Autonomous Environmental Mapping and Navigation System for Unmanned Ground Vehicles

Emilio Antoine Altamirano Mendoza, Cristobal Gallardo, Matheus Henrique Dias Cirillo, Smit Nareshbhai Kotadiya, Keshav Kumar, Sumit Mor, Aline Cynthia Yiagnigni Pokarou, and Gustavo Moura Scarenci de Carvalho Ferreira

Exchange and International Students, Department of Computer Science, University of Applied Sciences Hof, Hof, Germany

Abstract—This paper details the development of an autonomous environmental mapping system for an unmanned ground vehicle (UGV). Utilizing an RS Helios 16P LiDAR, visual sensors, and an Nvidia Jetson Orin Nano, the system enables autonomous traversal and path optimization. This report covers the hardware integration, organizational workflow, and specific engineering solutions for custom component fabrication.

Keywords—UGV, ROS2, LiDAR, SLAM, Jetson Orin Nano, 3D Printing.

I. INTRODUCTION The objective of this project is to develop a comprehensive environmental mapping system that enables an unmanned ground vehicle (UGV) to autonomously traverse terrain along an optimal path. This is accomplished through the integration of advanced sensing and computing technologies.

II. HARDWARE AND SOFTWARE COMPONENTS The system utilizes the following components: an AgileX Scout Mini rover as the mobile platform, an RS Helios 16P LIDAR for environmental perception, an ELB camera for visual data acquisition, and a Jetson Orin Nano as the onboard processing unit. The software stack comprises Ubuntu 22.04 with GUI, ROS2 Humble Hawksbill, Gazebo Fortress for simulation, and RViz for visualization.

III. PROJECT ORGANIZATION AND EXECUTION The team of eight members was organized into three specialized subgroups: the Design team, responsible for developing and fabricating hardware mounts for integration onto the UGV; the Hardware team, tasked with configuring and integrating the Jetson Orin Nano, Scout Mini, LIDAR, and camera systems; and the Simulation team, which developed a comprehensive testing environment using Gazebo, ROS2, and RViz.

IV. TECHNICAL CONTRIBUTIONS: HARDWARE-DESIGN *Emilio Antoine Altamirano Mendoza*

A. 3D Component Mounts The integration of diverse sensors onto the Scout Mini platform required custom structural solutions.

1) Challenge 1: Material Selection and Feasibility: Issue: Determining whether 3D printing was viable for component mounting or if alternative methods should be pursued. Solution: In collaboration with team member Cristobal Gallardo, analysis determined that 3D printing costs for small rover components would be minimal financially and material-wise. Previous experience with 3D printing confirmed this approach would simplify mount design and fabrication.

2) Challenge 2: Fabrication Resources: Issue: Locating suitable 3D printing facilities in the Hof area. Solution: University inquiry led to the Hof University Makerspace. René Göhring, facility supervisor, confirmed file submission requirements and provided guidance during initial prints, including the camera mount and Nvidia Jetson Orin Nano Board holders.

3) Challenge 3: LiDAR Mounting Design: Issue: Following the transition to RS Helios 16P LiDAR, a mounting solution was required for the large unit (152.5 mm diameter) with limited mounting points. The LiDAR featured only bottom-mounted screw holes (3 mm wide, 5 mm deep) without obstructing the 360° scanning field. A 10° inclination angle was specified by other teams for testing requirements. Solution: Following extensive research and iterative design development, the final mount (Fig. 1a, 1b) utilizes a two-piece configuration: the LiDAR secures to a ring component, which mounts via screws to a tilted base providing the required 10° inclination.

