The University of Manchester

Preliminary Design Review

This is a group assignment for
Robotic Systems Design Project
in the School of Engineering

Group Number
3

Student ID
11518806
11532270
11464161
14123340

School of Engineering

Contents

1 Introduction

The purpose of this document is to outline the preliminary design review of a project to ensure that the design program meets the required standards. A thorough analysis and review of the design allows potential issues to be identified and optimized in the early stages of the project to ensure that development progresses smoothly.

The problem statement for this project is 'Develop a robotic system capable of recognizing and grasping objects while safely avoiding obstacles'. The robot will be able to efficiently perform object recognition and classification tasks while ensuring safe operation over long periods of time. Through a high degree of collaboration between hardware and software, as well as innovative design strategies, the team will endeavor to meet all specifications and create a reliable and efficient robotics solution.

To achieve this goal, the project focuses on the construction of a complex mobile robotic system. Development of a four-wheeled robot equipped with the necessary peripherals and sensors. The robot autonomously recognizes and picks up color cubes and sorts them into appropriate storage bins. Implement autonomous path planning and environment mapping to ensure effective obstacle avoidance and optimize operational efficiency. The robot is able to transmit real-time task progress data to ensure transparent and monitorable operation. Ensure that the robot has the ability to operate autonomously for long periods of time while ensuring the safety and reliability of the operation.

1.1 Summary of Feedback Addressed from the Preliminary Design Review (PDR)

1.1.1 Mechanical Design

Missing annotations to identify components

- **Improvement Measures:**
  - Each of the components in the CAD drawing has been mentioned with arrows and text.

Blind spots caused by LiDAR placement

- **Improvement Measures:**
  - Changed the placement of LiDAR from under the chassis to the top of chassis avoiding blind spots.

1.1.2 System

Improving the Readability and Simplicity of System Block Diagrams

- **Improvement Measures:**
  - The general layout of the system block diagram was adjusted. Adjusted the size of the charts to make them easier to read and replaced the original colored backgrounds with more readable black text on a white background.
  - Added different colored legends to accurately categorize and label the robot components in the diagrams.

Optimization of Connections between Robot Components

- **Improvement Measures:**
  - The connections between the various components of the robot are categorized more clearly. There are three main components: power supply, signal control and data transmission which is marked on the diagram with different colored arrows.

Clarifying the Description and Illustration of Each Component of the Robot

- **Improvement Measures:**
  - A new component, called wired external power source, was added to the original to power the NUC and the robotic arm.
  - Clarified the functions of the camera and depth camera and how to interact with them.

Conclusion

Adjusted the overall layout of the system block diagram to make the connections between the robot components clearer and to clarify the camera function. Effectively improved the clarity of the diagram and made the relationship between the system components more intuitive.

1.1.3 Electrical System

Enhancing the Readability of the Electrical System Diagram (Formatting and Image Optimization)

- **Improvement Measures:**
  - Introduce a brand-new Electrical Device Table, consolidating previously scattered device information into a unified format.
  - This enhancement improves overall readability and standardization, making the information clearer and more intuitive.

Addressing Confusion Caused by Multiple Diagrams

- **Improvement Measures:**

- Redesigned the electrical connection diagram, adjusting the layout, and enlarging the size to ensure clarity for all components.
- Simplify content by removing unnecessary details, making core information more prominent.
- Consolidate device information into the Electrical Device Table, eliminating redundancy and improving overall readability and logical structure.

Providing an External Power Supply Solution for the NUC

- **Improvement Measures:**
  - Propose a specific external power supply solution, ensuring that the NUC and robotic arm have independent power sources, thereby enhancing system stability.
  - Purchase power extension cables to prevent wires from obstructing the robot's movement, optimizing the wiring arrangement.

Optimizing Power Supply for the Robotic Arm and Sensors

- **Improvement Measures:**
  - The robotic arm is now powered by an independent external power supply, improving safety and avoiding potential overload issues from sharing the main power source.
  - LiDAR and depth cameras remain connected to the NUC due to considerations regarding data transmission speed and their USB power supply method , with the external power supply providing unified power to the NUC to ensure stable data transmission.

Conclusion

These improvements effectively enhance the readability, logical structure, and safety of the electrical system while ensuring the reliability of the robot's operation.

1.1.4 Software Design

The RQT Diagrams are unreadable at 100% zoom

- **Improvement Measures:**
  - Separated the RQT graph from the system block diagram to ensure clarity at 100% zoom. Whenever possible, use the SVG format to maintain detail visibility when zoomed in.

For the system block diagram how decisions are made was not discussed

- **Improvement Measures:**

- Add a flowchart illustrating the robot's main actions, including the decision-making process for object recognition and grasping, clearly presenting each step and its corresponding execution flow.

The GIT repository appeared to have all the files, but there was no documentation explaining how to use/install the software.

- **Improvement Measures:**
  - Add a README.md file that provides a detailed step-by-step guide, from cloning the project to setting up the environment and installing dependencies. Additionally, it includes solutions for potential errors and a comprehensive guide on using the Git repository, with detailed code examples to help team members quickly understand and collaborate efficiently.

1.1.5 Analysis

Clarifying the relationship between each test and each requirement.

- **Improvement Measures:**
  - Match the six tests to the 13 requirements in the Requirements Matrix to ensure that each requirement is validated.

1.1.6 Project Plan

The Gantt chart was hard to read at 300% zoom. The project plan is sequential rather than parallel which is likley to cause issues.

- **Improvement Measures:**
  - The project schedule has been redesigned, with adjustments to the planning of Object Detection, Grasping, Mapping and Navigation tasks, adopting a parallel development approach where each team member is responsible for their own part. The first five weeks focus on coding and debugging, while the Gantt chart layout has been further optimized to clearly present the progress of each work package and its assigned team member. Compared to the previous version, the updated Gantt chart highlights milestones more effectively, improving readability. Additionally, the chart is now in SVG format to ensure clarity at different zoom levels.

1.2 Summary of modifications made since the Preliminary Design Review (PDR)

1.2.1 Mechanical design

- **Change in the placement of LiDAR**: It was placed under the chassis which caused blind spots that made it complicated for laser scanning. Now, it is placed on the top of the chassis below the manipulator platform. Blind spots are largely eliminated and the ov
- **Change in the placement of camera**: Initially, the camera was placed on the wrist of manipulator. It caused some inaccuracy in object detection and it was difficult for the camera to detect objects when the manipulator was at home position. It is now placed in the front of the robot on a ball-head mount so the angle of view can be adjusted manually.
- **Structural changes and addition of new parts**: Some parts were redesigned for better stability and due to the changes in placement of LiDAR and camera, a few rearrangements for other components had to be done.

1.2.2 Mapping and Navigation

The robot adopts `slam_toolbox` for mapping and navigation. Compared to the previous approach, this modification provides enhanced environmental perception and more stable localization performance.

`slam_toolbox` enables the robot to create maps in real-time in an unknown environment while supporting incremental mapping, reducing computational overhead during map updates.

New Features:

- Loop closure detection: When the robot recognizes a previously visited area, the system automatically adjusts the map, reducing accumulated errors and improving mapping accuracy.
- Pose graph optimization: Utilizes global optimization techniques to correct errors, making the navigation path smoother and more precise.
- Global/local costmap: Integrates sensor data (such as LiDAR and depth cameras) to enhance path planning and obstacle avoidance capabilities.

Improvements:

- Improved accuracy: Compared to the original navigation approach, `slam_toolbox` provides higher map construction accuracy, reducing pose drift issues.
- Enhanced navigation stability: By improving the path planning algorithm, the robot achieves more stable navigation in complex environments, avoiding excessive oscillations and path deviations.
- Increased adaptability: The system can better handle dynamic environments—even if surrounding objects change, it can quickly update the map and adjust the path accordingly.

1.2.3 Object detection

Initially, we considered using color recognition to distinguish blocks of different colors, but ultimately, we adopted deep learning to enhance the robot's object recognition capabilities. To achieve this, we developed a Python script to capture images of colored wooden blocks using a camera while ensuring that resolution and other factors did not interfere with object recognition. We built a high-quality training dataset by capturing over 1,200 images and manually annotating each target using labeling software. To improve the model's robustness, images were taken from various angles to simulate different perspectives the robot might encounter. The model was trained using YOLOv8n-seg, and after an initial 50 epochs, we observed that the recognition accuracy occasionally dropped below 90%. To address this, we expanded the training dataset and increased the number of training epochs. The final model performed well in object recognition tasks, with detailed analysis and evaluation provided in the object recognition module of the software design section.

1.2.4 Software Design

In terms of software design, we restructured the previous architecture by removing unnecessary control modules, as the robot no longer requires manual control via a joystick and instead operates autonomously. Additionally, we introduced an object recognition module and merged the mapping and navigation modules to reduce redundant data transmission. The restructured system enhances modular integration: the sensor module is solely responsible for external data collection, while the task control module handles data processing and directly manages both navigation and execution modules. This centralization of data processing allows other modules to focus solely on executing received commands, such as target positions or movement goals, thereby streamlining the overall system architecture and improving efficiency.

Furthermore, we optimized the Git repository by refining the project structure, adding a README.md file, and removing unnecessary files such as .idea, install, and build. These files were also added to .gitignore to prevent unnecessary clutter in the repository.

2 Sustainability Checklist

This checklist is based on the BS8622 Guide to Robot Sustainability (Draft) and covers 10 key dimensions to assess and improve the environmental impact of robots.

2.1 Materials

Identify the types of materials currently used in the design and possible ways to incorporate more environmentally friendly materials into the robot.

Current Material Types:

Our robot is constructed using a variety of materials, including metals(for the robotic arm and LeoRover), plastics (for the outer casing, including PLA 3D-printed components) and electronic components (for sensors and cameras).

More Eco-Friendly Material Options:

To align with the growing trend of technological sustainability, the team is actively exploring the use of recycled plastics (rPLA) and sustainably sourced metals to minimize the project's environmental impact. Additionally, we are committed to optimizing 3D printing designs to reduce printing waste and improve material efficiency, fostering a more eco-friendly manufacturing approach.

2.2 Software

How can software overhead (including power consumption, communication, data storage, and training) be minimized or made more sustainable?

Reducing Power Consumption:

- Optimize code to reduce unnecessary loop computations and improve computational efficiency.
- Optimize algorithms to lower the computational complexity of SLAM and path planning, reducing CPU/GPU workload.

Reducing Communication Energy Consumption:

- Implement edge computing to reduce cloud dependency by processing data locally and minimizing data transmission.
- Apply data compression to store and transmit only essential data, improving storage and communication efficiency.

Enhancing Storage Efficiency:

- Adopt intelligent data management to reduce redundancy and avoid storing unnecessary sensor data.

2.3 Energy

Has energy efficiency been considered within the design? What could be done to reduce energy consumption? What is the source of energy for the robot and are there more environmentally friendly options?

Energy Efficiency:

- The robot operates on an 11.1V Li-Ion 3S battery (8A) as its primary power source and utilizes the LEO Power Box for voltage distribution to optimize power management and reduce unnecessary energy consumption.

Measures to Reduce Energy Consumption:

- Smart power management: Optimize power supply strategies to reduce standby power consumption and prioritize critical components.
- Algorithm optimization: Improve SLAM and path planning algorithms to minimize unnecessary computations and enhance energy efficiency.
- Low-power components: Select more efficient sensors, motors, and processors to reduce overall power consumption.

Energy Source:

- The robot primarily relies on lithium-ion batteries for power, while some high-power components (such as the Intel NUC and robotic arm) still require an external power source.

More Environmentally Friendly Alternatives:

- Solar charging: Integrate solar panels to reduce dependence on the traditional power grid and improve energy independence.
- Battery recycling: Establish a battery recycling program to minimize the environmental impact of discarded batteries.

2.4 Waste

Identify if any waste products are generated by the robot during its operation and consider measures to reduce and handle them to minimize the environmental impact.

Sources of Waste:

- Battery disposal: Lithium-ion batteries have a limited lifespan and may cause environmental pollution when discarded.
- Electronic component replacement: Motors, sensors, and other components may require replacement after long-term use.

Measures to Reduce and Manage Waste:

- Battery recycling program: Collaborate with suppliers to establish a recycling system to minimize battery waste and pollution.

- Modular component replacement: Implement a detachable design to allow easy replacement of individual parts instead of discarding the entire system.
- Reducing 3D printing waste: Use recyclable PLA filament, improve printing efficiency, and minimize material waste.

2.5 Emissions

Identify possible sources of pollution generated by the robot and consider measures to reduce and handle them to minimize the environmental impact.

Sources of Pollution:

- The robot itself does not generate direct emissions during operation, but its charging power source may rely on fossil fuels.
- The manufacturing process for electronic components may involve pollutant emissions.

