ICWMC 2026

The Twenty-Second International Conference on Wireless and Mobile Communications

ISBN: 978-1-68558-347-7

March 8th –12th, 2026

Valencia, Spain

ICWMC 2026 Editors

Petre Dini, IARIA, USA

ICWMC 2026

Forward

The Twenty-Second International Conference on Wireless and Mobile Communications (ICWMC 2026), held between March 8-th, 2026 and March 12-th, 2026 in Valencia, Spain, continued a series of events on advanced wireless technologies, wireless networking, and wireless applications.

The event addressed wireless related topics concerning integration of latest technological advances to realize mobile and ubiquitous service environments for advanced applications and services in wireless networks. Mobility and wireless, special services, and lessons learnt from particular deployment complemented the traditional wireless topics.

We take here the opportunity to warmly thank all the members of the ICWMC 2026 technical program committee, as well as all the reviewers. The creation of such a high-quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to ICWMC 2026. We truly believe that, thanks to all these efforts, the final conference program consisted of top-quality contributions. We also thank the members of the ICWMC 2026 organizing committee for their help in handling the logistics of this event.

We hope that ICWMC 2026 was a successful international forum for the exchange of ideas and results between academia and industry for the promotion of progress in the field of wireless and mobile communications.

ICWMC 2026 Chairs

ICWMC 2026 Steering Committee
Dragana Krstic, University of Niš, Serbia
Rajat Kumar Kochhar, Ericsson, Sweden
Magnus Jonsson, Halmstad University, Sweden
Sonia Ben Rejeb, Higher Institute of Computer Science (ISI), University of Tunis El Manar (UTM), Tunisia

ICWMC 2026 Publicity Chairs
Francisco Javier Díaz Blasco, Universitat Politècnica de València, Spain
Ali Ahmad, Universitat Politècnica de València, Spain
José Miguel Jiménez, Universitat Politècnica de València, Spain
Sandra Viciano Tudela, Universitat Politècnica de València, Spain

ICWMC 2026
Committee

ICWMC 2026 Steering Committee

Dragana Krstic, University of Niš, Serbia
Rajat Kumar Kochhar, Ericsson, Sweden
Magnus Jonsson, Halmstad University, Sweden
Sonia Ben Rejeb, Higher Institute of Computer Science (ISI), University of Tunis El Manar (UTM), Tunisia

ICWMC 2026 Publicity Chairs

Francisco Javier Díaz Blasco, Universitat Politècnica de València, Spain
Ali Ahmad, Universitat Politècnica de València, Spain
José Miguel Jiménez, Universitat Politècnica de València, Spain
Sandra Viciano Tudela, Universitat Politècnica de València, Spain

ICWMC 2026 Technical Program Committee

Mohamed Abid, Al Yamamah University, KSA / University of Gabes, Tunisia
Bedoui Abla, CEDOC-2IT | INPT (National Institute of Posts and Telecommunication), Morocco
Iness Ahriz, CNAM, France
Khalil Aissaoui, Tunisia Polytechnic School (TPS), Tunisia
Wafa Akkari, University of Manouba, Tunisia
Ali Kadhum M. Al-Quraby, University of Babylon, Iraq
Diego Alberto Godoy, Universidad Gastón Dachary, Argentina
Adel Aldalbahi, King Faisal University, Saudi Arabia
Farman Ali, Qurtuba University of Science and IT, D.I. Khan, Pakistan
Adda Ali-Pacha, University of Sciences and Technology of Oran, Algeria
Firas Alsehly, Huawei Edinburgh Research Centre, UK
Karine Amis, IMT Atlantique, France
Tran Hai Anh, Hanoi University of Science and Technology (HUST), Vietnam
Antonio Arena, University of Pisa, Italy
Kamran Arshad, Ajman University, UAE
Nebojša Bačanin-Džakula, Singidunum University, Serbia
Nedia Badri, ENSI - University of Manouba, Tunisia
Corey E. Baker, University of Kentucky, USA
Chaity Banerjee, University of Central Florida, USA
Kamel Barkaoui, Cedric | Cnam, France
Dimitri Belli, National Research Council (CNR) - Institute of Information Science and Technologies (ISTI), Pisa, Italy
Hadda Ben Elhadj, SM@RTS | Higher Institute of Informatics | Monastir University, Tunisia
Sonia Ben Rejeb, Higher Institute of Computer Science (ISI) - Higher School of Communications of Tunis (SUPCOM), Tunisia
Emna Ben Slimane, ENIT - Tunis El Manar University, Tunisia
Djamila Bendouda, Ecole Nationale Supérieur de Technologie, Algeria

