

# Team RoboManipal

## RoboCup@Work

### Team Description Paper 2026

Praanesh V A, T S Saumyaa, Aanyaa Agarwal, Adwait Mahesh Pattar, Vikram Prabhu, Prahadish TN, Kunwar Abhuday Singh, Sourish Rishik Volety, Adithya Sriraman, Soham Chandrashekhar Bhanap, Tanush Ralhan, Vedant Kamath, Arnav Sharma, Vedant Agarwal, Sannidhi D Math, Nirja Parhi, Siri Sahiti Kedas, Prasad Shanmukh Panji, Dhruv Dabur, Veena Viswanathan, Manansh Pandey, Adithya Linga, Shubham Khandelwal, Kamalkant Thangaraju, Sanjay S, Aditya Zambare, Aditya Kumar Chaurasia, Yash Chaudhary, Jerome Varghese, Vivan Kulkarni, Moksh Balaji, Neehar S, Yashil Shetty, Vibhor Sachan, Krish Gandhi, Yashwanth Reddy, Vansh Pahwa, Shreyas B S, Dhruv Agrawal, Nilay Sasturkar

Manipal Academy of Higher Education  
Manipal, Karnataka, India

[team.robomanipal@gmail.com](mailto:team.robomanipal@gmail.com)  
[www.robomanipal.com](http://www.robomanipal.com)

## Abstract

This paper presents the Team Description Paper of Team RoboManipal for RoboCup@Work 2026. **KARMA**, the Team's autonomous mobile manipulation robot, is designed to perform industrial navigation and manipulation tasks in structured work environments. The system consists of a custom mobile base and a multi-degree-of-freedom robotic arm, supported by in-house electronics and distributed computing. Perception is achieved using RGB-D sensing combined with deep learning-based object detection and classical vision pipelines for task-specific features such as barrier tape detection. Autonomous navigation is realised using 2D SLAM, sensor fusion, and path planning, while manipulation is executed through motion planning and inverse kinematics. The overall software architecture is built on ROS 2, enabling modular integration of perception, navigation, manipulation, and task planning.

## 1 Introduction

RoboCup@Work is an international robotics competition focused on autonomous mobile manipulation in industrial-like environments. Robots are required to navigate structured arenas, identify and manipulate objects, and execute complex tasks with minimal human intervention. These challenges demand robust perception, reliable navigation, precise manipulation, and coordinated task execution under real-world uncertainties.

Team RoboManipal is the official undergraduate robotics team of Manipal Academy of Higher Education (MAHE), India. The Team participates in the RoboCup@Work league intending to develop a reliable, scalable, and industry-oriented autonomous robotic system. The Team primarily comprises undergraduate engineering students in their second and third years of study, representing multiple disciplines and contributing across perception, planning, control, and hardware domains. For the RoboCup German Open 2026, the Team focuses on improving perception accuracy, navigation stability, and seamless coordination between manipulation and mobility. This paper presents the system architecture, key design decisions, and the technical contributions of the Team.

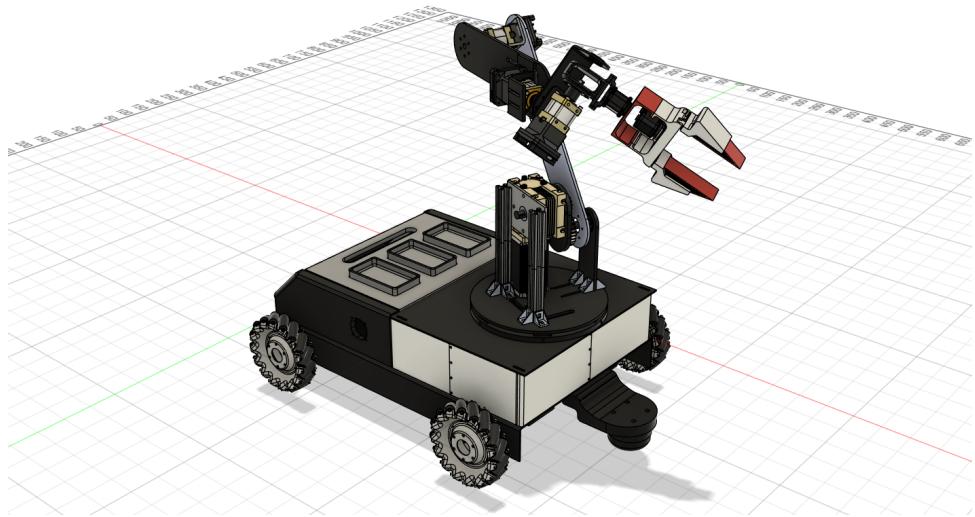


Figure 1: CAD model of **KARMA**, the complete mobile manipulation robot, showing the omnidirectional base, robotic arm, and overall mechanical integration.

## 1.1 Team Organisation and Achievements

Established in 2010 to foster innovation and excellence in robotics, Team RoboManipal consists of over 40 undergraduate students from diverse engineering backgrounds. The Team operates under the guidance of Dr Ishwar Bhiradi, Assistant Professor (Senior Scale) in the Department of Mechatronics Engineering at MIT, Manipal.

As part of its competitive journey, Team RoboManipal successfully qualified for **RoboCup 2025, Salvador, Brazil**, in the RoboCup@Work league. However, the Team was unable to participate in the event due to logistical constraints. Recent achievements are highlighted in Table 1.

