Nomad Technical Paper
Robotics Dojo Competition 2026

Nathan Kaelo, Mark Kitur, Muthomi Kioni, Emmanuel Wambua, Reinhard Bitok

Abstract
This paper presents the design and development of a four-wheeled autonomous mobile robot developed for the Robotics Dojo Competition 2026. The platform was designed for waste detection, collection, and discharge in an environment containing uneven surfaces and obstacles. It's hardware architecture combines four 12 V, 110 RPM JGB37-520 encoder motors, 85 mm off-road wheels, an ESP 32 controller, and a Raspberry Pi 5 running ROS 2. A two-dimensional RPLIDAR A1 provides range measurements for mapping, localization, and navigation, while a Raspberry Pi camera supports vision-based waste detection using a YOLO-based model with ONNX inference. The software architecture uses SLAM Toolbox for map generation and the ROS 2 Navigation2 framework, with NavFn as the global planner and Model Predictive Path Integral (MPPI) control for local trajectory selection. A rack-and-pinion mechanism driven by an N20 geared motor was designed to discharge collected objects. The development process included computer-aided design, subsystem integration, Gazebo simulation and RViz-based visualization. The paper documents the design rationale, principal calculations, software architecture, and validation approach.

I. Introduction

Autonomous Mobile Robots (AMRs) have emerged as a cornerstone of modern automation, redefining how complex logistics, material handling, and urban sanitation operations are executed. Unlike traditional fixed-infrastructure systems, AMRs leverage advanced perception, dynamic path planning, and onboard intelligence to navigate unguided, unstructured environments flexibly, safely, and efficiently.

Waste management facilities, recycling plants, and urban recovery zones present severe operational hurdles, characterized by challenging and diverse outdoor terrains—including gravel, sand, and grass patches—along with dynamic obstacles such as moving personnel and shifting debris. To operate effectively under these demanding conditions, this project introduces an advanced, ruggedized Autonomous Mobile Robot engineered for intelligent waste sorting and collection.

Powered by a Raspberry Pi 5 running ROS 2 and the Nav2 navigation stack, the robot combines robust hardware actuation with intelligent edge perception. It features a high-torque drivetrain driven by 12V 110 RPM JGB37-520 encoder motors coupled with 85mm diameter off-road tyres to ensure superior traction across uneven ground. For spatial awareness, the system integrates a 2D LiDAR for precise, real-time localization alongside a Raspberry Pi camera paired with an integrated AI vision model for automated waste classification from visual imagery. Once waste is identified and collected into its onboard loading container, the robot utilizes a motorized rack-and-pinion linear actuation mechanism driving a heavy-duty discharge plunger to execute a reliable, automated offloading sequence. Furthermore, the robot employs lidar-based SLAM to construct a 2D occupancy map of the environment, enabling accurate localization and autonomous navigation through complex and unstructured waste-handling landscapes.

A. Design Strategy

Our approach focuses on balancing high-level functional capability with field-proven robustness. Since waste management in real-world or unstructured outdoor environments introduces severe unpredictability such as shifting terrain and dynamic pedestrian traffic our engineering choices are guided by a philosophy of modularity, mechanical resilience, and calculated tradeoff management.

1) Approach to Competition Challenges

To successfully navigate and execute waste classification and collection, our strategy relies on separating complex perception tasks from core mobility functions. Rather than attempting a computationally heavy, fully integrated SLAM navigation stack on resource-constrained hardware, we distribute the workload intelligently:

Perception & AI: A Raspberry Pi camera runs an optimized edge-AI classification model to instantly identify waste types from visual inputs, separating classification logic from spatial pathing.

Navigation & Localization: We utilize a 2D LiDAR integrated with the ROS 2 Nav2 framework for real-time localization, path planning, and obstacle avoidance. LiDAR-based Simultaneous Localization and Mapping (SLAM) is employed to generate an occupancy map of the environment, which is subsequently used by the localization and navigation system for autonomous operation.

Mobility: To conquer unpredictable course conditions ranging from loose gravel and sand to turf and grass, we employ 12V 110 RPM JGB37-520 encoder motors coupled with 85mm off-road tyres, ensuring maximum traction and consistent odometry feedback.

