Team RoboManipal RoboCup@Work Team Description Paper 2025

Tejas Chinnappa Kodandera, Pratyush Agrawal, Dhruv Mehta, Mihir Kiran Kasibhotla, Triyan Mukherjee, Apoorv Khemani, Srikar Bharadwaj R, Neil Antony Dsouza, Parankusam Venkat Rohan, Tarun Murugan, Pranav Menon, Anirudh Swaminathan, Omer Mohiuddin Praanesh V A, T S Saumyaa, Aanyaa Agarwwal, Adwait Mahesh Pattar, Vikram Prabhu, Prahadeesh TN, Kunwar Abhuday Singh, Souri Rishik Volety, Adithya Sriraman, Soham Chandrashekhar Bhanap, Tanush Ralhan, Vedant Kamath, Arnav Sharma, Vedant Agarwal, Sannidhi D Math

Manipal Academy of Higher Education, Manipal, Karnataka, India
team.robomanipal@gmail.com
http://www.robomanipal.com

Abstract. This paper presents a solution to the challenges of RoboCup@Work, detailing the underlying software framework, developed capabilities, and hardware required for operation in industrial environments. The bot – 'KARMA' was designed and built entirely in-house, featuring custom control systems and an arm constructed from repurposed scrap metal. Material selection and manufacturing feasibility were prioritized, ensuring that all components could be fabricated using standard workshop techniques rather than industrial machinery, thereby reducing costs and simplifying production. Limited access to industrial motors and metals necessitated rigorous testing of power transmission methods to finalize an efficient solution. Structural integrity and weight optimization were critical considerations, requiring precise design of each link to balance strength, maneuverability, and long-term durability. From mechanical design to fabrication, every aspect underwent iterative refinement through persistence and innovation, incorporating lessons learned from extensive testing. The result is a fully functional, cost-effective robotic arm tailored to available resources, demonstrating the ability to overcome constraints through engineering ingenuity and adaptive problem-solving.

1 Introduction

1.1 About Team RoboManipal

Team RoboManipal is the official robotics team of Manipal Academy of Higher Education (MAHE), established in 2010 to pursue innovation and excellence in robotics. Comprising over thirty undergraduate engineering students from various disciplines, the team operates under the guidance of Dr. Ishwar Bhiradi, Assistant Professor (Senior Scale) in the Department of Mechatronics Engineering at MIT, Manipal.

Table 1: Competition Results Over the Years

Year	Location	Event/Competition	Result
2016	Delhi, India	ABU Robocon	9th
2016	Manipal, India	INK Makers at MIT, Manipal	2nd
2018	Ahmedabad, India	World Robotics Olympiad (WRO)	2nd
2021	Jharkhand, India	Takshak, IIT Dhanbad	1st
2021	Manipal, India	MIT Innovation Fest	2nd
2021	Erode, India	Disenyo PCB Challenge	1st
2022	Mumbai, India	Micromouse Challenge at IIT Bombay TechFest	3rd
2023	Delhi, India	RoboSoccer at Technoxian	3rd
2023	Goa, India	Line Follower at Goa Infofest	3rd
2023	Delhi, India	ABU Robocon	21st
2024	Delhi, India	ABU Robocon	22nd
2024	Manipal, India	Research Day at KMC Greens	2nd

2 Hardware Description

2.1 Power Electronics

The bot's power electronics and hardware are designed for efficiency and reliability, featuring a custom control board with an integrated power distribution system. This all-in-one board simplifies the management of power flow throughout the bot, reducing wiring complexity and enhancing performance. The system utilizes two 24V batteries, ensuring sufficient energy storage for prolonged operation. To meet the power requirements of various components, buck converters have been incorporated, which efficiently step down the voltage to the required levels. This setup ensures that the bot can operate smoothly under varying loads, providing a stable and robust power solution for all its functions.

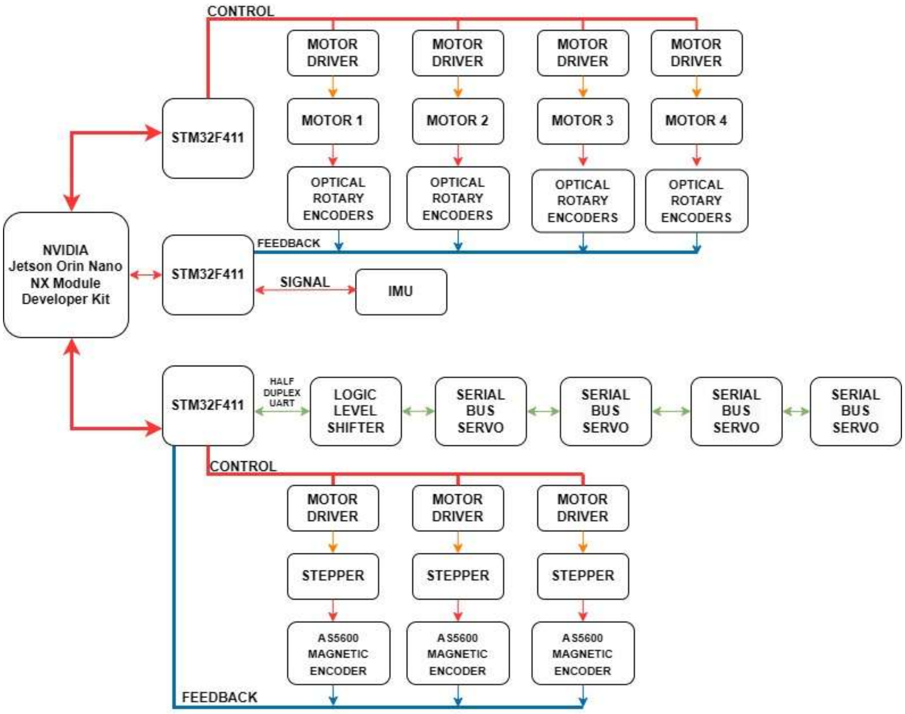
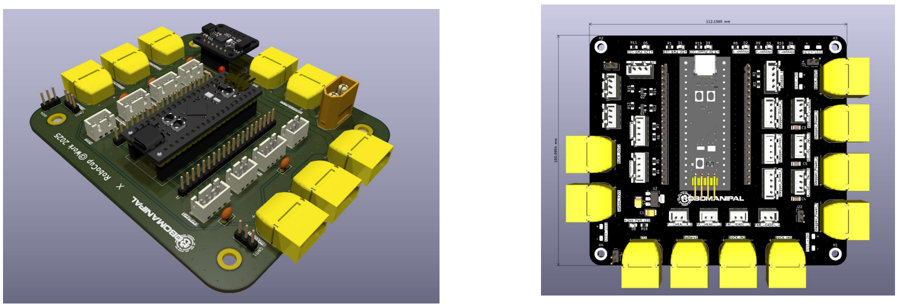


Fig. 1: This diagram shows the robotic system's electrical connections, featuring motors, sensors, MCUs, and power distribution, with key components like Cytron drivers and NVIDIA Jetson Orin. Color-coded lines indicate power and signal flow.



(a) (b)

Fig. 2: (a) Custom Control and Power Distribution Board for the Base. (b) Custom Control and Power Distribution Board for the Arm.

Table 2: Hardware Components and Pricing Breakdown (*The prices from INR to USD were calculated based on 5th February 2025 conversion rate.)

Components Used	Specification	Price in USD*
Nvidia Jetson Orin	Single Board Computer	915.97
STM32F411 MCU	Micro-controller unit	6.79
RpiLIDAR	2-Dimensional LIDAR	270.2
ZED 2i Depth Cam	Depth Camera	493.72
MDD30C Motor Driver	Single Channel Motor Driver	123.65
Rhino IG52	DC Motor	218.87
Waveshare ST3215/Dynamizel 18A	Serial bus servo	246.72
Orange Optical Encoders	Incremental Rotary Encoders	100.75
AS5600 Contactless Potentiometer	Absolute Rotary Encoders	3.41
TB67S109 Stepper Motor Driver	Stepper Motor Driver	13.87
Logic Level Shifter	Voltage shifter	0.14
LiPo Batteries	Power Source	274.79
	Total Cost	**2668.8**

2.2 Hardware Design

2.2.1 In-House Closed-Loop Stepper Motor Control

An in-house closed-loop stepper motor control system was developed with PID regulation and magnetic feedback for precise motion and dynamic torque control. Running on an STM32F4, it ensures real-time position correction, reducing step losses and enhancing efficiency for high-precision robotics applications. It controls the first two Degrees of Freedom.