Table 1: Competition Results Over the Years

Year	Location	Event / Competition	Result
2024	Delhi, India	ABU Robocon	22nd
2024	Manipal, India	Research Day at KMC Greens	2nd
2025	Chennai, India	Innovista Tech TED Challenge 2025	1st
2025	Bangalore, India	IISC Bangalore Robonautica -IISC Rhapsody (Mar 31)	3rd
2025	Bangalore, India	Byte Battles 2.0 (Ather Energy Hackathon)	1st
2025	Bangalore, India	Solstice 2025 – Line Follower (MIT Bangalore)	2nd
2025	Bangalore, India	Hardware Hackathon 2.0 – Journey to Mars (Dec 19-21)	1st

## 2 Hardware Description

### 2.1 Robotic Arm Design

The arm was designed using Fusion 360 and Solidworks, which provides 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, while Solidworks is better for URDF preparation which helps in simulation.

## 2.2 Base Mobility

The robot is built on a four-wheel Mecanum drive platform, enabling omnidirectional motion. The primary structural material used for the robot is mild steel (IS 2062 E250 grade), chosen for its strength and significantly lower cost compared to aluminium. The use of mild steel allowed us to achieve a rigid and durable structure while keeping overall manufacturing costs economical. To address corrosion concerns, the chassis components were treated with a protective primer coating followed by industrial-grade paint, which effectively prevents rust and enhances long-term durability. The wheels are driven by RMCS-2084 motors which provide sufficient torque directly.

## 2.3 Arm Dynamics

The robotic arm is designed with six degrees of freedom (6-DOF) to enable a wide range of motion, dexterity, and precise manipulation. The first degree of freedom is directly actuated using a NEMA 23 stepper motor coupled with a gearbox, delivering a total output torque of approximately 220 kg·cm. To ensure that the full structural load does not act directly on the motor shaft, a central bearing arrangement has been incorporated, significantly improving mechanical reliability and load distribution. The second degree of freedom also utilises a NEMA 23 stepper motor, and to meet the required torque demands, a 1:2 gear reduction has been implemented.

For the third, fourth, and fifth degrees of freedom, BLDC motors are employed to achieve smoother motion and higher efficiency. The BLDC motors used for the third and fourth joints provide a torque of 135 kg·cm, while the motor driving the fifth joint delivers 68 kg·cm, which is sufficient for the intended wrist and orientation movements. Structurally, the second and third joints are connected using an aluminium link to maintain strength and rigidity, while the sections from the third to the fifth degree of freedom utilise a combination of aluminium and PLA components. This hybrid material approach helps reduce overall weight and improve mass distribution, thereby enhancing dynamic performance and reducing actuator load.

The sixth degree of freedom, responsible for end-effector orientation, is actuated using a servo motor to achieve accurate positioning and closed-loop feedback, ensuring that torque and precision requirements are met effectively. Additionally, the gripper mechanism is driven by smart servo motors, allowing precise control and real-time feedback for reliable grasping and manipulation of objects. The overall mechanical architecture and kinematic layout of the robotic arm are inspired by the Franka Emika Panda robotic arm, incorporating similar design philosophies focused on modularity, lightweight construction, and smooth, controlled motion.

## 2.4 Electronic Compartment With Thermal Control

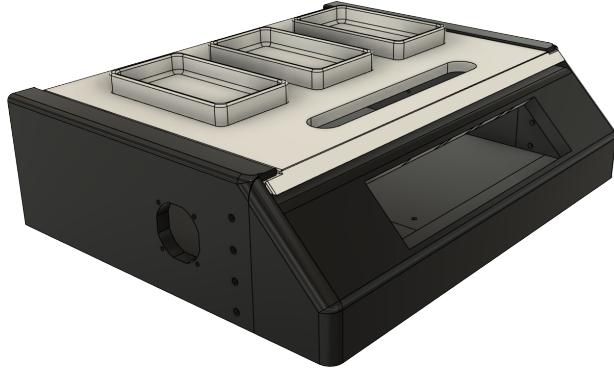


Figure 2: Electronics enclosure housing power distribution and control hardware

A dedicated electronics enclosure was designed to house all major electronic components, with the primary objective of simplifying wiring, improving fault detection, and enhancing the overall aesthetics of the robot. Centralising electronics into a single, well-organised compartment enables cleaner cable routing, easier maintenance, and quicker identification of wiring or component-related issues during

testing and operation. The enclosure is fabricated using PLA, which allows rapid prototyping, lightweight construction, and precise customisation to fit the system layout.

To ensure reliable operation and prevent overheating, the enclosure is equipped with two cooling fans, mounted on opposite sides to facilitate adequate airflow across the internal components. This active cooling arrangement helps dissipate heat generated by the electronics, maintaining safe operating temperatures and improving system reliability during prolonged use.

## 2.5 Power Electronics And Control System

The bot's power electronics and hardware are designed for efficiency and reliability, featuring a custom control board with an integrated power distribution system. The Base PCB is designed to distribute power from the main 24V battery to the base motors, the arm, and the Jetson. It also includes two electrically isolated STM32F411 microcontrollers that control the motors and send encoder and IMU data to the Jetson. To meet the power requirements of various components, buck converters have been incorporated to efficiently step down the voltage to the required levels.