2) Reliability vs. Complexity

The design prioritizes mechanical simplicity and operational reliability in demanding field conditions. This is reflected in the waste-discharge mechanism, where a rack-and-pinion linear actuator drives a heavy-duty plunger to eject collected waste. Compared with more complex multi-joint manipulators or gravity-tilt bins, this arrangement has fewer moving components and a simpler actuation process, reducing potential points of mechanical failure and the risk of jamming when handling bulky debris. Similarly, 2D LiDAR was selected as the primary sensor for navigation and obstacle detection. Its use provides consistent distance measurements without dependence on ambient lighting conditions, supporting reliable obstacle detection and real-time navigation in outdoor environments.

3) Capability vs. Robustness

We prioritized the reliability and robustness of the core autonomous functions over the implementation of more advanced features, such as real-time multi-robot fleet coordination and complex predictive routing. This approach allowed the design to remain focused on the essential operational requirements of a single autonomous unit, with particular emphasis on mechanical durability, drivetrain performance, and software stability. Considerable attention was therefore given to strengthening the chassis, optimizing wheel-ground contact for uneven terrain, and ensuring reliable communication between ROS 2 components. The use of durable acrylic structural components and buck-regulated power distribution further supported the design objective of maintaining mechanical integrity and stable electronic operation under the demanding conditions associated with outdoor environments.

B. Vehicle Design

While component specifications and competition guidelines provide the physical foundation of our Autonomous Mobile Robot, our engineering approach emphasizes rigorous design processes, iterative prototyping lessons, and systematic software frameworks. This section documents our methodology, key design evolution, and algorithmic strategies to serve as both technical justification for our Robotics Dojo entry and a knowledge-transfer repository for future teams.

DESIGN REQUIREMENTS AND INITIAL ASSUMPTIONS

1. Maximum robot dimensions are 300 x 300x 300mm
2. Unloaded robot is assumed to be around 3kg.
3. Operating terrain would have sand, gravel, wet grass and a 7-degree incline ramp.
4. Power supply not exceeding 24V.

1) Design Processes, Methodologies, and Component Selection

Our engineering workflow followed a structured, systems-engineering methodology, breaking the platform down into three interdependent subsystems: Mechanical, electrical and software. Rather than selecting components based solely on theoretical datasheets, we utilized a top-down requirements-driven design process coupled with physical prototyping:

Mobility & Drivetrain Selection: To conquer variable outdoor terrain (gravel, sand, grass), we evaluated several motor-wheel configurations. Initial calculations for stall torque and payload capacity led us to select **12V 110 RPM JGB37-520 DC encoder motors**.



JUSTIFICATION: to estimate the required motor torque, the 3kg unloaded mass and the maximum 7° ramp are considered. The force required to overcome gravity on the ramp is:

Fg=mgsin(70)=29.43sin(70)=**3.5866N**

Rolling resistance is also considered.Using a conservative preliminary rolling resistance coefficient of Crr=0.15 for mixed terrain:

Frr=Crrmgcos(7)=0.1529.43cos(7)=**4.3816N**

Total tractive force is therefore:

Ttotal=Fg+Frr=**7.9702N**

Total wheel torque required is:

Ttotal=Ftotalr=**0.3387Nm**

Since two motors are used:

Tmotor=0.33872=0.16935Nm 0.170Nm

Applying a factor of two as a preliminary design margin give a target torque of approximately 0.34Nm/motor.

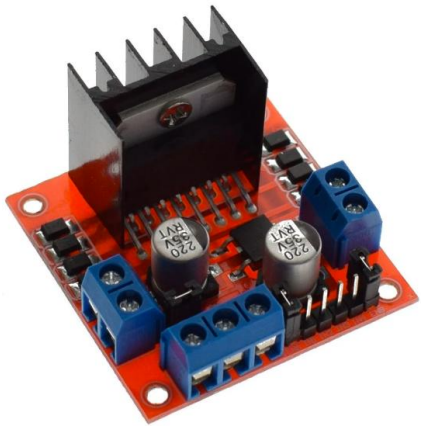
The selected motor is specified at approximately 10kg.cm of torque, which is approximately 0. 981Nm.Therefore the ratio between stated motor torque and design requirement is:

0.9810.34=2.885 which is approximately 2.9.

Hence, motors have a safety factor in case of overloading. Their integrated planetary gearboxes provide high reduction ratios necessary for low-speed high-torque maneuvering, while the built-in quadrature encoders deliver high-

resolution odometry feedback critical for precise dead reckoning.

