International Journal of Engineering Insights: (2025) Vol. 3, Nro.1, Regular Paper
https://doi.org/10.61961/injei.v3i1.87

Performance Evaluation of ROS2 and Micro-ROS for Real-Time Distributed Control of Omnidirectional Mobile Robots

Jonathan S. Toapanta · Carlos Julio Fierro-Silva

Received: 08 Jul 2025 / Accepted: 15 Oct 2025 / Published: 15 Nov 2025

Abstract: Omnidirectional mobile robots require efficient distributed control architectures to ensure reliable motion execution and real-time communication between high-level navigation software and embedded motor controllers. This paper presents the design and implementation of a ROS2- based distributed control system for an omnidirectional mobile robot using embedded communication platforms and intelligent actuators. The proposed architecture integrates ROS2, Micro-ROS, and ESP32-based communication to enable real-time interaction between software and hardware layers within a modular distributed robotic framework. A structured development methodology based on iterative SCRUM phases was employed for system design, implementation, testing, and validation. Experimental results demonstrate high functional reliability, robust closed-loop motion execution, and strong user acceptance, achieving a 98% satisfaction rate in perceived usability and functionality. The proposed architecture provides a scalable and cost-effective solution for omnidirectional robotic platforms in industrial and academic applications, contributing to the advancement of distributed ROS2- based mobile robotic systems.

Keywords ROS2 · micro-ROS · robotic control · intelligent actuators · ESP32 · distributed systems

1 Introduction

Mobile robotics has experienced significant growth in recent years, driven by advances in artificial intelligence, embedded systems, and real-time communication technologies. Among the various mobile robotic platforms, omnidirectional mobile robots (OMRs) have attracted increasing attention due to their superior maneuverability, holonomic motion capabilities, and suitability for complex environments such as industrial automation, logistics, and service robotics [1,2,3,4]. Their ability to move in arbitrary planar directions without changing orientation makes them particularly advantageous in dynamic and spatially constrained scenarios.

Despite these advantages, achieving accurate and robust motion control in OMRs remains a challenging task due to nonlinear dynamics, model uncertainties, external disturbances, and actuator coupling effects. Various advanced control strategies have been proposed to address these issues, including sliding mode control, adaptive control, and model predictive control [5]. Robust tracking controllers based on sliding mode formulations have demonstrated improved disturbance rejection and trajectory stability in omnidirectional robotic systems [6,7]. More recently, neural-network-based disturbance compensation and predictive control approaches have further enhanced trajectory tracking performance under uncertain operating conditions [8,9,10].

Beyond control design, accurate modeling and calibration remain essential to ensure reliable robot operation, particularly in systems affected by mechanical tolerances, wheel slippage, and sensor noise. Prior studies have investigated calibration and kinematic parameter identification methods to improve motion accuracy and localization precision in omnidirectional robotic platforms [11,12]. Additionally, efficient navigation in real-world environments requires the integration of global and local planning strategies for path tracking and obstacle avoidance, as demonstrated in several autonomous navigation frameworks for mobile robots [13,14,15,16].

In parallel with these algorithmic advances, the Robot Operating System 2 (ROS2) has emerged as a key middleware for modern robotic systems, enabling modular, scalable, and distributed real-time communication through the Data Distribution Service (DDS) protocol [17,18]. Recent developments in ROS2 have significantly expanded the mobile robotics ecosystem, par-

Jonathan S. Toapanta
Universidad Tecnológica Indoamérica, Ecuador.
E-mail: toapantajonathaning@gmail.com

Carlos Julio Fierro-Silva
Universidad Tecnológica Indoamérica, Ecuador.
E-mail: cfierro@indoamerica.edu.ec

ticularly through frameworks such as Nav2, which integrate localization, planning, and perception into unified navigation architectures [19]. Furthermore, ROS2-compatible simulation and deployment tools have facilitated the transition from algorithm development to real-world robotic implementation [20], while distributed robotic architectures have demonstrated improved scalability and interoperability in modern robotic ecosystems [21].

Despite these advances, implementing ROS2-based distributed control systems in real-time robotic platforms remains challenging. Communication delays, middleware overhead, and synchronization constraints can significantly affect control responsiveness and system reliability in embedded robotic applications. Recent studies have highlighted the importance of tracing and performance analysis frameworks for evaluating ROS2 execution behavior and communication latency in distributed systems [22, 23].