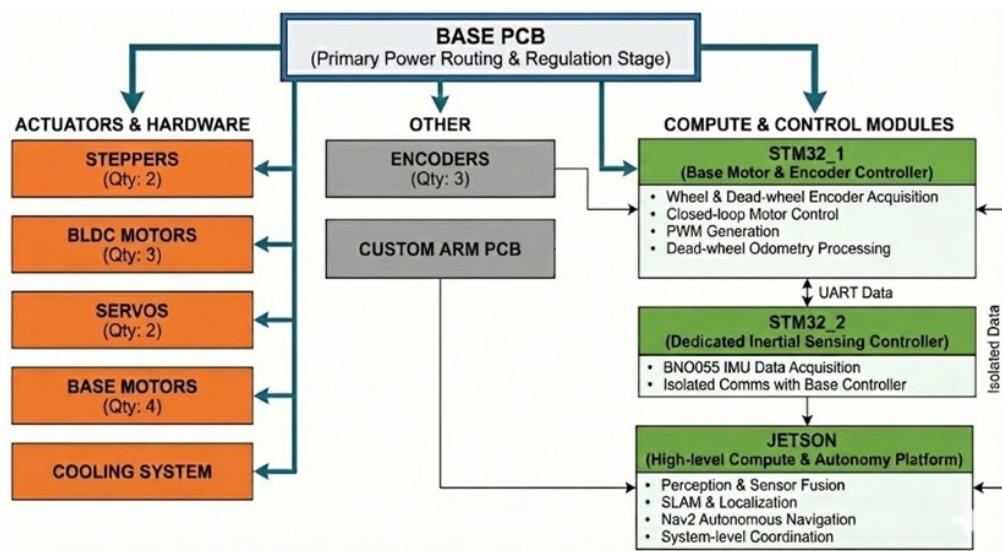


Figure 3: Electrical system overview showing interconnections between motors, sensors, microcontrollers, and power distribution units, including Cytron motor drivers and NVIDIA Jetson Orin compute modules.

### 2.5.1 In-House PCB for closed-loop arm control

The Arm PCB is a joint-mounted motor controller for both stepper motors and serial bus servos, designed for our 6-DOF robotic arm. Powered by an STM32F103, it utilizes an AS5600 encoder for precise closed-loop feedback. The board supports versatile connectivity, including a CAN transceiver, USB interface, and half-duplex UART for Dynamixel AX servos. Capable of running on 12V or USB power, it provides a compact, all-in-one solution for robust motion control and communication.

### 2.5.2 Serial bus servo motor control

Dynamixel servo was used to control 1 DOF of the 6 DOF robotic arm apart from the end effector. The servo motors are powered and managed by the STM32F1 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 movement.

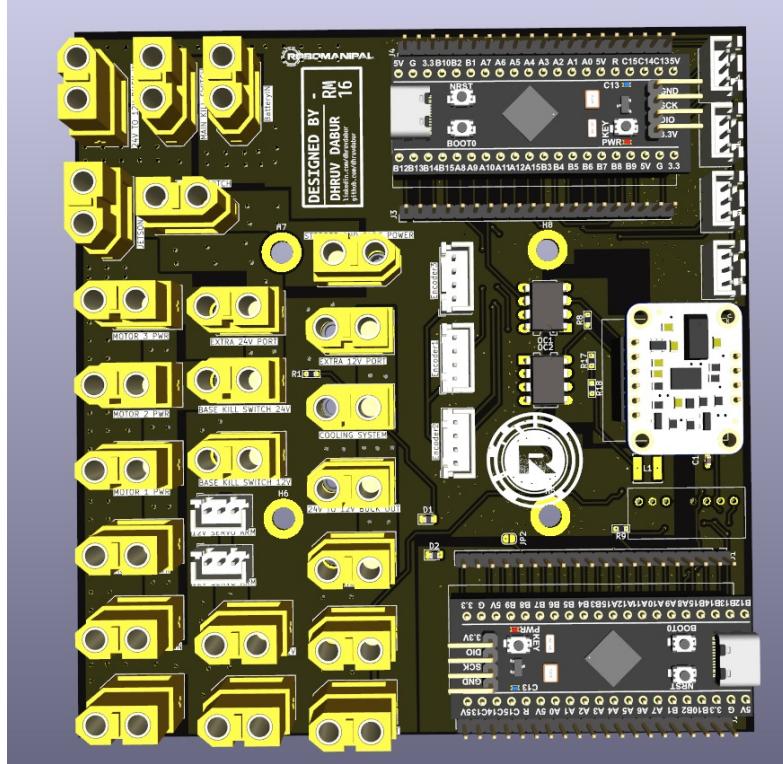


Figure 4: Custom control and power distribution board used for motor control, sensor interfacing, and regulated power delivery across the robotic system.