MOTOR DRIVERS
Dual channel L298N H-bridge motor diver.
Two motors on each side of the robot are connected to one motor driver.
Justification: the driver provides two independent motor channels and supports PWM control allowing the Arduino to control the motors.The motors have a stall current of 1A so the combined operating current would be 2A which is within the drivers 2A per channel current rating.

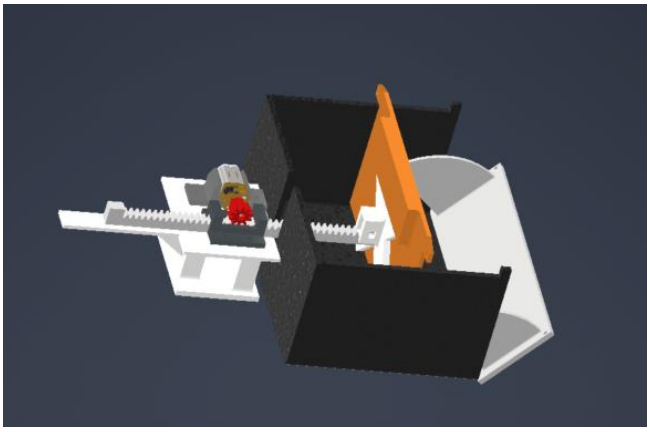


WHEELS
These were paired with **85mm diameter off-road tyres**, selected after researching various tread patterns to maximize contact patch friction on loose particulate ground.



JUSTIFICATION: the rubber wheels, wider wheel width and spiky surface allow for better traction on the terrain.

- **Structural & Actuation Iterations:** For the waste offloading mechanism, we researched on a simple and easy to integrate mechanism and after looking at a number of designs we decided to adopt a rack-and-pinion linear actuation mechanism driven by a high-torque N20 motor, coupled to a 3d printed reinforced discharge plunger. This converted rotational motor output into smooth, positive linear force, ensuring a jam-free offloading cycle regardless of payload mass.



JUSTIFICATION: the n20 motor is compact, lightweight.

Rack and Pinion Offloading Mechanism – Safe Load Calculations

1. **Design Parameters and Constants**

- Motor Stall Torque (T_{stall}) = 0.05 N·m
- Gearbox Efficiency (η) = 0.75 (75%)
- Pinion Pitch Radius (r) = 0.003 m (3 mm)
- Safety Factor (SF) = 3.0
- Coefficient of Sliding Friction (μ) = 0.25 (PLA on smooth surface)
- Gravitational Acceleration (g) = 9.81 m/s²

2. **Safe Working Torque (T_{safe})**
3. $T_{safe} = T_{stall} / SF$
4. $T_{safe} = 0.05$ N·m / 3.0 = 0.0167 N·m
5. **Safe Linear Pushing Force (F_{safe})**
6. $F_{safe} = (T_{safe} \times \eta) / r$
7. $F_{safe} = (0.0167$ N·m × 0.75) / 0.003 m = 4.17 N
8. **Maximum Safe Payload Weight (W_{safe})**
   $W_{safe} = F_{safe} / (\mu \times g)$
9. $W_{safe} = 4.17$ N / (0.25 × 9.81 m/s²) = 1.70 kg (1,700 g)

Hence the motor rating is more than enough to push the 150g object representing the waste.

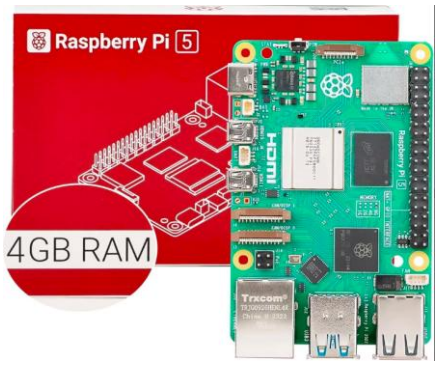
- **Compute & Sensing Integration:** The computing architecture separates deterministic motion control

from high-level perception. An ESP 32 microcontroller handles real-time motor control loops and low-level sensor polling, communicating via serial bus with a **Raspberry Pi 5**. For spatial awareness, a **2D RP LiDAR A1** provides robust, low-latency distance mapping for the Nav2 stack, while a **Raspberry Pi camera** feeds frames into an optimized edge-AI classification pipeline to identify waste categories on the fly.