To address embedded hardware limitations, Micro-ROS has emerged as an extension of ROS2 for microcontroller-based systems, enabling lightweight ROS2-compatible communication in resource-constrained devices [24]. Although previous studies have demonstrated the feasibility of ROS2 and Micro-ROS in distributed robotic applications, limited work has experimentally validated their integration in omnidirectional mobile robot platforms operating under real-world conditions.

Motivated by this gap, this paper presents the design and experimental validation of a ROS2-based distributed control architecture for an omnidirectional mobile robot integrating Micro-ROS, ESP32 embedded controllers, and intelligent actuator communication. The proposed system enables modular interaction between high-level ROS2 control nodes and low-level embedded hardware within a distributed robotic framework. Experimental results demonstrate reliable functional performance and strong user acceptance, supporting the feasibility of ROS2-based distributed architectures for omnidirectional mobile robotic systems.

2 Materials and Methods

2.1 System Architecture

The proposed system is based on a distributed robotic architecture designed to enable real-time communication and control between high-level software components and low-level embedded devices. The architecture follows a layered approach integrating sensing, processing, communication, and actuation modules within a ROS2-based ecosystem. Fig. 1 illustrates the general architecture of the proposed system, highlighting the interaction between the user interface, ROS2 control nodes, DDS middleware, embedded communication layer, and robotic actuators.

At the core of the system, ROS2 nodes are responsible for control logic, data processing, and system coordination. Communication between nodes is handled through the Data Distribution Service (DDS), which provides a scalable publish–subscribe mechanism suitable for real-time robotic systems.

To extend ROS2 capabilities to embedded platforms, Micro-ROS is integrated as a lightweight communication layer. This enables seamless interaction between the ROS2 network and resource-constrained devices. The embedded layer is implemented using an ESP32 microcontroller, which executes low-level control routines and interfaces directly with the actuators.

The actuation system is composed of Dynamixel intelligent servomotors, which provide position, velocity, and torque control along with real-time feedback. A user interface module allows command input and monitoring, enabling human interaction with the robotic system.

This architecture ensures modularity, scalability, and low-latency communication, which are essential for real-time robotic applications.

Table 1 Hardware components of the proposed system

Component	Description
ESP32	Embedded microcontroller for real-time control and communication
Dynamixel	Intelligent servomotors with integrated sensing
OpenCM	Dedicated controller for actuator management
Robotic platform	Omnidirectional mobile robot structure

2.2 Hardware Components

The hardware platform consists of embedded processing units, actuator modules, and the robotic structure. Table 1 summarizes the main components.

The ESP32 acts as an interface between ROS2 and the physical hardware, executing control commands and handling actuator communication. The Dynamixel actuators are selected due to their precision, reliability, and compatibility with embedded control systems.

2.3 Software Framework

The software architecture is built upon ROS2, which provides distributed communication, modularity, and

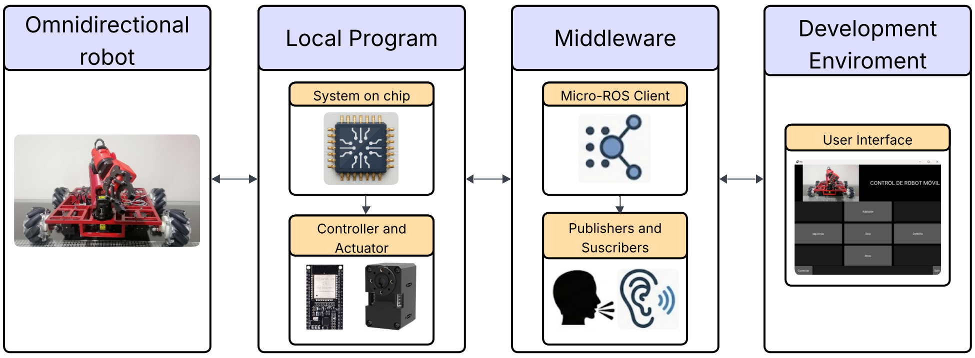

Fig. 1 General architecture of the ROS2-based distributed control system, showing the interaction between user interface, middleware (DDS), embedded devices, and robotic actuators.

Table 2 Software components

Software	Function
ROS2	Distributed middleware and communication framework
Micro-ROS	Embedded communication layer
Python	User interface and control scripts
Ubuntu 22.04	Operating system

scalability. The main software components are listed in Table 2.