Jordi Mongay Batalla, Warsaw University of Technology, Poland
Raúl Montoliu Colás, Institute of new imaging technologies (INIT) - Jaume I University, Spain
Fernando Moreira, Universidade Portucalense, Portugal
Alireza Morsali, McGill University, Canada
Mohamed M. A. Moustafa, Egyptian Russian University, Egypt
Sami Myllymäki, University of Oulu, Finland
Assia Naja, International University of Rabat, Morocco
Sameh Najeh, Higher school of Communication (Sup'Com) of Tunis, Tunisia
Leïla Najjar, Higher School of Communication of Tunis (SUP'COM), Tunisia
Monia Najjar, University of Tunis El Manar, Tunisia
Leila Nasraoui, National School of Computer Sciences (ENSI) | University of Manouba, Tunisia
Nejah Nasri, National Engineering School of Sfax (ENIS_LETI_Tunisia), Tunisia
Prasad Netalkar, Qualcomm, San Diego, USA
Armielle Ngaffo, Mediatron Laboratory, Tunisia
Maciej Nikodem, Wroclaw University of Science and Technology, Poland
Boubakr Nour, Beijing Institute of Technology, China
Diego Orlando Barragan Guerrero, Universidad Técnica Particular de Loja, Ecuador / ETS, Canada
Ekaterina Pakulova, Institute of Computer Science and Information Security of the Southern Federal University, Russia
Pablo Palacios, University of Chile, Chile
Tudor Palade, Technical University of Cluj-Napoca, Romania
Paulo Pinto, Universidade Nova de Lisboa, Portugal
Ivan Pires, Universidade da Beira Interior | Institute of Telecommunications, Portugal
Michele Polese, Institute for the Wireless Internet of Things | Northeastern University, USA
Valentin Radu, University of Sheffield, UK
Parisa Rafiee, George Washington University, USA
Adib Rastegarnia, Purdue University, USA
Heena Rathore, University of Texas, USA
Masood Ur Rehman, University of Glasgow, UK
Muhammad Atif Ur Rehman, Hongik University, South Korea
Yidong Ren, Qualcomm Inc., USA
Éric Renault, ESIEE Paris, France
Francesca Righetti, University of Pisa, Italy
Miguel Rodríguez-Pérez, University of Vigo, Spain
Elisa Rojas, University of Alcala, Spain
Haidar Safa, American University of Beirut, Lebanon
Hajer Saidi, National Engineering School of Sfax, Tunisia
Monia Salem, National School of Engineers of Tunis | University of Tunis El Manar, Tunisia
Varese Salvador Timóteo, Universidade Estadual de Campinas - UNICAMP, Brazil
David Sánchez-Rodríguez, University of Las Palmas de Gran Canaria, Spain
José Santa, Technical University of Cartagena, Spain
Adérito Seixas, Universidade Fernando Pessoa, Porto, Portugal
Oluyomi Simpson, University of Hertfordshire, UK
Soulayma Smirani, National Engineering School of Tunis (ENIT) | University of Tunis El Manar, Tunisia
Kaijun Song, University of Electronic Science and Technology of China, China
Marko Sonkki, Ericsson, Germany
Animesh Srivastava, Google, USA
Álvaro Suárez Sarmiento, Universidad de Las Palmas de Gran Canaria, Spain

Sheng Tan, Trinity University, USA
Fatma Tansu Hocanın, Cyprus International University, Lefkosa, TRNC
Rui Teng, Advanced Telecommunications Research Institute International, Japan
Hajer Tounsi, Ecole Supérieure des Communications de Tunis, Tunisia
Florian Tschorsch, TU Dresden, Germany
Sudhanshu Tyagi, Thapar Institute of Engineering & Technology | Deemed University, India
Rehmat Ullah, Hongik University, South Korea
Adrian Vidal, University of the Philippines Diliman, Philippines
Abdul Wahab, Queen Mary University of London, UK
Lei Wang, University of Connecticut, USA
Xianzhi Wang, University of Technology Sydney, Australia
You-Chiun Wang, National Sun Yat-sen University, Taiwan
Ulf Witkowski, South Westphalia University of Applied Sciences, Germany
Ouri Wolfson, University of Illinois at Chicago / University of Illinois at Urbana Champaign, USA
Diane Woodbridge, University of San Francisco, USA
Yuan Wu, University of Macau, Macau
Abid Yaqoob, Insight Centre for Data Analytics | Dublin City University, Ireland
Yilin Yang, Johns Hopkins University Applied Physics Laboratory, USA
Paul Yoo, University of London, UK
Sherali Zeadally, University of Kentucky, USA
Huanle Zhang, University of California, Davis, USA

Copyright Information

Table of Contents

A ROS2–Based Architecture for Indoor 3D Mapping and Autonomous Navigation with Azure Kinect and LiDAR