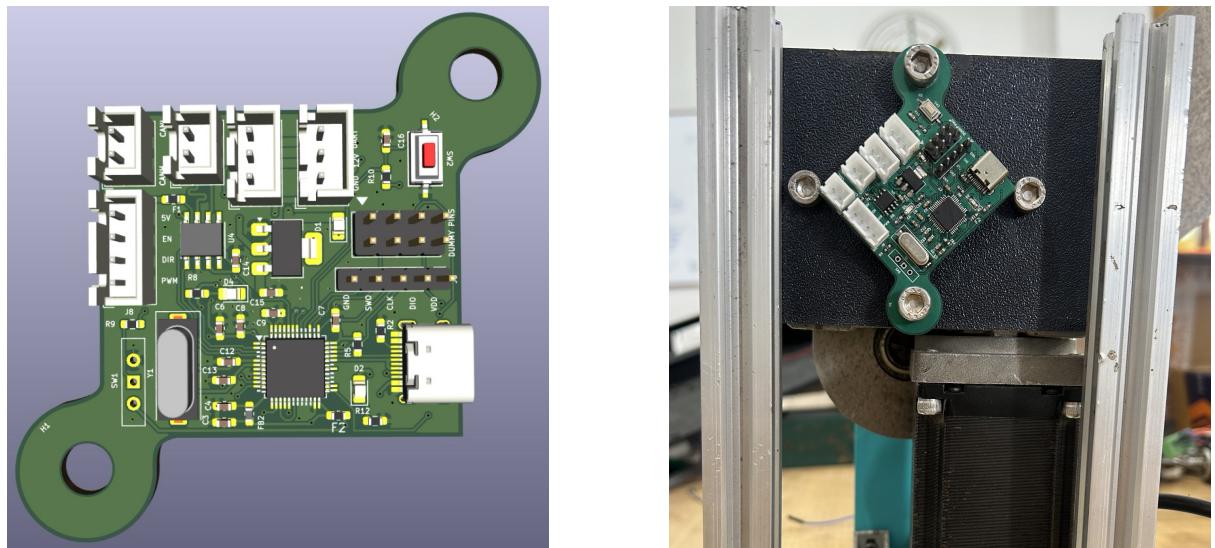


Figure 5: Closed-loop feedback mechanism using a magnetic encoder (contactless potentiometer). Left: PCB with integrated encoder setup. Right: encoder hardware used for joint position sensing.

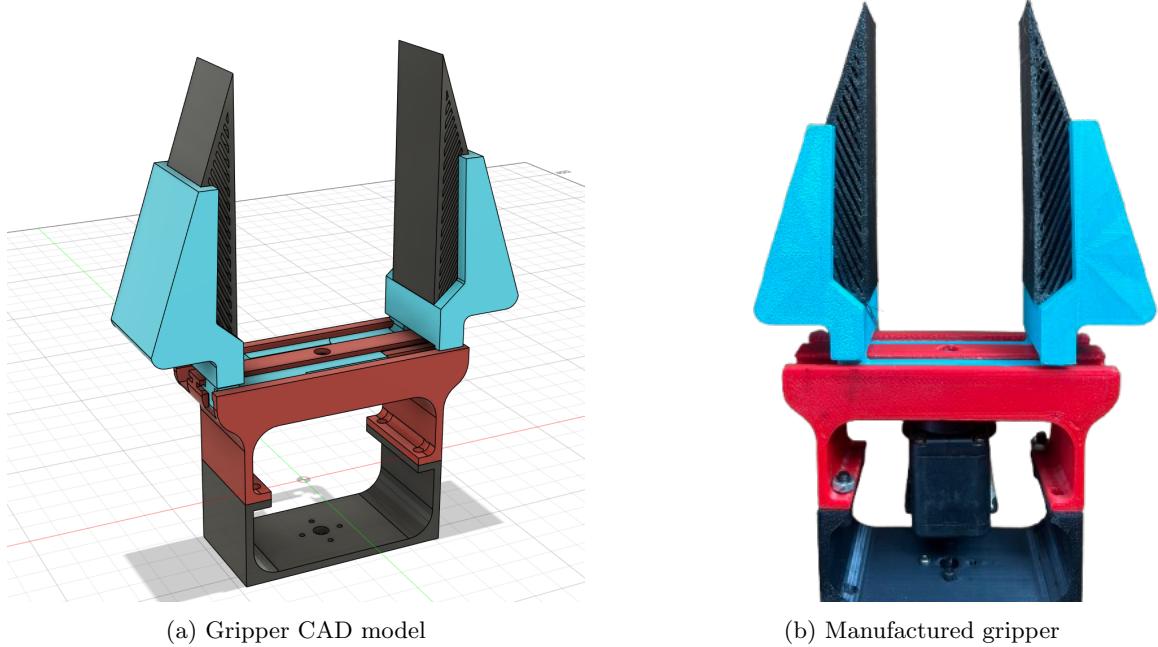


Figure 6: Comparison between gripper CAD design and manufactured prototype

### 2.5.3 Galvanic isolation and component protection framework

The electronic architecture of the robot is designed with a strong emphasis on electrical isolation, protection, and serviceability to ensure reliable operation in high-current, noise-prone environments. Power electronics, motor drivers, sensing modules, and computing units are treated as distinct electrical domains to minimize interference and prevent fault propagation. High-current motor subsystems are physically and electrically separated from low-power logic, sensing circuits, and onboard computing platforms, reducing the impact of voltage transients, ground loops, and electromagnetic interference generated during aggressive motion or rapid load changes. A centrally located emergency kill switch is integrated into the main power path and is protected using a glass fuse, ensuring rapid manual shutdown while providing an additional layer of overcurrent protection for the entire system.

Isolation is implemented at both the power and communication levels, with additional emphasis on protecting sensitive components from electrical faults and data corruption. Dedicated DC–DC converters provide galvanically isolated 5 V supply rails for microcontrollers and sensors, while the Jetson compute platforms are powered through an isolated supply derived from a custom power distribution and motor control board. Polyfuses and TVS diodes are strategically deployed across power and signal lines to safeguard sensitive electronics against overcurrent conditions, voltage spikes, and transient disturbances, thereby improving reliability and preventing unintended resets or data corruption. Communication between isolated domains is handled using optocouplers, allowing encoder data, control commands, and status signals to be exchanged without direct electrical coupling, preserving signal integrity and overall system safety.

At the system level, this isolation and protection strategy enables modularity, fault containment, and reliable long-term operation. Separate microcontrollers independently manage motor control, encoder processing, and inertial sensing, each operating within its own protected electrical domain. The electronics are housed in an easily accessible enclosure, complemented by a front control panel that allows individual subsystems to be selectively powered on or off. This capability greatly simplifies testing, debugging, and incremental bring-up by enabling isolated verification of each subsystem. In the event of a disturbance or failure in one domain, electrical isolation prevents cascading effects across the robot, allowing safe degradation or controlled shutdown. This layered isolation, protected power delivery,

and service-oriented design significantly enhance robustness and ensure stable autonomous operation in demanding industrial and RoboCup@Work environments.