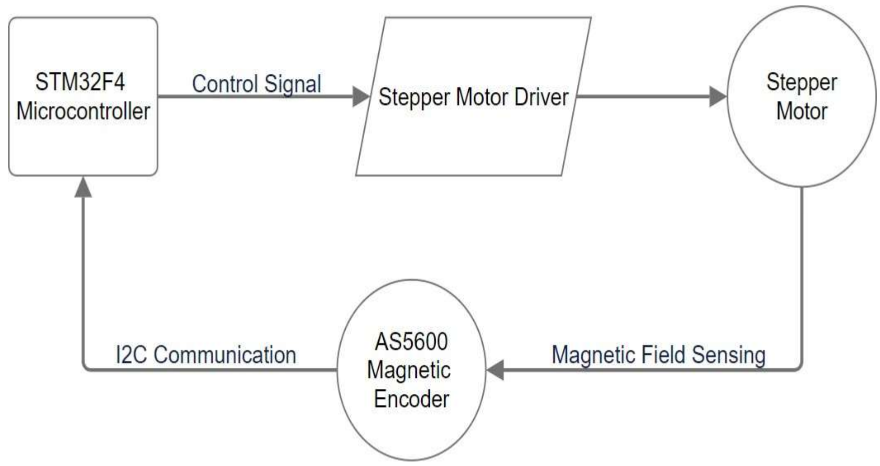


Fig. 3: Closed-loop feedback utilizing a magnetic encoder (wireless potentiometer).

2.2.2 Serial bus servo motor control

Dynamixel's serial bus servo was used to control 4 DOF of the 6 DOF robotic arm. The servo motors are powered and managed by the STM32F4 microcontroller, which is specifically configured for this application. An in-house custom firmware was developed to facilitate communication between the microcontroller and the servo motors, operating UART in half-duplex mode. This library is equipped with error detection features, ensuring reliable and accurate data transfer over a single communication line. This setup provides precise and dependable control over the robotic arm's movements.

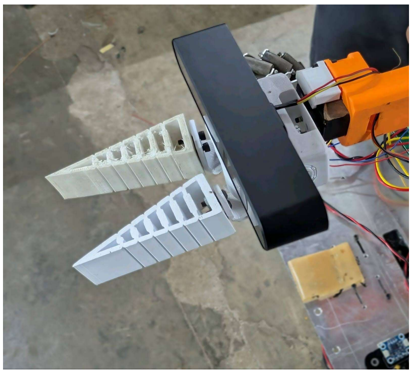


Fig. 4: Gripper Testing with Dynamixel.

2.3 Safety Features

2.3.1. Battery Protection Circuit (BPC):

A custom low- voltage cutoff circuit was developed, designed to protect Li-Po batteries from over-discharge. This circuit continuously monitors the battery voltage and automatically disconnects the power supply if the voltage falls below a predefined safe threshold.

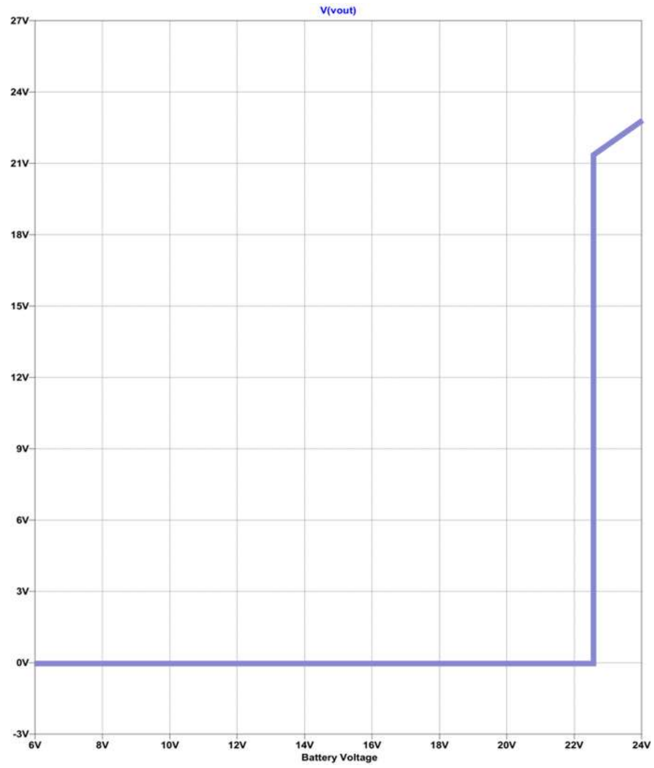


Fig. 5: (a) BPC simulation graph on Proteus.

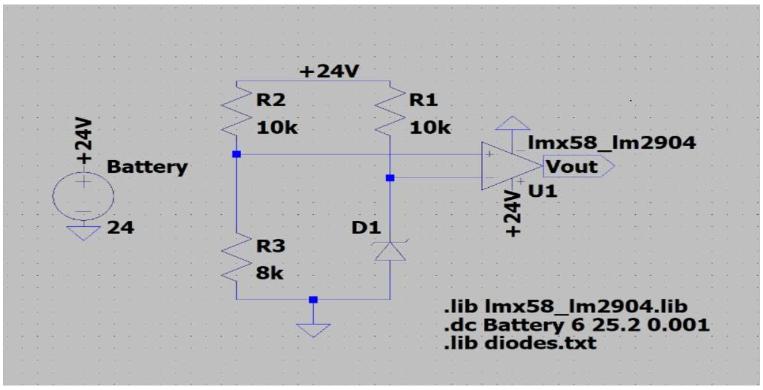


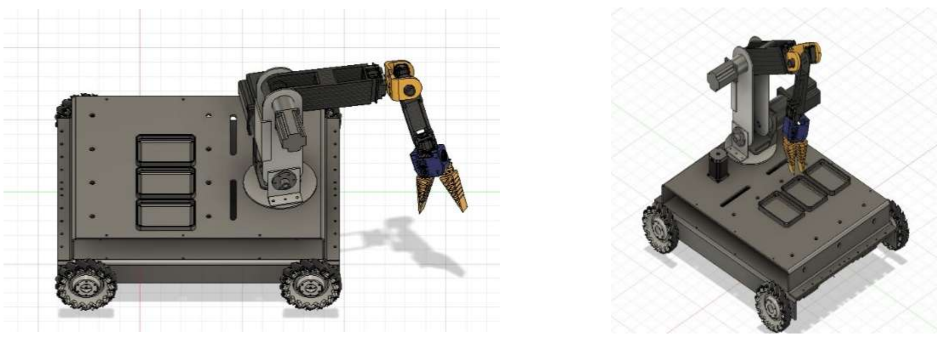
Fig. 5: (b) BPC on LTSpice (Simulation Software).

2.3.2. Serial Bus Servo:

Serial bus servos, like the Dynamixel servos, are equipped with multiple safety features to ensure reliable and secure operation. Running on in-house custom firmware that operates UART in half-duplex mode, these servos offer enhanced control and communication efficiency. Overload protection prevents damage by halting operation when excessive load or torque is detected, while overheating protection actively monitors internal temperatures and stops functionality if safe limits are exceeded. They also provide motion range limitations, allowing users to define safe angle boundaries to prevent mechanical damage. Additionally, overcurrent protection limits abnormal current surges, and voltage protection safeguards against undervoltage or overvoltage conditions, ensuring stable operation. Position error detection further ensures accurate responses by identifying discrepancies between the desired and actual positions.