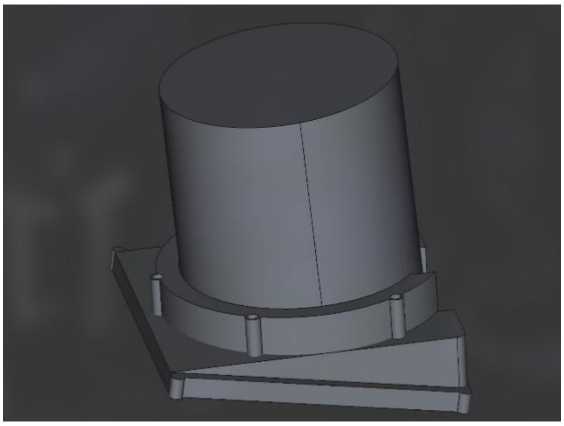

Fig 1a. Isometric view of the final iteration of the LiDAR mount. Fig 1b. Exploded view of the final iteration.

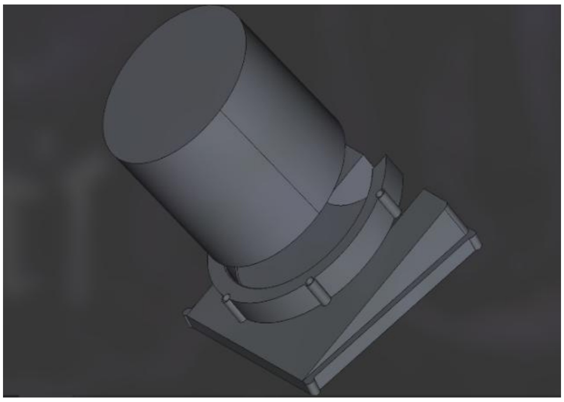

Fig 1b. Exploded view of the final iteration of the LiDAR mount

4) Challenge 4: Production Timeline Constraints: Issue: LiDAR mount design completion exceeded anticipated timeline. Upon submission to the Makerspace, all printers were occupied with estimated completion one week beyond the final presentation deadline. Solution: An expedited alternative design was developed using fundamental trigonometry. Given the 10° inclination requirement and known LiDAR dimensions (13.5 cm radius), calculations determined the necessary vertical offset:

tan θ = oc/ac → oc = ac × tan θ = 13.5 × tan (10°) = 2.38 ≈ 2.4 cm sin θ = oc/h → h = oc/sin θ = 2.4/sin (10°) = 13.82 cm

These calculations informed the simplified design (Fig. 2a, 2b), which was successfully fabricated (Fig. 2c) within the project timeline and integrated with all other components for operational testing.

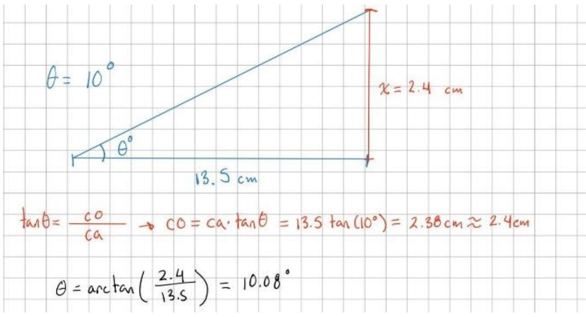

Fig 2a. Initial calculation made using the Pythagoras theorem.