ROS2 nodes manage trajectory generation, control execution, and data processing. Micro-ROS bridges communication between ROS2 and the ESP32, enabling efficient message exchange using lightweight protocols.

2.4 Kinematic and Control Model

The omnidirectional mobile robot is modeled using a holonomic kinematic framework, which allows independent motion along the planar axes and rotational movement. The robot state in the global reference frame is defined as:

$$\mathbf{v} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \tag{1}$$

where $\dot{x}$ and $\dot{y}$ represent the linear velocities in the Cartesian plane, and $\dot{\theta}$ denotes the angular velocity.

For a four-wheeled omnidirectional robot equipped with mecanum wheels, the relationship between the robot velocity and the wheel angular velocities is given by:

$$\boldsymbol{\omega} = \mathbf{J}\,\mathbf{v} \tag{2}$$

where $\boldsymbol{\omega} = [\omega_1\ \omega_2\ \omega_3\ \omega_4]^T$ is the vector of wheel angular velocities, and $\mathbf{J}$ is the Jacobian matrix defined as:

$$\mathbf{J} = \frac{1}{r}\begin{bmatrix} 1 & -1 & -(L+W) \\ 1 & 1 & (L+W) \\ 1 & 1 & -(L+W) \\ 1 & -1 & (L+W) \end{bmatrix} \tag{3}$$

where r is the wheel radius, and L and W represent the distances from the robot center to the wheel along the longitudinal and lateral axes, respectively.

The desired trajectory is defined as $\mathbf{v}_d = [\dot{x}_d\ \dot{y}_d\ \dot{\theta}_d]^T$. The tracking error is computed as:

$$\mathbf{e} = \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix} \tag{4}$$

A proportional control law is adopted to regulate the tracking error:

$$\mathbf{v} = \mathbf{K}_p\,\mathbf{e} \tag{5}$$

where $\mathbf{K}_p = \mathrm{diag}(k_x, k_y, k_\theta)$ is a diagonal gain matrix.

The computed velocity commands are transformed into wheel velocities using the kinematic model and transmitted through the ROS2 communication layer to the embedded controller.

This control strategy is selected due to its low computational complexity, making it suitable for real-time implementation in resource-constrained embedded platforms such as the ESP32, while still providing stable trajectory tracking performance.

2.5 Closed-Loop Control Scheme

The control architecture follows a closed-loop scheme, as illustrated in Fig. 2. The desired trajectory is generated at the ROS2 level and compared with the actual robot state to compute the tracking error.

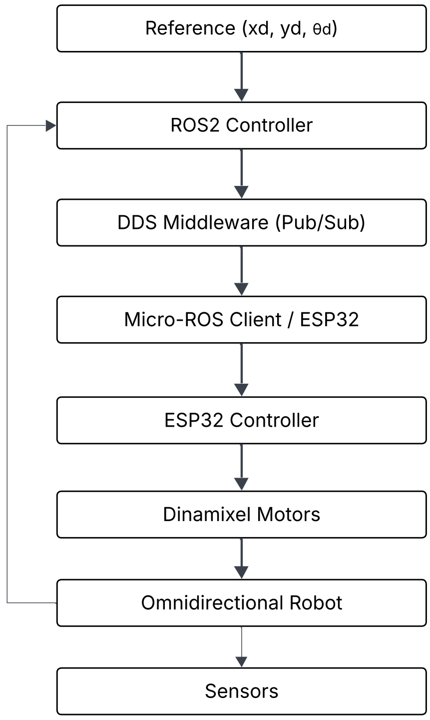

Fig. 2 Closed-loop control scheme of the ROS2-based omnidirectional robotic system.

Control commands are transmitted via DDS to the embedded system using Micro-ROS. The ESP32 executes actuator commands, while feedback from sensors is sent back to ROS2 nodes. This feedback loop enables real-time correction and ensures system stability.

2.6 System Workflow

The operational workflow of the proposed system follows a sequential process that ensures real-time interaction between the high-level control layer and the embedded system. The process begins with the initialization of ROS2 nodes and the configuration of communication interfaces.

Once the system is initialized, the DDS communication layer is established, enabling data exchange between ROS2 nodes and the embedded platform through Micro-ROS. After successful connection, the system waits for user commands, which are processed at the ROS2 level to generate appropriate control signals.