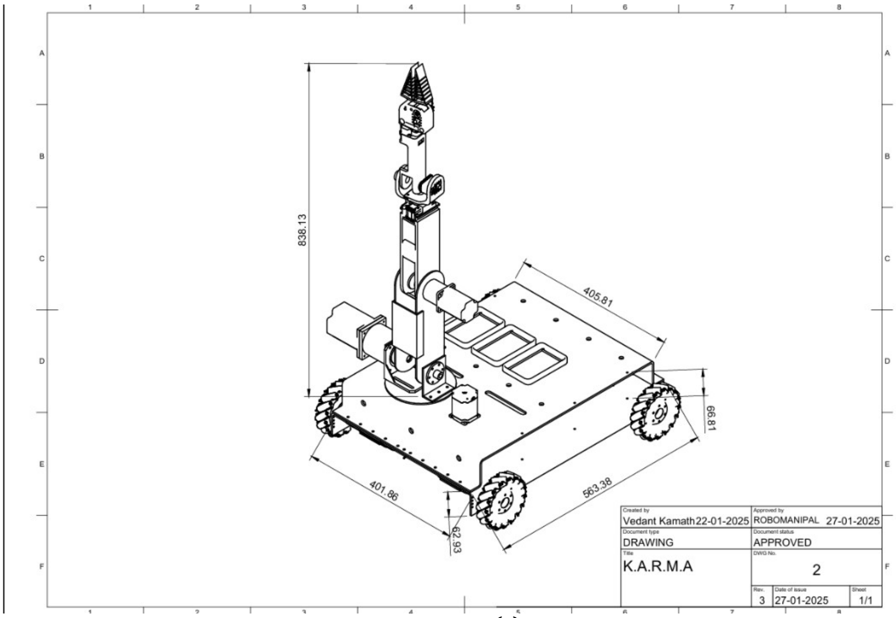
2.4 Structural Design

The arm was designed using Fusion 360, which provides optimized performance and dynamic modeling techniques essential for understanding its movement. This is crucial for Robot Operating System(ROS) implementations, as it ensures smooth integration. Fusion 360 also enables motion simulation, allowing users to verify performance and prevent collisions between parts.



(a) (b)

Fig. 6: (a) KARMA in object picking position. (b) KARMA in object drop and storage position.



(c)

Fig. 6: (c) Isometric drawing of the arm with scale.

The robot is built on a robust four-wheel Mecanum drive platform, enabling omnidirectional movement and precise maneuverability. The primary construction material for the robot is aluminum 6061(Alloy 61S), chosen for its lightweight properties, strength, and corrosion resistance. No gearboxes have been used with the RMCS-2084 motors driving the wheel, since enough torque was achieved by the selected motors.

For the first degree of freedom (DOF) of the arm, the load of the entire arm was initially exerted directly on the motor shaft, leading to potential reliability and performance issues. To address this, a timing pulley system supported by bearings was integrated. This setup effectively transferred the load away from the motor shaft, ensuring enhanced durability and smoother operation. Additionally, a jockey was employed to enhance tension on the belt, which helped prevent slippage during high torque requirements.

Initially, the arm links were designed with 3mm aluminum, but it was later changed to polylactic acid (also known as PLA) due to its lightweight nature and exceptional dimensional stability. A weight reduction of 85% was achieved by switching from aluminum links to 3D-printed PLA links. Initially, the four aluminum links weighed 4.88 kg, whereas the new PLA links weighed approximately 635 grams. This material also reduced the current draw from the servo motors, helping to prevent them from overheating quickly during operation.

2.4.1. Gripper Design

The gripper's design is intentionally simple and straightforward. It is primarily constructed from polylactic acid (PLA), while the fingers of the gripper are made from thermoplastic polyurethane (TPU). TPU is chosen for its excellent flexibility and grip, making it highly suitable for soft robotic applications. This material allows the gripper to effectively handle a wide variety of objects during pick and place operations. Additionally, the design features finger mounts that resemble a spur gear, enabling simultaneous actuation of the fingers using a single servo motor. This mechanism enhances efficiency and functionality, making the gripper versatile and effective for various applications.

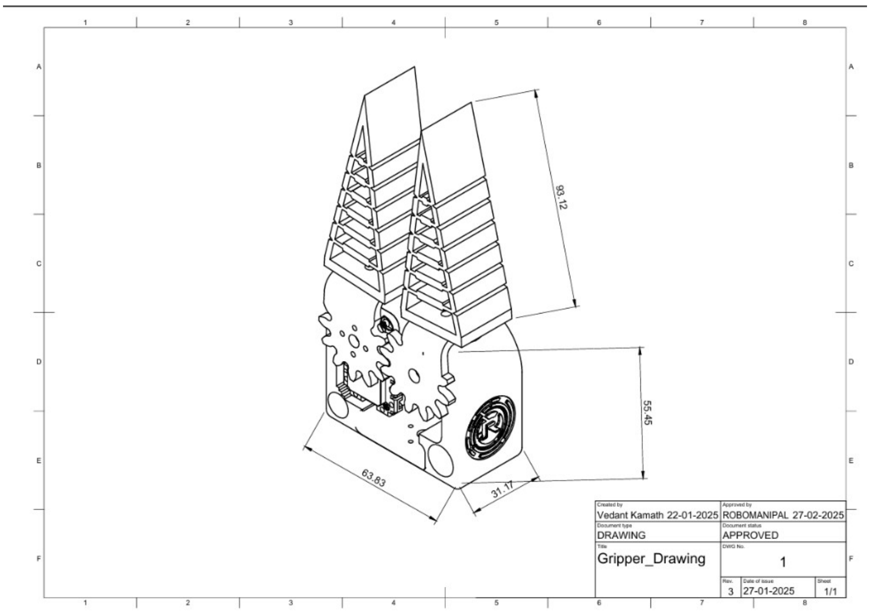
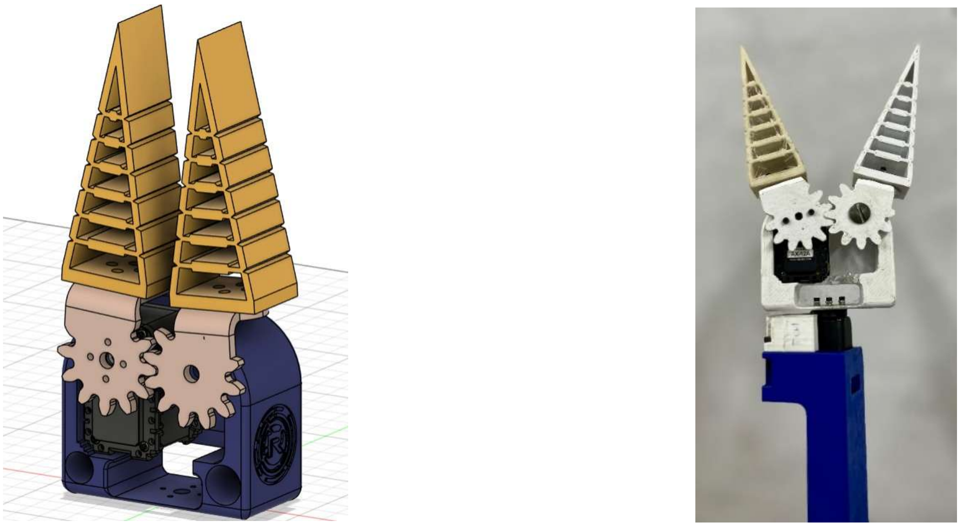


Fig. 7: Isometric Gripper Design.