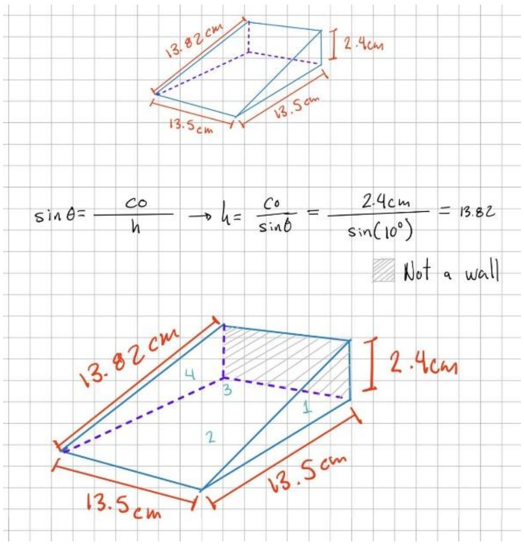

Fig 2b. Final design with measures

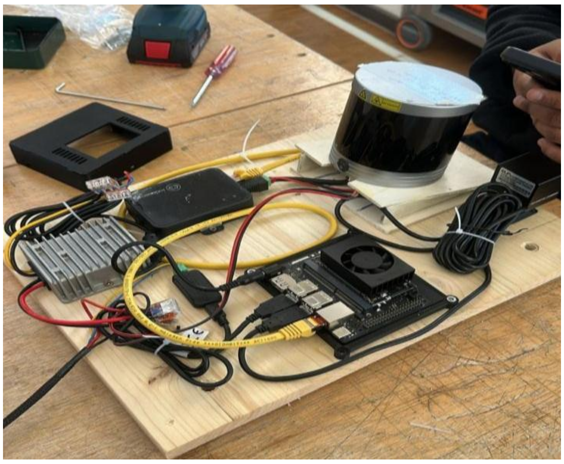

Fig 2c. Final mount made out of plywood.

V. MECHANICAL ASSEMBLY AND STRUCTURAL INTEGRATION *Cristobal Gallardo*

The physical construction of the UGV required a balance between rapid prototyping and long-term structural reliability. This section details the mechanical integration of the system components.

A. Hardware Assembly and Structural Integration *1) Issue:* Robot construction required optimization under significant time constraints and limited 3D printing equipment availability. *2) Solution:* A hybrid manufacturing workflow was implemented, dividing production between 3D printing and alternative fabrication methods. This ensured deadline compliance while maintaining the structural integrity required for field testing.

B. Strategic Manufacturing and Material Selection *1) Issue:* Limited 3D printer availability required prioritization of component manufacturing and identification of alternative materials for non-critical structures. *2) Solution:* 3D printing was reserved exclusively for components requiring complex geometries and high-precision sensor alignment (e.g., LiDAR and camera mounts). Hand-cut wood was selected for the remaining structural elements, providing a lightweight, easily modifiable, and durable alternative for mounting plates. This choice significantly accelerated the physical assembly phase.

C. Infrastructure and Tooling *1) Issue:* Professional-grade assembly required access to industrial tools and specialized fasteners to achieve high mechanical reliability. *2) Solution:* The Hof University MakerSpace provided comprehensive professional tooling, including industrial drills and precision fasteners. This infrastructure enabled the transformation of raw materials into a functional chassis with verified mechanical reliability and secure connections throughout.

D. Aluminium Rail Integration *1) Issue:* The robot base required secure integration of custom wooden components with the existing aluminium extrusion rails while maintaining platform rigidity for electronic component support. *2) Solution:* The modular aluminium rail architecture was leveraged using T-slot nuts and precision bolts. This method integrated the wooden base with the metal frame, creating a rigid platform capable of supporting high-density electronics without the risk of structural failure or component damage.

E. Component Distribution and Spatial Optimization *1) Issue:* Compact chassis dimensions necessitated precise spatial analysis for the placement of the Jetson board, LiDAR, cameras, and power systems. *2) Solution:* Detailed component mapping ensured all high-volume hardware and wiring fit within the chassis limits. The design prioritized thermal management for the Jetson Orin Nano, preservation of the LiDAR's 360° field of view, and accessibility to debugging ports. This high-density integration maximized available space while maintaining system balance and cooling capability.

VI. KEY OUTCOMES OF HARDWARE INTEGRATION

Hardware assembly encountered fewer technical obstacles than software integration but required immediate practical solutions for each mechanical challenge. Clear design logic and effective laboratory resource utilization enabled efficient problem resolution without major setbacks, ensuring on-time robot completion.

The project demonstrated that successful engineering requires adaptability beyond initial CAD designs, balancing technical requirements with real-world constraints through hybrid material strategies and creative spatial optimization.

