.

6 Electrical Design

6.1 Power connection diagram

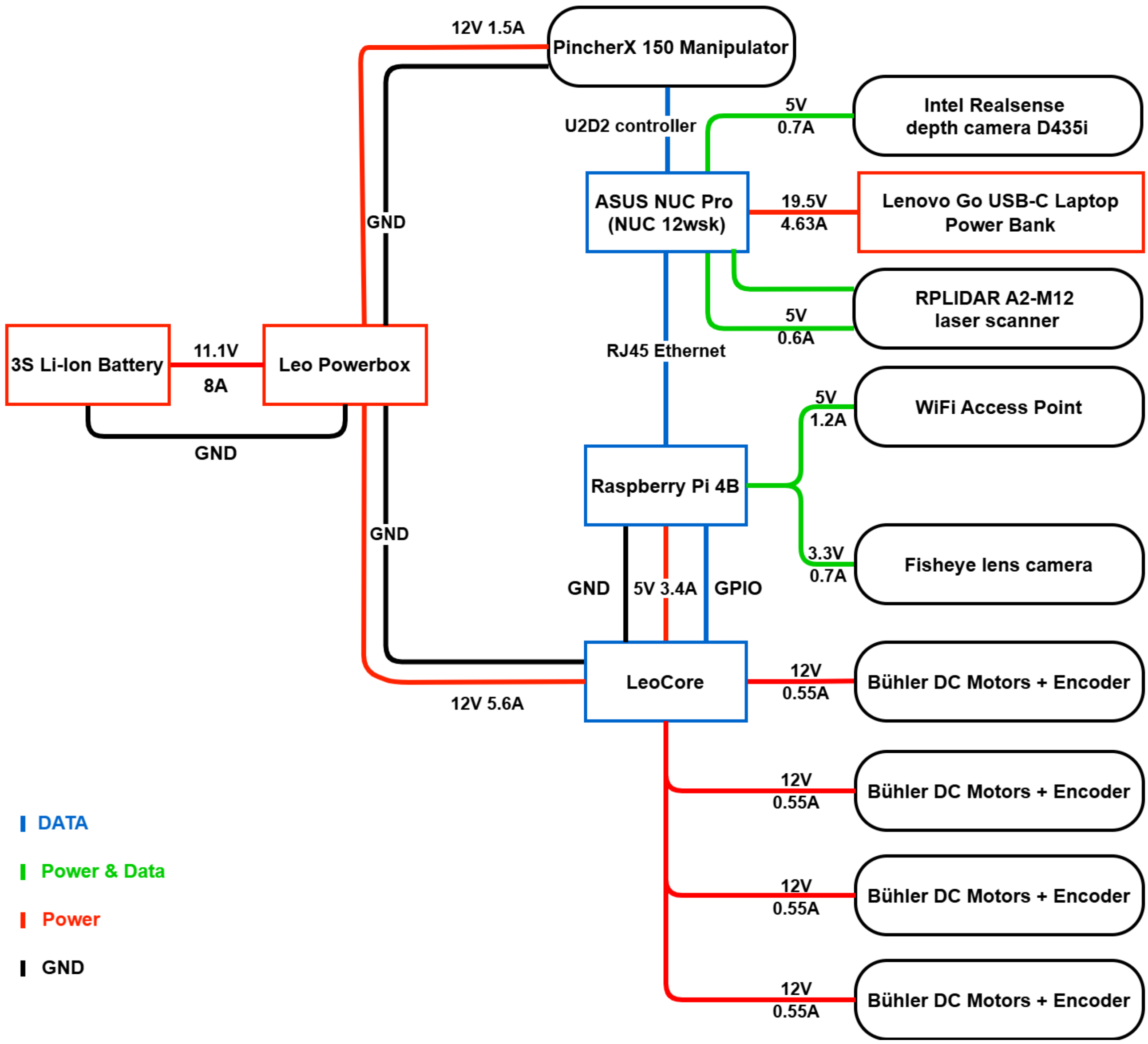


Fig. 24. Power connection diagram (original)

Fig. 24 illustrates the original connection design of NUC-integrated plan.An additional power bank will be implemented to power NUC. In this design, depth camera and LiDAR sensors will be coneected directly to NUC, utilizing the NUC's USB ports for both data transmission and power supply.

Based on further investigation, our measurements indicate that the power consumption of the NUC Pro varies from approximately 90W to 120W depending on the CPU configuration. For models equipped with an i7 or i5 processor, which is in our case, the unit requires a 20V DC supply at 6.0A (totaling approximately 120W). These values exceed the maximum rated output of our current power solution, which is a 65W Power Bank.

This discrepancy is critical, as our initial calculations were based on the power consumption figures provided by the power adapter, not accounting for the higher demands that occur during actual operation. Given the higher power draw, it is anticipated that the Power Bank may not provide sufficient energy, leading to potential under-voltage conditions or even system instability during peak loads.

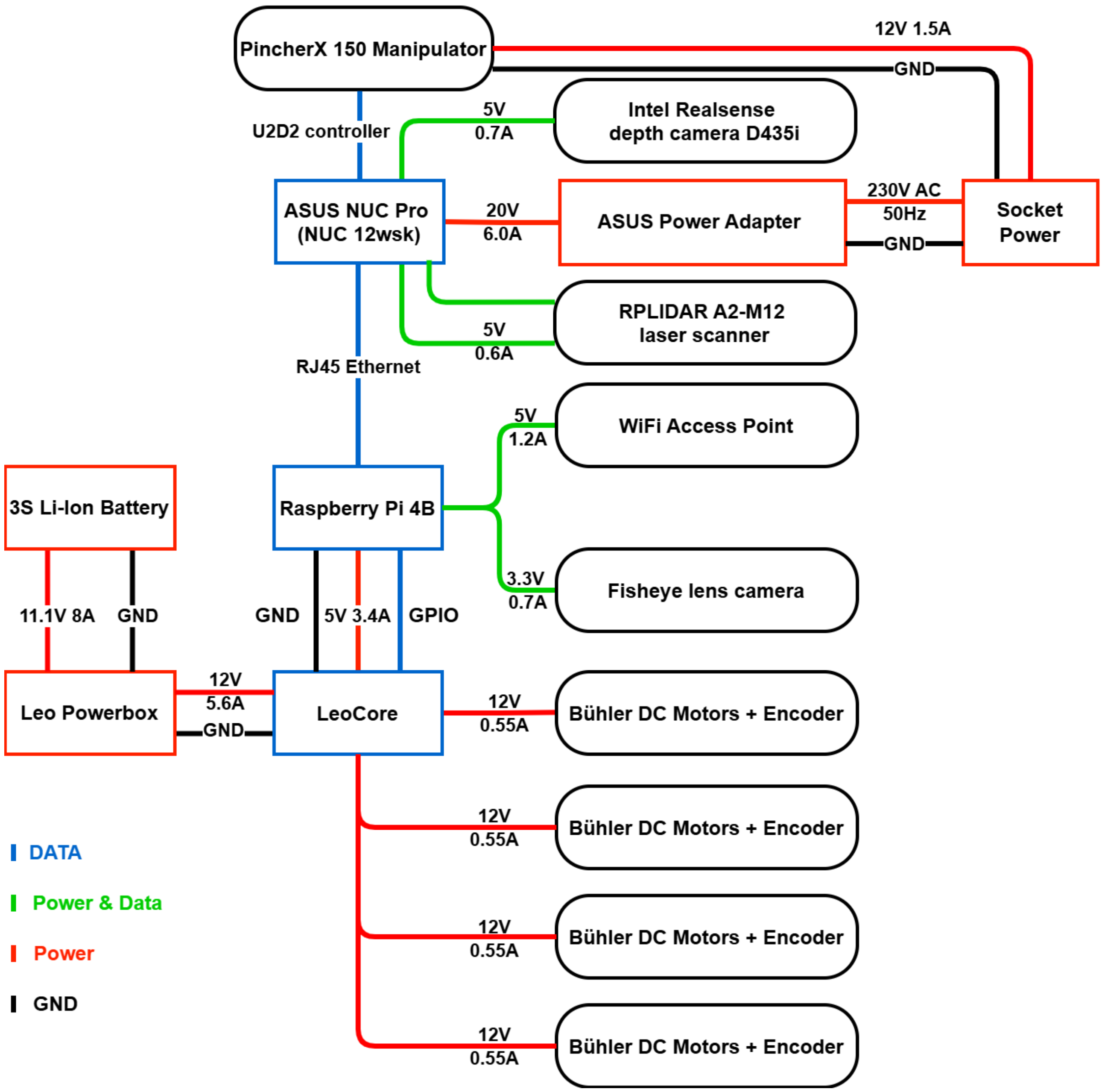


Fig. 25. Power connection diagram (updated)

To address this issue, we plan to conduct further testing with the existing Power Bank to assess its performance under realistic operating conditions. If these tests confirm that the Power Bank is unable to meet the NUC Pro's power requirements reliably, our backup strategy will be to connect the NUC Pro directly to an outlet power source, illustrated in Fig. 25. This alternative would ensure a stable power supply that can accommodate the higher consumption rates. It is also noteworthy that no viable method was identified for connecting the manipulator to a battery power source. Consequently, given this limitation, an external power supply analogous to the one used for the NUC Pro must be employed.

Table 1. Power Budget for Leo Rover Robot

Product	Supply Current (A)	Supply Voltage (V)	Rated Power (W)
Intel Realsense D435i	0.7	5	3.5
RPLIDAR A2-M12	0.6	5	3
ASUS NUC Pro (Computer)	4.63 - 6.0	19.5	90.29 - 117
PincherX 150 (Manipulator)	1.5 (est.)	12	18
Bühler DC Motors + Encoder	0.55 (x 4)	12	6.6 (x 4)
Fisheye Lens Camera	0.7 (est.)	3.3 (est.)	2.31
WiFi Access Point	1.2	5	6
Raspberry Pi 4B	1.5 (est.)	5	7.5
LeoCore (Microcontroller)	0.5 (est.)	12	6

6.2 Power Budget Analysis

6.2.1 Total Power Consumption

The power consumption of the Leo Rover robot is derived by summing the current requirements of all connected components, grouped by their operating voltage. Components powered at 5V, including the Intel Realsense D435i (depth camera), RPLIDAR A2-M12 (lidar), WiFi access point, and Raspberry Pi 4B (on-board computer core), consume a combined current of 4.0 A, resulting in a power requirement of 20.0 W. Similarly, components operating at 12V, such as the PincherX 150 (manipulator), Bühler DC Motors (actuator), and LeoCore (real-Time microcontroller), require a total of 4.2 A, translating to 50.4 W. Lastly, the fisheye lens camera, running on 3.3V, consumes 0.7 A, equivalent to 2.31 W (estimated value). Summing these values, the overall power requirement is approximately 72.71 W.

