Technical Design Paper for the Robotics Dojo Competition 2025

Paul Migwi, James Gathirwa, Zebby Akach (DEKUT AMR Mobile platform team)
Michael Machohi, Brian Kiprono (DEKUT AMR Navigation team)

***Abstract*—This paper outlines the journey of building the DEKUT AMR, an Autonomous Mobile Robot crafted for the 2025 Robotics Dojo Competition. Designed with a skid-steer system equipped with All Terrain-Wheels, it tackles challenging terrains like slopes, sawdust, rocky patches, and cardboard landscapes with impressive stability. The robot features a laser-cut acrylic chassis for strength, a 3D-printed servo-powered tipper for offloading, and a power setup with a 20V 2Ah Ingco battery and XL4015 buck converters (8-36V, 5A max) stepping down to 12V for motors and 5V for the Raspberry Pi 4. Enhanced with LiDAR and a Raspberry Pi camera, it handles navigation and computer vision tasks, identifying cube colors, loading ,offloading points, and diseased plants detected by an ML model. It integrates SLAM via the SLAM Toolbox principles to ensure predictable performance in dynamic environments. Our design choices prioritize reliability for this hefty robot, drawing on lessons from past issues with underpowered motors and overheating electronics to ensure long-lasting performance. Through iterative testing , we've shaped a robot ready to share its story and lessons with future teams, contributing to efficient warehouse-like navigation systems.**

I. INTRODUCTION

The purpose of the Robotics Dojo competition is to enhance the community of innovators capable of substantive contributions to the domain of autonomous unmanned systems. This enhancement is achieved by providing a venue and mechanism whereby the practitioners of the autonomous systems community may form new connections and collaborations, increase their proficiency and inventiveness, and foster their passion for robotics in the maritime domain.

A. Design Strategy

The core strategy was to design for robustness and adaptability. We began with a low–center-of-gravity, modular chassis powered by high-torque motors with conservative gear reduction and a power system sized with 30–50% headroom plus active thermal management. This was paired with a compliant drivetrain using wide, low-pressure tires to maintain traction across mixed surfaces such as cardboard, sawdust, ballast, and grass. The electronics were designed to remain serviceable for rapid field repairs.

While the core strategy guided the design, practical testing highlighted areas for refinement. Initially, we used medium-torque, high-RPM motors in a skid-steer setup, but the wide, low-pressure spiked tires generated excessive grip, preventing effective turning. To address this, we switched to high-torque rear motors and shifted much of the robot's weight toward the rear, which improved maneuverability and turning performance. This adjustment also required upgrading the electrical system to handle higher current demands. Throughout these iterations, cost-effective components were prioritized to stay within budget, as detailed below.

TABLE I

COMPONENT SPECIFICATIONS

Component	Description/Specifications
Encoded DC Motor	Motor with encoder for speed and position feedback
LM2596 Buck Converter	DC-DC step-down converter (adjustable output)
XL4015 DC-DC Buck Converter	High-current step-down converter (up to 5A)
Hex Connector (Coupling)	Mechanical coupling between motor shaft and load
Motor Mount	Bracket/frame for securing DC motor
18650 Lithium-Ion Battery	Rechargeable cell, commonly 3.7V nominal
Motor Driver L298N	Dual H-bridge driver for DC motors
Wheels	Circular mechanical parts for locomotion
Battery Charger	Charging unit for 18650 cells
Battery Holder	Housing for multiple 18650 cells

For mapping, we utilized SLAM Toolbox in *online_async* mode, combining LiDAR (/scan) and odometry (/odom) data through scan matching to build and update maps in real time. For localization, we also relied on SLAM Toolbox, running the localization node to load prebuilt maps and maintain the robot's pose estimate during operation.

Navigation was implemented using the Nav2 stack for path planning. During mapping, we ran the bringup launch file with SLAM Toolbox publishing the /map topic, while in localization mode we added AMCL to maintain pose estimates against the prebuilt map. Path planning used the A global planner and the DWB local planner. Twist Mux was configured to arbitrate between multiple velocity command sources, ensuring that user inputs could override Nav2 outputs when necessary, either for safety or manual control.

B. Vehicle design

This section delves into the practical realization of the DEKUT AMR, weaving together mechanical, electrical, and software elements while reflecting on the journey of design, lessons learned, and forward-looking insights.

I. Mechanical Design

The mechanical structure of the DEKUT AMR is built on a laser-cut acrylic chassis, selected for its ability to support the weight of a heavy AGV. Four motors were mounted on precisely drilled slots and paired with all-terrain wheels chosen for their grip and stability across varied surfaces. To enhance the skid-steer effect, the front motors were rated at 200 RPM while the rear motors were rated at 110 RPM, providing the torque balance and maneuverability needed for reliable operation. The wheel design further aids in distributing load and minimizing slippage on uneven terrain..
3D printing crafted custom parts, the LiDAR base, camera mount, and a servo-powered tipper,
allowing rapid prototyping and weight savings. The tipper, actuated by a servo with a 45° angle handles offloading efficiently. Early tests revealed vibrations loosening motor brackets during turning, teaching us to plan threaded inserts for future builds to enhance durability.
The camera and lidar are positioned at the top optimizing the field of view for both sensors to align SLAM data with visual inputs.