Measures to Reduce Environmental Impact:

- Use green electricity: Adopt solar or wind energy for charging to reduce the carbon footprint.
- Select environmentally friendly manufacturers: Source products from low-carbon supply chains, such as electronic components produced with sustainable manufacturing processes.
- Optimize charging strategy: Avoid overcharging, extend battery lifespan, and reduce energy waste.

2.6 Communication

Consider how much data is being transmitted to and from the robot. Is it all necessary? What are the data storage implications of data transmitted from the robot?

Data Transmission:

- The robot uses 2.4GHz Wi-Fi and wired connections for communication, transmitting only essential task data and remote monitoring information to reduce unnecessary bandwidth usage and power consumption. Thus, the data volume is reasonable.
- It is not necessary to transmit all data, as some robot modules (e.g., built-in cameras) are not actively used.

Local Data Processing:

- The robot processes data using edge computing, reducing communication energy consumption and storage costs, which helps conserve resources and protect the environment.
- Only necessary sensor data is stored to minimize data redundancy and improve storage efficiency.
- In the future, the robot can adopt low-power communication protocols, exploring alternatives such as LoRa and BLE (Bluetooth Low Energy) to further reduce communication energy consumption.

2.7 Modularity

How modular is the robot design? Can items be easily replaced or repaired? Consider both the hardware and software.

Modularity Level:

- The robot adopts a highly modular design, allowing individual components to be replaced or upgraded without the need for complete system disposal.
- Standardized interfaces ensure compatibility with components from different manufacturers, enhancing scalability and future adaptability.

Maintainability:

- Provide repair guides and maintenance documentation to encourage repair instead of full system replacement.
- Implement a parts recycling program to improve component reuse and reduce electronic waste.

2.8 Location/Placement

Are there any sustainability considerations with the deployment of the robot? Does the robot need to be transported using vehicles? If so, what is the most sustainable method of transportation?

Deployment Environment:

- The robot is primarily designed for indoor environments and is currently transported manually, making it an efficient and low-energy solution with minimal environmental impact.
- Since no long-distance vehicle transportation is required, the deployment process remains sustainable.

Sustainable Transportation Methods:

- When deploying in remote areas, electric vehicles or drones can be used to reduce carbon emissions.

2.9 Maintenance

How is the robot maintained? What checks are taken to ensure it doesn't break midmission? Does the robot have diagnostic software which monitors its performance, which can be used to identify when maintenance needs to be conducted?

Maintenance Methods:

- The robot is equipped with self-diagnostic software that monitors motor, sensor, and power system status, recording data for predictive maintenance.
- A modular design ensures that critical components can be quickly replaced, improving repairability.

Remote Maintenance:

- OTA (Over-the-Air) firmware updates minimize the need for manual maintenance intervention.
- Recyclable spare parts ensure that replaced components can be reused, reducing electronic waste.

2.10 Repurposing

Consider how the robot may be re-used at the end of the project. Can it be repurposed for other activities?

Multi-Purpose Applications:

- The robot can be modified for different applications, such as warehouse automation, educational robotics, and inspection tasks, preventing equipment from being left idle or scrapped.
- A modular structure allows the replacement of different sensors and actuators, enabling functionality upgrades without requiring full system disposal.

Recycling and Upgrading:

- The robot's hardware is designed for disassembly and future technology compatibility, allowing for component replacements to extend its lifespan.
- Standardized interfaces facilitate the integration of new technologies, such as advanced AI computing modules and low-power sensors.

Electronic Waste Management:

- Recover electronic components to prevent environmental pollution and establish a second-hand parts reuse program to promote component recycling.

- Refurbished robots can be used for research, education, and training, reducing resource waste.

Long-Term Sustainability:

- Open-source software and documentation enable continued improvements by different teams or organizations.
- The design is compatible with future upgrades, such as more efficient batteries and energy-saving computing units, enhancing long-term sustainability.

3 Cyber Security Considerations

In order to secure the robot and prevent unauthorized access, several network security measures have been implemented.

- **Wireless Connection Protection:** Any device attempting to connect to the cart wirelessly is required to enter a security password to prevent unauthorized access.
- **Hardware Access Protection:** The Intel NUC uses separate access passwords to restrict unauthorized personnel from logging in and ensure device security.
- **Software Security:** Install only official software and firmware updates to avoid malware invasion.
- **Periodic Security Review:** Regularly check security settings and logs to ensure that all security measures remain effective.
- **Security Incident Response:** If a security threat is detected, the password is immediately reset or the network is reconfigured.

By implementing these security measures, the robot is able to effectively defend against cyber threats, ensure operational security, and minimize security risks.

4 System

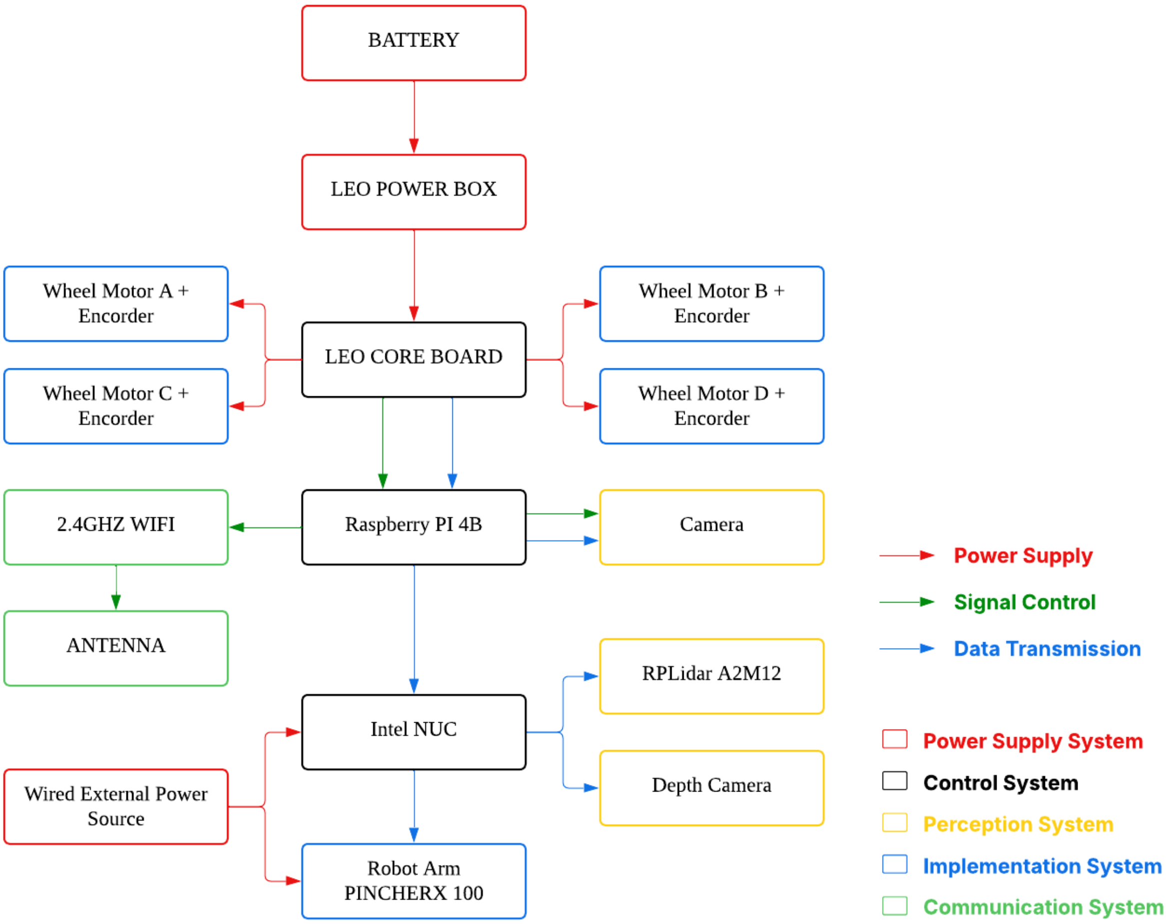

Fig. 1. System block-diagram of all the components in our robot.

All the components in the robot are described below. These components can be mainly categorized into: power system, control system, perception system, actuation system, and communication system. Each system has a clear division of labor and works together.

4.1 Power Supply System

1. **Battery:** Provides power to all subsystems.
2. **LEO Power Box:** As a power management unit, it is responsible for regulating current and voltage, and distributing power to other components to ensure the stability of power supply.
3. **Wired External Power Source:** Powers high power consumption devices such as robotic arm (PincherX 100) and Intel NUC.

4.2 Control System

1. **Intel NUC:** Main computing unit for processing complex data and running computationally intensive algorithms. For example, SLAM (Simultaneous Localization and Mapping) and navigation algorithms.

2. **Raspberry Pi 4B:** Processes sensor data, is responsible for communication with external devices and handles lighter tasks.

3. **LEO Core Board:** Reads data such as motion status, sensor readings, and controls motors and robot motion states.

4.3 Perception System

1. **RPLIDAR A2M12:** Provides a 360-degree scan of the environment and generates 2D maps.

2. **Intel RealSense Depth Camera:** Depth measurements provide access to the 3D structure of the environment, calculating the distance between the robot and an obstacle to accurately detect objects. Combined with RPLiDAR, it helps robots build maps and autonomously localize in unknown environments. Connects to the Intel NUC via USB and has the data processed by the Intel NUC.

3. **Camera:** Used for photography and visual processing tasks to aid in target detection and navigation.

4.4 Implementation System

1. **DC Motors and Encoders:** Four sets of DC motors control the robot's motion, and encoders provide real-time feedback of speed and position for precise control.

2. **Robot Arm (PincherX 100 Manipulator):** For gripping and manipulation tasks.

4.5 Communication System

1. **2.4GHz WiFi Module:** Provides a wireless communication function that supports two-way communication between the robot and external devices.

2. **External Antenna:** Enhances wireless communication signal and extended communication range.

5 Mechanical Design

The payload sled is designed in such a way that the robotic arm is mounted on a base plate. The base plate also has a bracket for the container. The container is placed on the front of the robot with an actuated bottom that opens. Lidar is placed on the under belly of the robot. Lidar's platform is supported by frames that are latched onto the robot. Frames hold lidar at the bottom of the robot and Intel NUC at the top of the robot. Four such frames are used to provide stability. Robot camera is mounted at the top of the arm with the help of a bracket.

5.1 3D CAD model of robot

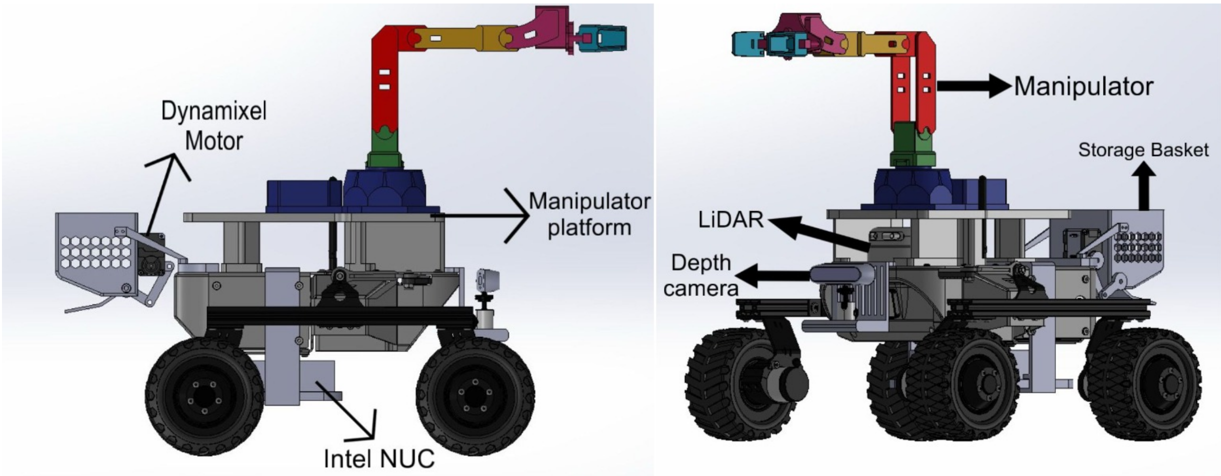

Fig. 2. 3D CAD model design

5.2 Design consideration and requirements

Considering the requirements of each and every sensor, the design is done in such a way that all sensors are able to function without any noises and hindrances. As sensors play a key role in autonomous navigation, it is important that they work accurately.