These control commands are transmitted through the DDS middleware to the embedded system, where the ESP32 executes the corresponding actuator control actions. The Dynamixel actuators perform the required motion, while sensor data is continuously acquired.

Feedback information is then transmitted back to the ROS2 nodes, allowing real-time monitoring and updating of the system state. This continuous exchange of information ensures closed-loop operation and enables dynamic response to changes in system conditions.

Table 3 Functional requirements validation

ID	Requirement	Attempts	Success	Status
RF1	Connection	40	38	Completed
RF2	Linear velocity	40	36	Completed
RF3	Angular velocity	40	36	Completed
RF4	Stop	40	40	Completed
RF5	Exit	40	40	Completed

2.7 Development Methodology

The system was developed using an iterative methodology inspired by the SCRUM framework, including planning, implementation, testing, and validation phases. Each iteration focused on integrating specific modules such as communication, control, and hardware interaction.

This approach allowed progressive system refinement and facilitated the identification of performance bottlenecks, particularly those related to communication latency and real-time execution.

3 Results

3.1 Functional Validation

The functional performance of the proposed system was evaluated through a series of execution tests, as summarized in Table 3. The table presents the number of attempts and successful executions for each functional requirement.

Figure 3 illustrates the experimental setup used during functional validation of the proposed control architecture. The interface provides real-time command execution for omnidirectional motion control, while the terminal output confirms successful ROS2-based communication and command processing during operation.

As shown in Table 3, the system achieved high reliability across all evaluated functions. The connection module reached a success rate of 95%, while motion control functions such as linear and angular velocity achieved 90%. The stop and exit functions demonstrated complete reliability with 100% success rates.

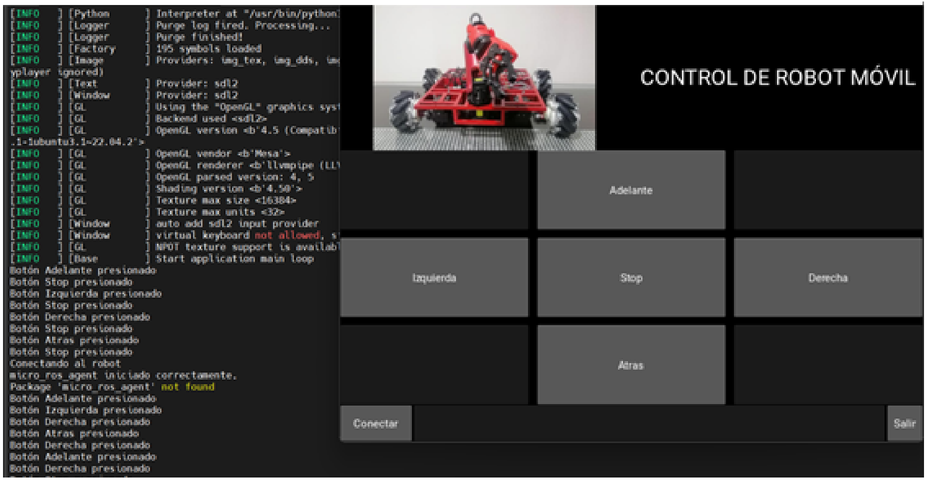

Fig. 3 Experimental functional validation setup of the proposed ROS2-based omnidirectional mobile robot control system. The figure shows the graphical user interface used for command execution, the physical omnidirectional robot platform, and the ROS2 terminal feedback during real-time operation.

Table 4 Likert scale used for TAM-based evaluation

Score	Description	Interpretation
1	Strongly disagree	Very negative
2	Disagree	Negative
3	Slightly disagree	Slightly negative
4	Neutral	Neutral
5	Slightly agree	Slightly positive
6	Agree	Positive
7	Strongly agree	Very positive

3.2 User Acceptance Evaluation

A user acceptance evaluation was conducted based on the Technology Acceptance Model (TAM), using a structured questionnaire. The evaluation was performed by a laboratory engineer, providing qualitative validation of system usability.

The rating scale used in the evaluation is presented in Table 4, ranging from 1 (strongly disagree) to 7 (strongly agree).

The results related to perceived usefulness are shown in Fig. 4. The participant assigned the highest rating (7) to most questions, indicating strong agreement regarding the usefulness of the developed system.