### 3 Software Architecture

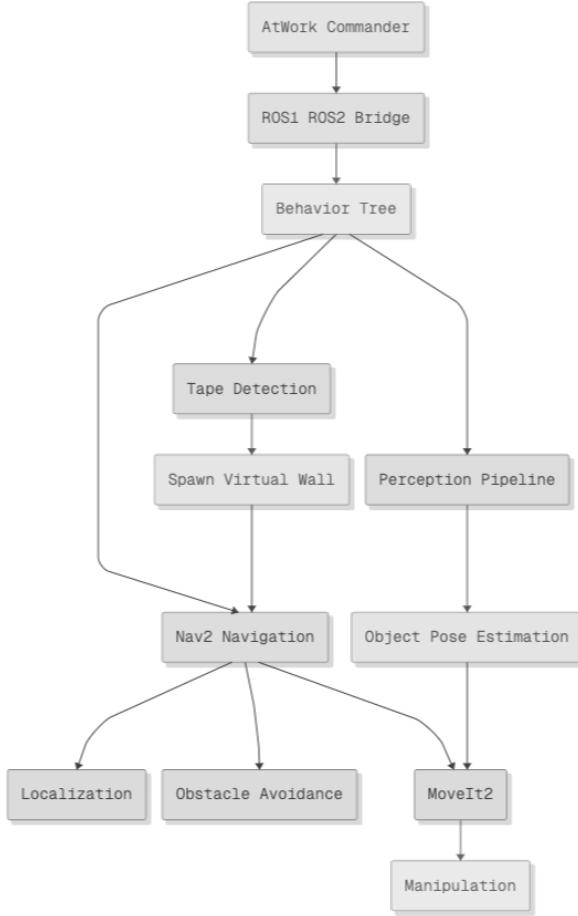


Figure 7: High-level software architecture of the robotic system showing perception, navigation, manipulation, and task coordination modules built on ROS 2.

The software system is designed as a modular and distributed architecture built entirely on ROS 2. The primary design objectives are robustness, real-time performance on embedded hardware, and clean separation between perception, navigation, manipulation, and task-level coordination. The architecture follows a layered approach, allowing individual subsystems to be developed, tested, and extended independently while maintaining tight integration during execution.

#### 3.1 System Overview

The robot operates using multiple ROS 2 nodes distributed across Jetson-class embedded computing platforms. Perception, navigation, and manipulation are implemented as loosely coupled subsystems that communicate through standard ROS 2 topics, services, and actions. This design enables parallel execution of computationally intensive tasks such as vision-based perception and motion planning while preserving deterministic control behavior at the actuator level.

Transform management follows standard ROS 2 conventions, with static transforms defining fixed sensor mounting relationships and dynamic transforms maintaining real-time pose estimates of the robot and its components. This unified transform tree enables consistent spatial reasoning across all software modules.

## 3.2 Perception System

The perception pipeline is implemented in ROS 2 and optimized for real-time execution on Jetson-class hardware. It processes synchronized RGB-D input from a stereo depth camera and is designed to robustly handle variations in lighting, surface reflectivity, and sensor noise commonly encountered in industrial and RoboCup@Work environments.

### 3.2.1 Barrier Tape Detection

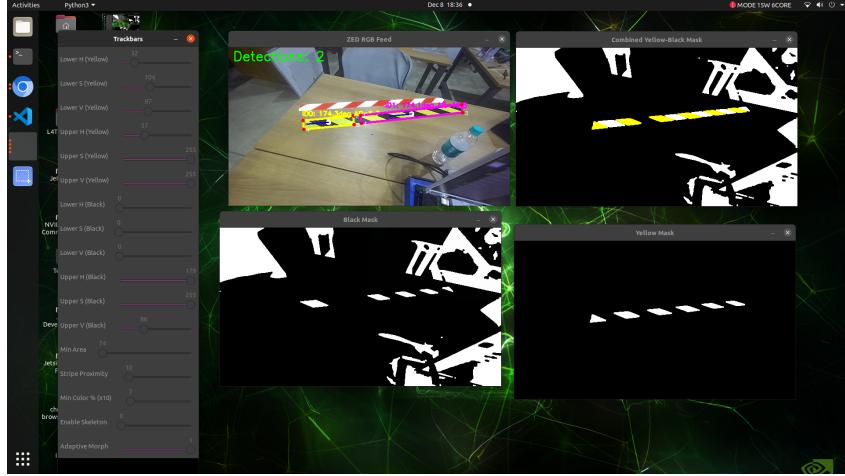


Figure 8: Detection of yellow–black safety tape used for navigation constraints and workspace boundary awareness.

Barrier tape detection is implemented as a lightweight vision–depth fusion pipeline to support safe navigation and workspace awareness. The system operates on synchronized RGB–D input and is designed to remain robust under reflective surfaces, uneven lighting, and sensor noise.

RGB images are preprocessed using luminance normalization to suppress glare, followed by color-based segmentation in HSV space to identify multiple tape types, including yellow–black striped, red–white striped, and solid green tapes. Basic morphological operations are applied to reconnect fragmented tape regions and remove noise.

Detected regions are filtered using simple geometric constraints to retain only elongated, tape-like structures. Valid detections are then associated with depth information and projected into 3D space using camera calibration parameters. The resulting 3D tape representations are published for downstream navigation and safety modules.

### 3.2.2 Object Perception and Pose Estimation

Object perception follows a dual-branch architecture that separates 2D semantic understanding from 3D geometric reasoning. In the 2D branch, object detection is performed using a TensorRT-optimized YOLO network, enabling low-latency inference on embedded GPU hardware. This branch produces tight 2D bounding boxes for objects of interest in the image plane.