6.2.2 Battery Capacity Analysis

The primary power source for the Leo Rover is a 3S Li-Ion battery with an output voltage of 11.1V, a capacity of 5800 mAh, and a maximum current rating of 8A. The battery's total energy capacity is calculated as:

$$\text{Capacity (Wh)} = \text{Voltage (V)} \times \text{Capacity (Ah)} = 11.1\,\text{V} \times 5.8\,\text{Ah} = 64.38\,\text{Wh}.$$

Under nominal operating conditions, the power draw of the robot is approximately 65 W. Using this, the estimated runtime of the robot can be computed as:

$$\text{Battery Life (hours)} = \frac{\text{Battery Capacity (Wh)}}{\text{Power Draw (W)}} = \frac{64.38}{72.71} \approx 0.89\,\text{hours}.$$

Our evaluation also revealed that the manipulator is not equipped with an interface to utilize the original Leo Rover battery. Consequently, the PincherX 150 Manipulator is designed to operate with an external power supply, which will be similar to NUC Pro. This configuration is anticipated to provide sufficient energy to support continuous operation for over 1 hour.

6.2.3 Conclusion

Given the estimated runtime, the 3S Li-Ion battery alone is sufficient to support the basic operation of robot for demonstration periods, which will be around 20 minutes. However, the voltage output by the battery is not suitable for supporting NUC Pro. Even if the NUC Pro is forced to be powered by the robot's battery, its maximum battery life will drop to less than half an hour, which is too risky for a demonstration of about 20 minutes.

To directly address this limitation, it is recommended to integrate an additional power source. One viable option is the Lenovo Go USB-C Laptop Power Bank, which provides an output of up to 20V and 65W, offering supplementary energy to meet the extended runtime requirements. However, further research has indicated that the output power provided by the power bank solution is unlikely to meet the operational requirements of the NUC Pro. Operating the host at a power level below its rated input may lead to system instability. Additionally, procuring a power supply device that both meets the necessary power specifications and provides appropriate monitoring indicators presents significant challenges. As an alternative, we propose using the original NUC Pro power adapter connected directly to the mains socket. This approach ensures that the host system is supplied with power within its specified rating, thereby enhancing overall stability and reliability.

Remotely deploying computers is also a possible solution, which can once and for all eliminate the need for the robot to consider powering the NUC. Using a wifi module and related protocols (such as SSH) as a communication bridge to control the robot may be a better solution than adding an additional power supply module. Nevertheless, this solution presents significant challenges regarding the transmission of sensor feedback and the processing of control signals. Additionally, uncertainty remains as to whether the Raspberry Pi motherboard is fully compatible with the aforementioned sensors.

By carefully managing the power budget and selecting appropriate supplementary energy sources, the Leo Rover robot can achieve reliable and sustained operation for its intended tasks.

7 Software Design

This section encompasses the development and integration of several critical functionalities, including object detection, manipulator control, SLAM (Simultaneous Localization and Mapping), and navigation. This section details how sensor data is processed and fused to enable the autonomous perception of the environment, while simultaneously facilitating precise control of the robot's manipulator. Object detection algorithms are employed to identify and classify coloured bricks within the operational environment, providing essential input for subsequent manipulation tasks. Concurrently, the robotic arm control subsystem translates high-level commands into low-level actuator instructions, ensuring accurate and repeatable grasping actions. Furthermore, the SLAM module utilizes data from various sensors—such as lidars and depth cameras—to build a reliable occupancy grid map and to continuously localize the robot within that map. Finally, the navigation subsystem leverages this localization information, along with real-time sensor feedback, to perform path planning and obstacle avoidance, thereby enabling the Leo Rover to maneuver autonomously through its environment. Together, these integrated software components form a robust framework for achieving the intended autonomous operations of the Leo Rover.

7.1 Object Detection

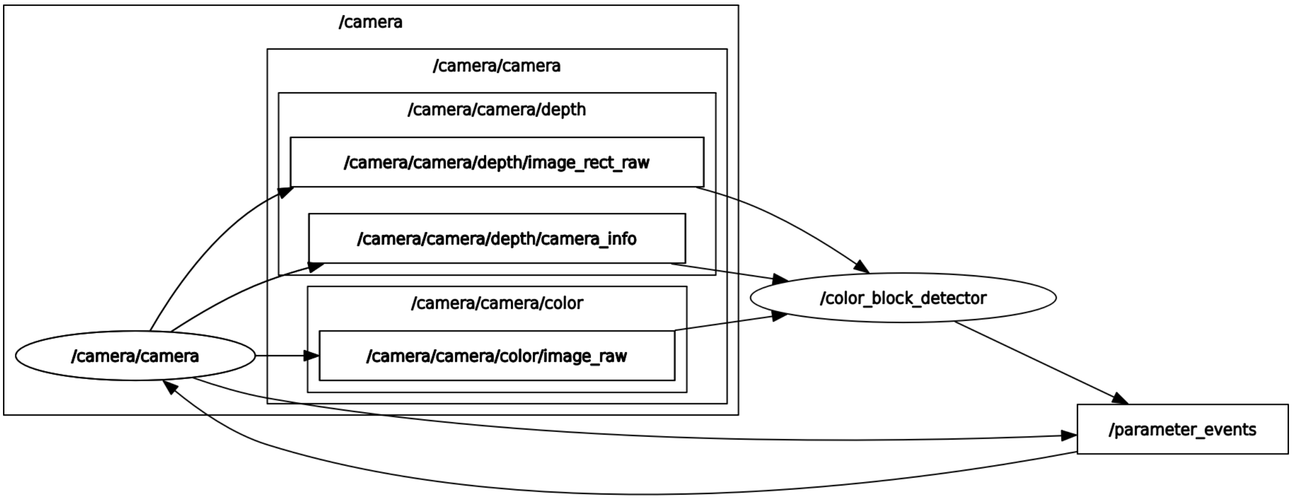


Fig. 26. RQT Graph depth camera section

By subscribing to three topics published by the Realsense Camera:

- ”/camera/camera/color/image_raw” for target object detection based on color,
- ”/camera/camera/depth/image_rect_raw” for determining the depth of the target object,
- ”/camera/camera/depth/camera_info” for coordinate transformation of the target object.

The ColorBlockDetector node processes and analyzes the target object’s position using the information from these topics.For objects detected by color recognition, shape detection is then carried out. The robotic arm can adjust its posture based on the specific shape of the object to achieve stable and secure grasping. Finally, the obtained coordinates and shape of the target object relative to the depth camera are published to the ”object_center_distance” topic for use by other devices, such as the robotic arm.

7.2 Manipulator

The PincherX-150 is a robotic manipulator arm developed by Trossen Robotics, which provides high torque, efficient heat dissipation, and enhanced durability within a compact form factor. In the context of our project, the arm addresses the need for performing tasks such as object retrieval and manipulation in indoor environments. Its design is particularly suited for applications requiring a combination of reach, payload capacity, and dexterity.

7.2.1 Simulation of the PincherX-150 Arm Using MoveIt

In this section, we detail the simulation process of the PincherX-150 robotic arm using MoveIt, a motion planning framework integrated with ROS (Robot Operating System). MoveIt facilitates the development of complex robotic applications by providing tools for motion planning, manipulation, 3D perception, kinematics, control, and navigation.

Initially, we constructed an accurate model of the PincherX-150 arm using the Unified Robot Description Format (URDF), specifying its physical properties such as joint types, link dimensions, and kinematic relationships. This model served as the foundation for the simulation environment. We employed Gazebo, a high-fidelity robotics simulator, to create a virtual workspace that closely mirrors the arm's intended operational setting. Gazebo's robust physics engine and sensor plugins enabled the replication of realistic interactions within the simulated environment.

To integrate the PincherX-150 arm with MoveIt, we utilized the MoveIt Setup Assistant to generate the necessary configuration files, including planning groups, end effectors, and virtual joints. This integration facilitated the definition of motion planning parameters and the selection of appropriate motion planners. MoveIt's seamless integration with the Open Motion Planning Library (OMPL) provided access to advanced motion planning algorithms, such as RRTConnect and PRM, which were utilized to compute feasible and efficient paths for the arm's movements.

In our simulation, we established a coordinate system based on the ”base_link” frame to define specific target positions for the robotic arm. This setup enabled us to direct the arm to reach precise coordinates within the simulated environment. We utilized MoveIt's motion planning capabilities to plan and execute the arm's movements accurately. Additionally, we conducted grasping tests to evaluate the arm's effectiveness in securely handling objects at designated positions. MoveIt's collision checking functionalities ensured that the arm's movements were safe, effectively preventing interactions that could lead to system errors or inefficiencies.

The simulation also incorporated trajectory planning and execution components, enabling the generation of time-parameterized paths that the arm could follow smoothly and accurately. As shown in figure below, real-time visualization tools in MoveIt provided immediate feedback on the arm's performance, allowing for adjustments and optimizations to be made promptly. This iterative process ensured that the arm's movements adhered to predefined goals and constraints, enhancing the reliability and effectiveness of the robotic system.