Similarly, Fig. 5 presents the results for perceived ease of use. The responses show consistently high scores, confirming that the system is easy to operate and meets user expectations.

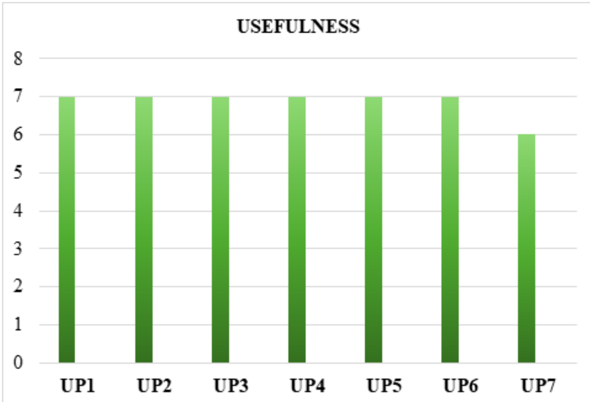

Fig. 4 Perceived usefulness evaluation results

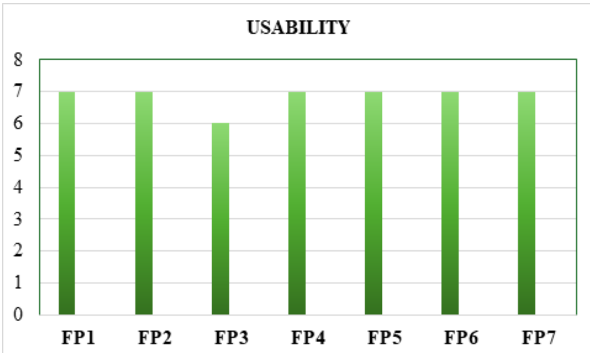

Fig. 5 Perceived ease of use evaluation results

4 Discussion

The experimental results demonstrate that the proposed ROS2-based distributed control architecture provides reliable real-time performance for omnidirectional mobile robot operation. Functional validation showed success rates above 90% for all evaluated control tasks, with critical commands such as emergency stop and shutdown achieving 100% reliability. These findings confirm that the proposed architecture is suitable for practical mobile robotic applications requiring robust low-latency communication.

The integration of ROS2 with Micro-ROS and ESP32 embedded controllers proved effective for enabling distributed communication between high-level motion planning modules and low-level actuator control. This result is consistent with previous studies reporting that ROS2 significantly improves modularity, scalability, and real-time interoperability in distributed robotic systems compared with ROS1-based implementations [19,24,22].

Regarding omnidirectional robotic platforms, the achieved motion reliability aligns with prior works highlighting the importance of accurate distributed control architectures for maintaining stable trajectory tracking and motion precision in holonomic robots [6,9,11]. The proposed architecture complements these control-oriented studies by providing a practical communication and embedded implementation framework capable of supporting real-time control strategies in physical omnidirectional platforms.

The TAM-based user evaluation yielded an overall acceptance rate of 98%, indicating that the developed platform is perceived as highly usable and effective. Similar high user acceptance has been reported in robotics interface studies where intuitive ROS2-based control environments improved operator interaction and deployment efficiency in laboratory and industrial settings.

Nevertheless, some limitations remain. The user acceptance evaluation was conducted with a limited sample size, which restricts the statistical generalizability of the usability results. Furthermore, although the functional performance was satisfactory, future studies should incorporate quantitative latency benchmarking and end-to-end timing analysis, as communication delays remain a critical factor in distributed ROS2 robotic systems [22,23].

Overall, the proposed architecture demonstrates that combining ROS2, Micro-ROS, and embedded controllers constitutes an effective framework for omnidirectional mobile robot control, offering a scalable and modular solution for next-generation distributed robotic platforms.

5 Conclusions

This paper presented the design and implementation of a distributed control system for an omnidirectional mobile robot based on ROS2 and embedded communication technologies. The proposed architecture integrates ROS2, Micro-ROS, and ESP32-based control, enabling efficient interaction between high-level software components and low-level hardware devices.

The experimental results demonstrated that the system achieves high reliability in functional execution, with success rates above 90% across all evaluated requirements. Critical control operations reached full reliability, confirming the robustness of the implemented architecture. Furthermore, the user acceptance evaluation based on the TAM model showed a high level of satisfaction, with an overall acceptance rate of 98%, indicating that the system is both effective and easy to use.