ESP 32 JUSTIFICATION

The ESP32-based expansion board was selected as the robot's low-level controller due to its processing capabilities, integrated peripheral interfaces and suitability for real-time embedded control applications. Its programmable GPIO pins, PWM capabilities and hardware timer support make it suitable for interfacing with motor drivers, reading encoder feedback and managing other hardware components. Additionally, its compatibility with wired communication interfaces allows it to exchange control commands and feedback data with the Raspberry Pi 5, which handles the computationally intensive tasks associated with ROS 2 and autonomous navigation. This separation of responsibilities reduces the processing burden on the main computing platform while providing a dedicated controller for hardware-level operations. The ESP32 also offers a compact and cost-effective alternative to the Arduino Mega, making it suitable for the robot's embedded control architecture.



JUSTIFICATION: Using a Raspberry Pi 5 has a quad-core Cortex-A76 processor and custom RP1 I/O chip, which together provide the high-performance compute and bus bandwidth necessary to run a native ROS 2 navigation stack, process high-frequency LiDAR and camera data, and manage low-latency serial communications with lower-level microcontrollers without throttling.



JUSTIFICATION: THE Pi 5 has a custom RP1 southbridge, which provides dedicated, high-bandwidth serial and USB throughput to handle continuous 360-degree laser scan streams without packet drops or buffer overruns. Additionally, the Pi 5's powerful quad-core processor effortlessly handles the computational overhead of converting raw scan data into ROS 2 LaserScan messages and running real-time 2D mapping and localization.

Power: We used a 20V Total power tool battery which we found to be cheaper compared to the alternative lipo battery.



Challenge: it requires designing an adapter since they are not produced as we are not using it for its designed purpose.

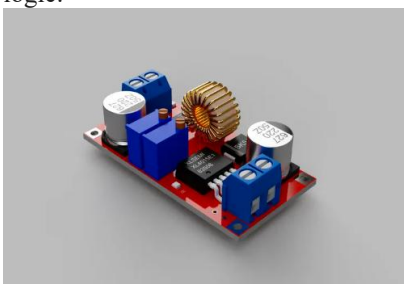
It also requires using buck converters to step down the high voltage to a safe value for the motors and the raspberry pi.

JUSTIFICATION: it has a high continuous discharge current capability, which easily handles heavy motor start-up transients and current spikes without severe voltage sag.

Additionally, its higher nominal voltage provides an efficient source for step-down buck converters to safely power both logic boards and high-torque actuators.

Power Distribution & Voltage Sags : During early integration testing, heavy motor current draws during sudden acceleration caused voltage sags on the main power rail, triggering reboots on the Raspberry Pi 5.

What didn't work: Relying on a single unconditioned battery tap for both actuators and logic. *The fix:* We redesigned the electrical architecture to incorporate dedicated high-efficiency **XL4015 buck converters** and isolated power distribution rails, completely separating noisy inductive motor loads from sensitive microcomputer logic.



JUSTIFICATION: it has the ability to efficiently step down the high, fluctuating voltage from the 20V power tool battery pack into stable, regulated operating voltages (such as clean 5V logic for the Raspberry Pi 5 and microcontrollers) with minimal thermal waste. Furthermore, their high switching efficiency and robust output current capacity (up to 5A) ensure they can handle sudden current spikes from the Pi 5, LiDAR, and sensors under heavy load without experiencing voltage sag or triggering unexpected brownouts.

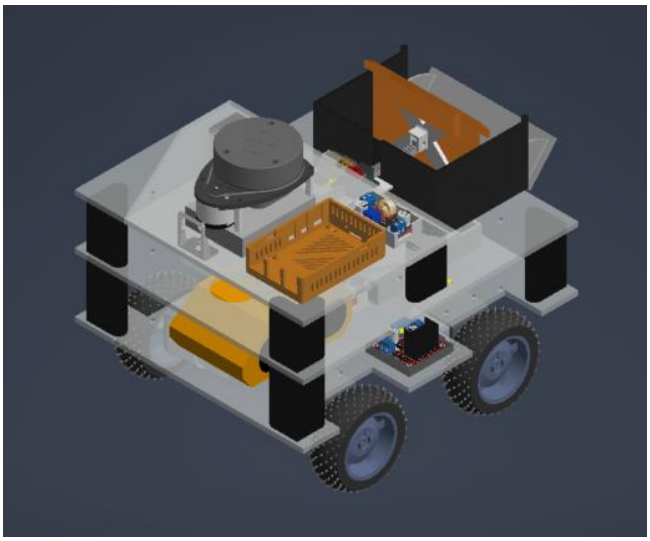
Motors: we set one buck converter to 12 v and routed this to the motors.
Raspberry pi: we set the other buck converter to 5.V which is the recommended power requirement.