Camera: The camera is placed in front of the robot so that object detection and picking can be accurate. Camera's main purpose is to detect objects so the robotic arm can pick them. Camera should be placed at a higher level from the ground so it can detect objects at a far away distance and pass signals to the robotic arm. It is mounted on a ball-headed stand so the angle can be adjusted manually. The whole setup is on a platform that is mounted to the top metal plate of the robot.

Lidar: It is placed at the top of the chassis so it is not obstructed by other components. As it is directly mounted on the top of the chassis, it is firmly seated. At this position, the lidar provides scans of better accuracy.

Intel NUC: NUC has to be placed such that all the cables can be connected to and from it so it is placed at the bottom of the robot on a platform mounted to the chassis. The cables are wires can be connected to it without disturbances. It is easily accessible to all the other components on the robot.

Payload container: It is placed at a height from the base of the rover. It is supported by links attached to the top of the robot. It is designed keeping in mind the bin in which the blocks have to be dropped into. It has an actuated opening bottom. The robotic arm does not interfere with the basket as its shoulder and wrist can be adjusted in such a way. The blocks that has to be picked are also to be kept on top of a platform (as per the requirement) so the arm can easily pick and drop it into the container. The size and payload of the container is designed in such a way that it can hold more than 2 blocks. It has a drop-down chute kind of opening on the bottom.

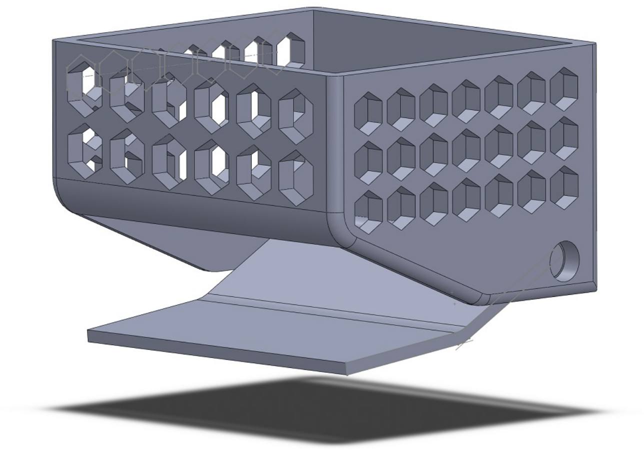

Fig. 3. Container CAD Model

5.3 Manufacturability

Manufacturing has to be done using latest technologies such as 3D printing and laser cutting. 3D printing enables rapid conversion of design concepts into physical components. It is also a sustainable way as it does not remove any material. Additive manufacturing only uses the material where required. These methods are the quickest and saves a lot of time. The parts are designed in such a way that it is light weight and also is rigid and stable. They are all manufacturable using 3D printing.

The design of container has been optimised for manufacturability with PLA (Polylactic acid) material to meet a balance between strength, lightweight properties, and ease of production. This model has a flat base that provides stability during printing and minimizes warping, hexagonal cutouts reduce material usage and weight without compromising structural integrity. The part orientation was carefully chosen to minimize overhangs in order to enhance manufacturability, and support structures were only used when necessary. A honeycomb infill pattern was selected for its excellent strength-to-weight ratio. It also includes adequate wall thickness and allowances to make the design durable and ensure smooth post-processing. These considerations taken altogether help in ensuring that the part is efficiently manufactured with quality while achieving the maximum capacity of PLA material.

Manipulator platform is also to be manufactured using PLA material. It has a long flat area for the manipulator base to be mounted and it is supported by 3 legs screwed to the top metal plate of the robots. Its thickness is 10 mm. It should be easy to print as it is not very complicated. Using PLA, both strength and less weight can be achieved. It will be able to handle the weight of the manipulator which is around 1kg as the structure is rigid and has enough pillars.

Platform for NUC and camera are both to be 3D printed using PLA material as well. Camera platform has some rectangular infill pattern to reduce the weight and material used.

6 Electrical Design

6.1 Main Modules and Connections

Effective power distribution is crucial in robotic system design. With varying voltage, current, and power needs across components, a stable power supply and optimized energy consumption are essential for system endurance. The table below outlines the electrical devices used, detailing their voltage, current, power requirements, and functions. This analysis helps optimize power management, ensuring system stability and efficiency.

6.2 Battery Runtime Analysis

The battery has a total capacity of 64.38Wh. Assuming the battery is fully discharged, the estimated runtime at full power consumption is:

$$\text{Runtime} = \frac{64.38\text{Wh}}{238.2\text{W}} \approx 0.27 \text{ hours} \approx 16.2 \text{ minutes} \tag{1}$$

However, the maximum output power of the battery is only 88.8W (calculated as $8A \times 11.1V$), which is significantly lower than the system's total power demand of 238.2W. Therefore, the battery alone cannot sustain the operation of the entire system.

Additionally, the Intel NUC requires 19V for operation, while the battery only provides 11.1V, leading to a voltage mismatch. Power supply optimization is necessary, and it is recommended to use an external power source for high-power components such as the Intel NUC and robotic arm.

6.3 Power Supply Optimization

To ensure stable and continuous system operation, the team optimized the power supply by dividing it into two sections based on the power consumption of each component and the battery's maximum output capability.

Device	Voltage (V)	Current (A)	Power (W)	Function
Battery Li-Ion 3s	11.1	8 (Max)	64.38Wh	Core power source of the system
LEO Power Box	12V/5V	-	-	Distributes battery energy; 12V powers motors & control boards; Step-down to 5V for low-voltage devices (Raspberry Pi, Wi-Fi module)
LEO Core Board	-	-	-	Controls motors and interacts with Raspberry Pi
Motors (4x)	12	2.2	26.4	Wheeled motors for mobility
Raspberry Pi 4B	5	3	15	Handles camera, Wi-Fi, and communication with Intel NUC
Wi-Fi Module	5	1.2	6	Provides wireless communication
Camera	3.3	0.3	1	Captures images for navigation or object detection
PincharX-100 Robotic Arm	12	5	60	Performs grasping tasks
Intel NUC	5V/19V	1A/6.32A	5/120	Handles high-performance computation tasks
RPLIDAR A2M12	5	0.5	2.5	Laser radar module for environmental perception
Depth Camera	3.3	0.7	2.31	Assists robotic arm operations with object recognition
Total Power Consumption	**-**	**-**	**238.2**	**Total power requirement for the system**

Table 1. Electrical Device Specifications

6.3.1 Section 1: Battery Power Supply

- Leo Rover chassis (including four wheeled motors and control system)
- Motors (4x)
- Raspberry Pi 4B
- Wi-Fi Module
- Camera

The total power consumption for this section is calculated as:

$$\text{Total Power Consumption} = \text{Motors (4x)} + \text{Raspberry Pi 4B} + \text{Wi-Fi Module} + \text{Camera} = 48.4W \quad (2)$$

Since 48.4W is lower than the battery's maximum output power of 88.8W, the battery can fully support this section's power needs, ensuring stable operation within this power range. According to Leo Rover's official website:

- In normal driving mode, the battery can last up to 4 hours.
- In video streaming mode, the battery can last up to 8 hours.

Based on this, the estimated power consumption in normal driving mode is:

$$\frac{64.38\text{Wh}}{4} = 16.09\text{W} \tag{3}$$

As the system's navigation power consumption is approximately 48.4W, which is higher than the normal driving mode estimate, the battery can still provide more than 1 hour of endurance, which is sufficient for most operational scenarios.

6.3.2 Section 2: External Power Supply

- Intel NUC (High-Performance Computing Unit)
- PincharX-100 Robotic Arm
- RPLIDAR A2M12 (LiDAR sensor) (for environmental perception)
- Depth Camera (for object recognition)

6.4 External Power Solution

Power Supply Method

- A wired external power source is used to directly supply Intel NUC and the robotic arm, preventing excessive load on the battery and ensuring long-term operation.
- Since Intel NUC requires 19V, while the battery only provides 11.1V, a separate power adapter is needed to meet the voltage requirements.

Cable Management

- As the robot needs to move freely, short power cables may restrict movement range or increase cable entanglement risks.
- To solve this issue, the team purchased and used extended power cables to ensure:
  - Continuous power supply to Intel NUC and the robotic arm.

- Unrestricted movement due to cable length constraints.
- Reduced dragging resistance and improved motion stability.

Based on the above analysis, we have developed the following electrical system architecture diagram. This diagram clearly illustrates the power distribution, power consumption requirements, and data communication connections of various components.

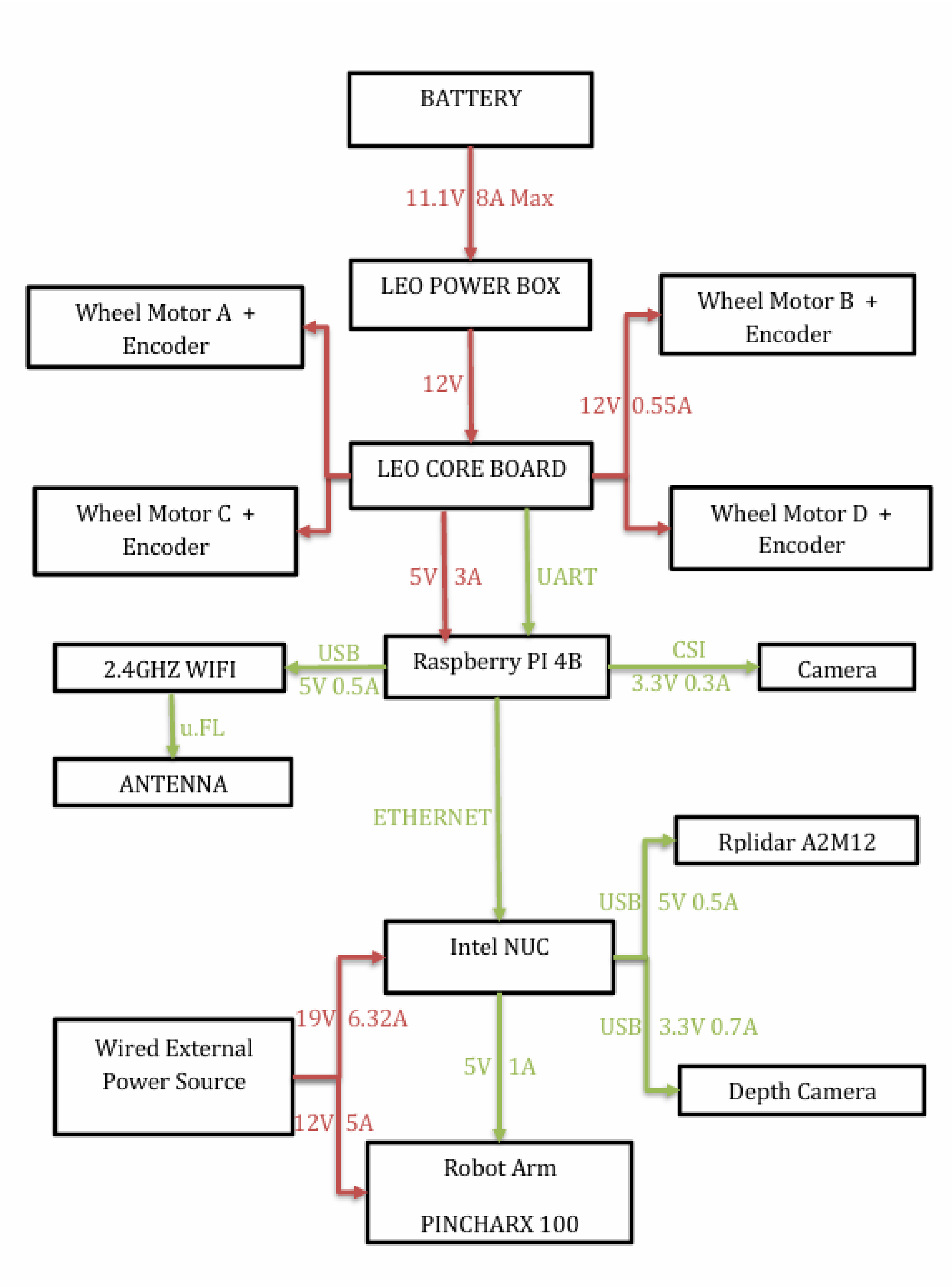

Fig. 4. Electrical System Architecture Diagram

6.5 Conclusion

Battery-powered section: Supports Leo Rover with a total power consumption of 48.4W, which is within the battery's power output capability, ensuring a reasonable endurance period.

Externally powered section: Supports Intel NUC the robotic arm and its sensor systems (LiDAR, Depth Camera), utilizing a wired external power source with extended cables to optimize mobility and ensure stable system operation.

With this optimized power supply strategy, the system can operate stably in different mission modes, leveraging battery power for flexibility while ensuring continuous power supply for high-power components, ultimately guaranteeing the reliability and sustainability of the platform.