The proposed approach offers significant advantages in terms of modularity, scalability, and real-time communication, making it suitable for applications in industrial automation and academic research. The integration of embedded systems with ROS2 expands the capabilities of distributed robotic systems, particularly in resource-constrained environments.

Future work will focus on extending the system to multi-robot scenarios, incorporating advanced control strategies, and performing detailed latency and performance analysis under different operating conditions. Additionally, further user studies will be conducted to validate the system with a larger and more diverse group of participants.

Conflict of interest

The authors declare no conflict of interest.

References

1. H. Taheri and C. X. Zhao, "Omnidirectional mobile robots, mechanisms and navigation approaches," *Mechanism and Machine Theory*, vol. 153, p. 103958, 2020.
2. K. Kanjanawanishkul, "Omnidirectional wheeled mobile robots: Wheel types and practical applications," *International Journal of Advanced Mechatronic Systems*, vol. 6, no. 6, pp. 289–302, 2015.
3. G. Campion, G. Bastin, and B. D'Andrea-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, pp. 47–62, 1996.
4. P. Muir and C. Neuman, "Kinematic modeling of wheeled mobile robots," *Journal of Robotic Systems*, vol. 4, no. 2, pp. 281–340, 1987.
5. M. Zhu, D. Gong, Y. Zhao, J. Chen, and J. Qi, "Compliant force control for robots: A survey," *Mathematics*, vol. 13, no. 13, p. 2204, 2025.
6. A. Azzabi and K. Nouri, "Design of a robust tracking controller for a nonholonomic mobile robot based on sliding mode," *International Journal of Advanced Robotic Systems*, 2021.
7. P. Q. Baban and M. E. Ahangari, "Adaptive terminal sliding mode control of a non-holonomic wheeled mobile robot," *International Journal of Vehicle Information and Communication Systems*, vol. 9, no. 3, pp. 1–15, 2024.
8. C.-G. e. a. Yun, "Trajectory tracking control using rbf neural network," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 2023.

9. T. e. a. Ding, "Trajectory tracking using mpc velocity control," *Mechatronics*, 2022.
10. C. e. a. Ren, "Conditional disturbance negation control," *IEEE Robotics and Automation Letters*, 2022.
11. A. T. e. a. Popovici, "Kinematics calibration of omnidirectional robot," *Sensors*, 2022.
12. B. e. a. Siciliano, *Robotics: Modelling, Planning and Control*. Springer, 2010.
13. S. e. a. Sahloul, "Hybrid navigation methods," *Journal of Robotics and Control*, 2021.
14. S. e. a. Thrun, *Probabilistic Robotics*. MIT Press, 2005.
15. F. Ahuraka, P. McNamee, Q. Wang, Z. N. Ahmadabadi, and J. Hudack, "Chaotic motion planning for mobile robots: Progress, challenges, and opportunities," *IEEE Access*, vol. 11, pp. 134 000–134 020, 2023.
16. Y. Shen, Y. Shen, Y. Zhang, C. Huo, and Z. Shen, "Research progress on path planning and tracking control methods for orchard mobile robots in complex scenarios," *Agriculture*, vol. 15, no. 18, p. 1917, 2025.
17. M. e. a. Quigley, "Ros: an open-source robot operating system," *ICRA Workshop*, 2009.
18. G. Pardo-Castellote, "Dds overview," in *IEEE ICDCSW*, 2003.
19. S. e. a. Macenski, "Survey of mobile robotics algorithms in ros2," *Robotics and Autonomous Systems*, 2023.
20. M. Z. e. a. Andreasen, "Maes simulator for ros2," *Artificial Life and Robotics*, 2023.
21. F. C. Refis, N. A. Mahammedi, C. A. Kerrache, and S. Dhelim, "From network sensors to intelligent systems: A decade-long review of swarm robotics technologies," *Sensors*, vol. 25, no. 19, p. 6115, 2025.
22. C. Bédard *et al.*, "ros2_tracing framework," *IEEE Robotics and Automation Letters*, 2022.
23. Y. e. a. Maruyama, "Exploring the performance of ros2," in *RTCSA*, 2016.
24. S. e. a. Garrido-Jurado, "Micro-ros for microcontrollers," *IEEE Robotics and Automation Magazine*, 2021.