This section details the hardware-level system integration, focusing on remote communication and sensor interfacing.

VII. SYSTEM INTEGRATION AND REMOTE OPERATION

Matheus Henrique Dias Cirillo

A. Rover Validation - Validation of the rover began with RF control testing. The controller required specific switch positions for the system to engage. The available documentation covered a different controller version and did not explain the specific model in use. Through trial and error, the correct switch configuration was identified, and control over the rover was successfully established.

B. Remote Operation - Initial remote setup encountered an operating system compatibility issue. Ubuntu 24.04 was originally installed on the Raspberry Pi, but it had to be downgraded to Ubuntu 22.04 to support ROS2 Humble. Additionally, the provided camera lacked a lens, rendering it unusable. While the CAN connection was established quickly using the official AgileX repository [1], the provided ROS2 node failed to compile. The official repository was identified as a fork, prompting a switch to the more updated, although not official, Weston Robot repository [2], which allowed the rover to move successfully. Connectivity challenges arose because the lab lacked a dedicated Wi-Fi network for SSH hostname broadcasting. This was resolved by creating a mobile hotspot and identifying IP addresses via a GUI. Finally, powering the Raspberry Pi proved problematic because the rover output 20V and the available step-down converter only reached 12V, limiting the system to tethered operation near an outlet.

It was suggested to replace the Raspberry Pi with an Nvidia Jetson Xavier to handle LiDAR processing. It soon became apparent that the Jetson was a full computer capable of replacing the Raspberry Pi entirely. The latest supported Jetpack version was based on Ubuntu 18.04, which is incompatible with ROS2 Humble, so an unofficial update to Ubuntu 22.04 was performed. This removed the system GUI and some official Nvidia features but allowed for rover control. To address the lack of video support over SSH, Motion was configured to transmit a video feed, achieving fully remote operation.

C. Full System Integration (LiDAR & Orin Nano) - Attempts were made to use a Leishen C16 LiDAR. While *tcpdump* verified data transmission, the ROS2 node received no data. Investigation revealed the driver was likely built for ROS2 Foxy. Debugging on the GUI-less Xavier was inefficient, so a newer Jetson Orin Nano running native Ubuntu 22.04 was sourced. The system was switched to an RS Helios 16p LiDAR and a USB Fisheye camera. The camera worked immediately with motion, but CAN communication failed due to missing kernel drivers. The kernel was rebuilt to enable proper CAN support. Finally, to avoid manually launching multiple nodes, a bash script was created that accepts commands like "rover <base|lidar|camera|teleop>" to automate the entire startup process. This resulted in a fully remote-operated rover with active camera and LiDAR feeds.

VIII. LIDAR INTEGRATION AND SENSOR CONFIGURATION *Smit Nareshbhai Kotadiya*

The integration of the primary perception sensor involved extensive network diagnostics and kernel-level modifications to ensure stable data acquisition for the SLAM algorithms.

A. Leishen LiDAR Data Transmission Failure *1) Issue:* The Leishen C16 LiDAR failed to transmit point cloud data on the Jetson Xavier (Ubuntu 22 CLI) despite exhaustive diagnostic efforts. *2) Attempted Solutions:* Procedures included default IP verification, SDK implementation, and network packet scanning for UDP/TCP traffic. Diagnostics via *curl* for admin panel access and ROSbag recording were also attempted to identify the *frame_id*. *3) Resolution:* The root cause remained unidentified, with a suspected requirement for trigger-based transmission. Consequently, the hardware was replaced with the RS Helios 16P LiDAR to ensure project continuity.

B. RS Helios LiDAR Network Configuration *1) Issue:* The RS Helios 16P initially failed to transmit data to the processing unit despite standard IP configuration and SDK deployment. *2) Solution:* Wireshark packet analysis was employed to monitor the network interface, revealing that the default factory IP address had been modified. Once the correct network parameters were identified through traffic analysis, the LiDAR was reconfigured and communication was established.