Itziar G. Alonso-González, Miguel Ángel Quintana-Suárez, David Sánchez-Rodríguez, Valeria Santana-Cardona, Francisco J. Cazorla-Hernández
Institute for Technological Development and Innovation in Communications (IDeTIC)
University of Las Palmas de Gran Canaria
Las Palmas de Gran Canaria, Spain
e-mail: itziar.alonso@ulpgc.es

Abstract— **High-fidelity 3D mapping is a prerequisite for deploying location-based services, particularly for accessible indoor navigation. While low-cost mapping platforms, such as configurations based on Raspberry Pi 4, are common, they present significant limitations in sensor throughput and onboard 3D reconstruction capabilities. This work introduces a new ROS2 Humble–based architecture running on the high-performance NVIDIA Corporation (NVIDIA) Jetson Orin NX platform. The sensing capabilities of the system are substantially enhanced through the integration of the Azure Kinect DK camera, while mapping and autonomous exploration rely on modern algorithms, such as Real-Time Appearance-Based Mapping (RTAB-Map) ROS2 and a custom LiDAR-based exploration script. The proposed architecture overcomes the constraints inherent to previous low-cost designs and enables a unified, efficient, and fully onboard processing pipeline. Functional validation confirms the architecture's capability to generate dense 3D maps and semantic data in real-time. Experimental results demonstrate stable Light Detection and Ranging (LiDAR) processing at 10 Hz and robust odometry updates at 10 Hz, ensuring the environmental detail required for cognitive accessibility applications.**

Keywords-ROS2; LiDAR; indoor mapping; SLAM; cognitive accessibility.

I. INTRODUCTION

Three-dimensional perception of indoor environments is a fundamental component of mobile robotics, particularly in tasks, such as Simultaneous Localization and Mapping (SLAM), path planning, and obstacle detection in assistive systems. In previous work [1], a low-cost prototype based on Do It Yourself (DIY) architecture and open-source software was developed, capable of generating 2D occupancy grid maps and 3D voxel models in real time. The system integrated a LiDAR sensor, a Time-of-Flight (ToF) camera, an Inertial Measurement Unit (IMU), and wheel encoders, all managed through the Robot Operating System (ROS) 2 in combination with Hector SLAM and OctoMap.

Despite the results obtained, the platform used—based on two Raspberry Pi 4 devices—presents significant limitations. These include the reduced range of the ToF sensor (PMD CamBoard PicoMonstar), the insufficient computational capacity to perform onboard 3D reconstruction, and the inability to incorporate information beyond the pure geometric structure of obstacles. These constraints highlighted the need for more robust architecture. Specifically, generating high-fidelity maps for cognitive accessibility requires a level of environmental detail and density that exceeds the computational capacity of standard low-cost embedded boards, such as identifying landmarks via *You Only Look Once* (YOLO). This dual workload of processing dense Red-Green-Blue-Depth (RGB-D) streams and running neural networks exceeds the computational capacity of standard low-cost embedded boards.

The present work addresses these needs through a comprehensive upgrade of the platform. The proposed system migrates to ROS2 Humble running on the NVIDIA Jetson Orin NX. This hardware upgrade is justified by the high computational demand of the ROS2 middleware and the processing of dense 3D point clouds required for high-fidelity mapping. Unlike consumer-grade assistive devices, this platform operates as a dedicated mapping instrument intended to generate the "ground truth" of the facility. This distinction justifies the hardware investment, as the system serves as a one-off infrastructure generator rather than a mass-market appliance. Therefore, the use of the Jetson Orin NX is necessary to handle the high bandwidth of the Azure Kinect and the 3D reconstruction pipeline without the latency bottlenecks observed in low-cost embedded boards. While this upgrade increases the unit cost compared to the previous Raspberry Pi-based prototype, it is a requisite for the system's role as a high-fidelity mapping instrument. Additionally, the Azure Kinect DK camera is integrated, significantly improving the quality and density of sensory data. The perception and exploration stack adopts RTAB-Map ROS2 for 3D mapping together with a custom autonomous exploration script based on LiDAR data and 2D occupancy grids. This approach enables robust environment coverage without relying on wheel-encoder-based navigation frameworks.

The remainder of this article is organized as follows: Section 2 reviews related work. Section 3 presents the system architecture, which integrates the hardware platform, sensor suite, and software framework with the proposed methodology. This section details the overall workflow,

ISBN: 978-1-68558-347-7

including calibration procedures, the autonomous navigation pipeline, and the processes employed for 3D mapping. Section 4 reports experimental results and discussion. Finally, Section 5 summarizes the conclusions and outlines future work

II. Related Work

To properly contextualize the contribution of this work, it is essential to review both the fundamental techniques in SLAM and the software frameworks that enable modern robotic systems.