In parallel, the 3D branch processes depth data to generate a point cloud representation of the scene. Dominant planar surfaces such as tables and workbenches are identified and removed using plane segmentation, retaining a clean object-level point cloud for further processing.

The outputs of both branches are fused by projecting 2D bounding boxes into 3D space to crop the corresponding regions from the point cloud. Background planar points are removed using the previously estimated plane equation, resulting in a minimal-clutter object cloud.

Full six-degree-of-freedom (6-DoF) object pose estimation is performed using a hybrid geometric approach. The centroid of the segmented cloud provides the translational components, while orientation is derived by aligning one axis perpendicular to the supporting plane, a second axis parallel to the plane, and the third axis with the principal geometric direction obtained from eigenvector analysis of the point cloud.

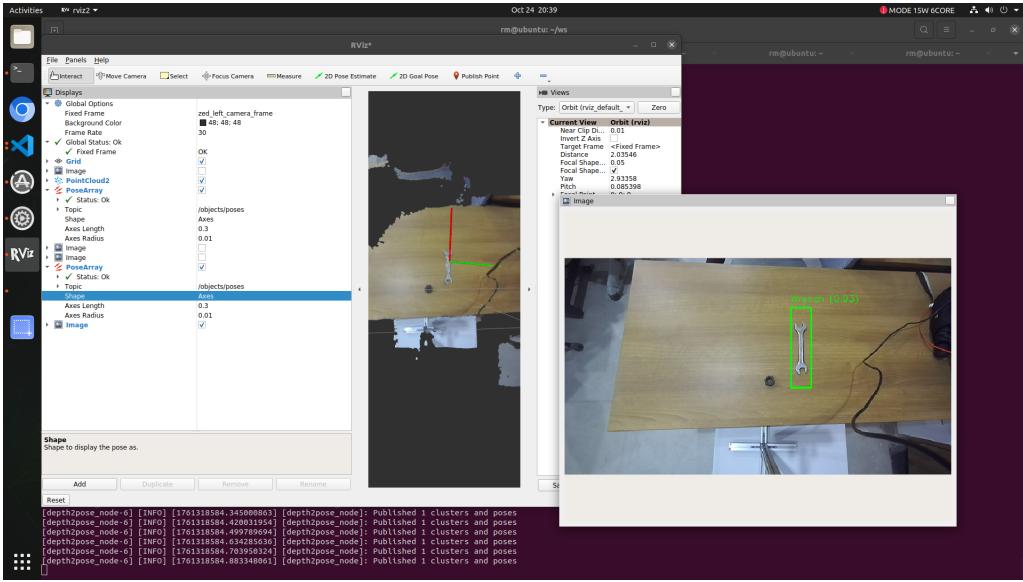


Figure 9: Perception pipeline output showing TensorRT-accelerated YOLO-based object detection and the corresponding estimated 6-DoF pose of a wrench using fused RGB–D data.

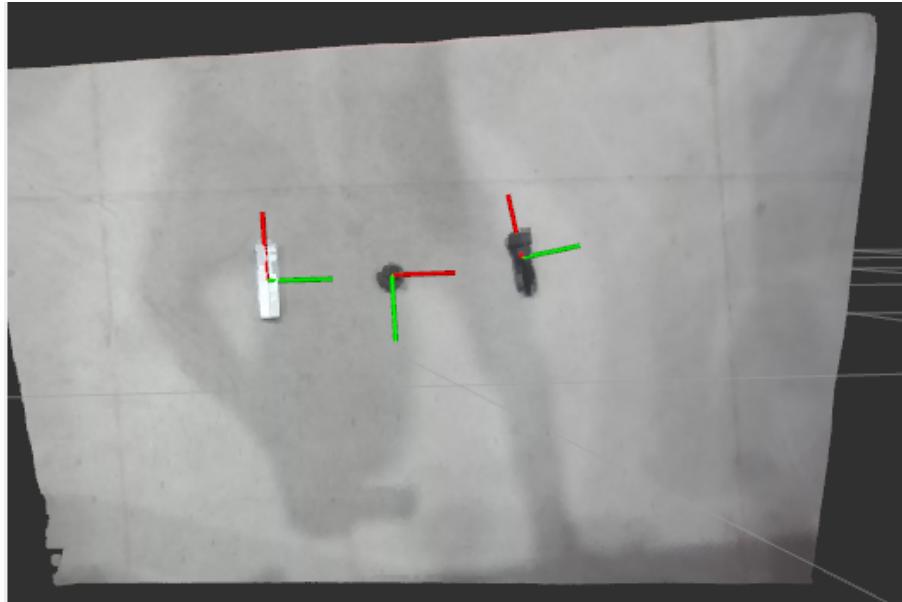


Figure 10: Overview of the 6-DoF object pose estimation pipeline combining 2D detection and 3D point cloud processing.