C. Point Cloud Coverage Optimization *1) Issue:* Initial data transmission from the LiDAR resulted in incomplete point cloud coverage, failing to provide the required 360° situational awareness. *2) Solution:* The system was migrated to the Jetson Orin Nano (Ubuntu 22 GUI) to utilize advanced debugging tools. Browser-based access to the LiDAR's local admin panel revealed a restricted horizontal FOV setting. Restoring the configuration to a full 360° sweep resolved the coverage issues.

D. Jetson Orin Kernel Integrity *1) Issue:* The Jetson Orin Nano kernel lacked the native CAN USB drivers necessary for communication with the Scout Mini base. *2) Resolution:* A complete kernel rebuild was performed. This low-level modification restored the required driver functionality, enabling the ROS2 stack to interface directly with the UGV chassis via the CAN bus.

E. SLAM Integration and Mapping *1) Issue:* Implementing Simultaneous Localization and Mapping (SLAM) and generating accurate environment maps presented significant integration challenges. *2) Attempted Solutions:* In collaboration with the simulation team, efforts focused on adjusting configuration parameters, verifying URDF/DAE file naming conventions, and modifying the *scout_description* and *master_launch* files. *3) Current Status:* All hardware components are currently operational. The system integration is nearing completion; however, while SLAM successfully generates maps, they currently exhibit suboptimal accuracy. Further refinement of the sensor fusion parameters is required to improve mapping precision.

IX. AUTONOMOUS ROVER DEVELOPMENT: SIMULATION AND NAVIGATION

Keshav Kumar, Sumit Mor, Aline Cynthia Yiagnigni Pokarou, Gustavo Moura Scarenci de Carvalho Ferreira

The simulation phase was the primary laboratory for validating high-level autonomy logic. Given the complexity of the AgileX Scout Mini and the RS Helios 16P LiDAR, the team adopted a "Simulation-to-Reality" (Sim2Real) methodology. This ensured that every software component—from the low-level motor controllers to the high-level path planners—was stress-tested in a mathematically deterministic environment before deployment on the physical chassis.

A. Simulation Environment Design

The team developed a comprehensive digital twin of the operational environment to mirror the geometric and physical constraints of indoor navigational spaces. The idea of the structure of the simulation environment and files like URDF and meshes were taken from official agileX simulation repo for scout_Mini on GitHub. These files were modified later for improvement and for some error corrections

1. **URDF and Xacro Modeling:** The rover's physical properties were defined using the **Unified Robot Description Format (URDF)**. This allowed the team to adjust sensor offsets (LiDAR height, camera tilt) in a single configuration file that updated both the visual model and the transform tree simultaneously.

2. **Inertial and Visual Geometry:** Each link—including the chassis, four drive wheels, and sensor mounts—was assigned precise inertial properties (mass, center of gravity, inertia tensor). The values were mostly taken from the official AgileX repo for scout mini on the GitHub. This was critical for Gazebo's ODE (Open Dynamics Engine) to accurately calculate the torque required for rotation and the friction generated during acceleration on different virtual surfaces.

3. **SDF World Generation:** The simulation world was constructed using **Simulation Description Format (SDF)**. The world included diverse obstacle types, ranging from primitive shapes (cubes, cylinders) to complex meshes. These obstacles were strategically placed to test the LiDAR's ability to resolve narrow corridors and handle "blind spots" created by the rover's own structural pillars.

B. Technical Hurdles in simulation design

These are some of the most significant technical hurdles which were faced during the design of the simulation environment.

The Problem: Initial tests showed that when the rover performed a 360 turn, the LaserScan data in RViz would "smear." This indicated that the system's odometry believed the robot had turned more (or less) than it actually had in the physics engine.
Steps taken to resolve the error:

a. **Parameter Tuning:** The team performed a deep dive into the **libgazebo_ros_diff_drive** plugin. The investigation identified that the default wheel separation parameter did not account for the "effective" track width caused by the Scout Mini's wide tires.
b. **The Empirical Solution:** The wheel separation value was incrementally adjusted and validated through repeated testing. This correction significantly improved alignment between simulated motion and visualization, resulting in accurate LiDAR scans and consistent SLAM performance.