In recent years, LiDAR-based SLAM has advanced considerably through multisensor fusion techniques, designed to improve accuracy and robustness. Xu et al. [2] provides a comprehensive overview of 3D LiDAR SLAM principles and compares several algorithms using real-world datasets, while LiDAR Odometry and Mapping (LOAM) [3] stands out as an optimized solution for real-time odometry and mapping, offering higher accuracy than comparable approaches. This progress in LiDAR-based SLAM has evolved in parallel with research on vision-based systems.

Indeed, numerous visual approaches with different sensor configurations have been proposed. Macario Barros et al. [4] analyze visual–inertial, monocular, and RGB-D methods, and ORB-SLAM3 [5] extends these capabilities through inertial integration and multi-map management, achieving improved robustness in low-feature environments. Other visual methods [6] offer versatility but remain sensitive to illumination and range, while KinectFusion [7] popularized dense real-time RGB-D reconstruction, albeit with scalability limitations and the absence of loop closure.

Alongside these algorithmic advances, research on accessible sensors has fostered comparative evaluations of low-cost devices. Gupta and Li [8] assess RGB-D and stereo sensors for indoor applications, whereas Tee and Han [9] compare three widely used 2D SLAM algorithms, such as Hector SLAM, Cartographer, and Gmapping, on a robot equipped with LiDAR, IMU, and wheel odometry. Complementarily, Takaya et al. [10] examine simulated 2D and 3D mapping using Hector SLAM and OctoMap, and Raveendran et al. [11] apply 3D SLAM in disaster environments, demonstrating its applicability under extreme conditions.

Although the present work focuses on indoor environments, there is methodological continuity with outdoor mapping systems. These environments introduce additional challenges—environmental variability, spatial scalability, and illumination changes—yet share common techniques for sensor fusion, filtering, and state estimation. Yin et al. [12] address these challenges by proposing a robust Kalman filter for Global Navigation Satellite System (GNSS)–IMU fusion in Unmanned Ground Vehicles (UGVs), while [13] focuses on yaw and velocity error estimation. In parallel, the pursuit of computational efficiency has encouraged the use of deep learning. For example, Zhang et al. [14] employ PointNet to down sample LiDAR point clouds while maintaining low reconstruction error.

In addition to algorithmic progress, the underlying software ecosystem has undergone significant transformation. The evolution of ROS has redefined the capabilities of autonomous systems. The transition from ROS 1 to ROS2 represents a fundamental shift toward more robust and real-time–capable architectures. Unlike its predecessor, ROS2 utilizes the Data Distribution Service (DDS) middleware, which imposes higher serialization and processing overheads that can overwhelm low-power embedded devices when handling high-bandwidth sensor streams. With its new DDS-based communication layer, ROS2 improves modularity, determinism, and safety—critical features in complex and collaborative environments [15]. In parallel, the ROS2 ecosystem has matured quickly, consolidating advanced tools, such as the Nav2 navigation stack, which incorporates modern planners, recovery behaviors, and a modular plugin-based architecture [16]. Benchmarking platforms, such as Arena-Rosnav [17] and performance evaluation frameworks [18][19] have further contributed to reproducible testing and accelerated research in 3D mapping and dynamic navigation. These developments have supported recent work in hybrid robotics and low-cost mapping systems [2][20].

Within the reviewed literature, two relevant gaps are evident—both addressed directly by the present work. First, although LiDAR-based multisensor fusion is well established, the joint integration of Time-of-Flight cameras with LiDAR and wheel odometry remains relatively unexplored. Existing studies do not detail specific ToF sensor configurations, highlighting a need for better characterization of hybrid architectures that combine precise range with active depth sensing. Second, despite the prevalence of ROS and the growing interest in standalone C++ frameworks, none of the works surveyed document the migration of LiDAR-based 3D reconstruction systems to ROS2. This absence is notable, given that ROS2 offers substantial improvements in deterministic communication, modularity, and real-time performance—features that are especially critical in mobile platforms operating under computational constraints.

Building on these observations, the architectural evolution presented in this work is oriented toward a cognitive perception platform running ROS2 Humble on the NVIDIA Jetson Orin NX. This transition enables the replacement of the original mapping stack (Hector SLAM and OctoMap) with advanced tools, such as RTAB-Map ROS2 for visual SLAM and Nav2 for autonomous navigation, both designed for distributed systems operating under high sensory load. Additionally, the exploration of cognitive perception techniques aligns with emerging trends in literature toward semantically enriched scene understanding through deep learning methods, such as YOLOv7 and YOLOv8 [21].