7 Software Design

The following section explores in detail the visual representation of the nodes and topics in our project and shows the corresponding System Block Diagram. The overall architecture of our project consists of five functional modules, each of which has a specific role in the system, ensuring efficient execution of tasks and seamless collaboration between modules.

- Sensor Module
- Object Detection Module
- Task Control Module
- Execution Module
- Mapping & Navigation Module

Each module plays a specific role in the system and communicates efficiently through standardised ROS2 themes. The perception module collects key sensor data from the IMU, LIDAR, and camera nodes to provide the robot with information on attitude, environment, and object detection, which are the basis for path planning and task execution. The task control module is responsible for setting high-level mission goals and monitoring mission progress in real-time, while the mapping and navigation module builds an accurate model of the environment by fusing data from multiple sensors, and performs path planning and velocity control to ensure that the robot reaches its target efficiently and accurately. The execution module translates the navigation and mission commands into concrete physical actions in collaboration with the chassis and robotic arm.

Meanwhile, the System Block Diagram (SBD) visualises the communication and data flow between modules, providing a clear reference for understanding the system architecture. Next, we will introduce each module in more detail.

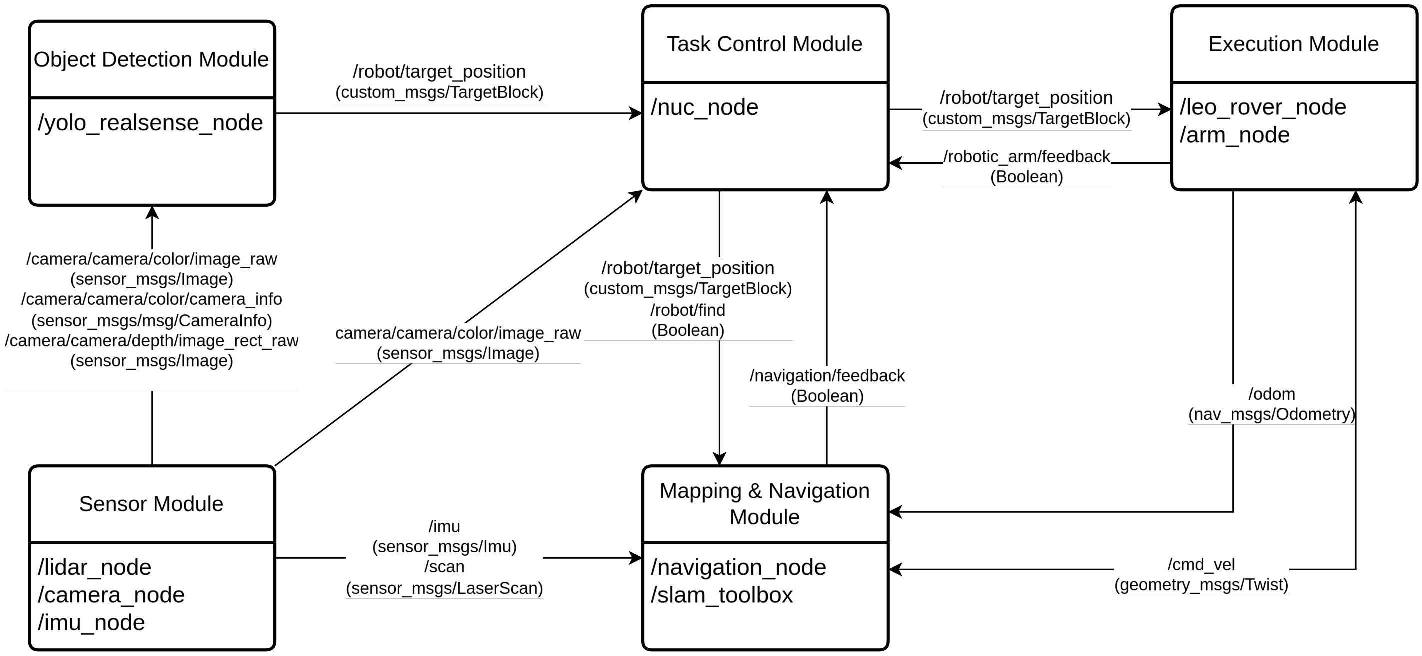

Fig. 5. Project System Block Diagram (SBD)

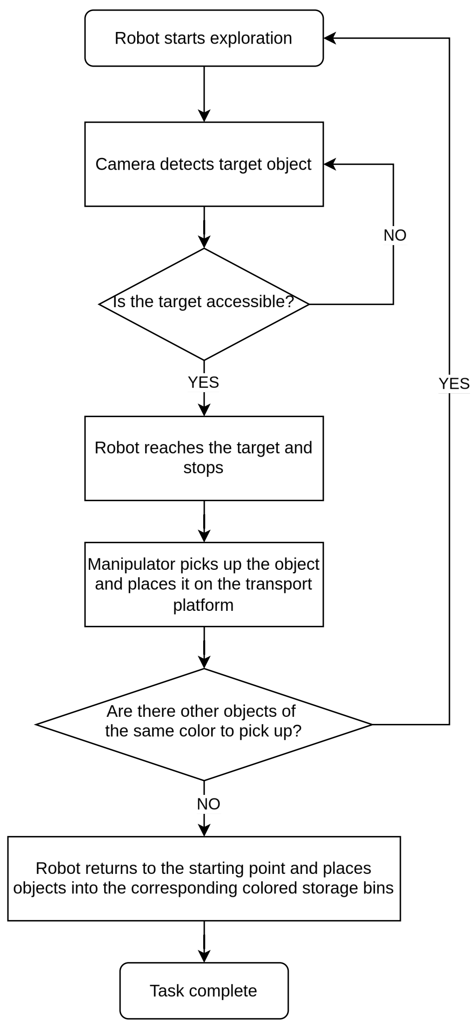

Fig. 6. Project RQT Graph

7.1 Sensor Module

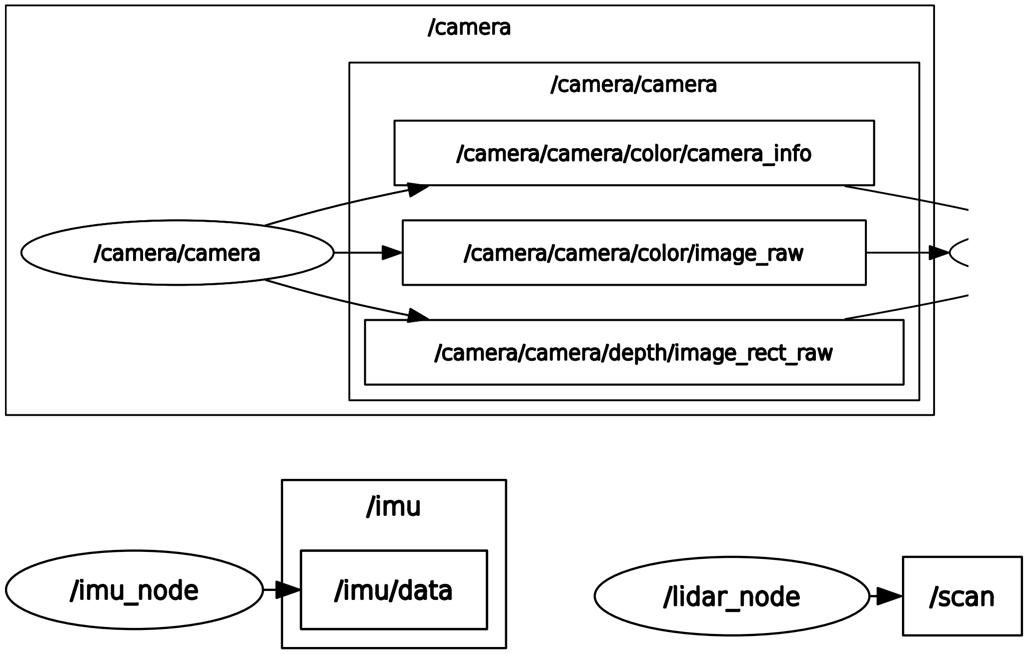

Fig. 7. Sensor Module RQT Graph

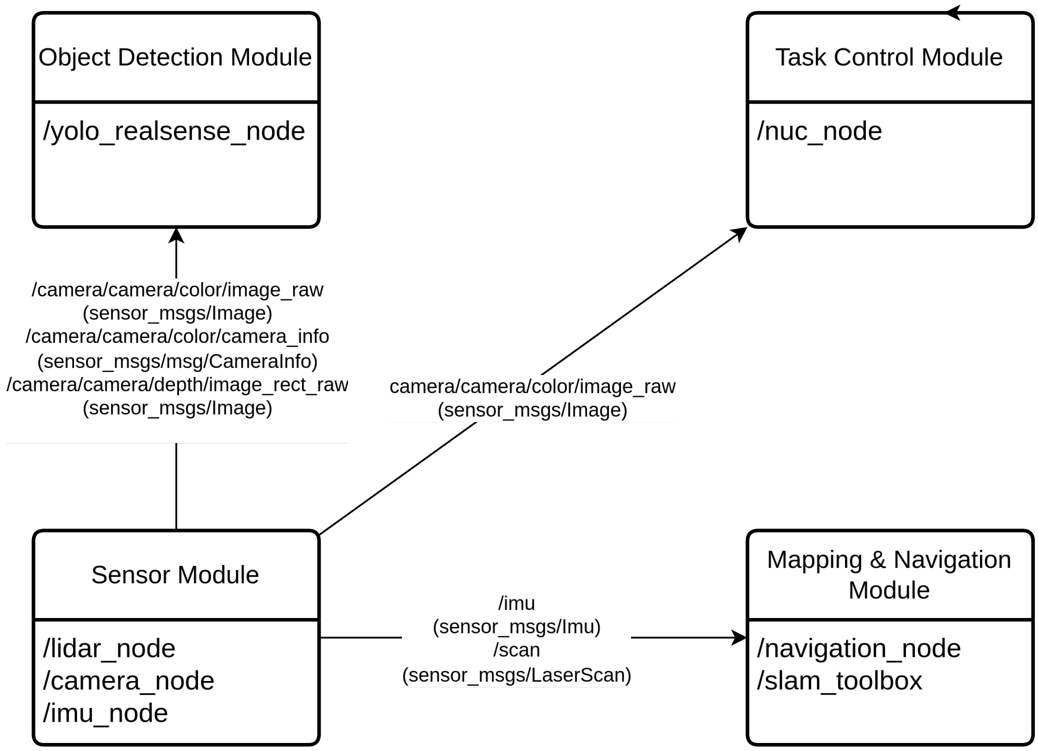

Fig. 8. Sensor Module SBD

The sensor modules include the IMU, LIDAR, and camera, which are used to provide data on the robot's kinematic state and environment awareness.The IMU node publishes `sensor_msgs/Imu` messages via `/imu/data`, which contain information on the robot's attitude (quaternion), angular velocity, and linear acceleration. This data is critical for navigation path planning and attitude control. The LIDAR node publishes `sensor_msgs/LaserScan` messages via /scan, which provide distance and angle data on surrounding obstacles for obstacle avoidance and map generation. The camera node publishes `/camera/camera/color/image_raw (sensor_msgs/Image)`, `/camera/camera/color/camera_inf` `(sensor_msgs/msg/CameraInfo)`, `/camera/camera/depth/image_rect_raw (sensor_msgs/Image)` to provide real-time image data of the environment.