The Problem: Missing LiDAR Plugin.

Main Issue: Initial simulation models lacked a functional LiDAR sensor, preventing SLAM and perception testing.

Steps taken to resolve the error: The team integrated the **Ray Sensor** plugin into the URDF file to emulate the lidar. The plugin was configured to match the hardware's specific horizontal resolution, vertical channels (16), and scanning frequency which was later changed as the Slam Toolbox require 2D scan.

The Problem: Autonomous Exploration Without Manual Teleoperation (during Mapping)
Steps taken to resolve the error: A custom autonomous exploration node was developed using a Nav2 Action Client. The node periodically generated random goal positions within a defined range, enabling stochastic exploration of unknown areas. This approach allowed the SLAM algorithm to progressively convert unknown space into known free space.

The Problem: Obstacle Detection and Safety in Simulation (during Mapping)
Steps taken to resolve the error: LiDAR scan data was divided into front, left, and right sectors. Safety thresholds were defined to intercept velocity commands when obstacles were detected within a critical distance. Additional escape logic was implemented to recover from situations where the rover became stuck in confined areas.

The Problem: Exploration in Unknown Environments (during Mapping)
Steps taken to resolve the error: Dynamic goal generation was employed to allow the rover to explore unknown space incrementally. By continuously selecting new goal positions, the rover was able to expand the known map without relying on pre-existing global information.

The Problem: Runaway Robot on Program Termination (during Mapping)
Steps taken to resolve the error: Signal handling mechanisms were implemented to detect shutdown events. A safe shutdown routine repeatedly published zero-velocity commands to ensure the rover reliably stopped upon program termination.

H. Simulation Outcomes and Sim2Real Transition

The final simulation trials resulted in a robust, self-sufficient system. The UGV successfully demonstrated **Fully autonomous mapping** of a complex indoor environment and **Dynamic pathfinding** that bypassed both static and simulated dynamic obstacles.

The code developed within this 4-page equivalent of simulation work was packaged into a modular ROS2 workspace. Because the simulation used the exact same topic names (/scan, /cmd_vel, /odom) and message types as the physical Scout Mini and RS Helios LiDAR, the transition to hardware was "plug-and-play," significantly reducing the time required for final on-site integration.

X. INTEGRATION AND FINAL RESULTS

A. System Integration Overview System integration involved the unification of mechanical assemblies, hardware components, and software modules into a single operational entity. The development followed a V-model approach: individual subsystems—including custom sensor mounts, LiDAR drivers, camera interfaces, SLAM algorithms, and navigation logic—were first validated independently. Upon achieving subsystem stability, they were integrated via a centralized ROS2 launch configuration to coordinate perception, mapping, and motion control.

B. Integrated Launch and Runtime Operation To ensure operational reliability, a master launch file was developed to initialize all essential ROS2 nodes simultaneously. This included the AgileX Scout Mini base driver, the RS Helios 16P LiDAR driver, the SLAM Toolbox, and supporting visualization nodes. This centralized architecture reduced configuration errors and ensured consistent startup behavior across both the simulation environment and the physical deployment.

C. Results in Simulation in the Gazebo environment, the UGV successfully demonstrated fully autonomous exploration and mapping.

- **Mapping:** The SLAM Toolbox reliably generated high-resolution 2D occupancy grids.
- **Navigation:** The custom autonomous exploration logic enabled dynamic traversal of unknown virtual environments.
- **Reliability:** Obstacle avoidance and emergency safety routines functioned as intended, providing stable and repeatable performance throughout extended test cycles.