Overall, this review of related work highlights recent advances in perception, mapping, navigation, and robotic middleware, and establishes the foundations that justify the proposed architecture as a solution to the processing, sensing, and semantic limitations identified in prior systems.

III. System Architecture

A. Hardware Platform

Since its publication in [1], the system architecture has evolved from an initial proof-of-concept prototype into a more advanced platform oriented toward cognitive perception, addressing the functional limitations identified in the original design. The primary objective is to deploy the robot as a mapping instrument to generate high-fidelity 3D semantic maps. These maps serve as the "ground truth" infrastructure required to support future location-based services for people with functional diversity.

Consequently, the system is designed for offline map generation rather than consumer-grade real-time assistance. The original prototype fulfilled this purpose by producing simultaneous 2D and 3D maps through sensor fusion of a 2D LiDAR, a Time-of-Flight (ToF) camera, an IMU, and wheel odometry, all managed in ROS (Legacy) using Hector SLAM and OctoMap.

However, the initial Raspberry Pi 4 architecture struggled with the ROS2 stack and 3D reconstruction pipeline. The combined bandwidth of the LiDAR (~70 KB/s) and RGB-D streams saturated the CPU, causing serialization bottlenecks, sparse maps, odometry drift, and dropped DDS messages. Furthermore, insufficient resources prevented semantic analysis onboard, crucial for annotating accessibility information. These constraints necessitated a complete architectural redesign to eliminate performance bottlenecks and enable cognitive perception.

The new architecture migrates to ROS2 Humble, leveraging its native support for deterministic real-time communication. The platform upgrades to a high-performance NVIDIA Jetson Orin NX, essential for executing real-time computer vision and deep-learning models like YOLOv7/v8. Additionally, an Azure Kinect DK is integrated to provide dense RGB, depth, and point-cloud streams, significantly enhancing data quality compared to the previous ToF sensor (see Figure 1 and Table I).

TABLE I. Components in both Prototypes

Specification	Original Prototype (ROS 1)	Migrated Platform (ROS2)
Processing Unit	2x Raspberry Pi 4 Model B (Legacy)	NVIDIA Jetson Orin NX (Proposed)
Depth / Vision Sensor	PMD CamBoard PicoMonstar (ToF)	Azure Kinect DK (RGB-D)
LiDAR Sensor	Delta LiDAR 2A	RPLidar S2
Low-level Control System	ATMega2560	Arduino UNO + MotorShield (QGPMarker v5.3)

Integrating the Azure Kinect DK on an ARM64 architecture required manual compilation of the vendor's Software Development Kit (SDK) and optimization of the sensor pipeline. To manage the estimated 5.7 Mbps system-wide bandwidth without saturating the DDS middleware, the RGB-D acquisition rate was fixed at 15 FPS, while the LiDAR stream operates at 10 Hz. For system stability, the acquisition rate was fixed at 15 FPS. All sensor configurations were encapsulated in independent ROS2 launch files to allow fine-grained resource allocation. During the migration, inconsistencies were also detected in the publication of the */odom* topic and in the TF tree, which were corrected by adjusting publishing rates and recalibrating wheel-encoder parameters.

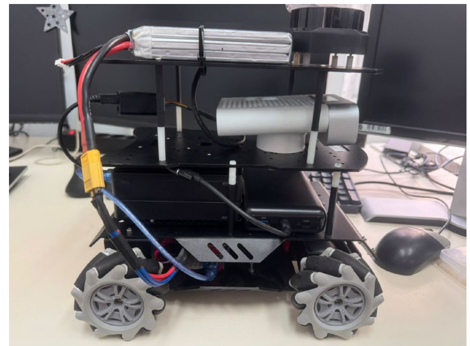

Figure 1. Robot implementation

The new architecture incorporates a modern ROS2-optimized stack for SLAM and navigation:

- RTAB-Map ROS2 for 3D mapping and loop-closure detection.
- LiDAR-based autonomous exploration using a custom frontier-driven strategy operating on 2D occupancy grids.
- YOLOv7/YOLOv8 (experimental) for cognitive perception and classification of structural and functional barriers.

This consolidated architecture provides a robust foundation for a complete real-time perception, mapping, and navigation pipeline, enabling advanced environment interpretation. Building on this foundation, the system's operational workflow—which replaces the traditional methodology section—is presented as a direct consequence of the architectural design.

B. Laser-Based Odometry Using RF2O

Due to persistent wiring issues in the mecanum wheel encoders, reliable wheel odometry could not be obtained. As a result, the system adopts rf2o_laser_odometry (Range Flow-based 2D Odometry) as its primary odometry source. This method estimates the robot's planar motion (x, y, yaw) by analyzing consecutive LiDAR scans, without relying on wheel encoder feedback. The rf2o node operates at 10 Hz, synchronized with the /scan topic published by the RPLidar S2, and publishes the transformation between the odom and base_footprint frames. This configuration provides robust local odometry suitable for SLAM and autonomous exploration in indoor environments, even in the absence of functional wheel encoders.