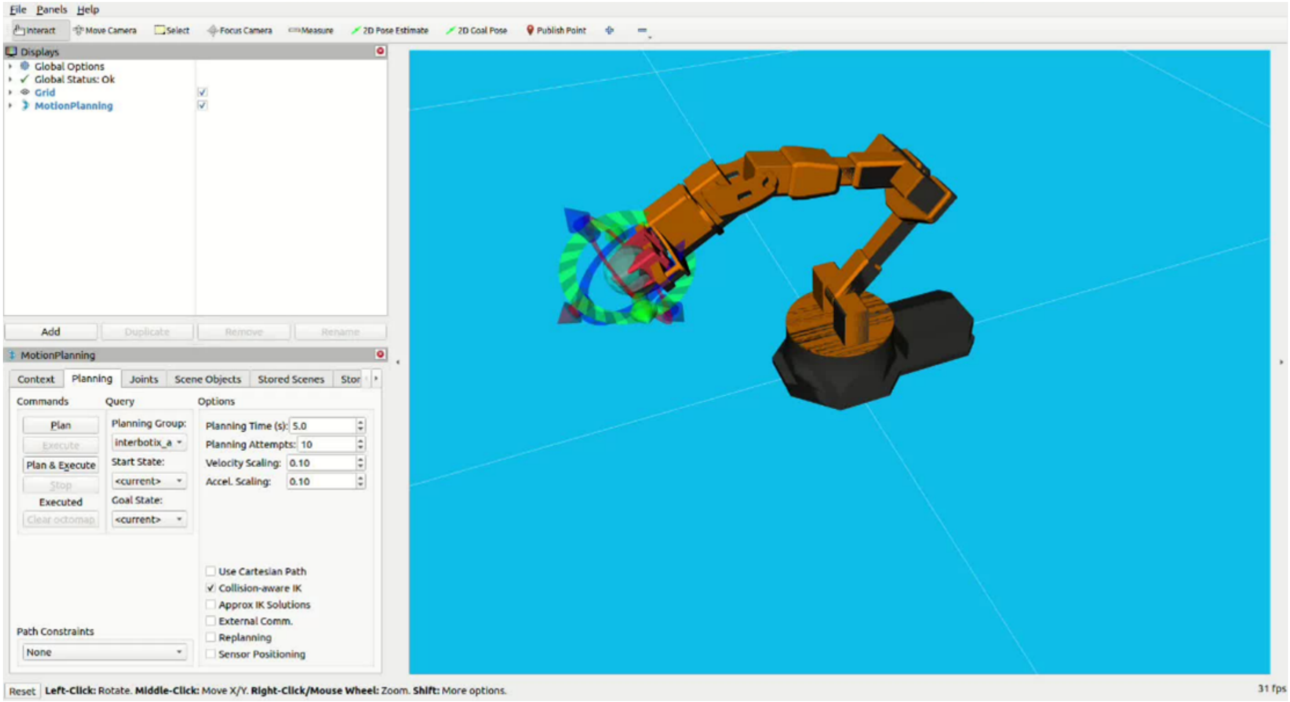


Fig. 27. Simulation in MoveIt

The simulation provided valuable insights into the arm's performance, allowing for the validation of design specifications such as reach, dexterity, and payload capacity. Various tasks, including object manipulation and path following, were successfully executed within the simulation, demonstrating the arm's capabilities and the effectiveness of the motion planning algorithms.

7.2.2 Real-World Implementation of the PincherX-150 Arm

In our real-world experiment, we commenced by setting up the necessary software environment on the Intel NUC host computer. This involved installing ROS 2 and other relevant packages.

Following the software setup, we established a communication link between the host computer and the robotic arm. This connection enabled the transmission of control commands from the computer to the arm, allowing for precise movement and task execution.

To enable the robotic arm to reach specific coordinates determined by the depth sensor, we developed Python scripts utilizing ROS 2's capabilities. The depth sensor played a crucial role in detecting the position of objects, such as blocks, within the environment. It captured real-time spatial data and published the coordinates to a ROS 2 topic. The Python scripts subscribed to this topic, received the coordinates, and commanded the robotic arm to move accordingly.

The sequence of operations began with the depth sensor identifying the location of a block. It transmitted the coordinates to the ROS 2 topic, where the Python script received the data. The script then instructed the robotic arm to navigate to the specified location, grasp the block, transport it to the destination, and release it. Upon completing the task, the arm returned to its default position, ready for the next operation.

Throughout this process, RViz, a 3D visualization tool for ROS as shown in figure below, was employed to monitor the state and movements of the robotic arm in real-time. RViz provided a graphical interface that displayed the arm's position, orientation, and trajectory, enhancing our ability to debug and optimize the system. By visualizing the data published by the depth sensor and the corresponding movements of the arm, we ensured the accuracy and efficiency of the task execution.

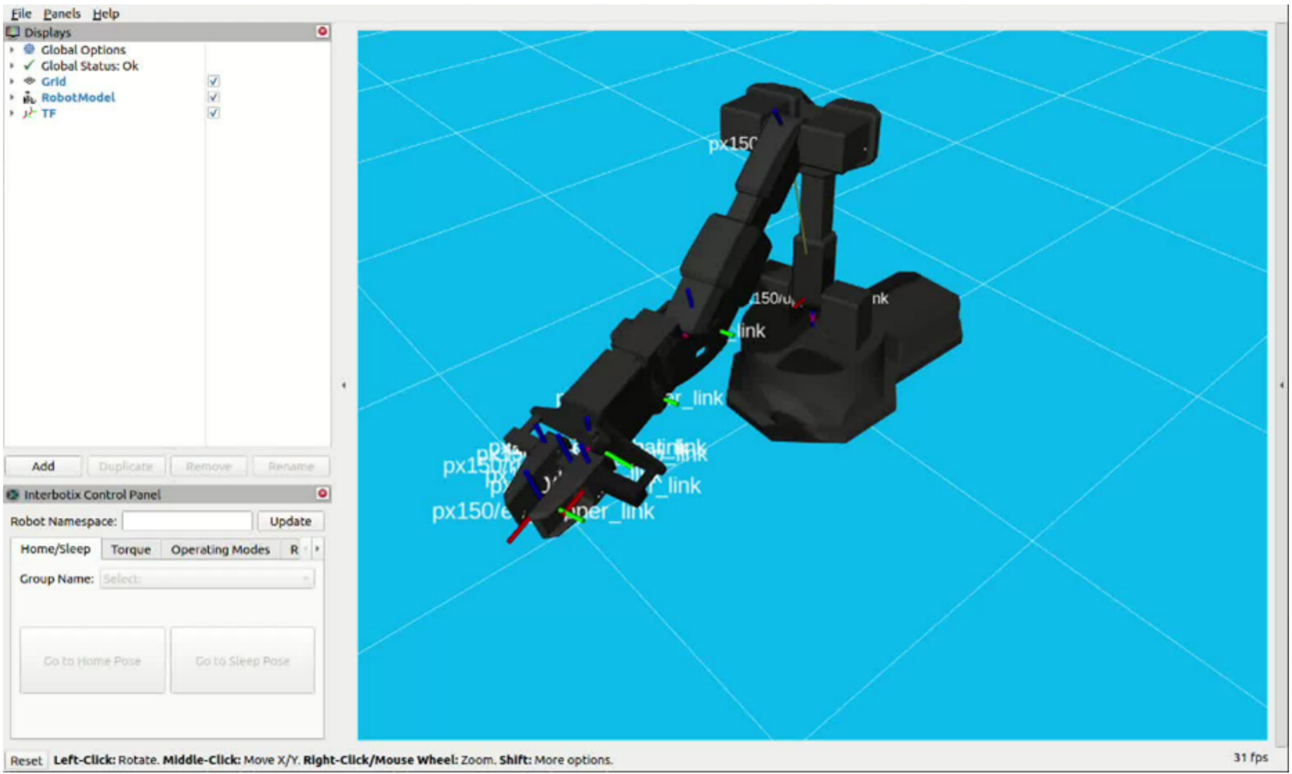


Fig. 28. Visualization in RViz

Fig. 29. Manipulator test in real-world

As a result, the robotic arm successfully demonstrated precise manipulation of objects as shown in figure above. It was able to accurately move to the coordinates provided by the depth sensor, perform the grasping operation, and deliver the object to a designated location. Additionally, the task was completed smoothly without collisions or errors in movement. The arm successfully returned to its default position after completing the operation, indicating the accuracy and effectiveness of the motion planning and execution process.

7.2.3 RQT Graph of the PincherX-150 Arm

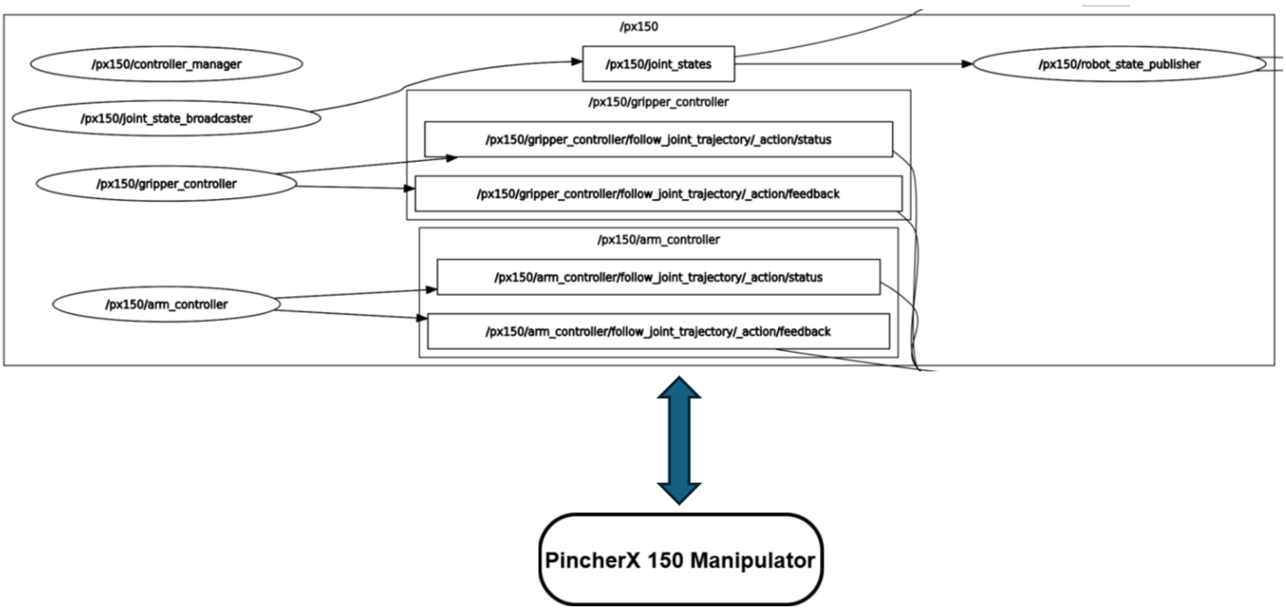


Fig. 30. RQT Graph of the Manipulator

Figure above shows the RQT graph of the robot arm and its connection to the system block diagram. The RQT graph shows the ROS communication structure of the PincherX-150 robot arm. It contains several key nodes that ensure the control and state management of the robot arm. The ”/px150/joint_state_broadcasting” node is responsible for publishing real-time joint position data through the ”/joint_states” topic, which is essential for maintaining the state of the arm and supporting visualization, simulation, and control. The ”/px150/controller_manager” node acts as a central coordinator, managing the control system and receiving joint state information to ensure coordinated control of the movement.