### 3.3 Navigation and Localization

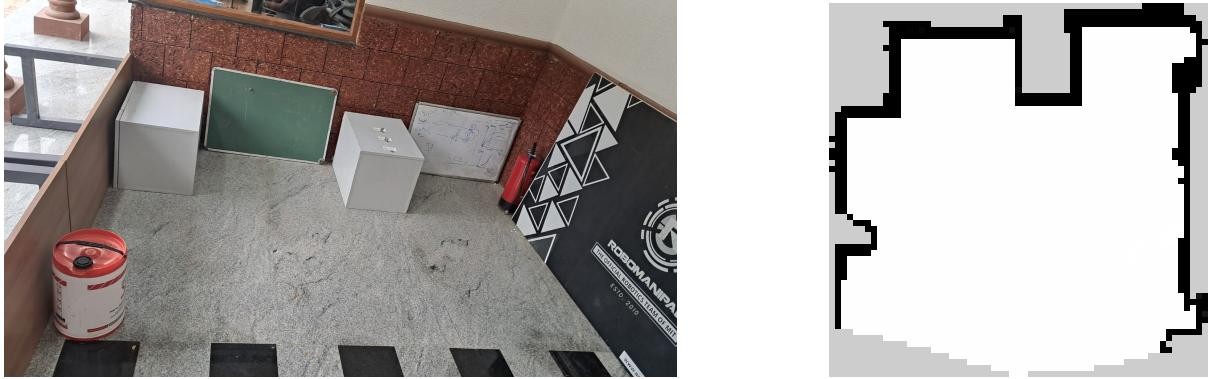


Figure 11: (Left) Experimental arena used for autonomous navigation tasks. (Right) Occupancy grid map generated autonomously by the robot during SLAM-based exploration.

**Video:** Mapping and navigation demonstration

Autonomous navigation and localization are achieved through the integration of dead-wheel odometry, LiDAR-based SLAM, and the ROS 2 Navigation (Nav2) stack. This combination provides accurate short-term motion estimation, drift-corrected global localization, and reliable path execution in structured indoor environments.

#### 3.3.1 Dead-Wheel Odometry



Figure 12: Dead-wheel odometry setup

The robot employs a three-wheel dead-wheel odometry system consisting of three passive omnidirectional encoder wheels arranged in a triangular configuration. Two longitudinal wheels are mounted parallel to the robot’s forward direction and placed symmetrically on either side of the chassis to measure forward motion and rotation. A third lateral wheel is oriented perpendicular to the forward direction to measure sideways motion.

As the robot moves, encoder ticks from each wheel are converted into linear displacements using calibrated wheel radii and encoder resolutions. Forward displacement is estimated from the average of the left and right wheel measurements, while rotational motion is derived from their difference. Lateral motion is computed from the front wheel measurement with compensation for rotation based on the wheel’s offset from the robot’s center of rotation.

The resulting motion increments are integrated over time to estimate the robot’s pose. Since the tracking wheels are passive and not driven, this approach significantly reduces odometry errors caused by wheel slip during acceleration, braking, or uneven floor contact. An Extended Kalman Filter (robot\_localization) fuses the dead wheel encoder odometry and the 9 axis IMU to produce a smooth, drift reduced odometry estimate in the odom frame that provides better pose accuracy for local planning.

### 3.3.2 SLAM and Localization

Simultaneous Localization and Mapping is performed using SLAM Toolbox in an online asynchronous pose-graph configuration. Front-facing LiDAR scan data is filtered to retain only a  $150^\circ$  field of view, spanning from  $75^\circ$  to  $+75^\circ$  relative to the robot's forward axis, removing rear-facing measurements that may intersect the robot's structure or introduce mapping artifacts. These filtered scans are fused with dead-wheel odometry to provide an initial motion estimate between successive observations.

Incoming scans are aligned against the accumulated global map using scan-to-map matching, allowing correction of odometry drift. Loop closure detection identifies revisited areas and triggers pose-graph optimization, which redistributes accumulated error across historical poses and maintains long-term map consistency.

The transform hierarchy follows standard ROS 2 conventions, with dead-wheel odometry providing smooth short-term motion estimation and SLAM Toolbox publishing map-to-odometry corrections for global consistency. AMCL(Adaptive Monte Carlo Localization) gives us LiDAR based, map frame pose estimation through Nav2 by performing scan to map matching and provides robust map relative poses and map to odom corrections for reliable global localization and path execution.

This process produces a drift-corrected occupancy grid map and accurate localization suitable for autonomous navigation.

### 3.3.3 Navigation Planning and Control

Path planning and execution are handled by the ROS 2 Navigation (Nav2) stack. A dual-costmap architecture is used, with a global costmap operating in the map frame for long-range planning and a rolling local costmap in the odometry frame for real-time obstacle avoidance.

Global paths are generated using the Navfn planner employing A\* algorithm, while local trajectory execution is performed by the TEB (Timed Elastic Band) local planner. The TEB local planner optimizes trajectories in time and space for holonomic motion suitable for the mecanum-wheel drivetrain and evaluates them using multiple cost criteria, including obstacle proximity, path alignment, and goal progress. A behavior-tree-based navigator coordinates planning, control, recovery behaviors, and replanning, while a velocity smoother enforces acceleration limits to ensure smooth and dynamically feasible motion.

Together, the integration of dead-wheel odometry, SLAM Toolbox, and the Nav2 stack enables reliable, drift-resistant autonomous navigation in complex indoor RoboCup@Work environments.