(a) (b)

Fig. 8: (a) CAD of Gripper Design. (b) Gripper Design

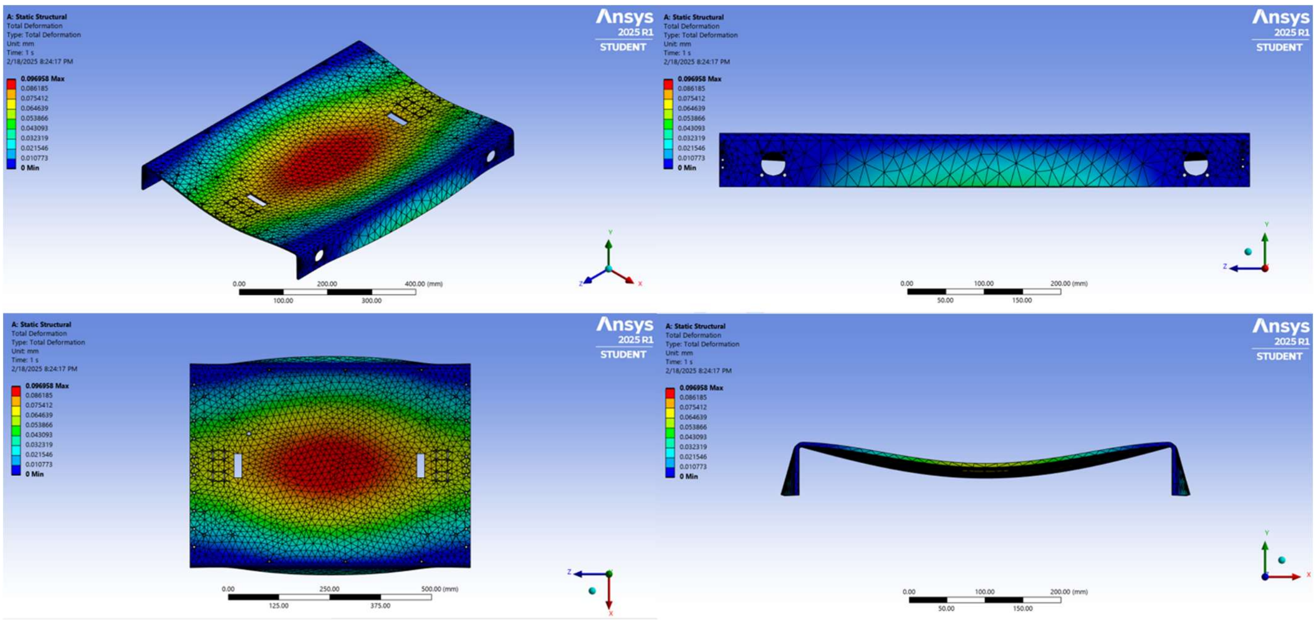


Fig. 9: FEA Analysis of the base.

Model details	Base (5mm Aluminium sheet)
Loadings	Point load of 11kg at the mounting point of the arm
Type of meshing	Tetrahedral mesh
Mesh Details	5mm edge sizing

3 Software Description

3.1 Pose

3.1.1 YOLOv8

The robot employs a ZED RGB-D camera, strategically mounted near the end-effector of its robotic arm, for object detection and recognition tasks. The camera's captured images are processed using YOLOv8, a state-of-the-art neural network optimized for real-time object detection. This model has been trained on the RoboCup Objects 2024 dataset, enabling the identification of objects across 22 distinct classes and providing precise coordinates of detected objects in real-time, as depicted in Figure 10. To enhance inference speed and simplify integration with C++ frameworks, the trained YOLOv8 model is converted into the ONNX (Open Neural Network Exchange) format. This conversion ensures compatibility with highly efficient inference engines, significantly boosting the real-time performance of the robot. Additionally, the streamlined training process, which takes approximately two hours on a GPU, allows for rapid re-training to adapt to new objects at competitions, ensuring the robot's detection capabilities remain up-to-date and effective.

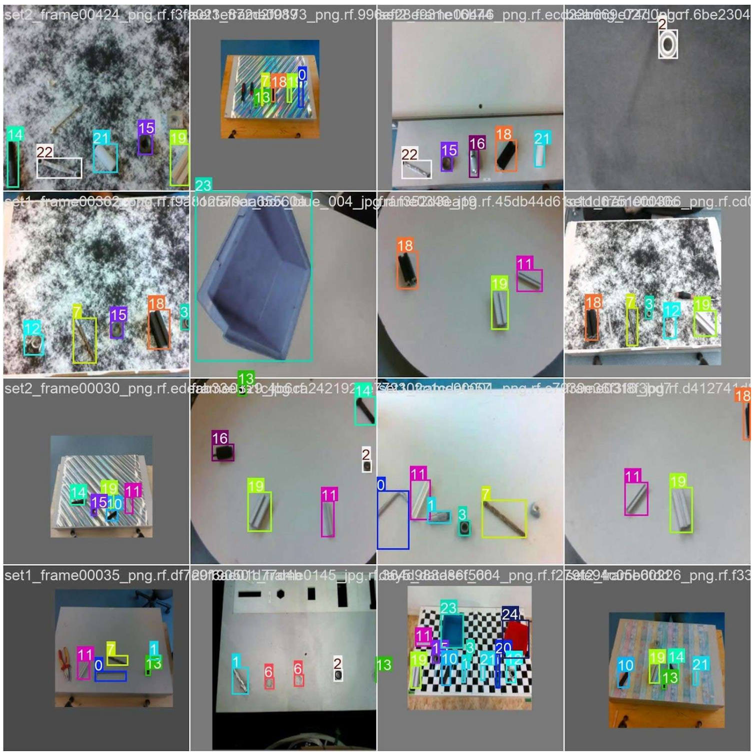


Fig. 10: Identification of various RCup@Work objects using bounding boxes, with each class assigned a unique ID.

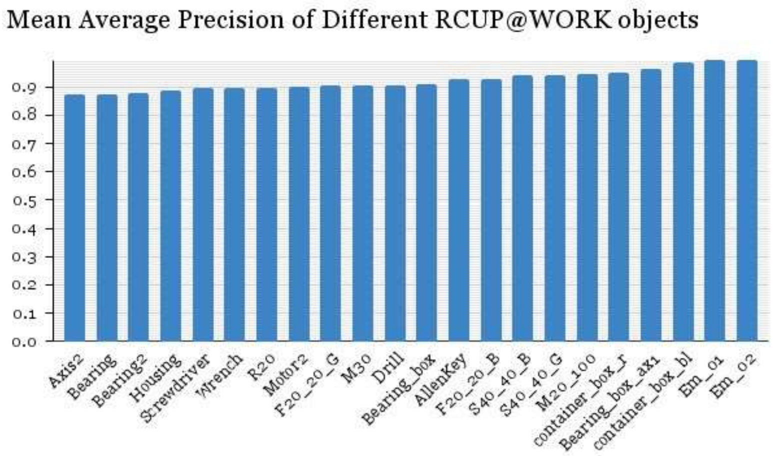


Fig. 11: Mean average precision of different RCUP @Work objects.

3.1.2 Barrier-Tape Detection

The vision system detects and highlights specific objects in video frames using OpenCV. It processes each frame by converting it to the HSV colour space and creating binary masks for regions of interest based on predefined thresholds. The masks are refined with morphological operations, including opening to remove noise and closing to fill gaps. Contours are then extracted and combined if they are close to each other, using bounding boxes to merge nearby regions. Final bounding boxes are filtered based on their aspect ratio and size to eliminate non-relevant shapes. Detected regions are labelled and displayed on the video feed in real-time, showcasing the system's ability to analyze and classify objects dynamically.

3.1.3 Empty Space Detection

To place an object, the robot identifies empty areas in the workstation by processing 3D point cloud data from the RGB-D camera. Edge detection and Hough Transform techniques are used to detect horizontal planes. The data is converted to grayscale, edges are detected, and lines marking boundaries of potential free spaces are identified. Contour approximation refines these spaces into rectangles to evaluate their suitability.

The system then evaluates neighbouring areas around candidate points, ensuring sufficient free space. This process is repeated until at least two viable placement locations are identified. This is accomplished by a custom script that utilises OpenCV and Open3D libraries.

3.2 Movement

3.2.1 Simultaneous Localization and Mapping (SLAM)

The **RPLiDAR A2M8** is utilized to gather approximately **180- degree** laser scan data of the robot's surroundings. This data facilitates **obstacle detection** and **environment mapping**. The scan data is processed by the **slam_toolbox**, which generates a real-time map of the environment, a critical component for autonomous navigation.

Simultaneously, the **Nav2 framework** is employed for **path planning** and **motion control**, leveraging the map generated by SLAM. To ensure precise localization within the map, the **Adaptive Monte Carlo Localization (AMCL)** algorithm is integrated into the system. Additionally, an **Extended Kalman Filter (EKF)** fuses data from multiple sensors, including wheel odometry and IMU measurements. This fusion reduces the effects of noise and drift in individual sensors.