In addition, the ”/px150/gripper_controller” and ”/px150/arm_controller” nodes are action servers that control the gripper and the robot arm, respectively. They receive trajectory commands through their respective ”followJoint_trajectory” topics and provide status and feedback through the ”action/status” and ”action/feedback” subtopics. This allows real-time monitoring and adjustments to ensure precise control of the robot arm and gripper.

7.3 Situational awareness and mobility

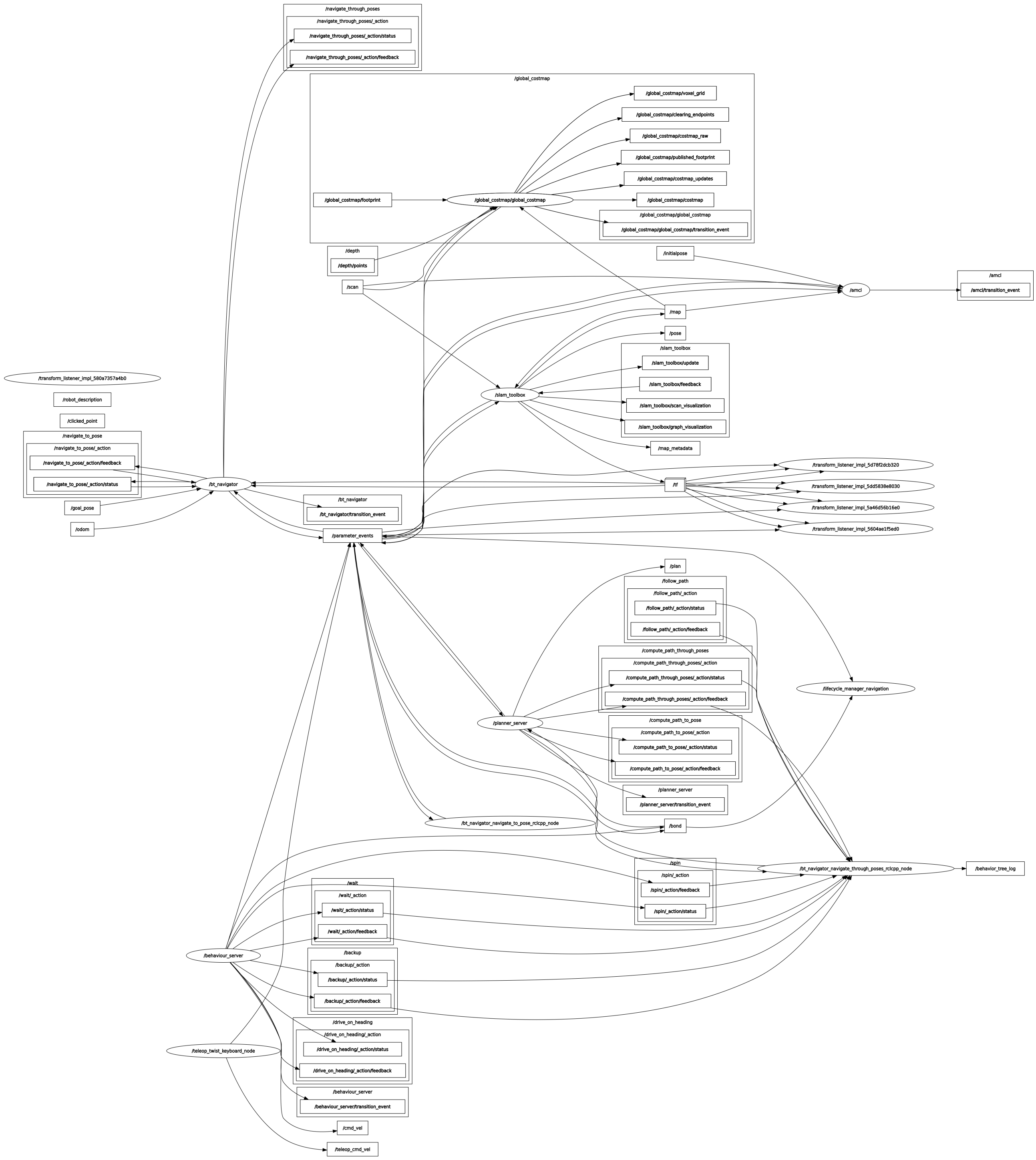


Fig. 31. RQT Graph of SLAM and Navigation

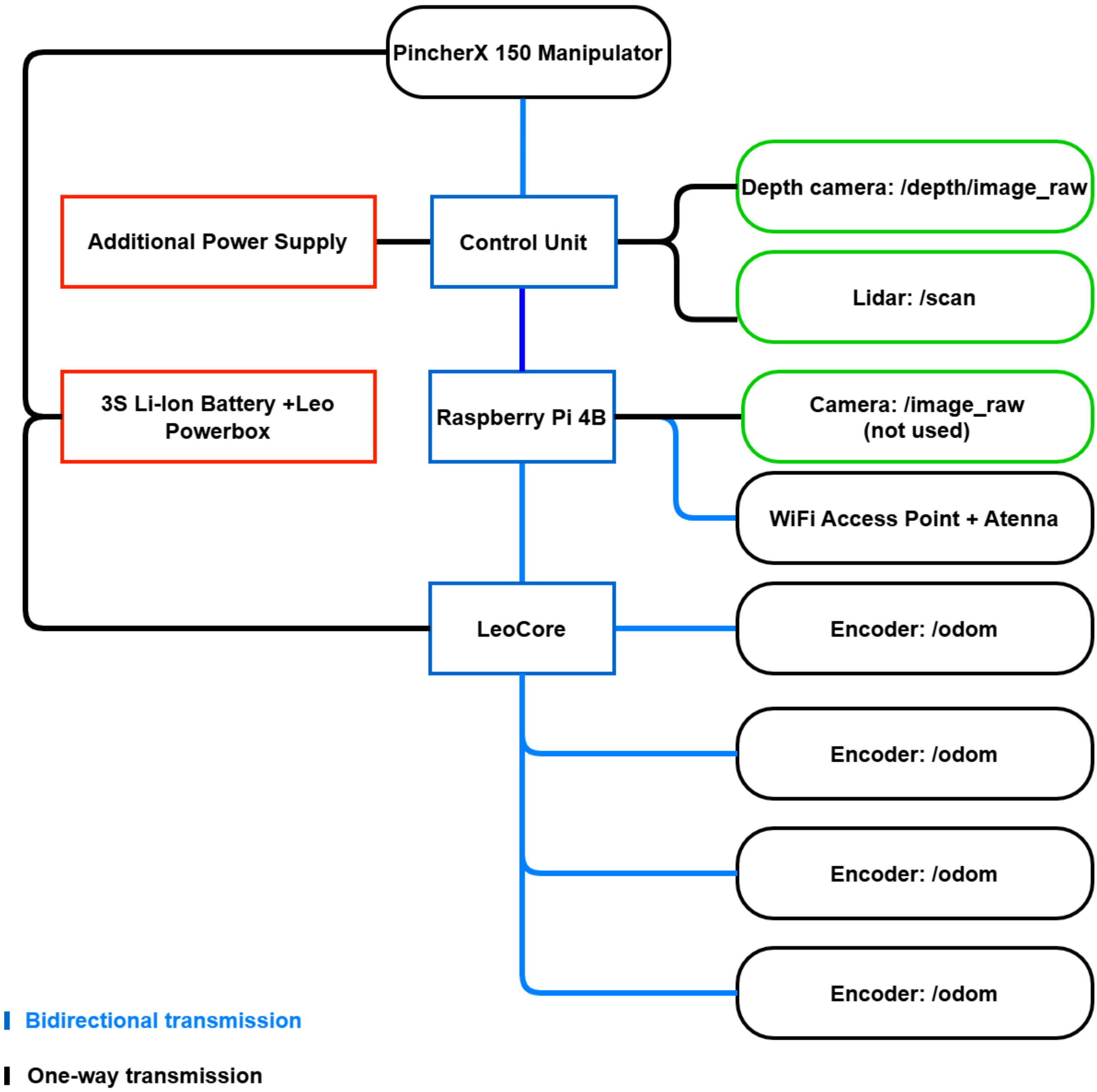


Fig. 32. System Block and Navigation Software Relationship Diagram

This section provides a detailed overview of the software architecture for localisation, mapping, and navigation within the Leo Rover system. It describes how the subsystem is designed to process sensor feedback—from devices such as lidars, cameras, and other sensors—to accurately perceive the environment. The processed data is then used to generate control commands, which are transmitted to the robot's actuators via the Leocore interface. In essence, this design enables the system to translate real-time sensor information into precise navigational actions, thereby supporting autonomous operation.