The XL4015 buck converter has a maximum current rating of 5A which fits the Raspberry pi requirements and also for the motors.
4 motors with a stall current of 1A each.

Raspberry Pi Power requirement challenge: the raspberry pi power requirements being 5.1V/5A, the power bank options in the market had a maximum of 5V/3A hence our decision to go with the buck converter method.

Chassis Rigidity vs. Weight: We utilized 5mm thickness acrylic plates to form our chassis to support the components and mounted out motors to this chassis.

FINAL CAD DESIGN



2) Algorithmic Approaches and Software Architecture

A. ROS 2 architecture
The software is organized around ROS 2, with separate functions for sensor processing, mapping, localization, navigation, and low-level hardware control. This modular arrangement allows the navigation system to issue velocity commands while the embedded controller handles motor interfacing and returns feedback. The robot model is represented using a Unified Robot Description Format (URDF) description, supporting visualization and simulation of the platform and its sensors.

B. Mapping and localization
SLAM Toolbox is used during mapping to combine LiDAR observations and odometry into a two-dimensional occupancy-grid representation of the environment. After a map has been generated and saved, the robot can use the stored map for localization and navigation. Adaptive Monte Carlo Localization (AMCL) estimates the robot pose within the known map using LiDAR observations and the particle-filter localization approach.

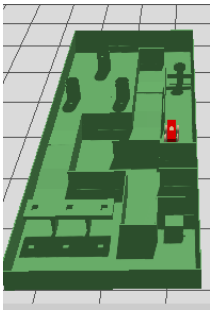
C. Global planning and local control
Autonomous navigation is implemented using the ROS 2 Navigation2 (Nav2) framework. The global planner is NavFn, which computes a path through the global costmap from the robot's estimated pose to the requested goal. The local motion controller is Model Predictive Path Integral (MPPI). MPPI evaluates sampled motion trajectories over a prediction horizon and selects control commands based on their associated costs, including the influence of obstacles and motion constraints represented in the controller configuration. The selected command is then sent to the robot's velocity interface.

D. Low-level hardware interface
A custom ROS 2 hardware interface communicates with the ESP 32 over a serial link. The interface sends left- and right-side wheel-velocity commands and receives wheel position and velocity feedback. The design notes also describe IMU orientation, angular velocity, and linear acceleration in the returned data. The wheel sets are exposed through a differential-drive control interface, with the front and rear wheels on each side grouped as the left and right sides of the platform.

E. Task-level perception and waste detection
A camera-based object-detection pipeline is used to identify waste categories. The documented implementation uses a YOLO-based model with ONNX inference. The perception subsystem is intended to distinguish categories relevant to the sorting task, including plastic, organic and metal waste. The perception output is intended to inform the task-level decision process, while navigation is responsible for moving the robot to the selected location.

F. Task level autonomous decision making
The high-level task execution is implemented using a behavior-tree architecture based on py_trees. The behavior tree provides a structured representation of the competition task and determines which navigation action should be executed based on the detected waste type and gate condition.

G. Simulation environment
A Gazebo simulation environment was developed to support software integration and pre-deployment testing. The simulated setup includes the robot model, competition field, and representations of relevant sensors and motion interfaces. RViz is used to inspect LiDAR data, map output, costmaps, and navigation behavior. Simulation provides a controlled environment for identifying model, frame, sensor, and navigation-configuration issues before testing on the physical platform.



Robot simulation.

3) Iterative Testing, Lessons Learned, and Design Failures

To ensure system reliability and validate performance under competitive conditions, our testing methodology followed a phased, rigorous progression from isolated component checks to full simulation and hardware integration.

1. Unit Testing
Individual subsystems were isolated and benchmarked to verify baseline performance before physical assembly:

Drivetrain & Actuation Testing: The 12V 110 RPM JGB37-520 encoder motors and motor drivers were evaluated across variable PWM duty cycles. Quadrature encoder feedback was logged to validate speed regulation, odometry tracking consistency, and stall-torque limits under load.