For visualization and monitoring, **RViz** provides a graphical interface displaying the robot's position, map updates, and navigation progress. To ensure seamless integration of sensor data and navigation logic, static and dynamic transforms maintain proper alignment and communication between various frames, such as the base link, laser, and odometry.

This comprehensive setup enables the robot to **autonomously navigate** unknown environments, effectively **avoiding obstacles** and following planned paths with **high precision**.

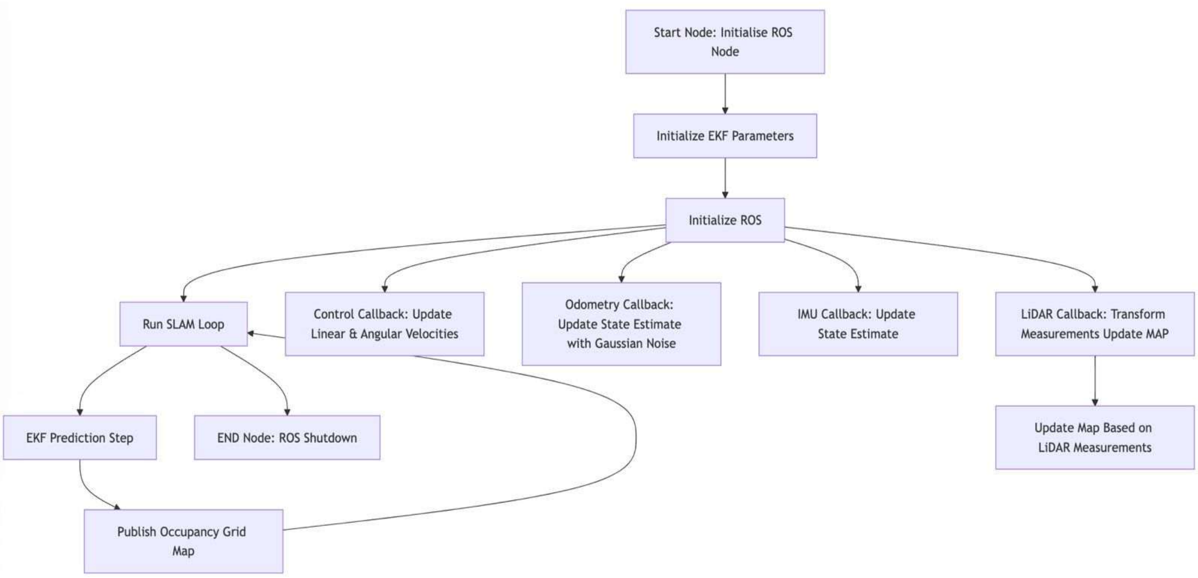


Fig. 12: Flowchart representation of EKF used to understand SLAM.

Simulated navigation using SLAM and Nav2:
https://drive.google.com/file/d/1keDHFtPm7nLRP8Xg8YAC8_NZv90Eh5P9/view?usp=drive_link

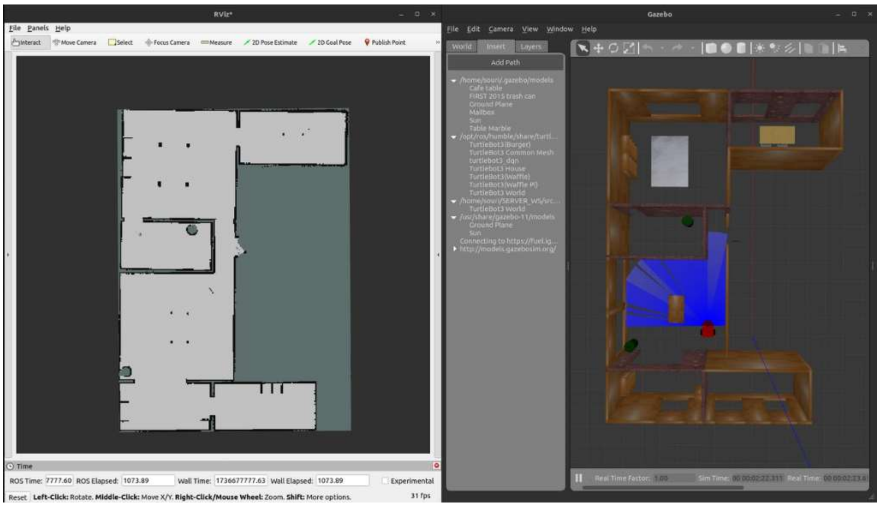


Fig. 13: The worlds were completely mapped using the slam_toolbox package.

3.2.2 IMU and Odometry Integration

Initially, the MPU6050 IMU with a custom library was used for its DMP functionality on STM32, but accuracy and drift issues led to a switch to the Bosch BNO055. The BNO055's onboard sensor fusion significantly improved orientation stability and reduced drift.

Integrated with four optical encoders, the IMU provided precise odometry data, processed in real-time on the STM32. This data was sent via USB to the Nvidia Jetson for SLAM and navigation, enabling accurate positioning and path planning in dynamic environments.

3.2.3 Navigation with AMCL

The **Nav2 stack** facilitates autonomous navigation by allowing the robot to navigate to a specified Nav goal pose within the environment. The process begins with the user defining a goal position and orientation on the map, typically through a graphical interface like **RViz**. The Nav2 stack leverages the map generated by the SLAM Toolbox which is preloaded as a static map for global planning.

The Global Planner computes an optimal path to the goal pose, considering static obstacles, while the Local Planner dynamically adjusts the path to account for moving obstacles and environmental changes. It also ensures smooth and collision-free motion by generating velocity commands for the robot. This robust navigation framework enables the robot to reach the designated 2D goal autonomously pose efficiently and reliably, even in dynamic and complex environments. The figure below (Fig14) shows autonomous navigation using Nav2 based on initial pose given to AMCL.

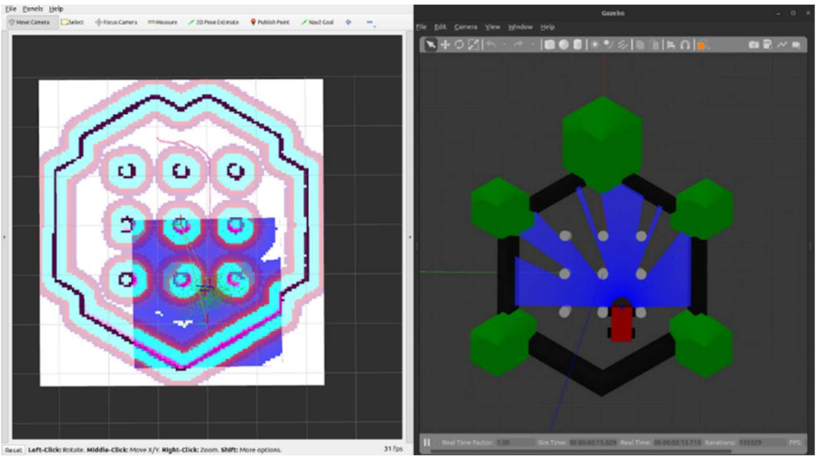


Fig. 14: Movement of robot based on the 2D goal pose provided by Nav2 stack.

The rqt_graph (Fig.15 and Fig.16) provides a high-level visualization of the ROS2-based navigation stack, showcasing the interaction between nodes, topics, and action servers. At its core, the system integrates sensor data, localisation, mapping, path planning, and motion control to enable autonomous navigation. Nodes such as **//slam_toolbox, /amcl,** and **/ekf_filter_node** play crucial roles in processing laser scans, odometry, and IMU data to create accurate maps and state estimates. These outputs are essential for real-time localization and obstacle detection, forming the foundation for efficient navigation.

The system employs a dual-layer costmap strategy, with **/global_costmap** handling long-term path planning and **/local_costmap** focusing on real-time obstacle avoidance. The **/controller_server** node coordinates motion control, while nodes like **/follow_path** ensure precise execution of planned paths. Behavior tree-based nodes, such as **/bt_navigator_navigate_to_pose_rclcpp_node**, oversee the navigation process and handle recovery actions like spinning or backing up when obstacles disrupt the robot's path. This modular approach ensures flexibility and reliability in dynamic environments.