7.3.1 Localisation and mapping

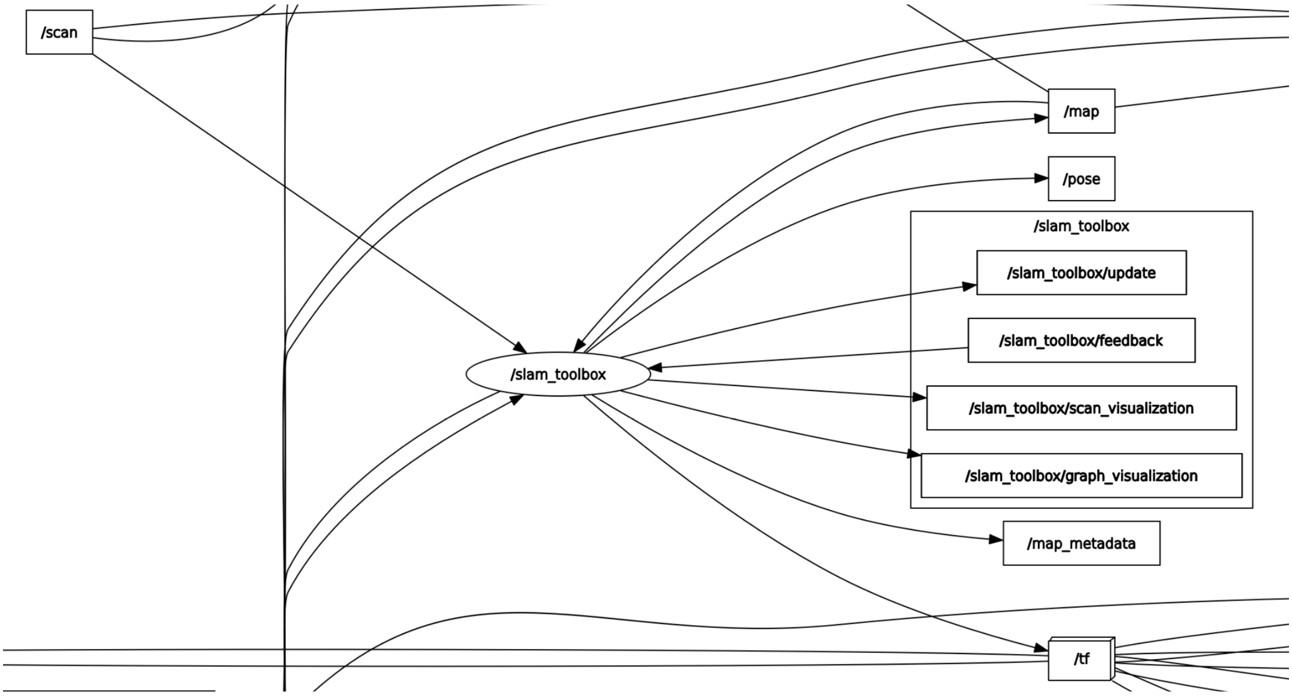


Fig. 33. RQT Graph of SLAM

For the Leo Rover, accurate environmental perception is achieved primarily through mapping and localisation, which serve as the foundation for subsequent navigation decisions. In robotic systems, these functions are typically integrated as Simultaneous localisation and Mapping (SLAM). In our ROS2-based system, the SLAM Toolbox is employed as the core module for sensor fusion, relying on feedback from topics such as /scan and /odom to build a two-dimensional occupancy grid map of the environment and to estimate the robot's pose. The SLAM Toolbox publishes this map on the /map topic, which is then used by other subsystems such as navigation and obstacle avoidance.

Specifically, the SLAM process involves the following key topics and data flows:

- **/map**
  Produced by the SLAM Toolbox, this topic provides the computed occupancy grid map data.
- **/pose**
  Published by the SLAM Toolbox, this topic contains the real-time pose estimation of the Leo Rover.
- **/tf and /joint_state**
  These topics, published by the robot_state_publisher node, describe the relative coordinate transformations among the various components of the Leo Rover.
- **/scan**
  Provided by the lidar sensor, this topic delivers laser scan data that capture environmental features.

- **/odom**
  Originating from the odometry data (typically from motor encoders), this topic conveys information on the robot’s position and velocity.

7.3.2 Navigation

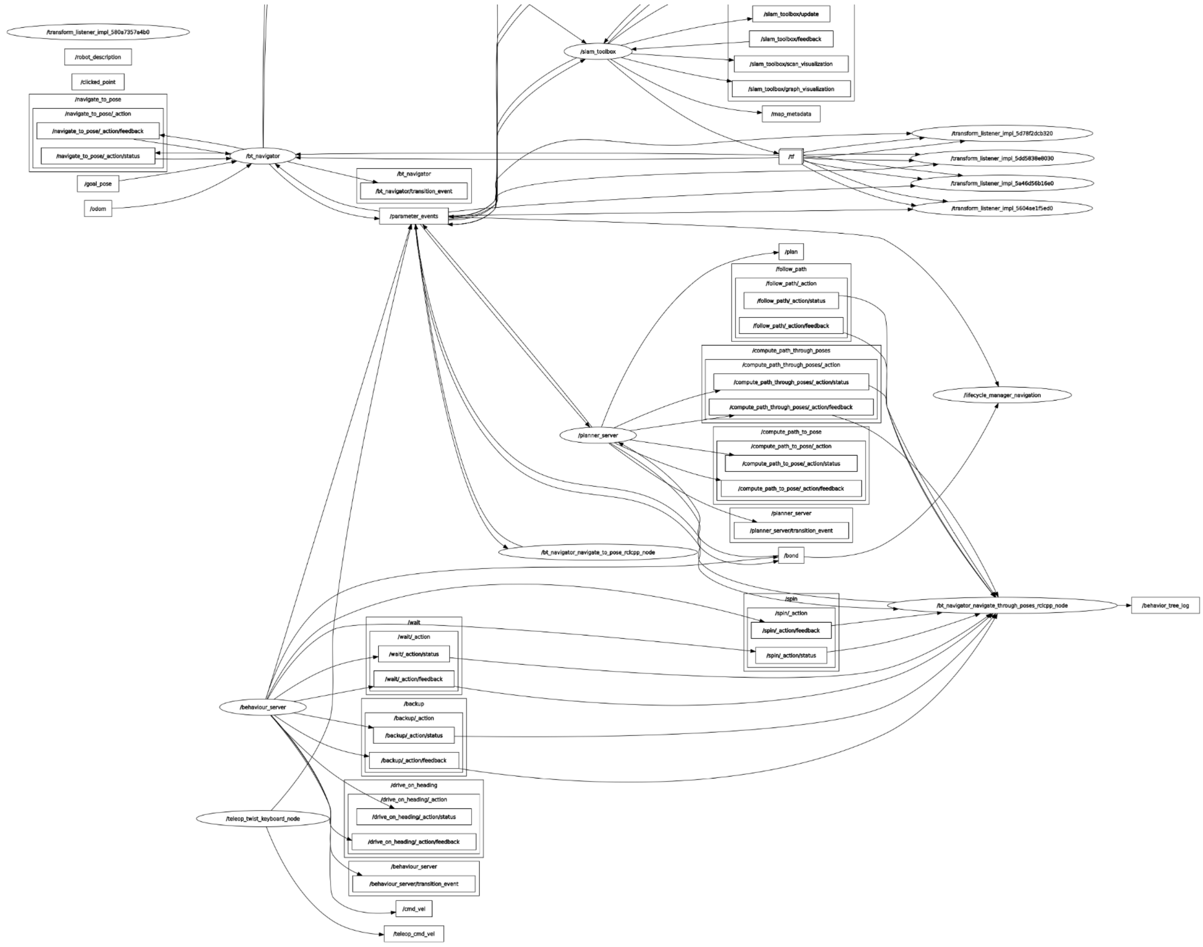


Fig. 34. RQT Graph of Navigation

In parallel with SLAM, the navigation system is responsible for moving the Leo Rover from its current location to a specified target while avoiding obstacles. The navigation system in the ROS2 Navigation (Nav2) stack consists of several core nodes, as is shown in Fig. 34, including the planner_server, controller_server, behavior_server, and bt_navigator. These nodes collaboratively implement path planning, action decision-making, and lifecycle management for the robot. For example:

- **planner_server**
  Receives the current pose and target goal as input, and outputs a collision-free global path using algorithms such as A* or Dijkstra’s algorithm.
- **controller_server**

Processes the global path and computes the appropriate velocity commands based on the current odometry and pose measurements, outputting commands to the /cmd_vel topic.

- **behavior_server**
  Manages high-level behavioral decisions, such as recovery actions when obstacles are encountered.

- **bt_navigator**
  Orchestrates the overall navigation process through a behavior tree, integrating inputs from the planner and controller to execute a coherent navigation strategy.

The Nav2 stack utilizes multiple costmaps to represent the environment. The /global_costmap integrates static environmental features, while the /local_costmap provides a dynamic representation of nearby obstacles. Additional topics such as /path, /plan, and /local_plan facilitate communication among these nodes, ensuring that the robot's path planning and motion control processes are tightly integrated with the mapping and localisation components provided by the SLAM Toolbox.

For instance, the SLAM Toolbox fuses data from lidar scans (/scan), odometry (/odom), and transformations (/tf) to generate a reliable occupancy grid map. This map, along with the robot's real-time pose, forms the basis for the global and local planning operations within the Nav2 stack. The planner generates an optimal path from point A to point B, while the controller computes the necessary motion commands to follow this path. In addition, the behavior tree framework provided by bt_navigator governs complex navigation behaviors such as obstacle avoidance and recovery actions.

In summary, for a static environment where colored bricks are to be detected and retrieved, our system first employs SLAM Toolbox to build an accurate map and localize the robot. Subsequently, the Nav2 stack takes over, utilizing a combination of global and local planning strategies to navigate the environment reliably. The integration of these systems ensures that the Leo Rover can autonomously navigate while compensating for sensor limitations and environmental challenges.

7.4 Github repository

https://github.com/Junzhe-Yan/RobotDesignProject_Team11/tree/main

8 Analysis

The components provided by the university of Manchester, can be utilised to present a unique solution to the problem statement, which team 11 aims to achieve in their robot. To analyse the effectiveness of the team's robotic solution, requires a closer inspection of the team's design decisions. During the planning and fabrication stages of the robots creation, the verification matrix was examined and heavily referenced throughout. By going through and analysing the verification matrix each requirement has been rigorously considered in the design choices and can be clearly explained:

8.1 Functional Environment Requirements

The robot will operate in a smooth, flat concrete surface environment. The requirement for this environment (Req No-1.6) is that ‘the robot shall operate on a smooth, flat surface in the laboratory testing area.’.The Leo Rover is equipped with 130mm diameter rubber wheels, and its overall design is capable of handling rough terrains, far exceeding the requirements of the experimental environment.

8.2 Functional Object Requirements

The functional object requirements refer to the object that the robot needs to retrieve from the environment. The basic requirement is that the robotic system ”shall recognize the blocks and bins through sensors.”(Req No-2.1) and ”shall obtain the position information of the blocks and bins based on equipped sensors.” (Req No-2.2). The object to be retrieved is an opaque wooden cube, small and light enough to be handled by the onboard Trossen PincherX 150 manipulator. The object will be within the reach of the manipulator’s 450 mm arm and will weigh less than 50 grams. The robot will detect the object using the Intel Realsense Depth Camera, which is mounted on the same shelf as the Intel NUC. The Intel Realsense Depth Camera can capture the color of the target object and calculate its depth(Req No-3.1). Through the object detection algorithm, it will classify the recognized objects(Req No-3.2) and return the coordinates of the three blocks (Req No-2.3)in an environment with sufficient lighting conditions(Req No-3.3).

8.3 Functional Basic Mobility

8.3.1 Requirement 1.1 The robot shall be able to perform forward movement, backward movement, and rotation with regard to z-axis.

This requirement was verified through tests using joystick control. To satisfy Req No-1.1, the robot’s mobility system was designed to provide multidirectional movement, including linear translation (forward and backward) and rotational motion about the vertical axis (z-axis). Verification was conducted using a joystick interface that allowed an operator to manually command Leo Rover, thereby directly observing the robot’s ability to execute the required maneuvers. The test results confirmed that the robot demonstrated the necessary responsiveness and precision in movement.