Since the wheel encoders are not operational, the system publishes simulated joint_states at 10 Hz. These values do not represent real wheel positions but are used solely to

 ISBN: 978-1-68558-347-7

maintain correct visualization of the robot model in RViz2 and ensure consistency in the TF tree.

C. Operational Workflow Defined by the Architecture

The system operation is structured into four consecutive phases that directly reflect the interaction between the architectural modules under ROS2. Unlike classical navigation pipelines, the workflow is designed for robust autonomous exploration and data capture, rather than goal-driven navigation. Figure 2 illustrates the complete workflow of the proposed approach.

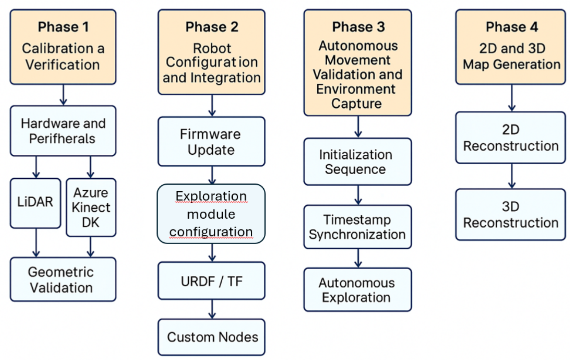

Figure 2. Workflow of the proposed approach.

1) Phase I. Initial System Calibration and Verification

Within the new architecture, initial calibration ensures proper interaction among the Jetson Orin NX, the Arduino UNO motor-control board, and the primary sensors. The RPLidar S2 and Azure Kinect DK were integrated and validated, ensuring stable publication of */scan* and configuring RGB-D capture at 15 FPS to prevent system load peaks (which previously exceeded 90% on the legacy hardware). The robot's geometric model was verified in RViz2 using an updated Unified Robot Description Format (URDF). The transformation tree (TF) is maintained by the robot_state_publisher and static_transform_publisher nodes, ensuring consistent links between the base, LiDAR, and camera frames.

2) Phase II. Robot Configuration and Integration in ROS2

During this phase, the full ROS2 computation graph is launched. Sensor drivers, RF2O odometry, SLAM Toolbox, RTAB-Map, TF publishers, and visualization nodes are integrated. Simulated *joint_states* are published at 10 Hz to ensure compatibility with RViz2 and robot_state_publisher. No wheel-encoder-based odometry or Nav2 components are used in this phase.

3) Phase III. Autonomous Motion Validation and Environment Capture

Autonomous motion is achieved using a custom exploration script (smart_explorer.py). The script analyzes the 2D occupancy grid generated by SLAM Toolbox, detects unexplored frontier regions, and generates velocity commands on /cmd_vel. Obstacle avoidance is performed directly using LiDAR range data. This strategy enables reliable exploration without requiring accurate wheel odometry.

All sensor data, including LiDAR scans, RF2O odometry, depth images, dummy RGB images, and TF transforms, are recorded using rosbag for offline analysis and map generation.

4) Phase IV: Environment Reconstruction

The 2D occupancy grid was generated using outputs from SLAM Toolbox, while the 3D reconstruction was obtained from the Azure Kinect DK point clouds.

This process demonstrates that the proposed architecture enables robust environment capture, modeling, and navigation in structured indoor spaces.

IV. RESULTS AND DISCUSSION

The implemented architecture was validated through a series of functional tests designed to assess the stability of the sensing pipeline, the correctness of the ROS2 integration, and the operation of the autonomous navigation modules. The results obtained confirm the viability and robustness of the proposed design.

1) Sensor Integration and ROS2 Pipeline Validation

All hardware components—RPLidar S2, Azure Kinect DK, and the motor controller—were successfully integrated into ROS2 Humble. The system was configured to operate without wheel-encoder-based odometry due to persistent wiring issues in the mecanum wheels. Consequently, rf2o_laser_odometry was adopted as the primary odometry source, providing planar motion estimation based solely on consecutive LiDAR scans at 10 Hz, synchronized with the publication rate of the /scan topic.

The ROS2 communication pipeline demonstrated stable performance under sustained operation, handling multiple high-bandwidth data streams without message loss or DDS instability. The LiDAR sensor published laser scans at 10 Hz, with an average throughput of 135–138 KB/s, while the RF2O odometry node generated /odom updates at 10 Hz with a bandwidth of approximately 6–7 KB/s. The Azure Kinect DK was configured in depth-only mode at 15 FPS, resulting in a stable data rate of approximately 5.2 MB/s. Transform messages were published at 35–40 Hz, with a bandwidth of 6–7 KB/s, maintaining a consistent and conflict-free TF tree. These quantitative observations—specifically the stability of the 10 Hz odometry loop—validate that the Jetson Orin NX correctly handles the ROS2 DDS communication overhead, solving the latency bottlenecks of the previous architecture.