The architecture is designed for seamless integration, with components like **/map_server** and **/lifecycle_manager_localization** ensuring proper initialization and data flow. Visualization tools like RViz allow real-time monitoring of the robot's position, map updates, and navigation status, enhancing system usability and debugging capabilities. The use of sensor fusion through the **/ekf_filter_node** minimizes noise and drift, ensuring accurate localization and smooth navigation. Overall, this architecture balances efficiency, scalability, and robustness, enabling autonomous robots to operate effectively in complex and dynamic environments.

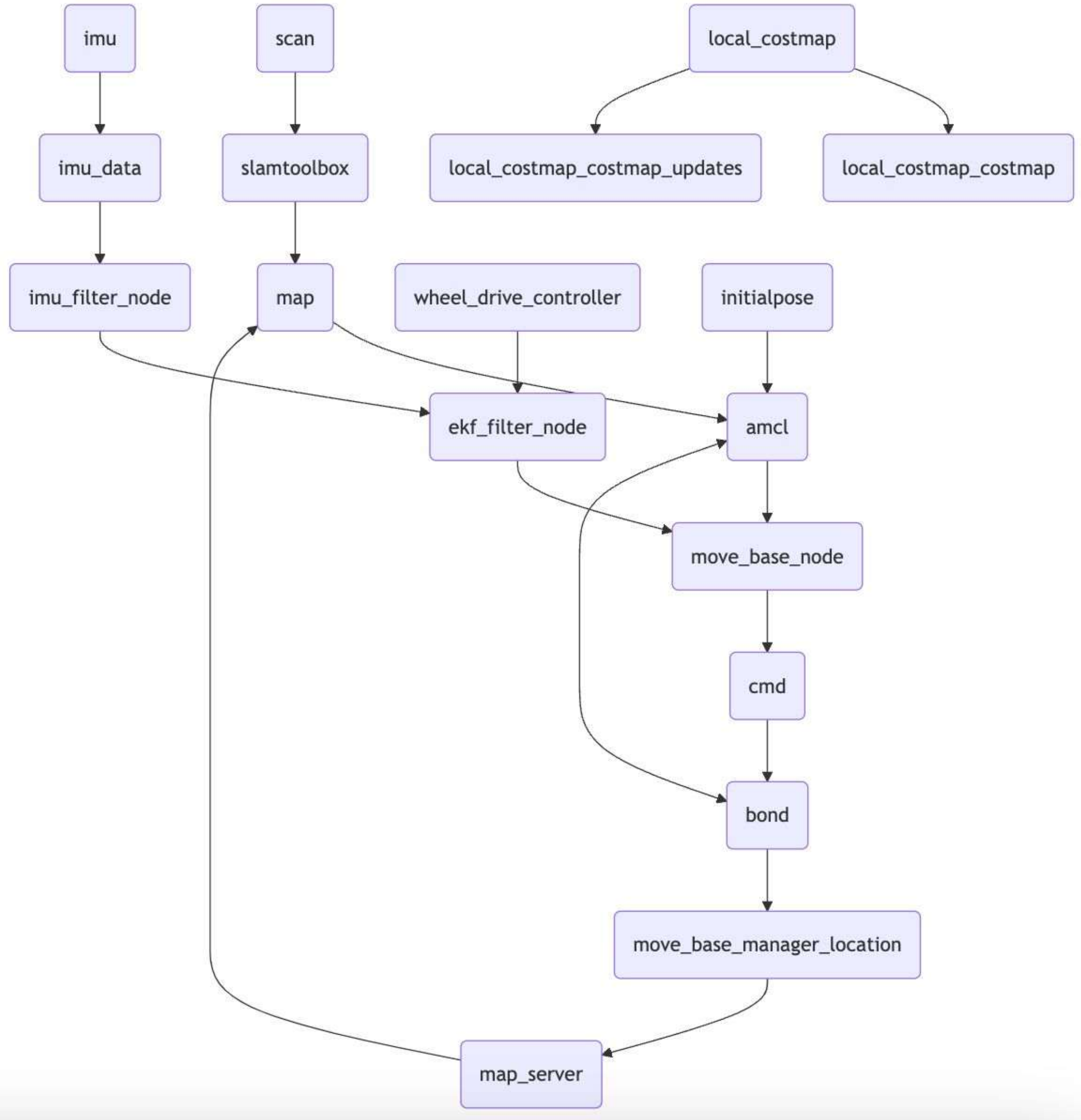


Fig. 15: RQT_graph Part – 1 of the overall robot.