7.2 Object Detection Module

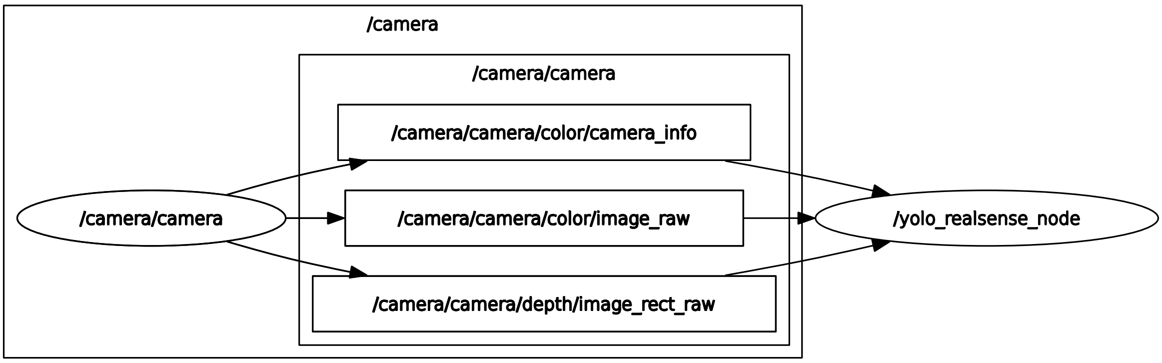

Fig. 9. Object Detection Module RQT Graph

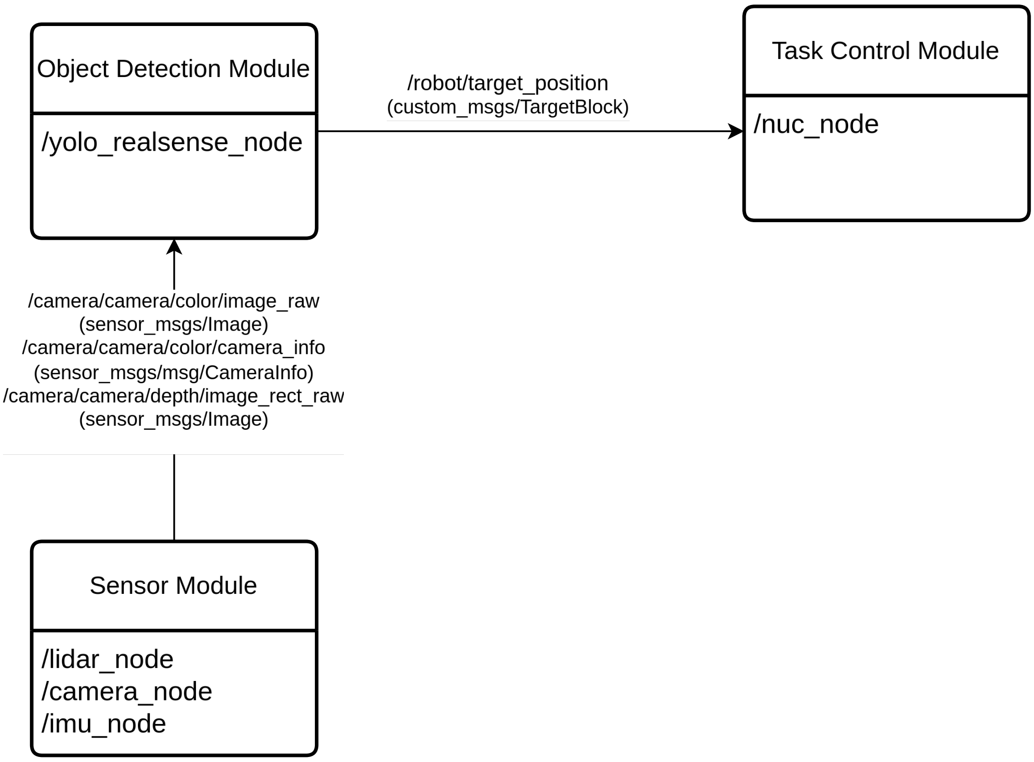

Fig. 10. Object Detection Module SBD

In the object recognition part, we used Deep Learning Algorithm to help the robot use the camera to recognize different colored wooden blocks. Specifically, I used the YOLOv8n-seg model for training to achieve efficient and accurate target detection and segmentation tasks.

In order to construct a high-quality dataset, more than 1200 images were captured using the camera, and all images were labeled by the X-AnyLabeling software to ensure the accuracy of the training data. During the model training process, I set training rounds (epochs) = 100 and batch size = 8 and completed the model optimization based on these parameters. Eventually, the model successfully converged and achieved high recognition accuracy on the target detection task. This node will publish a custom_msgs message containing the object's x, y, z position, color, and type (either bin or block).

The following figure shows the trend of loss change during the model training process and the performance evaluation results.

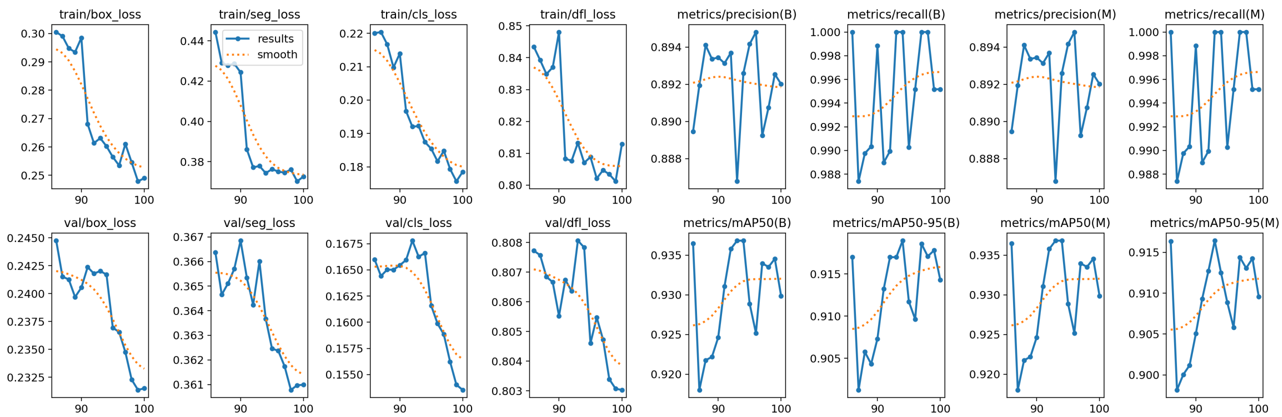

Fig. 11. Result of Deep Learning Algorithm

7.2.1 Training Losses and Validation Losses Analysis

During training, we focused on tracking the following four key loss terms:

- **Boundary box loss (box_loss)** - measures the accuracy of target box predictions
- **categorization loss (cls_loss)** - measures the error in category prediction
- **Segmentation Loss (seg_loss)** - computes loss for semantic segmentation tasks
- **Distributed Focal Loss (dfl_loss)** - further optimizes target box accuracy

Trends in training loss and validation loss:

- **Training loss** is gradually decreasing, indicating that the model parameters are continuously optimized and the learning effect is good.
- The **validation loss** also shows a decreasing trend and stays close to the training loss, indicating that the model has a better generalization ability and there is no obvious overfitting phenomenon.
- **box_loss and cls_loss have a larger decrease**, indicating that the accuracy of target box prediction and target classification have been significantly improved.
- There is **a small fluctuation in dfl_loss**, probably due to the fact that the model is still adjusting the parameters during the bounding box optimization process.

Summary: The simultaneous decrease of training loss and validation loss indicates that the model is converging stably and has better generalization ability.

7.2.2 Evaluation indicators (Precision, Recall, mAP)

In order to fully evaluate the model's target detection performance, we analyzed the following key metrics:

- **Precision**: how many of the targets predicted by the model are correct (the higher the better)
- **Recall**: the proportion of all real targets that are correctly detected by the model (the higher the better)
- **mAP50 (mean Average Precision @ IoU 0.5)**: the average precision calculated under conditions of IoU ≥ 0.5
- **mAP50-95**: average precision calculated across multiple thresholds between IoU 0.5 and 0.95, measuring the model's performance under more stringent criteria

Analysis of evaluation results

- **Precision stays above 0.89**, indicating that the model has a low false detection rate and strong predictive stability.
- **Recall is close to 1.0**, indicating that the model is able to detect most of the target objects with a low miss detection rate.
- **The mAP50 continues to rise**, reflecting that the model is able to maintain high detection accuracy under looser IoU thresholds.
- **The mAP50-95 is also increasing**, indicating that the model's performance remains stable even under more stringent IoU criteria.

Although there are some fluctuations in Precision and Recall during the training process, the overall trend is upward and there is no significant decrease, indicating that the model has good stability.

Summary: The evaluation results show that the model performs well in the target detection task, and the recognition precision, recall, and mAP metrics are all at a high level, with good training results.

7.2.3 Possible directions for optimization

Although the current model training is better, there is still room for further optimization. The following are a few possible directions for improvement:

- **Expanded dataset**: It is observed that the model has already converged to a better performance at training up to around 80 rounds (epochs). Further increasing the diversity of the dataset may be more effective than simply extending the number of training rounds, in particular, collecting more samples under different lighting conditions, angles, and backgrounds can help to improve the generalization ability of the model.
- **Adjusting the learning rate**: The training loss fluctuates slightly in some epochs, probably because the learning rate is not stable enough. You can reduce the learning rate to improve the convergence stability of the model and avoid drastic parameter updates.
- **Enhanced Data Preprocessing and Data Enhancement**: Adopt data enhancement techniques, such as: Random Cropping, Rotation, Brightness/Contrast Adjustments, Color Jittering. Improve model robustness and reduce the risk of overfitting through data enhancement. and reduce the risk of overfitting.

7.3 Task Control Module

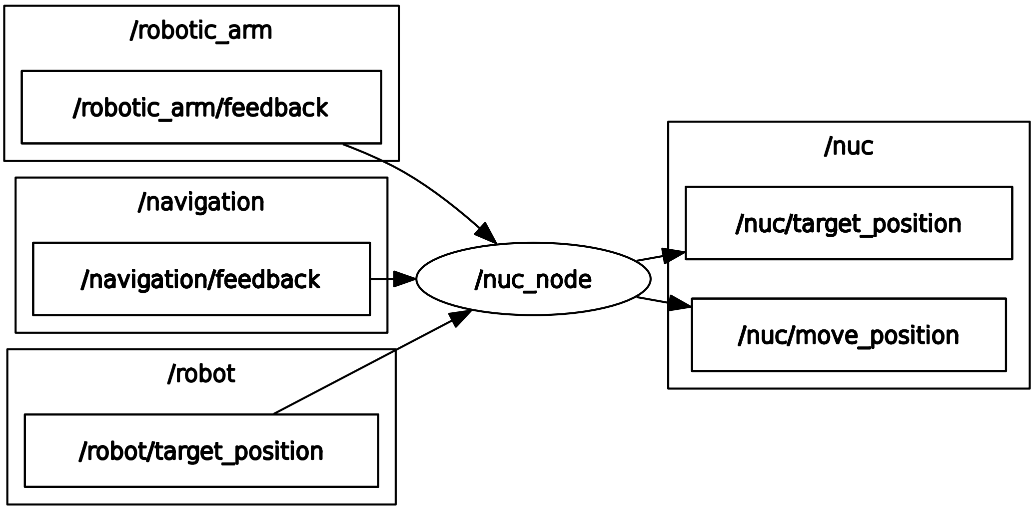

Fig. 12. Task Control Module SBD

The task control module consists of `/nuc_node` and is responsible for high-level task management and coordination. It specifies the goal task to be accomplished by the robot, such as navigating to a specified location, by sending a message of type `custom_msgs/TargetBlock` to the `/nuc/move_position` topic. The module also sends messages to the `/nuc/target_position` topic (`custom_msgs/TargetBlock`) to perform fine-grained operations on the robot arm, such as grasping or placing objects. In addition, `/nuc_node` subscribes to `/robot/target_position` to use the object detection information provided by the camera for task tuning, improving the accuracy and reliability of task execution. The module also receives navigation status feedback via `/navigation/feedback (Boolean)`, which is used to monitor the progress of the robot's movement, and also receives robotic arm status feedback via `/robotic_arm/feedback (Boolean)`, which is used to dynamically adjust the task.