2) 2D Localization and Mapping

Using SLAM Toolbox, the robot successfully generated local 2D occupancy maps in confined indoor environments. The mapping node produced updates at 0.5 Hz with a resolution of 0.05 m/cell and an average message size of 160 KB, maintaining global consistency without CPU saturation.

During these tests:

- The robot-maintained pose estimates with low drift over short trajectories,
- The SLAM graph remained globally consistent,
- Loop-closure events were registered in cases where the mechanical instability did not significantly perturb odometry.

These experiments validate the front-end sensing and back-end optimization components of the architecture.

3) *Autonomous Exploration*

Although the ROS2 Navigation2 (Nav2) stack was initially considered, it was not deployed due to unreliable mecanum wheel encoder data, which prevented the use of wheel-based odometry required by Nav2. The system instead relies on LiDAR-based odometry (rf2o_laser_odometry) and a lightweight autonomous exploration script (smart_explorer.py) that operates on LiDAR scans and the 2D occupancy grid generated by SLAM Toolbox. This approach enables robust environment coverage for mapping purposes without dependency on wheel odometry.

4) *RTAB-Map 3D Perception Pipeline (Partial Validation)*

RGB-D data from the Azure Kinect DK was streamed into RTAB-Map ROS2, enabling:

- Successful initialization of the RGB-D SLAM modules.
- Generation of local point clouds aligned with LiDAR and odometry data,
- Registration of loop-closure hypotheses under favourable motion conditions.

The 3D perception pipeline was verified to operate correctly under stationary and short-motion tests.

V. Conclusion and Future Works

The migration to ROS2 Humble running on the NVIDIA Jetson Orin NX has proven to be a strategic and effective response to the limitations of the original architecture. This transition enabled a system that is significantly more powerful, robust, and scalable, capable of supporting advanced perception and mapping stacks, such as RTAB-Map ROS2, together with a lightweight autonomous exploration strategy suitable for platforms with unreliable wheel odometry. By distributing the workload across 18 active ROS2 nodes, the architecture eliminates previous computational bottlenecks, handling a system-wide bandwidth of ~5.7 Mbps while maintaining stable 10 Hz odometry and 10 Hz LiDAR processing—metrics that are critical for consistent SLAM convergence. The new technological foundation establishes the basis for using the robot as a high-fidelity mapping instrument. Unlike consumer-grade robots, this platform prioritizes map density over component cost, allowing for the offline generation of semantically enriched cognitive maps that serve as the "ground truth" infrastructure for separate low-cost user devices.

The results obtained confirm that the architectural redesign is sound and that the sensing and navigation modules operate correctly under controlled conditions, paving the way for future deployment in complex, dynamic scenarios.

Acknowledgment

This work is the result of research projects partially funded by the Canary Islands Agency for Research, Innovation and the Information Society (ACIISI) and by the European Regional Development Fund (ERDF) under the Canary Islands FEDER Program 2021–2027. Project [PROID2024010006]

References

[1] C. M. Mesa-Cantillo, I. Alonso-González, D. Sánchez-Rodríguez, M. A. Quintana-Suárez, and K. D. García-Mederos, “A Methodology Based on Lidar, Time-Of-Flight Camera and Odometry for a Real-Time 3D Model Generation,” Preprints 2023, 2023060303, 2023. [Online]. Available: https://doi.org/10.20944/preprints202306.0303.v1 [retrieved: January, 2026].

[2] X. Xu et al., “A Review of Multi-Sensor Fusion SLAM Systems Based on 3D LIDAR,” Remote Sens., vol. 14, no. 12, p. 2835, 2022. https://doi.org/10.3390/rs14122835

[3] P. Zhou, X. Guo, X. Pei, and C. Chen, “T-LOAM: Truncated Least Squares LiDAR-Only Odometry and Mapping in Real Time,” IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1–13, 2022. https://doi.org/10.1109/TGRS.2021.3083606

[4] A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel, “A Comprehensive Survey of Visual SLAM Algorithms,” Robotics, vol. 11, no. 1, p. 24, 2022. https://doi.org/10.3390/robotics11010024

[5] C. Campos, R. Elvira, J. J. Gómez, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM,” IEEE Transactions on Robotics, vol. 37, no. 6, pp. 1874–1890, 2021. https://doi.org/10.1109/TRO.2021.3070826