8.3.2 Requirement 1.5 The maximum speed of the robot shall not exceed 0.5 meters per second.

This was achieved by incorporating constraints in the configuration. In order to adhere to the speed limitation imposed by Requirement 1.5, the robot’s control software was modified to include explicit speed constraints. The configuration parameters were adjusted so that the motion commands would not exceed the threshold of 0.5 m/s. Controlled experiments will be conducted wherein the robot's velocity was continuously monitored using both sensor feedback and diagnostic tools. These tests shall confirm that the speed remained within the prescribed limit under a variety of operating conditions.

8.4 Functional Rover Path Planning Requirements

8.4.1 Requirement 1.2 The robot shall self-navigate based on detected target locations.

To achieve Req No-1.2, the robot uses a LiDAR mounted at the bottom and a visual camera at the front to detect the environment and construct a virtual map(Req No-1.2). The system employs SLAM Toolbox to construct an accurate occupancy grid map of the environment. This map is subsequently used by the navigation algorithm to determine the optimal path toward the detected target locations. The global costmap, which aggregates the environmental data acquired via lidar and other sensors, serves as the primary reference for path planning. In testing, the system successfully identified target coordinates and generated a collision-free trajectory, thereby validating the robot's self-navigation capability. This approach ensures that the robot can autonomously navigate to its targets without human intervention, meeting the design specifications.

8.4.2 Requirement 1.3 The robot shall autonomously avoid obstacles and maintain a distance of no less than 5 cm from obstacles.

The system utilizes SLAM to build a map and the local costmap to detect and avoid obstacles through the navigation algorithm. The design integrates the SLAM Toolbox to continuously update the map with real-time sensor data, including that from the lidar, which is essential for identifying obstacles. The local costmap is configured to incorporate dynamic obstacles detected in the immediate vicinity of the robot. The navigation algorithm uses this local costmap to enforce a minimum clearance of 5 cm from any detected obstacles, thereby ensuring safe passage. Rigorous simulation tests confirmed that the robot could successfully detect, plan around, and avoid obstacles while maintaining the specified clearance, thus meeting the requirement for autonomous obstacle avoidance.

8.4.3 Requirement 1.4 The robot shall return to its starting position after placing all blocks in the bins.

The design records the initial starting position and employs a counter, based on the known number of recycled blocks, to trigger the return navigation. To satisfy Req No-1.4, the system records the initial position of the robot at the commencement of its task. As the robot collects and places blocks into the corresponding bins, a counter tracks the number of recycled blocks. Once this count reaches the predetermined total, the control system initiates a return-to-home navigation sequence using the global costmap. The same SLAM and navigation algorithms used for self-navigation are leveraged to guide the robot back to its starting position. Simulated trials have demonstrated that this mechanism reliably returns the robot to its home base, ensuring compliance with the requirement for post-task repositioning.

8.5 Functional Object Manipulation and Retrieval Requirements

In this section, we will analyze how the design of the PincherX-150 manipulator meets the Functional Object Manipulation and Retrieval Requirements outlined in the Requirements Verification Matrix, ensuring the robot's ability to autonomously retrieve and manipulate objects. The requirements focus on three primary tasks: object identification, secure manipulation, and precise placement. Each of these objectives must be thoroughly validated to confirm that the manipulator is both effective and reliable in executing the retrieval process.

8.5.1 Requirement 4.1: The robot shall be equipped with a manipulator.

To meet this requirement, the PincherX-150 manipulator is mounted securely on the robot's platform. The manipulator was selected based on its high torque and precision, necessary for handling the relatively light but potentially fragile objects the robot is designed to manipulate. The manipulator's robust design and high durability allow it to reliably perform the repeated tasks of grasping and moving objects within the robot's operational environment. The manipulator is interfaced with the ROS 2 architecture, allowing it to receive commands from the system and operate seamlessly with other subsystems such as the sensors and the mobility system. This ensures the robot is fully equipped to handle the object manipulation process.

8.5.2 Requirement 4.2: The robot shall grab the blocks securely through the manipulator.

The object manipulation process begins with the secure grasping of the blocks. The design of the PincherX-150 manipulator incorporates a gripper with adjustable force control, which allows the manipulator to securely hold objects with varying textures and shapes. The gripper's force can be calibrated to ensure that it does not apply too much pressure, which might damage the objects, nor too little, which might cause the robot to fail in grasping them. The manipulator is also programmed with a feedback loop that monitors the status of the object during the grasp, verifying that the block is indeed held securely before any motion occurs. The system includes real-time feedback from the manipulator's sensors, ensuring that the robot can detect and rectify any instability in the grasp before proceeding with further tasks. This requirement is verified through testing where the robot picks up various objects, maintaining a secure grasp throughout the movement process.

8.5.3 Requirement 4.3: The robot shall grab the block to ensure that the displacement of the block is less than 1 mm during its movement before placing it.

To ensure precision during manipulation, the robot is designed to minimize any displacement of the block during transport. The manipulator is carefully calibrated to maintain stability during the lifting, carrying, and placing phases. During object retrieval, the robot performs a calibration sequence where it checks the stability of the block after it has been picked up. This is accomplished using the feedback provided by the depth camera and the Raspberry Pi for real-time monitoring. The manipulator uses fine-tuned control algorithms to adjust its grip on the object and ensure that no more than 1 mm of displacement occurs. Additionally, the robot's control system constantly adjusts the arm's motion based on real-time sensor data, accounting for any unexpected movements or shifts in the object's position. This requirement is validated by running tests where the object is moved across various distances, and the maximum displacement is measured to ensure it remains below the 1 mm threshold.

8.5.4 Requirement 5.1: The robot shall autonomously place the retrieved block into the corresponding colored storage bin.

Once the object has been retrieved, the robot's manipulator must place the block into the correct storage bin based on the color detected by the sensors. The robot is equipped with a color recognition algorithm that integrates data from the Intel RealSense D435i depth camera. This allows the system to distinguish the color of the block in real-time and map the correct bin for placement. The manipulator is guided to the appropriate location, ensuring that the block is placed into the correct bin. The process is executed autonomously, with no human intervention required. The accuracy of placement is continually monitored through the system's sensors, ensuring that the block is dropped precisely into the bin without errors. This requirement has been thoroughly tested in both simulated and real-world environments to verify that the robot consistently places blocks into the right bins.

8.5.5 Requirement 5.2: The robot shall accurately position the object to ensure a minimum clearance of 2 cm from the edge of the hole in the bin.

Precision in placement is critical to ensure that the block is properly placed within the designated storage area without obstructing the bin's opening. The manipulator is programmed to move the block with high accuracy, ensuring that there is a 2 cm clearance from the edge of the hole. The robot's arm uses feedback from the depth camera to adjust the final positioning of the block before releasing it. The system checks the position of the block and compares it to the bin's geometry in real-time to ensure that the clearance is maintained. This functionality is validated by conducting placement tests in which the robot places blocks and the distance from

8.5.6 Requirement 5.3: The robot will need to place the blocks in a 15 cm-diameter hole on a 20 cm-side cubic bin.

The robot is designed to work with bins that have a specific size and shape. The manipulator must be capable of accurately positioning the block into a 15 cm-diameter hole on a 20 cm-side cubic bin. The robot's control system integrates geometric calibration of the target placement area, allowing it to calculate the exact location of the center of the hole within the bin. During the placement phase, the robot adjusts the arm's position to ensure that the object is dropped accurately into the hole. This is ensured through continuous verification by the system, which uses the camera data to verify the target area and adjust for any misalignment. The accuracy of this requirement is tested in real-time by measuring the drop position and confirming the object is placed within the hole without exceeding the bin's boundaries.

8.6 Functional Degree of Autonomy Requirements

The robot's degree of freedom ranges from being fully human-controlled to fully autonomous. Team 11's goal is to enable the robot to autonomously complete path planning, obstacle detection and avoidance, and object grasping, without human intervention during the process.

8.7 Product Design Size, Shape, Mass, and Style Requirements

The Leo Rover meets the size, shape, and weight requirements outlined in the problem statement, with dimensions of 447 x 433 x 249 mm and a weight of 6.5 kg. The combined weight of the components on the payload sled, including the Trossen PincherX 150 manipulator, RPLidar A2M12, and Intel Realsense Depth Camera, is 1.81437 kg (4 lbs) , 0.19 kg , and 0.072 kg , respectively. This brings the total weight to 9.25437 kg, which is well below the 15 kg limit. The remaining weight capacity allows for the addition of a larger payload sled capable of accommodating all the listed peripherals.

8.8 Product Design Cost Requirements

In this project, there are no additional expenses apart from creating the payload sled. This aligns with the mission cost requirements.

9 Project Plan

Below is the Gantt chart for the AERO62520 Robotics Systems Design Project, our group co-operated with each other as well as allocated tasks wisely and completed most of the tasks in the first semester and second semester, The project consists of 15 key tasks. The early phase focuses on hardware assembly (Robot assembly) and software development (Basic ROS, Robot software, Steering with Joystick, and Following ARTag), spanning from late September to early November. Following this, tasks such as ROS2 installation, Intel NUC setup, and sensor integration are scheduled, ensuring the fundamental robotic system is operational.

The design phase (System, Mechanical, Electrical, and Software Design) occurs in early December, involving core system architecture refinements. The final phase, spanning February to early

March 2025, includes advanced functionalities like Mapping and Navigation, Object Detection, and Grasping, marking the transition from system setup to functional implementation.