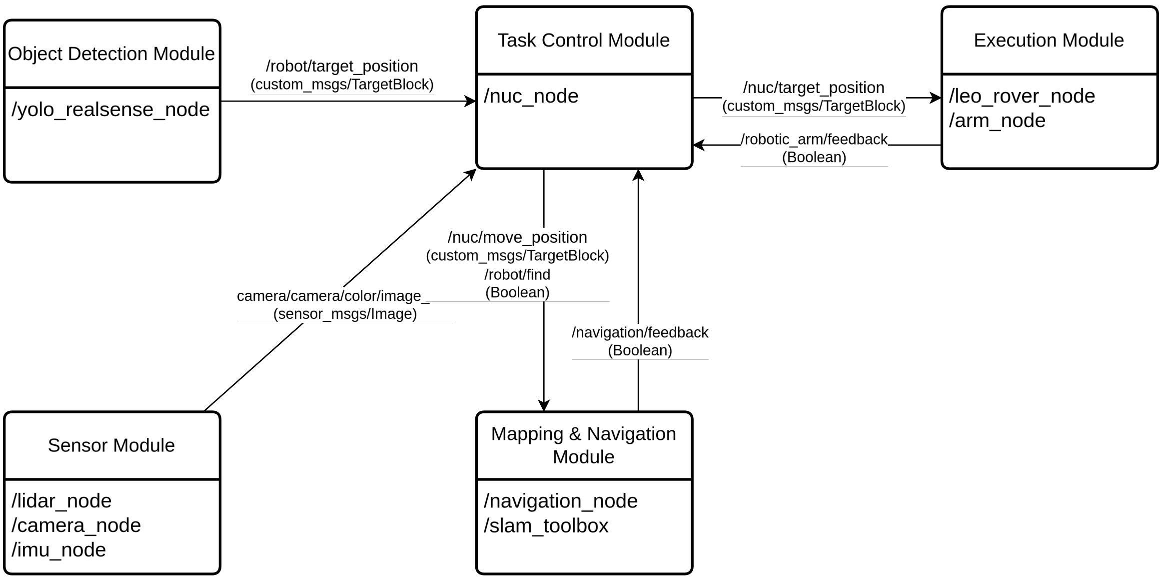

Fig. 13. Task Control Module SBD

7.4 Execution Module

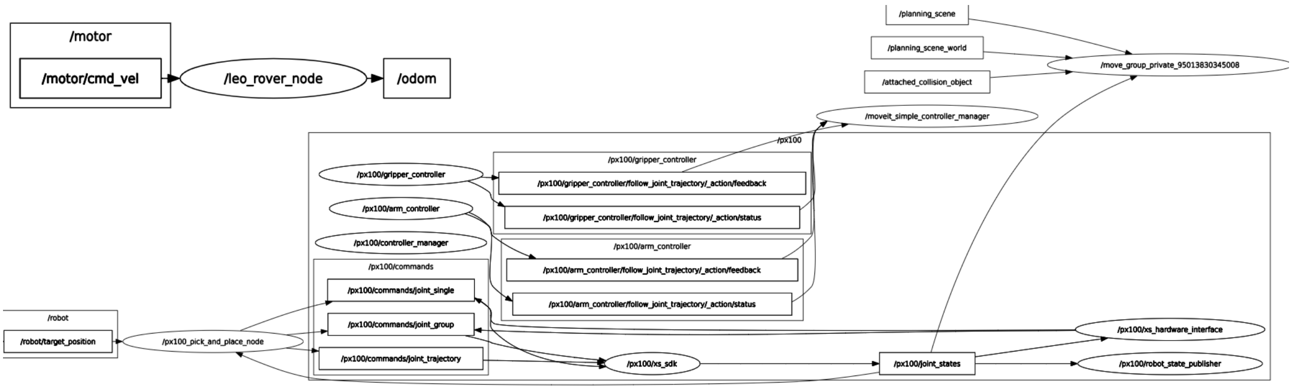

Fig. 14. Execution Module RQT Graph

The execution module consists of `/leo_rover_node` and `/arm_node`, which are responsible for chassis motion control and robotic arm operation respectively. `/leo_rover_node` subscribes to `/cmd_vel` (`geometry_msgs/Twist`), which converts velocity commands into motor control signals to drive chassis motion. At the same time, robot position information is published via `/odom` (`nav_msgs/Odometry`), which provides feedback on the current odometer status for navigation and task monitoring. `/arm_node` is responsible for robotic arm operation, subscribing to `/nuc/target_position` (`custom_msgs/TargetBlock`) to perform tasks such as grasping, placing, etc., and providing feedback on the arm's operating status via `/robotic_arm/feedback` (`Boolean`), supporting real-time monitoring and adjustment of the task control node.

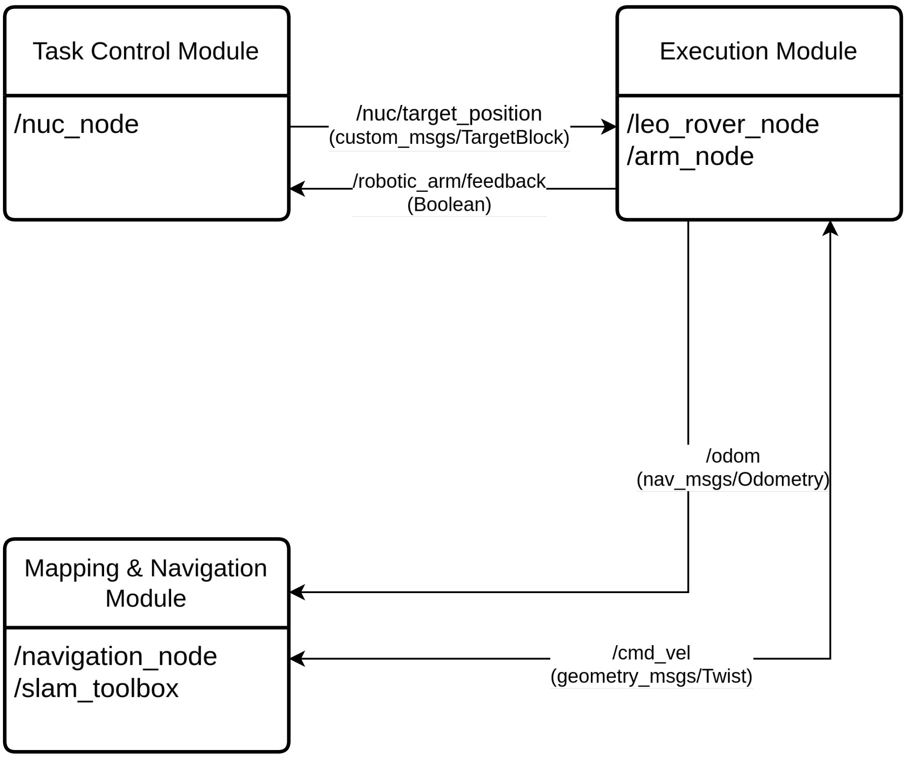

Fig. 15. Execution Module SBD

7.5 Mapping & Navigation Module

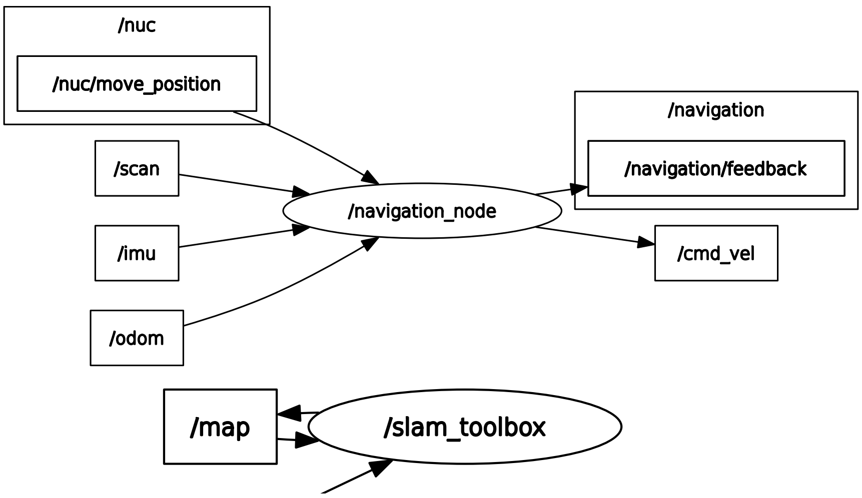

Fig. 16. Mapping & Navigation Module RQT Graph

The mapping and navigation module consists of `/navigation_node` and `/slam_toolbox`, which are responsible for map construction and path planning. `/slam_toolbox` subscribes to `/scan` data to generate a global occupancy grid map, which provides the environmental information base for navigation. `/navigation_node` is responsible for path planning, speed control and navigation task execution, and performs path stability correction by subscribing to the attitude information provided by `/imu/data`, and generates paths by subscribing to the target information provided by `/nuc/move_position`. planning, and send velocity commands through `/cmd_vel` (`geometry_msgs/Twist`) to control the robot chassis movement and get the chassis' real-time motion status. At the same time, the navigation module will report the real-time status of the navigation task to the task control

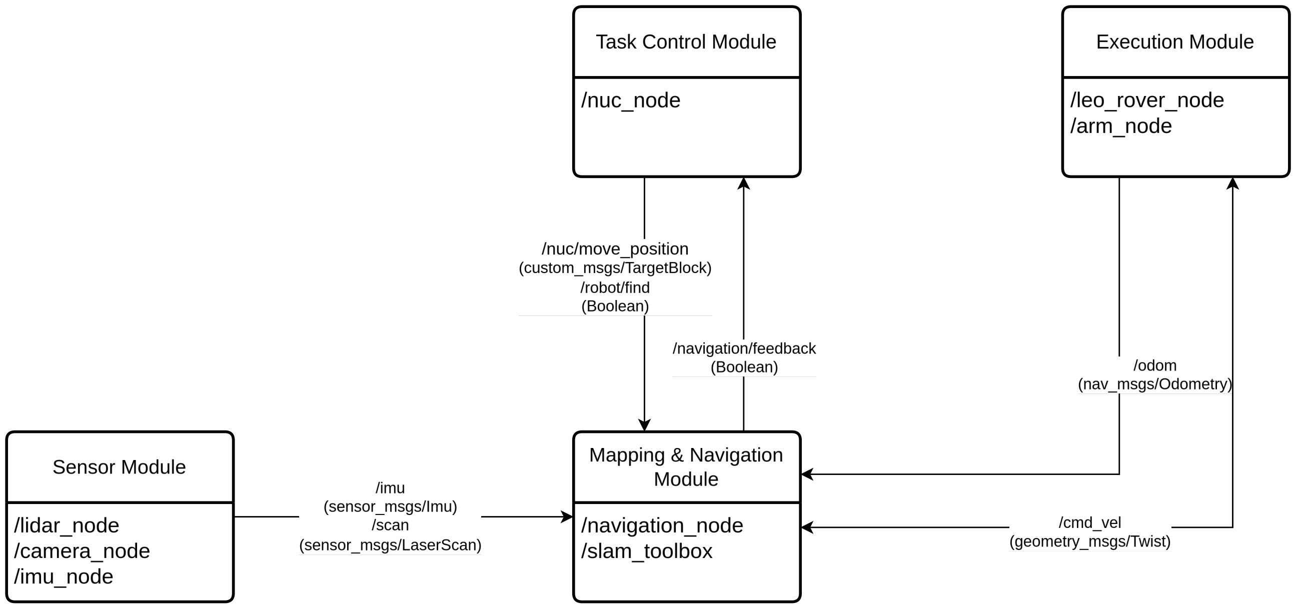

Fig. 17. Mapping & Navigation Module SBD

module, realizing closed-loop feedback and improving the reliability and controllability of task execution.

7.6 Team3's Git repository

https://github.com/Haoran-01/team3-project

8 Analysis

8.1 The robot shall operate smoothly and reliably in the specified environment.

(Environment Requirements Verification Matrix 3.4.1, 3.4.2 and 3.4.3)

8.1.1 Verification method

Test Environment: Dry, level indoor floor. During the test, ensure that the robot is not subject to external disturbances, such as collision of objects or influence of people.
Functionality Test: Run the robot in the test site and observe its motion status and adaptability. Test the robot's ability to operate on both flat surfaces and slightly rough surfaces (e.g., man-made slight bumps) to verify its terrain adaptation. The test scope covers at least 10 round-trip tasks of the robot.
Performance Metrics: Smoothness and stability of the robot's movement in the environment. No obvious slipping, jamming or tipping during operation.
Variance Analysis: If the test results do not meet expectations, the reasons need to be analyzed, such as hardware or tire configuration problems and instability of motion control algorithms. Adjust and retest to ensure that the functional requirements meet the standard.

Success Criteria: The robot is able to operate stably and efficiently in the environment with 100% task completion. There are no environmentally induced failures or deviations from the path during operation.

8.1.2 Realization Conditions

Hardware Configuration: Equipped with large diameter (130 mm) rubber tires for enhanced terrain adaptability. The chassis is designed with four-wheel drive to ensure stable driving on rough ground. Appropriately equipped with IMU for attitude adjustment and ground condition monitoring.
Environment Control: Ensure that the robot's operating environment is indoor, with a dry and level floor. Keep no one or no object interference during operation.

8.2 The robot shall be able to recognize objects accurately

(Object Recognition Requirements Verification Matrix 3.1.1, 3.1.2 and 3.1.3)

8.2.1 Verification method

Test Method: Multiple blocks with different colors and shapes are arranged in the experimental site. Capture RGB images via Intel RealSense depth camera and run the image processing and recognition algorithm. Record the recognition results (color, shape) of the objects in each test.
Comparison of results: Compare the results obtained by the recognition algorithm with the characteristics of real objects: verify the correctness of color classification, verify the accuracy of shape classification, and verify the ability of multi-target detection.
Discrepancy analysis: If there are errors or omissions in the recognition results, analyze whether the parameter settings of the depth camera are appropriate (e.g., field of view, resolution), check whether the color segmentation and shape extraction algorithms are processed accurately, adjust the algorithm thresholds or optimize the model, and re-test.
Success criteria: 100% accuracy in object recognition, including color and shape classification, and correctness in multi-target detection. The robot is able to recognize multiple objects in real time.