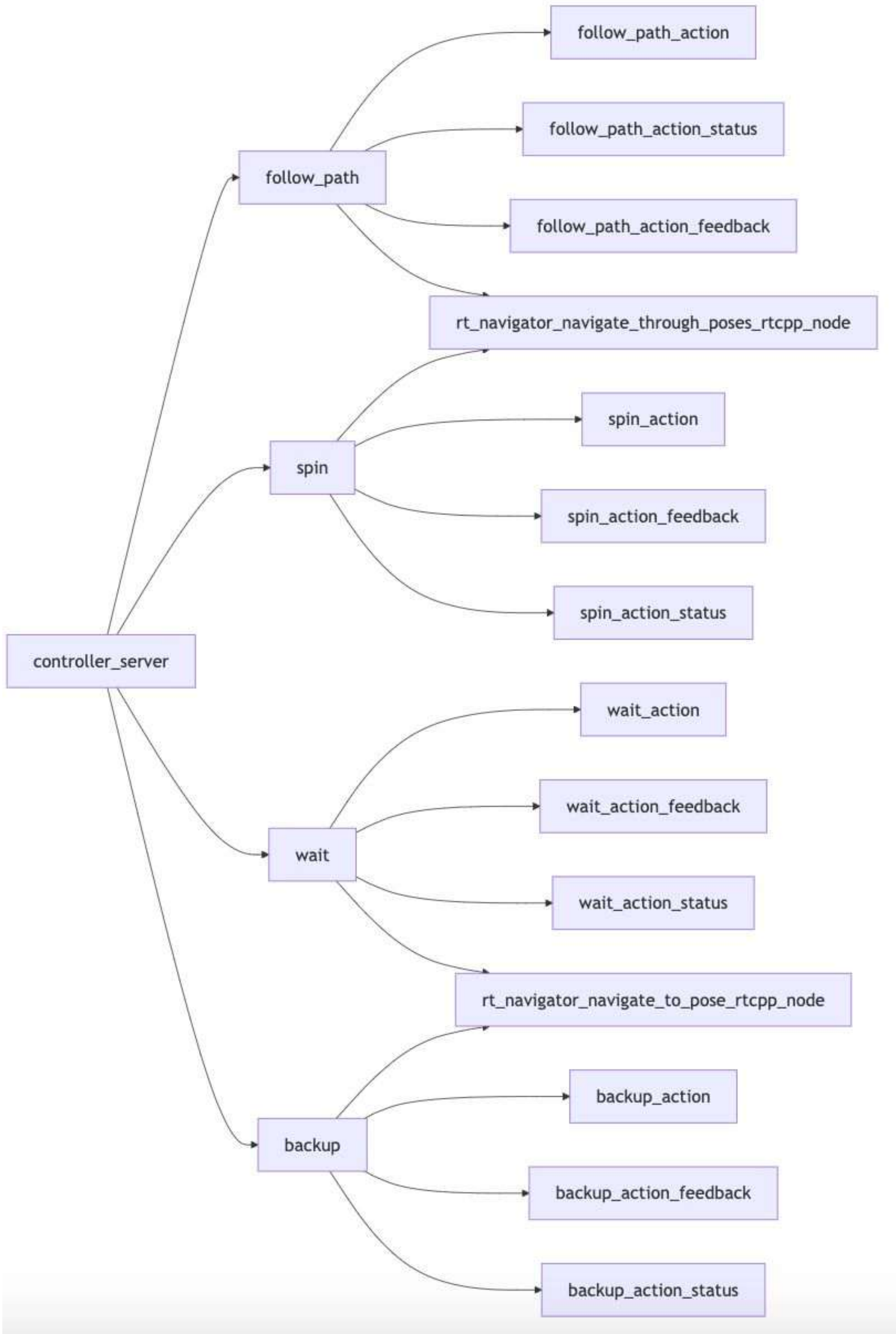


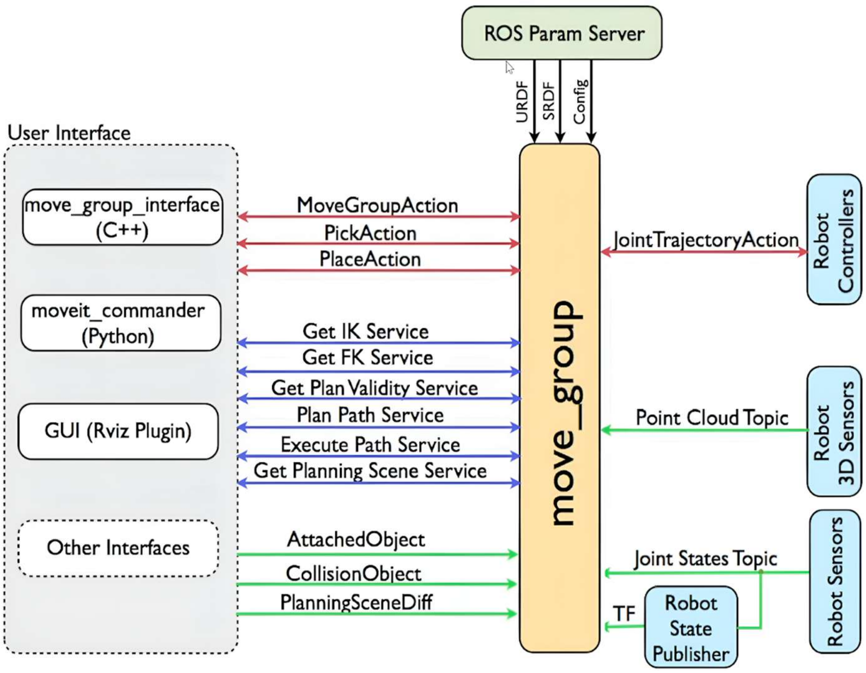
Fig. 16: RQT_graph Part - 2 of the overall robot.

3.3 Arm Control

3.3.1 MoveIt2:

An advanced open-source motion planning framework initially developed by Willow Garage, has been chosen for implementing arm kinematics. This robust framework integrates key functionalities such as **motion planning**, **kinematics**, **collision detection**, and **dynamic 3D environment representation**, enabling precise and efficient robotic arm control.

Tailored for **ROS2**, MoveIt2 ensures seamless compatibility with existing ROS2 packages and tools, including **RViz**, which enhances visualization and interaction capabilities. Its modular design and extensive features make it a powerful solution for complex robotic applications, providing reliability and ease of use in dynamic environments.



(a)

(b)

Fig. 17: (a) Workflow of MoveIt2. (b) RVIZ visualization of robotic arm using MoveIt2.

The **StereoLabs Zed-2i depth camera** serves as the vision system, delivering **point cloud data**, **6-DOF pose information**, and **plane segmentation outputs**. This rich dataset is seamlessly integrated into the **inverse kinematics module** within **MoveIt2**, enabling the robotic arm to perform precise **pick-and-place operations**. The integration allows the system to identify and interact with objects in real-world environments with high accuracy. Furthermore, MoveIt2's compatibility with **RViz** is leveraged to simulate and visualize the arm's movements in real-time, ensuring accurate planning and reliable execution of tasks.

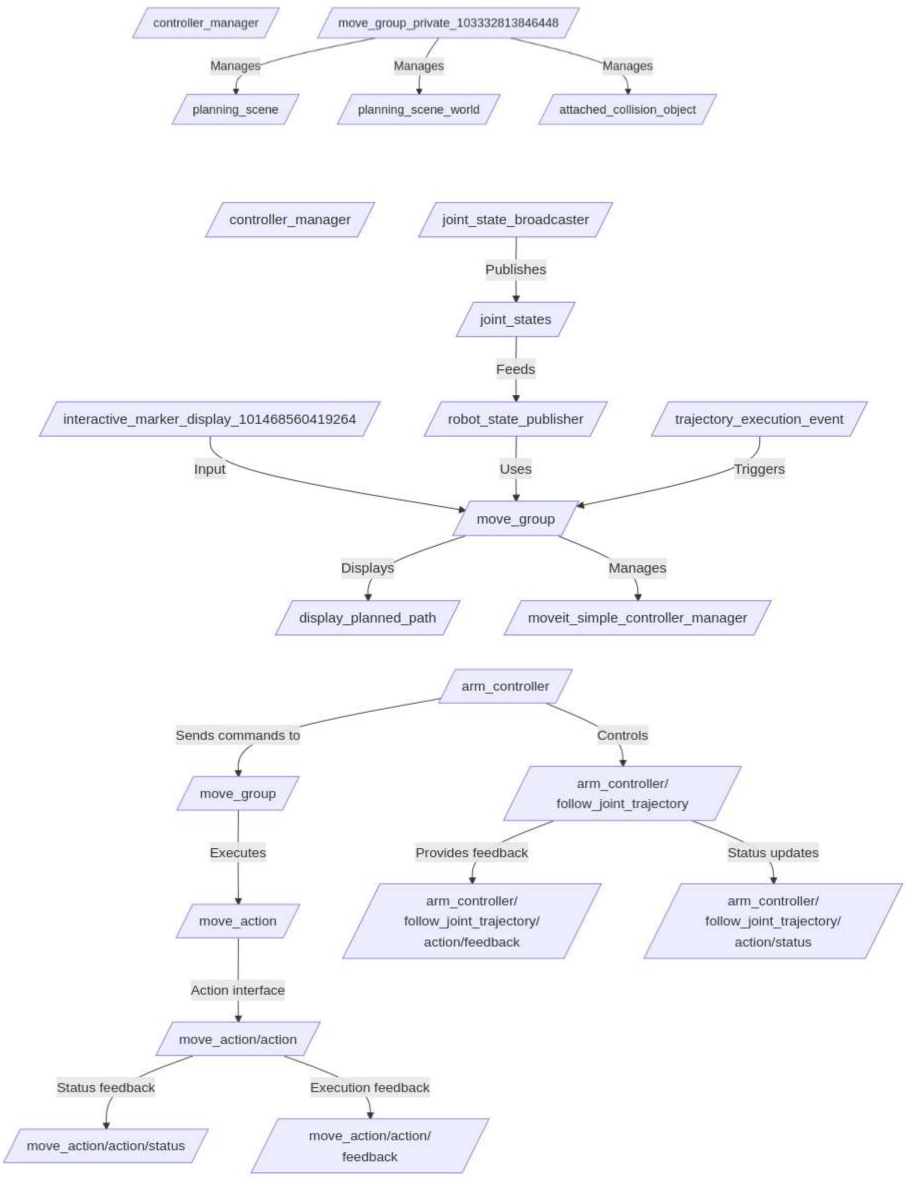


Fig. 18: Node architecture for Arm Simulation using MoveIt2

Simulation of arm_xyz in Rviz using MoveIt2 - Simulation of KARMA arm in Rviz using MoveIt2 : https://drive.google.com/file/d/1-RzNobyaHWOWv80olhhknhlq2UzB9EiP/view?usp=drive_link

4 Task Planner

Our RoboCup robot features a distributed computing architecture utilizing two Raspberry Pi 5s, a Jetson, and multiple STM32 microcontrollers to efficiently handle SLAM-based navigation and object manipulation.

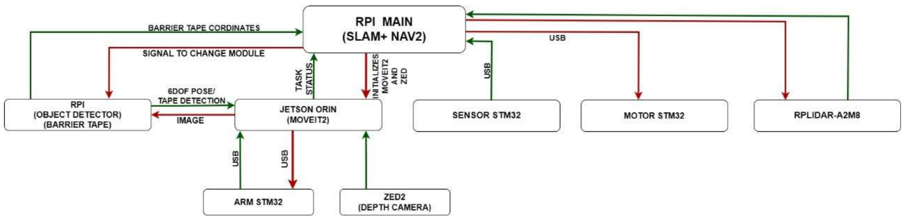


Fig. 19: Task Planner for proper completion of given tests.