Sensor & Perception Validation: The RPLiDAR A1 sensor was calibrated in both static and dynamic environments. Range accuracy was verified by benchmarking LiDAR scan distances against known physical measurements and calibrated test obstacles.

Edge-AI Classification Benchmarks: The Raspberry Pi camera module and integrated AI model were tested independently under various ambient lighting conditions to ensure consistent inference latency and high classification accuracy for waste identification.

2. Integration Testing
Once subsystems were mounted onto the chassis, integration tests focused on inter-module communication, electrical stability, and closed-loop control:

Hardware-software synchronization: Validated seamless communication loops between the ESP 32, the Raspberry Pi 5 running ROS 2, and the Nav2 path-planning stack.

Power distribution & system stability: Tested battery drain profiles and voltage regulation under simultaneous high-load conditions (motor acceleration, LiDAR spinning, and AI inference) to ensure the isolated buck-converter architecture prevented system burnouts or unexpected thermal shutdowns.

Closed-loop motion control: Fine-tuned PID control loops using real-time quadrature encoder feedback to ensure precise heading maintenance, linear velocity tracking, and accurate dead reckoning.

3. Simulation & Validation

Prior to full field deployment, virtual testing environments were heavily leveraged to stress-test software logic, navigation algorithms, and sensor fusion pipelines:

Gazebo Physics Simulation: Utilized for testing physical interactions, robot dynamics on simulated rough surfaces, dynamic obstacle avoidance around human-representing mannequins, and collision detection thresholds.

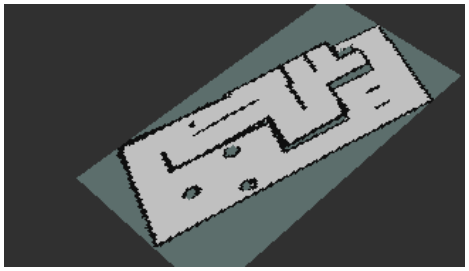
RViz Visualization: Deployed to monitor real-time LiDAR point-clouds, costmap layers, and local/global planner trajectories, allowing for rapid debugging of navigation behaviors before running live hardware trials.

Lessons Learnt:

1. Using accurate dimensions during the CAD designs.
2. Being detailed in the requirements of every step to prevent bottlenecks.
3. Thorough testing for all application cases and not assuming it will work since it worked in a similar case.

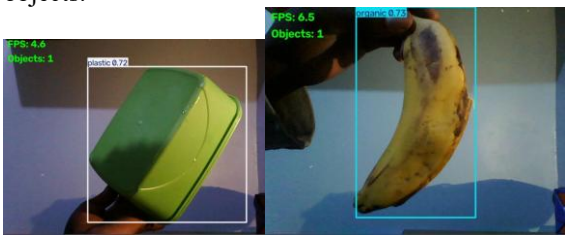
C. Experimental Results

SLAM and localization performance: The LiDAR-based SLAM system was evaluated by mapping the competition environment. The generated occupancy-grid map was inspected to determine whether the major environmental features, boundaries, and navigation paths were correctly represented.



Map generated while driving the robot in the gamefield.

Waste detection and classification: The vision-based waste detection system was evaluated using representative waste objects under practical operating conditions. Figure X presents an example detection obtained using the developed YOLO-based object detection pipeline with ONNX inference. The system successfully identified the presented objects.



Real time waste detection using YOLO based vision system.

D. Acknowledgements.

We sincerely thank Dedan Kimathi University of Technology (DeKUT) for its support, Jomo Kenyatta University of Agriculture and Technology (JKUAT) for hosting us and our project supervisor for their invaluable guidance, technical advice, and continuous support throughout the development of this project.

E.References

[1] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," Science Robotics, vol. 7, no. 66, p. eabm6074, 2022, doi: 10.1126/scirobotics.abm6074.

[2] S. Macenski, F. Martín, R. White, and J. G. Clavero, "The Marathon 2: A navigation system," in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), 2020, pp. 2718–2725, doi: 10.1109/IROS45743.2020.9341207.

[3] S. Macenski and I. Jambrecic, "SLAM Toolbox: SLAM for the dynamic world," Journal of Open Source Software, vol. 6, no. 61, p. 2783, 2021, doi: 10.21105/joss.02783.

[4] T. Moore and D. Stouch, "A generalized extended Kalman filter implementation for the Robot Operating System," in Proc. IEEE Int. Conf. Intell. Autom. Syst., 2014.