## 3.4 Arm Manipulation Software

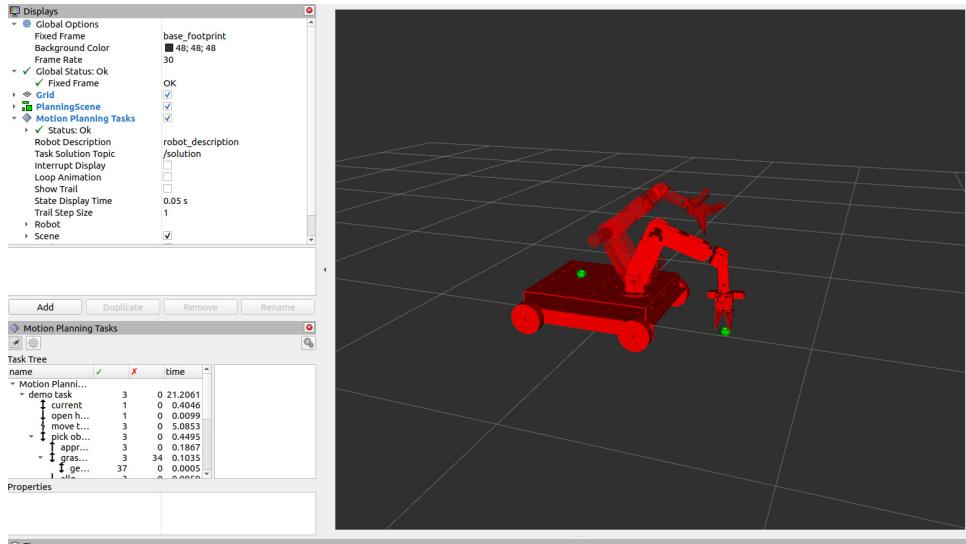


Figure 13: MoveIt 2-based manipulation pipeline showing motion planning, collision checking, and trajectory execution for the robotic arm.

**Demonstration video:** Pick-and-place execution

Manipulation is implemented using MoveIt 2, which provides motion planning, inverse kinematics, collision checking, and trajectory generation for the robotic arm. The planning scene maintains a real-time 3D representation of the robot and its surroundings, enabling collision-aware motion generation.

MoveIt 2 interfaces with the ROS 2 Control framework for trajectory execution. Planned joint trajectories are time-parameterized and streamed to low-level controllers, which manage joint-level actuation and feedback. This separation between planning and execution enables high-level autonomy while maintaining deterministic timing and safe control behavior.

A custom hardware interface manages all the actuators of the robotic arm and exposes their positions to the planning system enabling real-time monitoring of the arm’s motion and behavior.

### 3.5 System Integration and Safety

System-wide safety is enforced through multiple software layers. Homing sequences establish absolute joint references at startup, while software-defined joint limits and workspace constraints prevent over-travel. Watchdog mechanisms monitor communication health between computing platforms and controllers, triggering controlled stops in the event of timing violations or sensor faults.

The tight integration between perception, planning, and control enables safe interruption, replanning, or recovery during execution. This capability is critical for RoboCup@Work scenarios, where dynamic environments and shared human–robot spaces require robust and predictable behavior.

## 4 Key Improvements and Contributions

Compared to the previous iteration, the current system improves robustness, component safety and reliability across both hardware and software. The electronics box has been reorganized for power, computation, and communication components, improving maintainability. The perception pipeline has been redesigned by combining deep- learning-based object detection with geometric reasoning and conservative depth validation, resulting in more reliable performance under difficult lighting and reflective surfaces. Localization accuracy has been improved through the integration of dead-wheel odometry, significantly reducing pose drift caused by wheel slip.

For the hardware, the drive system has been changed from stepper/servo motors to BLDC motors, providing higher efficiency and smoother velocity control. The software stack is deployed across multiple Jetson-class embedded platforms for parallel execution, while a ROS 2-based architecture cleanly separates perception, navigation, manipulation, and task coordination.

## 5 Future Work

Future development will focus on further enhancing perception and navigation capabilities of KARMA. Object pose estimation accuracy can be improved through depth-aware refinement techniques and by expanding training datasets to cover a wider range of object variations and lighting conditions. Improving the interaction between SLAM, planning, and local control will also be essential for achieving stable localization and *reliable* obstacle avoidance in dynamic and cluttered workspaces.

Another major direction involves advancing manipulation and coordination between the mobile base and the robotic arm. Planned improvements include adaptive grasp selection, recovery strategies for failed grasps, and tighter base–arm synchronization to enable efficient execution of complex manipulation tasks. Finally, additional efforts will be directed toward system reliability and safety by extending monitoring, fault detection, and recovery mechanisms to support long-term autonomous operation under competition conditions.

## 6 Conclusion

This technical description of KARMA presented the complete design and system architecture of Team RoboManipal’s Work robot, detailing the integrated hardware platform, distributed ROS 2-based software stack, and key system-level improvements over previous iterations. Unlike earlier work that primarily focused on planned advancements, the current system demonstrates a consolidated and deployable solution for autonomous manipulation and navigation in industrial-like environments.

Consistent depth-aware perception, improved localization through dead-wheel odometry, and collision-aware manipulation collectively enable dependable task execution with minimal human intervention. The

modular software architecture and structured hardware design further enhance system maintainability, scalability, and operational safety.

The presented system provides a strong foundation for RoboCup@Work participation, representing a stable and well-integrated robotic platform. The achieved improvements in reliability, modularity, and safety position the system as a dependable baseline for continued development and competitive deployment.

## 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>
- [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 preprint 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>
- [7] Bosch Sensortec, “BNO055 Intelligent 9-axis Absolute Orientation Sensor,” Datasheet, 2020.
- [8] STMicroelectronics, “STM32F4 Series Reference Manual,” RM0090 Reference Manual, 2021.
- [9] NVIDIA, “Jetson Orin NX Technical Brief,” NVIDIA Documentation, 2023.
- [10] StereoLabs, “ZED SDK Programming Guide,” Version 3.8, 2023.
- [11] “RoboCup@Work Objects Dataset 2024,” Roboflow Universe. [Online]. Available: <https://universe.roboflow.com>
- [12] Slamtec, “sllidar\_ros2,” GitHub repository, 2024. [Online]. Available: [https://github.com/Slamtec/sllidar\\_ros2](https://github.com/Slamtec/sllidar_ros2)
- [13] Open Robotics, “ROS 2 Documentation,” [Online]. Available: <https://docs.ros.org>