8.2.2 Realization Conditions

Sensor Configuration: Equipped with an Intel RealSense depth camera that has 2% depth accuracy and provides RGB images. The depth camera is mounted above the front of the robot, covering the main activity area in front.
Image Acquisition and Preprocessing: The depth camera acquires RGB images. Noise reduction is performed on the images to improve data quality.
Image Segmentation: Use color segmentation algorithm to extract different colored object regions. Convert RGB image to HSV space and set appropriate threshold range for color filtering.

Feature Extraction and Classification: Extract 3D shape features (e.g., size, edge contour) of objects based on depth map data. Apply image recognition algorithms (e.g. CNN or Support Vector Machine SVM) to classify the objects.
Multi-target detection and tracking: Implement multi-target detection using YOLO (You Only Look Once) or Faster R-CNN.
Data Processing and State Estimation: Realize real-time object detection algorithm, fusion processing of depth data and RGB images.Based on the sensor input, the recognition results are dynamically adjusted by the algorithm to improve real-time and robustness.
Autonomous Recognition and Response: After recognizing an object, the robot is able to perform subsequent tasks, such as obstacle avoidance, based on the recognition results.

8.3 The robot shall be able to detect obstacles in the environment and create a map

(Autonomous Navigation Requirements Verification Matrix 3.2.1)

8.3.1 Verification method

Map Comparison Test: Compare the map generated by the robot with an accurate map of the known environment (at least 4 tests). Analyze the match between the point cloud features and the actual environment.
Difference Analysis: If significant discrepancies exist, analyze potential causes, such as inaccurate sensor data or misconfigured SLAM algorithms. Perform adjustments and retests as needed.
Success Criteria: Point cloud features in the map must match the actual environment with an accuracy rate exceeding 95%.

8.3.2 Realization Conditions

Sensor Configuration: Equipped with IMU and LiDAR sensors to perceive the robot’s posture and gather environmental information.
SLAM Package: Use an AMCL algorithm-based SLAM package for map construction.
Data Processing and State Estimation: SLAM algorithms process sensor data via Kalman or particle filters to estimate the robot’s state.
Autonomous Localization and Navigation: Using the generated map, the robot shall navigate autonomously, dynamically scan, and update the environment map.

8.4 The robot shall locate itself in the map, navigate autonomously, and adjust its path and speed dynamically without colliding with obstacles

(Autonomous Navigation Requirements Verification Matrix 3.2.2)

8.4.1 Verification method

Localization Accuracy Test: Place the robot at a known position and use SLAM (e.g., AMCL) to detect its location. Validate accuracy by comparing the robot's estimated position on the map to its actual position.
Success Criteria: Localization errors shall remain within a specific range (e.g., less than 5% error rate), ensuring precise positioning on the map.

8.4.2 Path Planning and Obstacle Avoidance Validation

Path Planning Test: Simulate path planning tasks in different environments, such as navigating from a starting point to a target object, and verify the robot can find unobstructed paths.
Obstacle Avoidance Validation: Place various obstacles in the robot's path and observe its avoidance strategy and performance. Conduct 5 experiments with obstacles in different locations, ensuring they can be detected by LiDAR.
Difference Analysis: If obstacles are not avoided, analyze possible causes, such as detection failure or planning errors, and optimize accordingly.
Success Criteria: The robot must accurately detect obstacles, construct environmental maps, and avoid collisions during navigation.

8.4.3 Realization Conditions

Sensor Configuration: Equipped with LiDAR and depth cameras for real-time obstacle detection and environmental data acquisition.
Path Planning Algorithm: Combine A* (global path planning) and DWA (local path planning) algorithms to dynamically adjust paths and avoid obstacles.
Real-Time State Estimation: Employ sensor fusion (e.g., IMU and LiDAR) to ensure accurate and stable path planning.

8.5 The robotic arm shall firmly grasp opaque wooden cubes weighing below to 50g and safely place them into a top-mounted box, which it can then empty

(Manipulation of Robotic Arms Requirements Verification Matrix 3.3.1, 3.3.2, 3.3.3 and 3.3.4)

8.5.1 Verification method

Grasping Test: Test the robotic arm's ability to securely grasp opaque wooden cubes weighing up to 50g. Place cubes at varying heights, angles, and distances (e.g., ±50mm offset) and observe if the arm can stably grasp and maintain its grip on different objects. The system shall retry after a failed grasp.

Placement Accuracy Test: Test if the arm can accurately place the cubes into the box, achieving continuous success in more than 5 attempts.
Grasping and Placement Test: Validate the robotic arm's stability during continuous grasping and placement cycles. Perform 10 consecutive cycles and check for jamming or anomalies.
Box Emptying Test: Verify the box can empty its contents. Conduct 10 consecutive trials, checking for smooth operation without jamming or anomalies.
Success Criteria: The grasp success rate shall exceed 90% (including retries). Wooden cubes shall accurately enter the box, and the robotic arm shall operate smoothly without anomalies.

8.5.2 Realization Conditions

Hardware Design:
Pincher X100 Robotic Arm: Offers a flexible range of motion to cover different heights, angles, and distances.
End Effector (Gripper): Equipped with clamps suitable for gripping cubes weighing up to 50g.
Box Installation: Designed with a top opening to allow the robotic arm to place cubes smoothly. Fitted with a motor to empty cubes into a designated area.
Vision Recognition Design: Integrate real-time target detection algorithms (based on color or shape recognition) using an Intel NUC mini-PC.

8.6 The robot shall autonomously transport objects to the drop-off point and return to the starting point without human intervention.

(Autonomous Navigation Requirements Verification Matrix 3.2.3)

8.6.1 Verification method

End-to-End Autonomous Task Test: Design a complete scenario where the robot starts from an initial position, identifies the target object, avoids obstacles, grasps the object, transports it to the drop-off point, and returns to the starting position. No human intervention is allowed throughout the process.
Multi-Task Cycle Test: Complete multiple pick-and-place tasks within 20 minutes, recording completion rate, error rate, and efficiency metrics (e.g., tasks completed per unit time).
Complex Environment Test: Test the robot in low-light, high-obstacle-density, or complex-path environments to validate stability and efficiency.
Long-Duration Stability Test: Run the robot for extended periods to observe potential issues like battery depletion, grasp failures, or path planning errors.
User Intervention Rate Test: Record the need for manual adjustments during operation (e.g., path correction or restarts). The goal is to achieve zero manual intervention.
Success Criteria: The robot must complete the end-to-end process without human intervention, achieving a task success rate of over 95%, and operate smoothly and stably.

8.6.2 Realization Conditions

Complete all prior tasks:

Task 1: Robot environmental adaptability testing.

Task 2: Identify colors and shapes of objects.

Task 3: Detect obstacles and create a map.

Task 4: Localize, navigate autonomously, and dynamically adjust paths and speed. Avoid collisions.

Task 5: Securely grasp opaque wooden cubes and safely place them into the top-mounted box, which can then empty its contents.

9 Project Plan

The Gantt chart below clearly shows the timetable, task allocation (Hisham, Yan, Lyu, Zhang) and key milestones for each work package for the next term, visually reflecting the overall planning and milestones of the project.

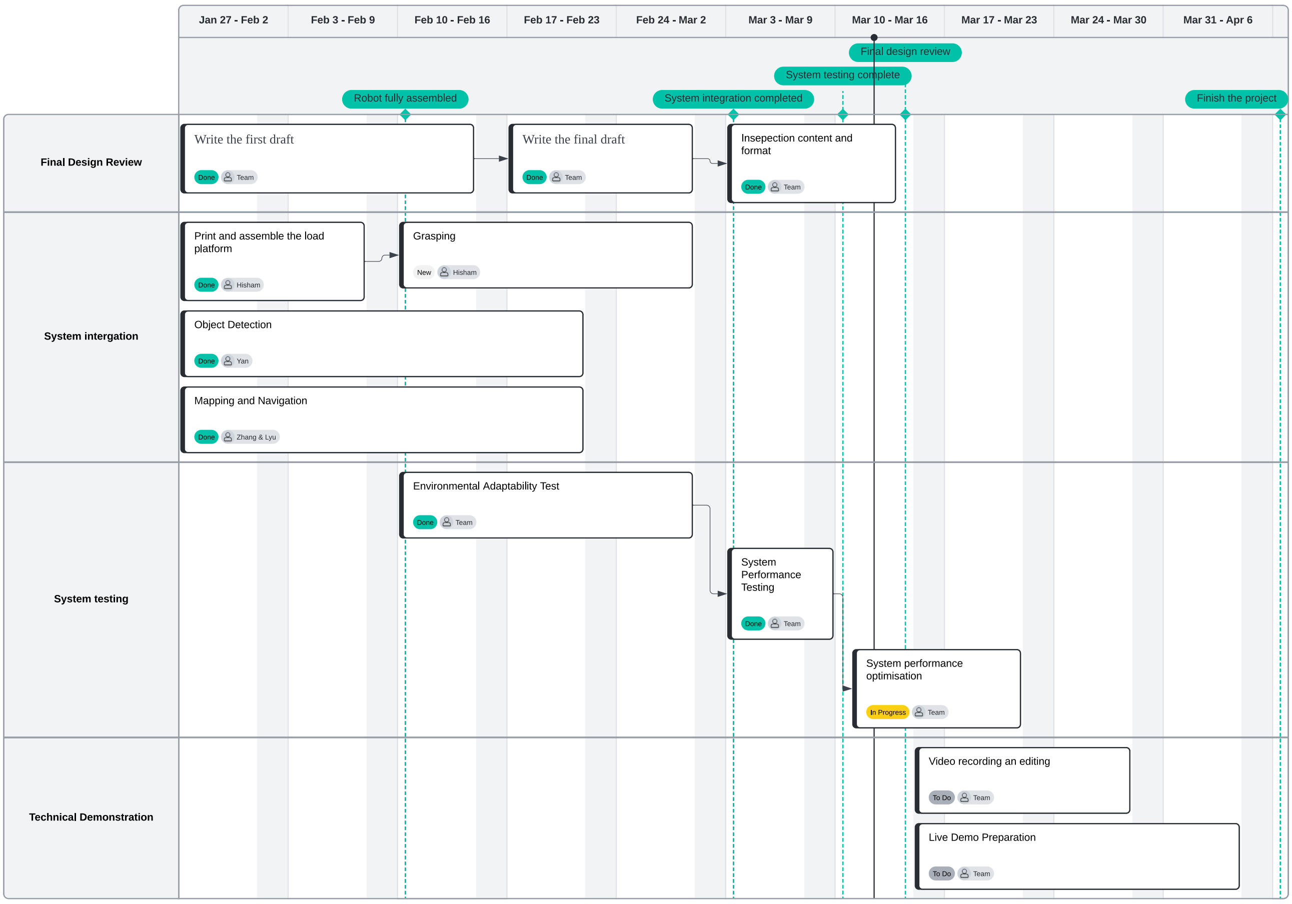

Fig. 18. Project Plan Gannt Chart

A Design Requirements

A.1 Stakeholder Engagement

Our group's project aims to:

Develop a robot capable of autonomously retrieving colored objects from the environment and placing them into corresponding storage bins at the starting point."

At the initial stage of the project, we conducted discussions with the user to understand their specific requirements. Through two 15-minute face-to-face meetings, we documented the following key user needs.

Engagement	Customer Requirement
What is the shape, size, and color of the objects?	The robot will handle blocks of specific geometric shapes (e.g., cubes), with standard sizes (e.g., 10 cm^3), and multiple colors such as red, blue, green, and yellow. The robot must identify these with high accuracy.
In what environment will the robot operate?	The robot will navigate a dynamic environment with obstacles like walls. It must efficiently reach blocks placed randomly and return them to storage bins without collisions.
What are the specifications of the manipulator?	A third requirement is that the robot should be equipped with a manipulator for retrieving the colored blocks placed randomly in the environment. It should also be able to accurately place it in the designated storage bins.
What is the expectation on controller and interface?	The customer requires the robot's control, operation, and hardware to be constrained to the components provided with the Leo Rover kit.

Table 2. Requirements Verification Matrix

A.2 Problem Statement

Develop an autonomous robot capable of retrieving colored blocks scattered within a designated environment and placing them into corresponding storage bins based on color. The storage bins are located in the starting point. The robot must be able to accurately detect, differentiate, and handle blocks of various colors, navigate the environment while avoiding obstacles, and operate independently for extended periods without human intervention. The solution should achieve high accuracy in color sorting, and ensure safe interaction within the environment.