Project title	MSc Robotics AERO62520 Robotic Systems Design Project
Start date	2024/9/27

No.	Project name	Members	Start time	End time	Days of duration
1	Robot assembly	All Members	2024/9/27	2024/10/11	15
2	Robot software	Dongze LI	2024/10/16	2024/10/25	10
3	Basic ROS	Shuyan Zhang	2024/10/16	2024/11/6	22
4	Steering with Joystick	Junzhe Yan	2024/10/25	2024/11/6	13
5	Following ARTag	Jiaqi Yang	2024/10/25	2024/11/6	13
6	Installing ROS2	Shuyan Zhang	2024/11/8	2024/11/22	15
7	Intel NUC Setup	Jiaqi Yang	2024/11/8	2024/11/22	15
8	Using Sensors	Dongze Li	2024/11/8	2024/11/22	15
9	System Design	Jiaqi Yang	2024/12/4	2024/12/13	10
10	Mechanical Design	Dongze Li	2024/12/4	2024/12/13	10
11	Electrical Design	Shuyan Zhang	2024/12/4	2024/12/13	10
12	Software Design	Junzhe Yan	2024/12/4	2024/12/13	10
13	Mapping and Navigation	Shuyan Zhang	2025/2/4	2025/3/6	31
14	Object Detection	Jiaqi Yang & Dongze Li	2025/2/4	2025/3/6	31
15	Grasping	Junzhe Yan	2025/2/4	2025/3/6	31

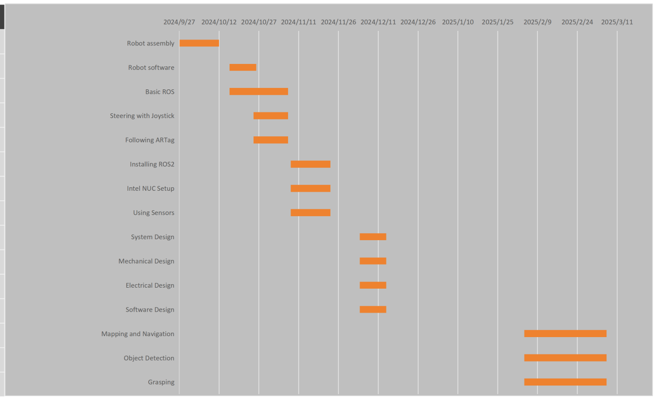


Fig. 35. Project plan Gantt chart

Appendices

A Design Requirement Analysis

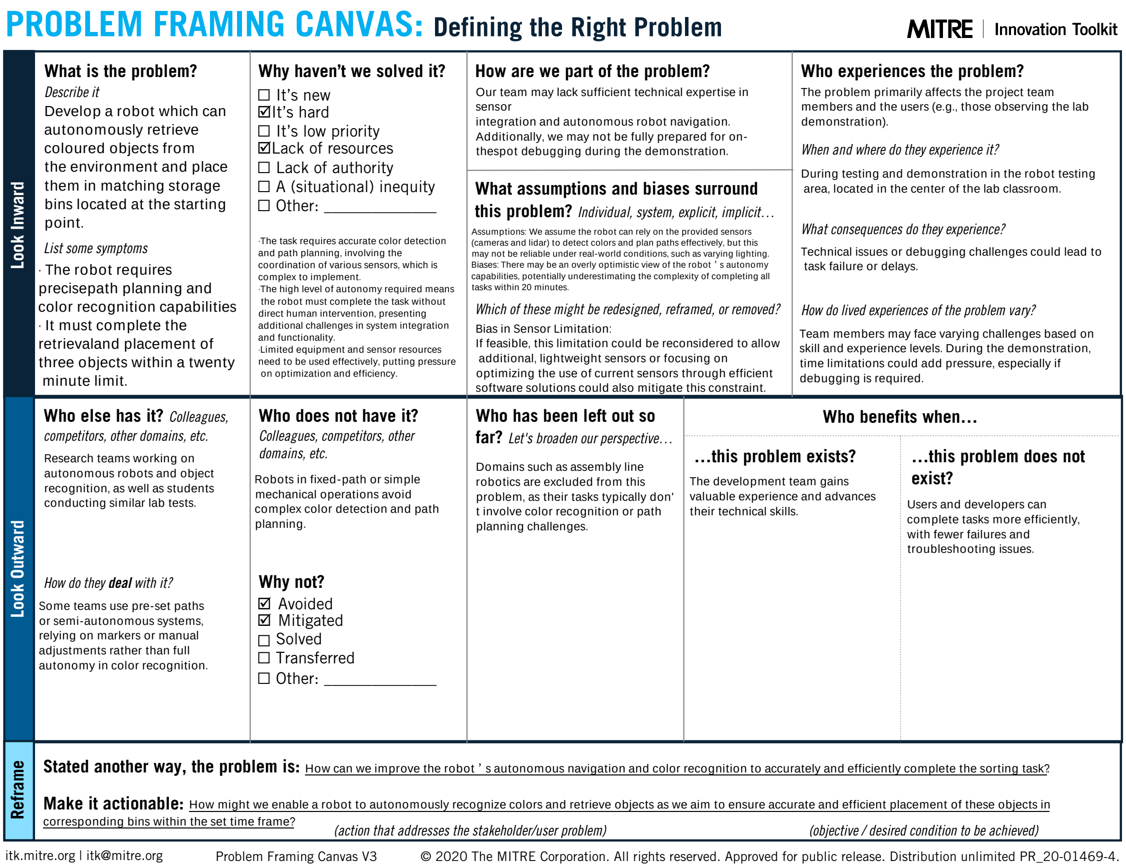
Appendix: Design Requirement Analysis

1 Stake Holder Engagement

Based on the requirements provided by the client, Team 11 members summarized the project goals as follows:

”To develop an autonomous robot capable of identifying and retrieving coloured objects.”

The initial project description lacked sufficient detail to establish a clear and actionable engineering objective. Consequently, Team 11 developed a structured approach to clarify and define the project goals. This process involved two in-person question-and-answer sessions with the client, each lasting approximately 10 minutes, complemented by an online consultation. Through these iterative engagements, the team progressively refined and formalized the specific objectives of the project. The summarised objectives are listed in Fig. 1.


PROBLEM FRAMING CANVAS: Defining the Right Problem

MITRE | Innovation Toolkit

Look Inward

What is the problem?
Describe it
Develop a robot which can autonomously retrieve coloured objects from the environment and place them in matching storage bins located at the starting point.

List some symptoms
· The robot requires precisepath planning and color recognition capabilities
· It must complete the retrievaland placement of three objects within a twenty minute limit.

Why haven’t we solved it?
☐ It’s new
☑ It’s hard
☐ It’s low priority
☑ Lack of resources
☐ Lack of authority
☐ A (situational) inequity
☐ Other: ______________

·The task requires accurate color detection and path planning, involving the coordination of various sensors, which is complex to implement.
·The high level of autonomy required means the robot must complete the task without direct human intervention, presenting additional challenges in system integration and functionality.
·Limited equipment and sensor resources need to be used effectively, putting pressure on optimization and efficiency.

How are we part of the problem?
Our team may lack sufficient technical expertise in sensor
integration and autonomous robot navigation. Additionally, we may not be fully prepared for on-thespot debugging during the demonstration.

What assumptions and biases surround this problem? *Individual, system, explicit, implicit…*
Assumptions: We assume the robot can rely on the provided sensors (cameras and lidar) to detect colors and plan paths effectively, but this may not be reliable under real-world conditions, such as varying lighting.
Biases: There may be an overly optimistic view of the robot ’ s autonomy capabilities, potentially underestimating the complexity of completing all tasks within 20 minutes.

Which of these might be redesigned, reframed, or removed?
Bias in Sensor Limitation:
If feasible, this limitation could be reconsidered to allow additional, lightweight sensors or focusing on optimizing the use of current sensors through efficient software solutions could also mitigate this constraint.

Who experiences the problem?
The problem primarily affects the project team members and the users (e.g., those observing the lab demonstration).

When and where do they experience it?
During testing and demonstration in the robot testing area, located in the center of the lab classroom.

What consequences do they experience?
Technical issues or debugging challenges could lead to task failure or delays.

How do lived experiences of the problem vary?
Team members may face varying challenges based on skill and experience levels. During the demonstration, time limitations could add pressure, especially if debugging is required.

Look Outward

Who else has it? *Colleagues, competitors, other domains, etc.*
Research teams working on autonomous robots and object recognition, as well as students conducting similar lab tests.

*How do they **deal** with it?*
Some teams use pre-set paths or semi-autonomous systems, relying on markers or manual adjustments rather than full autonomy in color recognition.

Who does not have it?
Colleagues, competitors, other domains, etc.
Robots in fixed-path or simple mechanical operations avoid complex color detection and path planning.

Why not?
☑ Avoided
☑ Mitigated
☐ Solved
☐ Transferred
☐ Other: ______________

Who has been left out so far? *Let's broaden our perspective…*
Domains such as assembly line robotics are excluded from this problem, as their tasks typically don' t involve color recognition or path planning challenges.

Who benefits when…

…this problem exists?
The development team gains valuable experience and advances their technical skills.

…this problem does not exist?
Users and developers can complete tasks more efficiently, with fewer failures and troubleshooting issues.

Reframe

Stated another way, the problem is: How can we improve the robot ’ s autonomous navigation and color recognition to accurately and efficiently complete the sorting task?

Make it actionable: How might we enable a robot to autonomously recognize colors and retrieve objects as we aim to ensure accurate and efficient placement of these objects in corresponding bins within the set time frame?
(action that addresses the stakeholder/user problem) *(objective / desired condition to be achieved)*

itk.mitre.org | itk@mitre.org Problem Framing Canvas V3

Fig. 1. Problem Framing Canvas

2 Problem statement

This project aims to develop a Leo-Rover-based robot that can autonomously retrieve coloured blocks and place them into designated storage bins that match the colour of the retrieved block. The robot will operate within the test field of course lab and must navigate, identify, retrieve, and sort bricks by colour without human intervention. Key issues identified include adequate environmental perception, essential mobility, reliable object recognition, and the manipulator's accurate grasping and releasing motions.

Team 11's goals can be broken down into the following objectives:

1. The robot shall have essential mobility and navigation capabilities.
2. The robot shall be able to detect blocks and bins.
3. The robot shall be able to identify the colour of blocks and bins.
4. The robot shall have retrieval capability.
5. The robot shall be able to place the retrieved bricks into bins.

3 Functional requirements and performance requirements

The following functional requirements describe the robotic system's capabilities to achieve the objectives outlined in the Problem Statement. Performance requirements, subsets in functional requirements, define how well the system must perform the specified functionalities.

1. Mobility: The robot shall have essential mobility and navigation capabilities.
    - 1.1 The robot shall be able to perform forward movement, backward movement, and rotation with regard to z-axis.
    - 1.2 The robot shall navigate in an unmapped environment.
    - 1.3 The robot shall autonomously avoid obstacles and keep a distance of no less than 5 cm from obstacles.
    - 1.4 The robot shall return to its starting position after placing all the blocks in the bins.
    - 1.5 The maximum speed of the robot shall not exceed 0.5 meters per second.
    - 1.6 The robot will operate on a smooth, flat surface in the laboratory testing area.
    - 1.7 The robot should complete all actions within the specified 20 minutes.
2. Detection: The robot shall be able to detect blocks and bins.
    - 2.1 The robot shall recognize the blocks and the bins through sensors.
    - 2.2 The robot shall obtain the position information of the blocks and the bins based on equipped sensors.
    - 2.3 The robot will need to retrieve 3 blocks from the field.
3. Identification: The robot shall be able to identify the colour of blocks and bins.
    - 3.1 The robot shall employ sensors to identify the colour of blocks and bins.
    - 3.2 The robot shall associate and classify blocks and bins based on the colour information obtained.