II. Electrical Design

Powering the DEKUT AMR is a 20V 2Ah Ingco battery, a significant upgrade from the underperforming 18650 batteries that once limited runtime. Two XL4015 buck converters (8-36V input, 5A max) step down voltage that exploits benefits of a heatsink and higher current output: one step down to 12V for the motors to prevent overload, and another to 5V for the Raspberry Pi 4, resolving past overheating issues with the LM2596 converter. The Arduino Mega draws 5V from the Pi, while the servo SG90 (for the tipper), LiDAR, and camera share this line. This centralized setup simplified wiring but required careful current monitoring to avoid drops, a lesson now integrated into our design process. The battery supports approximately 2 hours of operation under full load, a critical factor for competition endurance.

Wiring is organized with color-coded cables and fused junctions to protect against shorts, when on different terrains. The raspberry pi 4 powers the LiDAR by a USB cable and camera is powered via the camera ribbon flex CSI cable, ensuring stable 5V delivery, while motor drivers receive 12V directly from its respective buck.

III. Software Design

The software stack integrates ROS2 with Arduino-based low-level motor control, SLAM Toolbox for mapping and localization, Nav2 for path planning, and perception nodes for cube identification and plant health monitoring.

At the lowest level, an Arduino handles PWM motor actuation and encoder feedback. Communication with ROS2 is achieved via a single serial port, where velocity commands are transmitted to the Arduino and encoder counts are returned for odometry. To avoid blocking issues on this shared serial interface, a ROS2 multithreaded executor was implemented, enabling concurrent forward (ROS → Arduino) and reverse (Arduino → ROS) communication. Encoder counts are processed into the /odom topic (nav_msgs/Odometry), providing continuous wheel odometry.

For LiDAR input, an RPLiDAR node runs on the Raspberry Pi, publishing /scan (sensor_msgs/LaserScan) data. SLAM Toolbox, configured in online_async mode, fuses /odom and /scan data via scan matching to build and update maps as the AGV navigates. Localization is achieved either by running SLAM Toolbox in localization mode (loading serialized maps) or through Nav2 with AMCL (amcl node), depending on the operational context. A URDF model generated in Fusion360 defines the robot's kinematic structure (odom → base_link), and care was required to align Nav2 parameters, since the stack defaults to base_footprint; parameter resets ensured consistency with base_link. The TF tree was continuously verified to prevent runtime errors in RViz.

Navigation is handled through Nav2. During teleoperation, two velocity command sources are available: a custom keyboard teleop node (keyboard_vel) and teleop_twist_joy (joy_vel). Both, along with Nav2-generated commands, are passed through twist_mux, which prioritizes inputs and publishes a unified cmd_vel_out topic. This topic is subscribed by the serial bridge node, which relays commands to the Arduino for actuation. Path planning combines the A* global planner with the DWB local planner, enabling route generation on loaded maps and reactive avoidance of terrain irregularities.

Perception tasks were implemented using the Raspberry Pi camera. For cube handling, OpenCV with HSV thresholding detected cube colors after loading by an external manipulator. This information was fused with SLAM-based localization to trigger servo-actuated offloading into color-coded zones. For plant inspection, a trained convolutional neural network model processed the camera stream to classify plant health states (e.g., healthy vs. diseased), extending the AGV's functionality beyond logistics.

Challenges encountered included serial synchronization between Arduino and ROS, parameter mismatches in Nav2 (base_link vs. base_footprint), and occasional delays in map loading from SLAM Toolbox. Future improvements include encoder redundancy for more robust odometry and upgraded servos for higher torque during offloading operations..

C. Experimental Results

This section presents the results of hands-on testing for the DEKUT AMR, conducted over 50 hours across different terrains to validate performance, reliability, and system integration under conditions similar to the Robotics Dojo Competition.

Initial tests evaluated the effectiveness of the revised motor setup. The skid-steer configuration was validated by confirming that the high-torque rear motors (110 RPM) and medium-speed front motors (200 RPM) could provide adequate turning capability on different terrains. Steep, rough terrain trials further confirmed the drivetrain's torque capacity. Motor calibration included measurement of linear speed (0.425 m/s) and maximum angular velocity (2.99 rad/s), which informed the development of custom Arduino control code for deadzone handling, velocity clipping, and scaling.

Endurance tests involved running the AGV continuously for two hours to evaluate battery capacity and thermal stability. The Inco 20 V 2 Ah battery performed reliably, with its built-in thermal sensor (thermostat/thermistor) enabling real-time monitoring. XL4015 converters were tested under full load, revealing a safe duty cycle limit to prevent overheating. The Raspberry Pi buck converter exhibited mild heating, mitigated effectively by a heatsink.