Objectives

- The team will use a four-wheeled rover, assemble it, and install the various peripherals required.
- The team will utilize the robot to successfully retrieve specified items in the shortest possible time and return the robot to its starting position successfully.
- The team’s goal is to enable the robot to autonomously plan paths and draw maps in its environment.
- The goal of the group is to be able to access data from the rover in order to understand the progress of its work in real time.
- The goal of the group is to enable the robot to operate safely and reliably.

A.3 Functional Requirements & Performance Requirements

A.3.1 Object Recognition

1. The robot shall be able to detect objects in the environment
2. The robot shall be able to recognize multiple objects
3. The robot shall be able to recognise the colour of the object and its shape

A.3.2 Autonomous Navigation

1. The robot shall detect obstacles in the environment and create a map.
2. The robot shall localize itself within the map framework and navigate autonomously while dynamically adjusting its path in real-time.
3. The robot shall autonomously transport objects to the drop-off point and return them to the collection point without human intervention.

A.3.3 Manipulation of robotic arms

1 The robotic arm shall firmly grasp opaque wooden cubes of various shapes.

2 The robotic arm shall safely place the objects into a top-mounted box.

3 The robot shall be able to pick up objects on the ground using a robotic arm

4 The box shall be capable of emptying its contents.

A.3.4 Environment

1 The robot shall be operated indoors.

2 The robot shall be operated without interference from external objects or other people.

3 The robot shall be operated only on dry and level ground with no highly visible undulations or traps.

Object Recognition Requirements Verification Matrix					
Requirement No.	Shall Statement	solution	Verification Success Criteria	Verification Method	Result
3.1.1	Detect objects in the environment.	Use a depth camera to capture object shapes and locations.	The robot detects all objects in its vicinity.	Conduct object detection tests in controlled environments.	
3.1.2	Recognize multiple objects.	Implement a multi-object detection algorithm.	The robot correctly identifies at least 95% of objects in its field of view.	Test object detection on datasets with multiple objects.	
3.1.3	Recognize the color and shape of objects.	Use image processing techniques for color filtering and shape analysis.	The robot correctly identifies the color and shape of 95% of objects.	Conduct tests with colored and shaped objects in varying environments.	

Autonomous Navigation Requirements Verification Matrix					
Requirement No.	Shall Statement	solution	Verification Success Criteria	Verification Method	Result
3.2.1	Detect obstacles in the environment and create a map.	Use LiDAR and SLAM to create a map of the environment.	The robot generates a map with >90% accuracy.	Test obstacle detection and mapping in a cluttered indoor environment.	
3.2.2	Localize itself and navigate autonomously while dynamically adjusting its path in real-time.	Implement real-time path adjustment using SLAM and odometry.	The robot maintains localization accuracy within 5 cm and avoids obstacles dynamically.	Perform navigation tests with dynamically placed obstacles.	
3.2.3	The robot shall autonomously transport objects to the drop-off point and return them to the collection point.	Use a predefined pick-and-place task combined with path planning algorithms.	The robot successfully completes transport tasks without human intervention in 95% of attempts.	Conduct task completion trials in a controlled environment.	

Manipulation of Robotic Arms Requirements Verification Matrix					
Requirement No.	Shall Statement	solution	Verification Success Criteria	Verification Method	Result
3.3.1	Firmly grasp opaque wooden cubes of various shapes.	Design a gripper with adaptable grip force and precision.	The robotic arm securely grasps 95% of wooden cubes in test trials.	Perform grasping tests with cubes of various shapes and weights.	
3.3.2	Securely place the objects in the box on the load platform.	Implement a trajectory planning algorithm for precise placement.	Objects are placed into the box without displacement in 95% of cases.	Perform placement trials with objects of varying sizes and weights.	
3.3.3	Pick up objects on the ground using a robotic arm.	Design the arm to reach ground level and use visual feedback for positioning.	The robotic arm successfully picks up objects on the ground in 95% of trials.	Conduct ground pick-up tests with varying object positions.	
3.3.4	The box shall be capable of emptying its contents.	Use a tilt mechanism or automatic release system for the box.	The box successfully empties its contents in 100% of trials.	Conduct box emptying tests with varying loads.	

Environment Requirements Verification Matrix					
Requirement No.	Shall Statement	solution	Verification Success Criteria	Verification Method	Result
3.4.1	The robot shall be operated indoors.	Ensure the robot’s components are designed for indoor conditions.	The robot functions as expected in various indoor scenarios.	Conduct functional tests in different indoor environments.	
3.4.2	The robot shall be operated without interference from external objects or people.	Use sensors to detect interference and halt operations as needed.	The robot halts or avoids interference 100% of the time.	Test the robot’s behavior in environments with moving objects or people.	
3.4.3	The robot shall be operated only on dry and level ground.	Ensure the wheels and sensors are calibrated for flat surfaces.	The robot operates without slipping or tilting in 100% of trials on flat surfaces	Conduct mobility tests on dry, flat terrain.	

B Workplace Charter

B.1 Our Purpose

- **Objective:** This document outlines the rules, commitments, and procedures to ensure smooth and effective collaboration for the successful completion of the ROS robot project.
- **Usage:** It serves as a reference to maintain team cohesion, resolve conflicts, and keep track of each member's responsibilities.

B.2 Statements of Principles and Commitments

- **Equity:** We are committed to creating a fair working environment for every team member, ensuring that everyone has an equal opportunity in their role. We will allocate resources appropriately based on individual circumstances to ensure they receive the support they need to realize their full potential and make an outstanding contribution.
- **Diversity:** We value the rich variety of cultural and educational backgrounds within our team as our strength. We encourage the expression of diversity and believe that different perspectives can bring more innovation and creativity to projects. We are committed to fostering an environment where every member can actively participate and ensure that their voices are heard and respected in discussions, resulting in more holistic and inclusive team decisions.
- **Inclusion:** We are committed to fostering an inclusive work environment that ensures that every member, regardless of their background or identity, feels accepted and respected. We encourage team members to actively collaborate and work together to achieve team goals through open and transparent communication. Everyone's contribution is equally important in our team and we are committed to building a culture of mutual trust, support and collaboration.
- **Accessibility:** We are committed to providing equitable access to resources for all team members, ensuring that everyone can participate in team activities and work without barriers. Whether the need is physical or cognitive, we will provide the appropriate support and tools to ensure that each member can participate productively and to the fullest extent of his or her abilities.

B.3 Team Rules

- **Language requirements:** All team members must communicate in English to ensure that everyone on the team can fully understand and participate in discussions.
- **Working hours:** The team will work from Monday to Friday from 10:00 to 18:00. Members are expected to actively participate in teamwork during these hours to ensure successful completion of tasks.

- **Communication channels:** Communication during working hours requires the use of **Microsoft Teams**, and outside of working hours communication will be via **WhatsApp** for quick communication and informal matters. Important documents and formal information must be transmitted through **Microsoft teams** to ensure secure documentation and easy tracking.
- **Regular meetings:** A weekly team meeting will be held to review the week's progress, plan tasks for the following week, and assign specific responsibilities. Members are free to exchange ideas and offer comments or suggestions at the meeting.
- **Meeting absence notification:** If a group member is unable to attend a regular meeting due to an unforeseen event, the member must notify the group leader at least 24 hours prior to the start of the meeting, stating the reason. Exceptions are made for emergencies.
- **Task completion communication:** If a team member anticipates not being able to complete an assigned task on time, he/she is required to contact the Team Leader as early as possible to explain the reason. It is the responsibility of the Team Leader to adjust task assignments in a timely manner and assess the overall contribution of team members to ensure that the project schedule is not compromised.
- **Task refusal and consequences:** If a member refuses to complete an assigned task and still refuses to fulfill their duties after a conflict resolution process, the Team Leader has the right to reassign the task and adjust the member's contribution accordingly, notifying the team member and documenting this.
- **Feedback and collaboration:** All team members should actively provide feedback on their progress and problems encountered in the project to ensure information transparency and to promote efficient team collaboration.
- **Respect and support:** Team members should respect each other's opinions and ways of working, and actively provide support to create an open and collaborative team environment.

B.4 Daily Activity Conflict Avoidance

In order to ensure successful cooperation within the team and to avoid escalation of conflict into a formal complaint process, the team will take the following precautions in its daily activities:

1. **Maintain communication:** Team members should be actively involved in communication during working hours to discuss project progress, task assignments, and potential problems. All members should be proactive in expressing their views and insights to ensure that the team has a common understanding of each individual's work and that any possible misunderstandings or grievances are resolved in a timely manner.
2. **Clear division of labor:** The team will ensure that all tasks are assigned with clear responsibilities and deadlines to avoid work conflicts due to unclear responsibilities. The team leader will ensure that the division of labor is fair and that tasks are reasonably adjusted according to each member's ability and workload.

3. **Respect for differences of opinion:** Diversity of opinion will be promoted within the team, and all members will be expected to respect each other's ways of working and differences of opinion. Any technical or design decisions will be discussed to reach a consensus, and in case of disagreement, the team will resolve the issue based on a democratic voting mechanism.

4. **Rapid problem solving:** If a problem is identified in the course of daily work, team members are required to bring it to the table and discuss it in a timely manner. The team leader or designated mediator will intervene as soon as possible to help the team reach a consensus and avoid escalation of the conflict.

5. **Transparency and feedback:** A high degree of transparent communication will be maintained within the team, and each member will be required to update the progress of the task on a regular basis and seek feedback on any difficulties in the work. Through open communication and feedback mechanisms, the team will be able to quickly resolve problems at their earliest stages to avoid deterioration.

B.5 Conflict Resolution Flowchart

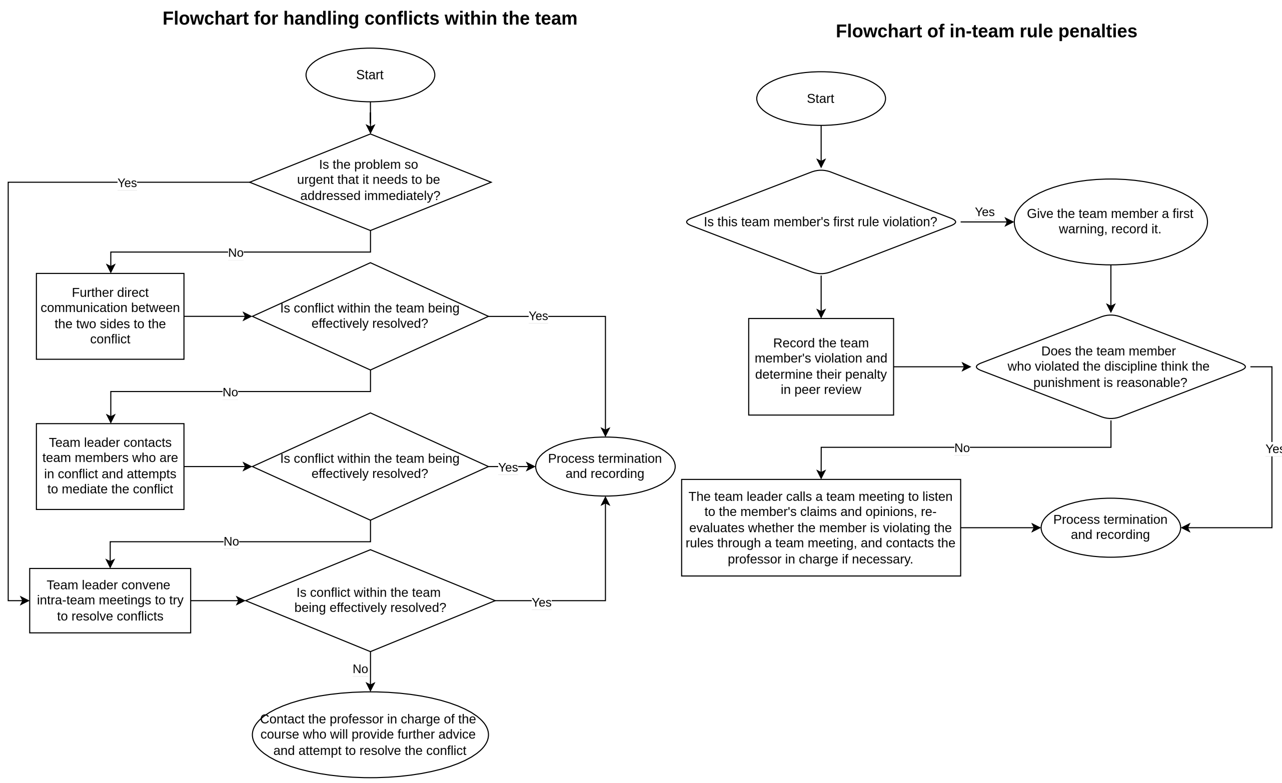

Fig. 19. Project RQT Graph