[6] R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras,” IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255–1262, 2017. https://doi.org/10.1109/TRO.2017.2705103

[7] R. A. Newcombe et al., “KinectFusion: Real-time dense surface mapping and tracking,” in 2011 10th IEEE International Symposium on Mixed and Augmented Reality, Basel, Switzerland, 2011, pp. 127–136. https://doi.org/10.1109/ISMAR.2011.6092378

[8] T. Gupta and H. Li, “Indoor mapping for smart cities — An affordable approach: Using Kinect Sensor and ZED stereo camera,” in Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2017, pp. 1–8. https://doi.org/10.1109/IPIN.2017.8115909

[9] Y. K. Tee and Y. C. Han, “Lidar-Based 2D SLAM for Mobile Robot in an Indoor Environment: A Review,” in Proceedings of the 2021 International Conference on Green Energy, Computing and Sustainable Technology (GECOST), 2021, pp. 1–7.
https://doi.org/10.1109/GECOST52368.2021.9538731

[10] K. Takaya, T. Asai, V. Kroumov, and F. Smarandache, "Simulation environment for mobile robots testing using ROS and Gazebo," in Proceedings of the 2016 20th International Conference on System Theory, Control and Computing (ICSTCC), 2016, pp. 96–101. https://doi.org/10.1109/ICSTCC.2016.7790647

[11] R. Raveendran, S. Ariram, A. Tikanmäki, and J. Röning, "Development of task-oriented ROS-based Autonomous UGV with 3D Object Detection," in Proceedings of the 2020 IEEE International Conference on Real-time Computing and Robotics (RCAR), 2020, pp. 427–432. https://doi.org/10.1109/RCAR49640.2020.9303034

[12] Z. Yin, M. Fu, and K. Shen, "A Novel Robust Kalman Filter for Unmanned Ground Vehicles Positioning under GNSS Abnormal Measurements," in 2020 39th Chinese Control Conference (CCC), Shenyang, China, 2020, pp. 3427–3432. https://doi.org/10.23919/CCC50068.2020.9189178

[13] X. Xia et al., "Estimation on IMU yaw misalignment by fusing information of automotive onboard sensors," Mechanical Systems and Signal Processing, vol. 162, p. 107993, 2022. https://doi.org/10.1016/j.ymssp.2021.107993

[14] X. Zhang, Y. Li, H. Yin, and R. Xiong, "3D LiDAR Map Compression Using Deep Neural Network," in Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2019, pp. 1430–1434. https://doi.org/10.1109/ROBIO49542.2019.8961606

[15] J. Petershans, J. Herbst, E. Mittag, M. Rueb, and H. D. Schotten, "Advancing Telerobotics: Evaluating ROS2 in a Real-World Communication Test Environment," in Mobilkommunikation; 29. ITG-Fachtagung, Osnabrück, 2025, pp. 97–102.

[16] T. Schwörer, J. E. Schmidt, and D. Chrysostomou, "Nav2CAN: Achieving Context Aware Navigation in ROS2 Using Nav2 and RGB-D sensing," in 2023 IEEE International Conference on Imaging Systems and Techniques (IST), Copenhagen, Denmark, 2023, pp. 1–6. https://doi.org/10.1109/IST59124.2023.10355731

[17] L. Kästner et al., "Arena-Rosnav 2.0: A Development and Benchmarking Platform for Robot Navigation in Highly Dynamic Environments," arXiv preprint arXiv:2302.10023, 2023. [Online]. Available: https://arxiv.org/abs/2302.10023 [retrieved: January, 2026].

[18] G. M. C. van den Hoven, "Introducing a Performance Observation Framework to ROS2," Master's thesis, Eindhoven University of Technology, 2024. [Online]. Available: https://research.tue.nl/en/studentTheses/introducing-a-performance-observation-framework-to-ros2 [retrieved: January, 2026].

[19] M. Ryalat, G. Al-refai, N. Almtireen, and H. ElMoaqet, "Design of a ROS2-Based Hybrid Aerial-Ground Robot for Autonomous Inspection Applications," IEEE Access, vol. 13, pp. 1–1, 2025.
https://doi.org/10.1109/ACCESS.2025.3582653

[20] D. D. Yanyachi et al., "Laser_RobMap: An open source ROS2 compatible tool for 3D mapping using a Mobile Robot and 2D LiDAR," SoftwareX, vol. 30, p. 102142, May 2025. https://doi.org/10.1016/j.softx.2025.102142

[21] Ultralytics, "Discover YOLO models | Cutting-Edge Computer Vision," 2025. [Online]. Available: https://www.ultralytics.com/yolo [retrieved: January, 2026].

 ISBN: 978-1-68558-347-7