4.1 Primary Raspberry Pi (Main Controller)

- Runs **SLAM and the Nav2 stack** to handle navigation.
- Directly connected to the **LiDAR**, which provides real-time mapping and localization data.
- Interfaces with two STM32 microcontrollers:
  - **Motor Control STM**: Processes commands from the RPi to drive the motors.
  - **Sensor Fusion STM**: Collects data from the **IMU and wheel encoders** and sends it back for SLAM processing.

4.2 Jetson (Motion Planning, Coordination & Vision Processing Unit)

- Runs **MoveIt2** for robotic arm control.
- Runs an **ONNX-based model** for object detection and tape detection.
- Simultaneously executes the tape detection model to assist in navigation.

4.3 Task Execution Flow

- The main RPi navigates the robot using **SLAM and Nav2**.
- Once the goal is reached, it signals:
  - The Jetson to **start MoveIt2 operations** for object manipulation.
  - The secondary RPi is to **switch from tape detection to object detection** for precise object handling.
- This ensures the **separation of navigation and object-handling tasks**, preventing interference.

4.4 Arm Control

A dedicated **STM32 microcontroller** drives the robotic arm's motors, executing movement commands based on values computed by MoveIt2 on the Jetson.

This architecture ensures smooth coordination between navigation and object handling while optimizing computational efficiency across different hardware components.

5 Future Scope

The capabilities of the robotic system are planned to be extended by leveraging the MoveIt Task Constructor to implement advanced manipulation functionalities such as pulling, pushing, and twisting. This will enhance the system's versatility in handling diverse tasks within the RCUP@WORK environment.

Looking ahead, a custom tech stack is planned to be developed, tailored to specific requirements and integrating a fine-tuned GraspNet implementation optimized for RCUP@WORK objects. This will enable more precise and efficient grasp planning, improving object handling in real-world scenarios.

For enhanced perception and navigation, 3D LiDAR and a depth camera are planned to be integrated with RTAB-Map for 3D SLAM, creating a dense, feature-rich representation of the environment while ensuring loop closure detection to minimize long-term drift. To enable real-time obstacle avoidance and smooth navigation, a **local planner** is essential. While a **global planner** like **SmacPlanner** can generate high-level paths using the 3D map, a **local planner** such as **TEB (Timed Elastic Band)** or **MPC (Model Predictive Control)** can refine these paths dynamically. The depth camera will enhance obstacle detection beyond LiDAR's range, improving safety in cluttered spaces. Additionally, for **3D path planning**, frameworks like **OctoMap, Voxblox, or FIESTA (Fast Incremental Euclidean Distance Fields for Online Motion Planning)** will allow the robot to navigate complex terrains, detect overhanging obstacles, and traverse multi-level environments efficiently. These upgrades will be particularly beneficial for **warehouse automation, search-and-rescue missions, and autonomous navigation in unstructured terrains**, where precise 3D perception and adaptive path planning are crucial.

For simulation and refinement, Isaac Sim is planned to be incorporated, enabling realistic physics-based testing and validation of robotic models. Additionally, DenseFusion will be integrated to combine RGB and depth data, enhancing object pose estimation for more accurate and reliable object interactions.

In the mechanical domain, the future of our robotic arm lies in transitioning from aluminum and PLA to composites, which will enhance strength while reducing weight. This weight reduction will allow us to minimize bulky power transmission methods like timing pulleys and gears, improving efficiency. A lighter structure will also make the arm more compact and stable, enhancing precision and responsiveness. Additionally, lower weight will reduce energy consumption and increase payload capacity, making the arm more efficient for

various applications. With these improvements, the robotic arm will be stronger, more agile, and better suited for industrial and research use.

Advancements in electronics are bringing exciting improvements to boost performance and functionality. One key area is better sensor fusion, which will help systems process data from multiple sensors more accurately and reliably. Adding features like Electrostatic Discharge (ESD) protection will also make devices more durable and resistant to environmental damage. The shift to BLDC motors for actuation will allow for smoother motion, better control, and greater efficiency, extending the lifespan of mechanical components. Additionally, moving to a swerve drive system will make movement more precise and agile, especially for applications that need complex maneuvers. Altogether, these upgrades will take electronic systems to the next level.

6 Conclusion

Ongoing and future developments focus on creating a highly efficient, adaptive, and intelligent robotic system capable of tackling complex manipulation and navigation tasks. By integrating advanced perception, motion planning, and control strategies, the system will be better equipped for real-world applications such as warehouse automation, search-and-rescue operations, and autonomous navigation in dynamic environments. Additionally, improvements in hardware and simulation tools will enhance performance and reliability, ensuring seamless integration into practical use cases. These continuous innovations aim to exceed the demands of the RCUP@WORK environment, setting new standards for precision, efficiency, and adaptability in robotics.

7 References

1. S. Chitta et al., "MoveIt 2.0: Modern Task and Motion Planning for Robotic Manipulators," *IEEE Robotics & Automation Magazine*, vol. 29, no. 1, pp. 20-29, 2022.

2. Ultralytics, "YOLOv8: State-of-the-Art YOLO Object Detection," *GitHub repository*, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics. [Accessed: Nov. 11, 2024].

3. T. Moore and D. Stouch, "A Generalized Extended Kalman Filter Implementation for the Robot Operating System," in *Proc. 13th Int. Conf. Intelligent Autonomous Systems (IAS-13)*, 2014.

4. S. Macenski et al., "Nav2: A New Generation of Navigation Servers," *arXiv*, arXiv:2303.08663, 2023.

5. S. Macenski, F. Martín, R. White, and J. G. Clavero, "The Marathon 2: A Navigation System," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2020.

6. ROBOTIS, "Dynamixel SDK," *GitHub repository*, 2023. [Online]. Available: https://github.com/ROBOTIS-GIT/DynamixelSDK. [Accessed: Dec. 15, 2024].

7. Bosch Sensortec, "BNO055 Intelligent 9-axis absolute orientation sensor," *BNO055 Datasheet*, Nov. 2020.

8. STMicroelectronics, "STM32F4 Series Reference Manual," *RM0090 Reference Manual*, 2021.

9. NVIDIA, "Jetson AGX Orin: Technical Brief," *NVIDIA Technical Documentation*, 2023.

10. StereoLabs, "ZED SDK Programming Guide," *Documentation*, v3.8, 2023.

11. M. Bazavan, "Mastering Robotics Software Engineering with ROS: Develop and Deploy Advanced Manufacturing Solutions," Learn Robotics with ROS School. [Online]. Available: https://learn-robotics-with-ros-s-school.teachable.com/p/mastering-robotics-software-engineering-with-ros-develop-and-deploy-advanced-manufacturing-ssolution. [Accessed: Dec. 13, 2024].

12. "RoboCup Objects 2024 Dataset," Roboflow Universe. [Online]. Available:https://universe.roboflow.com/dirwg01w/robocup_objects_2024/dataset/1. [Accessed: Nov. 27, 2024].

13. "BNO055 STM32 Library," GitHub - Ivyknob. [Online]. Available: https://github.com/ivyknob/bno055_stm32. [Accessed: Dec. 13, 2024].

14. "AX-12A Actuator Manual," ROBOTIS e-Manual. [Online]. Available: https://emanual.robotis.com/docs/en/dxl/ax/ax-12a/.[Accessed: Feb. 1, 2025].

15. Slamtec, "sllidar_ros2," GitHub repository, 2024. Available: https://github.com/Slamtec/sllidar_ros2 .[Accessed: Dec. 1, 2024]
16. "AX-18A Actuator Manual," ROBOTIS e-Manual. [Online]. Available: https://emanual.robotis.com/docs/en/dxl/ax/ax-18a.[Accessed: Feb. 1, 2025].
17. B. J. Hardy, "Small Satellite Trends and Future Directions," *AIAA/USU Conference on Small Satellites*, Aug. 1993. [Online]. Available: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=5160&context=smallsat. [Accessed: Feb. 6, 2025].