Appendix: Design Requirement Analysis

3.3 The robot will operate in an environment with sufficient lighting conditions over 1,000 lx.

4. Retrieval: The robot shall have retrieval capability.

   4.1 The robot shall be equipped with a manipulator.

   4.2 The robot shall grab the blocks securely through the manipulator.

   4.3 The robot shall grab the block to ensure that the displacement of the block is less than 1 mm during its movement before placing it.

5. Placement: The robot shall be able to place the retrieved bricks into bins.

   5.1 The robot shall autonomously place the retrieved block into the corresponding coloured storage bin.

   5.2 The robot shall accurately position the object to ensure a minimum clearance of 2 cm from the edge of the hole in the bin.

   5.3 The robot will need to place the blocks in a 15cm-diameter hole on a 20cm-side-cubic bin.

4 Requirements Verification Matrix

The following table outlines the verification criteria and verification methods associated with each requirement to ensure compliance with the design specifications.

Table 1. Requirements Verification Matrix

Req No	Reference Para	Shall Statement	Verification Criteria	Verification Method	Responsible Party	Results
1.1	Mobility	The robot shall be able to perform forward movement, backward movement, and rotation with regard to z-axis.	Test the robot's movement in all specified directions in the lab environment. The test is deemed successful if the robot can complete the corresponding movement according to the input instructions.	Test	Group 11	TBD

Continued on next page

Appendix: Design Requirement Analysis

Req No	Reference Para	Shall Statement	Verification Criteria	Verification Method	Responsible Party	Results
1.2	Mobility	The robot shall self-navigate based on detected target locations.	Verify autonomous navigation by placing target locations in the testing area and observing self-navigation. The test is deemed successful if the robot can autonomously map and plan its route in the environment using SLAM.	Demonstration	Group 11	TBD
1.3	Mobility	The robot shall autonomously avoid obstacles and keep a distance of no less than 5 cm from obstacles.	Place obstacles in the robot's path and verify whether it can avoid them while maintaining at least 5 cm distance.	Test	Group 11	TBD
1.4	Mobility	The robot shall return to its starting position after placing all blocks in the bins.	Test whether the robot returns to the starting point after block placement.	Demonstration	Group 11	TBD
1.5	Mobility	The maximum speed of the robot shall not exceed 0.5 meters per second.	A command exceeding the predefined maximum speed threshold is issued. The test is deemed successful if the robot's actual speed remains at or below the specified upper limit	Inspection	Group 11	TBD
1.6	Mobility	The robot will operate on a smooth, flat surface in the laboratory testing area.	Confirm setup and operation on the specified surface.	Inspection	Group 11	TBD

Continued on next page

Appendix: Design Requirement Analysis

Req No	Reference Para	Shall Statement	Verification Criteria	Verification Method	Responsible Party	Results
1.7	Mobility	The robot should complete all actions within the specified 20 minutes.	Measure total task completion time during testing. The test is deemed successful if the robot can complete all tasks within the specified time.	Test	Group 11	TBD
2.1	Detection	The robot shall recognize the blocks and bins through sensors.	Test recognition capability by using sensors to identify blocks and bins.	Test	Group 11	TBD
2.2	Detection	The robot shall obtain the position information of the blocks and bins based on equipped sensors.	Validate position data acquisition for blocks and bins through sensor testing. The test is deemed successful if the robot's onboard sensors can successfully transmit back the target's position information.	Analysis	Group 11	TBD
2.3	Detection	The robot will need to retrieve 3 blocks from the field.	Confirm that there are indeed three coloured blocks.	Inspection	Group 11	TBD
3.1	Identification	The robot shall employ sensors to identify the colour of the blocks and bins.	Test colour detection based on the returned value from detecting blocks and bins. The test is deemed successful if the robot's onboard sensor can successfully transmit back the target's color information.	Test	Group 11	TBD

Continued on next page

Appendix: Design Requirement Analysis

Req No	Reference Para	Shall Statement	Verification Criteria	Verification Method	Responsible Party	Results
3.2	Identification	The robot shall associate and classify the blocks and bins based on the colour information obtained.	Verify classification accuracy based on whether the robot can match detected colours of a certain range. The test is deemed successful if the robot can successfully classify objects of similar colors based on the sensor readings.	Test	Group 11	TBD
3.3	Identification	The robot will operate in an environment with sufficient lighting conditions over 1,000 lx.	Confirm test field under adequate lighting.	Inspection	Group 11	TBD
4.1	Retrieval	The robot shall be equipped with a manipulator.	Check manipulator installation on the robot.	Inspection	Group 11	TBD
4.2	Retrieval	The robot shall grab the blocks securely through the manipulator.	Test grasp security by having the manipulator hold a block.	Test	Group 11	TBD
4.3	Retrieval	The robot shall grab the block to ensure that the displacement of the block is less than 1 mm during its movement before placing it.	Verify grip stability by having the manipulator hold a block during robot movement. The test is deemed successful if the robot can maintain the grasp of the block for more than 10 minutes without interference.	Demonstration	Group 11	TBD

Continued on next page

Appendix: Design Requirement Analysis

Req No	Reference Para	Shall Statement	Verification Criteria	Verification Method	Responsible Party	Results
5.1	Placement	The robot shall autonomously place the retrieved block into the corresponding coloured storage bin.	Observe autonomous placement accuracy during testing. The test is deemed successful if the robot is able to drop the block into the bin.	Demonstration	Group 11	TBD
5.2	Placement	The robot shall accurately position the object to ensure a minimum clearance of 2 cm from the edge of the hole in the bin.	Measure clearance from the hole's edge after placement. The test is deemed successful if the robot places the block into the bin without the block or manipulator coming into contact with the bin.	Test	Group 11	TBD
5.3	Placement	The robot will need to place the blocks in a 15 cm-diameter hole on a 20 cm-side cubic bin.	Observe the specified bin dimensions.	Inspection	Group 11	TBD

B Team Workplace Charter

Appendix: Team Workplace Charter

Team11 Workplace Charter

1. **Purpose of the Document**

This Workplace Charter outlines the guidelines and expectations for how the team in *AERO62520 Robotic System Design Project* will function. It serves as a reference for team members on work protocols, communication standards, and conflict resolution methods. The document also acts as a preventive measure against team conflicts by establishing clear principles and guidelines to promote harmonious and productive collaboration. It will be used to ensure that all members operate under agreed-upon standards and that any disputes or issues can be addressed efficiently.

2. **Statement of Principles and Commitments**

Our team is committed to:

- Ensuring a professional, respectful, and supportive working environment.
- Encouraging open, honest, and timely communication to resolve issues and prevent misunderstandings.
- Creating an inclusive and equitable space where contributions of each team member are valued.
- Achieving high standards in all assigned tasks while promoting collaboration and teamwork.

3. **Team Rules**

- **Working Hours**: 10 a.m. to 6 p.m. & Monday to Friday
- **Language**: English should be the primary language for all members.
- **Communication Channels**: All work-related communication and progress synchronization will be conducted on private channel via Microsoft Teams. WeChat will be used for informal discussions or to clarify details quickly.
- **Subtask Assignment**:
    - Subtasks will be divided according to the project's needs and assigned to team members. Members will take full responsibility for their respective tasks.
    - Subtasks will be assigned based on team members' expertise. If there is no major match or the task volume is large, it will be assigned randomly. If the number of team members required for all current subtasks is less than the number of team members, the unassigned team members should be ready to assist other team members at any time.
- **Communication Responsiveness**: During working hours, members must reply to team communications within a day. In cases of emergencies or force majeure, members should notify the team and the instructor in advance.

The content of this document was drafted by team members, the text was polished and formatted with the help of AI, and was reviewed and approved by all team members.

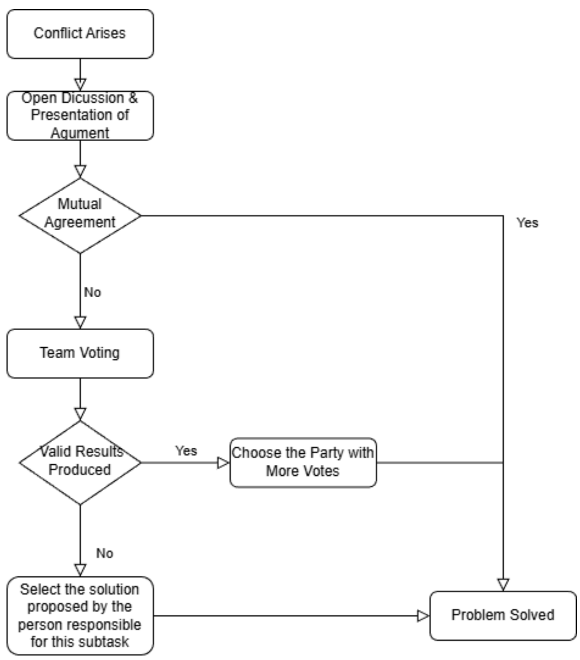
4. **Daily Preventive Measures**

To avoid conflicts and maintain a productive environment, our team will:

- **Active collaboration**: All team members should maintain an open communication attitude to ensure that everyone is clear about the progress of the task and the current challenges.
- **Clear Subtask Allocation**: Before each task, the content will be clearly assigned to team members in the form of sub-tasks to avoid confusion caused by overlapping responsibilities.
- **Avoid Excessive Interference**: Each team member is encouraged to help each other, but should avoid interfering with others' work. This includes frequent unconstructive comments, unnecessary physical contact, and using lab session time to handle personal matters.
- **Pre-class Meeting**: Before each Lab session, progress synchronization, member status assessment and opinion exchange will be conducted face-to-face to ensure that all team members synchronize work information and inform other team members of their ideas.
- **Immediate Resolution of Minor Issues**: Any minor issues or misunderstandings should be addressed promptly, ideally on the same day, to prevent escalation.

5. **Conflict Resolution Process**

- Any form of verbal or physical aggression is strictly prohibited.
- The conflict resolution process should refer to the following flowchart.



The content of this document was drafted by team members, the text was polished and formatted with the help of AI, and was reviewed and approved by all team members.