D. Results on the Physical Rover On the physical hardware, SLAM-based mapping was successfully achieved using real-time data from the RS Helios 16P LiDAR. The UGV generated accurate environmental maps while under remote supervision. However, while the Nav2 stack was fully validated in simulation, its deployment on the physical hardware remained a challenge and was not fully integrated within the project timeframe.

E. Functional Capabilities Achieved By the conclusion of the project, the system successfully realized the following:

- Reliable high-bandwidth acquisition of LiDAR and camera data.
- Real-time SLAM-based mapping in both simulation and physical environments.
- Autonomous exploration and navigation behaviours validated in simulation.
- Safe motion control with integrated obstacle-aware hardware interlocks.

F. Limitations and Partial Implementations Despite the successful integration of core functionalities, full autonomous navigation with dynamic path planning on the physical rover remains a pending objective. Furthermore, additional sensor fusion tuning is required to improve map accuracy and navigation robustness in complex, non-structured real-world environments. These limitations are attributed to the high complexity of real-time hardware integration and project time constraints rather than fundamental architectural flaws.

XI. SYSTEM-LEVEL DISCUSSION

A. Key Technical Challenges The project highlighted the inherent complexity of integrating mechanical design, embedded hardware, and autonomous software. Hardware-related hurdles—specifically LiDAR driver compatibility and Jetson Orin Nano kernel configuration—required extensive troubleshooting. Parallelly, simulation challenges regarding sensor synchronization and kinematic stability demanded precise parameter tuning. These experiences underscored the necessity of systematic debugging and modular system design.

B. Design Decisions and Trade-offs A strategic decision was made to prioritize the reliability of SLAM-based mapping over the deployment of full autonomous navigation on the physical rover. This trade-off ensured a high-quality mapping performance, acknowledging the significant overhead required for real-world Nav2 deployment. Additionally, the decision to replace the initial Leishen C16 LiDAR with the RS Helios 16P significantly improved system reliability, albeit at the cost of additional integration time.

C. Lessons Learned and Reflection The project provided several critical insights:

1. **Early Validation:** Simulation-first development significantly mitigates hardware integration risks.
2. **Compatibility:** Documentation quality and hardware-software compatibility are primary drivers of development efficiency.
3. **Safety:** Multi-level safety mechanisms are mandatory for reliable autonomous behavior.
4. **Collaboration:** Specialized subgroups (Design, Hardware, Simulation) enabled efficient parallel progress, while regular inter-group communication was essential to align multidisciplinary decisions.

XII. CONCLUSION The project successfully developed a functional UGV platform capable of environmental mapping. The simulation environment demonstrated the full scope of autonomous navigation, including the implementation of the SLAM Toolbox and the Nav2 navigation stack. While the physical implementation achieved complete SLAM functionality and remote-operated mapping, the transition of the Nav2 stack to the physical system remains a goal for future development. These results fulfil the core objective of creating a robust, perception-aware environmental mapping system.

REFERENCES

[1] AgileX, "ROS2 Packages for Scout Mobile Robot," [Online]. Available: https://github.com/agilexrobotics/scout_ros2.

[2] Weston Robots, "ROS2 Packages for Scout Mobile Robot," [Online]. Available: https://github.com/westonrobot/scout_ros2.

[3] AgileX, "Scout Mini User Manual," 2023. [Online]. Available: https://global.agilex.ai/.

[4] RoboSense, "RS-Helios-16P User Guide," 2023. [Online]. Available: https://www.robosense.ai.

[5] Open Robotics, "ROS 2 Humble Hawksbill Documentation," 2024. [Online]. Available: https://docs.ros.org/en/humble/.

[6] S. Macenski and et al, "SLAM Toolbox for Real-Time Simultaneous Localization and Mapping," *Journal of Open Source Software,* 2021.

[7] Nvidia, "Jetson Orin Nano Developer Kit User Guide," 2023. [Online]. Available: https://developer.nvidia.com/embedded/jetson-orin-nano .