The servo-powered tipper was tested for a 50 g payload, successfully achieving a 45° tilt on cardboard terrain. The RPLiDAR was assessed for /scan range consistency in sawdust, while SLAM Toolbox parameters in online_async mode were tuned, particularly resolution and travel distance settings to improve map quality. The Pi camera with OpenCV HSV thresholding (hue 0–10, saturation >50, value >50) reliably detected red cubes under varying lighting conditions.

SLAM robustness and navigation accuracy were primary focuses. The Nav2 stack required extensive parameter tuning due to the robot's weight and unique dynamics. Default configurations led to deadband issues and control instability, which were addressed by adjusting parameters in the behavior server, velocity smoother, and controller server. Final navigation settings achieved reliable performance with maximum velocities of 0.25 m/s linear and 1.9 rad/s angular. These values were applied consistently across Nav2 and teleoperation modes. Teleop testing also incorporated teleop_twist_joy with turbo mode enabled, allowing full-throttle operation when required.

D. Acknowledgements

We're grateful to Dedan Kimathi University of Technology for providing foundational knowledge and lab access, Jomo Kenyatta University of Technology for hosting us, JICA for organizing and funding the competition, and our supervisor Prof. Michael M. Gichane for his invaluable guidance, technical insights, and continuous support driving our success.

E. References

[1] Al-Abboodi, A., Al-Abboodi, M., & Al-Mamouri, A. (2020). *Comprehensive Study of Skid-Steer Wheeled Mobile Robots: Development and Challenges*. *Industrial Robot: An International Journal*, 47(5), 665-694. doi:10.1108/IR-03-2020-0049. A detailed examination of skid-steer robot mechanics, focusing on torque distribution and terrain adaptability, with practical insights into failure prevention.

[2] Craig, J. J. (2005). *Introduction to Robotics: Mechanics and Control* (3rd ed.). Upper Saddle River, NJ: Pearson Education. A foundational text covering robotic kinematics, dynamics, and control systems, with extensive sections on motor control and path planning relevant to AGV design.

[3] Corke, P. (2017). *Robotics, Vision and Control: Fundamental Algorithms in MATLAB* (2nd ed.). Cham, Switzerland: Springer. Offers in-depth algorithms for vision-based navigation and control, including practical implementations for sensor integration in mobile robots.

[4] Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous Localization and Mapping (SLAM): Part I The Essential Algorithms. *IEEE Robotics & Automation Magazine*, 13(2), 99-110. doi:10.1109/MRA.2006.1638022. Provides a comprehensive overview of SLAM algorithms, emphasizing real-time mapping techniques without prior environmental data.

[5] Everett, H. R. (1995). *Sensors for Mobile Robots: Theory and Application*. Wellesley, MA: A.K. Peters/CRC Press. A thorough resource on sensor technologies, detailing LiDAR and encoder applications in autonomous navigation across varied terrains.

[6] Fossen, T. I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. Chichester, UK: Wiley. Includes advanced control strategies applicable to terrestrial robots, with detailed chapters on motion stability under load.

[7] Fox, D., Burgard, W., & Thrun, S. (2006). *Probabilistic Robotics*. Cambridge, MA: MIT Press. A seminal work on probabilistic methods in robotics, offering deep insights into odometry and mapping algorithms used in SLAM implementations.

[8] Groover, M. P. (2015). *Automation, Production Systems, and Computer-Integrated Manufacturing* (4th ed.). Boston, MA: Pearson. Explores automation principles, with detailed sections on power management and mechanical assembly for robotic systems.

[9] Horn, B. K. P. (1986). *Robot Vision*. Cambridge, MA: MIT Press. A classic text on computer vision, providing foundational knowledge for cube detection and environmental analysis using camera systems.

[10] Huang, H. M., & Messina, E. (2005). *Robotics and Automation Handbook*. Boca Raton, FL: CRC Press. A comprehensive guide covering robotic design, control, and testing methodologies applicable to AGV development.

[11] Kelly, A. (2013). *Mobile Robotics: Mathematics, Models, and Methods*. Cambridge, UK: Cambridge University Press. Detailed mathematical models for mobile robot navigation, including skid-steer kinematics and sensor data processing.

[12] Kozak, M., & Gordon, B. (2019). Skid Steering of Robotic Vehicle for Autonomous Applications. In *Proceedings of the IEEE International Conference on Automation Science and Engineering*, 1-6. Vancouver, Canada: IEEE. Analyzes skid-steer control strategies with practical data on torque and turning efficiency.

[13] Latombe, J. C. (1991). *Robot Motion Planning*. Boston, MA: Springer. A detailed exploration of path planning algorithms, including A* and local planners, relevant to Nav2 implementation.