Autonomous Navigation Vehicle (GAMYA Robot) - A Cost-Effective ROS-Based Platform for Real-Time Mapping and Obstacle Avoidance

T. Lakshmibai,
A.P/CSE, Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, Kanchipuram, TamilNadu, India

Abstract—This paper presents *GAMYA*, a cost-effective autonomous navigation vehicle developed using the Robot Operating System 2 (ROS2) framework. The system integrates a Raspberry Pi 4 as the central processing unit, an Arduino Uno R3 for motor control, and a 2D LiDAR sensor for environmental perception. GAMYA is designed to perform real-time mapping, localization, and obstacle avoidance through the implementation of the ROS2 Navigation2 stack and SLAM Toolbox. A differential drive mechanism is employed, and odometry is computed using encoder feedback to support accurate navigation. The system is tested in a simulated environment using Gazebo, where it successfully navigates dynamic and static obstacles. Results demonstrate reliable autonomous performance with minimal latency, showcasing the potential of using low-cost components to develop robust robotic platforms suitable for academic and research applications.

Keywords—Autonomous Navigation, ROS2, LiDAR, SLAM, Gazebo Simulation, obstacle avoidance, Raspberry Pi, Arduino, Mobile Robot

I. Introduction

Autonomous navigation has become pivotal in robotics, enabling applications in logistics, agriculture, and service industries. Traditional systems often require expensive sensors and high-performance computers. GAMYA (Geared Autonomous Mobile sYstem for Applications) addresses these challenges by leveraging affordable components—Raspberry Pi 4 and Arduino Uno R3—coupled with ROS2 and a 2D LiDAR to achieve reliable mapping, localization, and obstacle avoidance. This paper outlines GAMYA's design, implementation, and evaluation in a simulated environment.

II. Literature Review

Autonomous Mobile Robots (AMRs) are widely used in sectors like industrial automation and smart mobility. Their functionality depends on efficient integration of perception, planning, and control systems. The Robot Operating System (ROS) provides a modular platform to integrate various robotic components. Quigley et al. [1] introduced ROS as a standard for scalable robotic development. ROS2, the updated version, enhances real-time communication, security, and multi-robot operations [2]. Low-cost platforms using Raspberry Pi and Arduino have been investigated in recent research. Mohanraj et al. [6] designed an indoor robot using ultrasonic sensors but lacked SLAM capabilities. LiDAR-based approaches, like those by Ohno and Okada [4], offer better localization accuracy.

SLAM algorithms (Gmapping, Cartographer, SLAM Toolbox [7]) enable real-time 2D mapping. Macenski et al. [2] emphasized ROS2's Navigation2 stack for modular path planning. Holz et al. [5] discussed the importance of feedback in cluttered navigation environments. Despite progress, cost and complexity remain challenges for institutions with limited resources. GAMYA addresses this gap using affordable hardware and open-source tools.

It demonstrates that effective autonomous navigation is achievable on a budget.
This positions GAMYA as a practical solution for education, research, and real-world deployment.

III. System Architecture

A. Hardware Components

- **Raspberry Pi 4**: Onboard computer for ROS2 nodes.
- **Arduino Uno R3**: Low-level motor control.
- **2D LiDAR (RP-Lidar A1)**: 360° scanning for environment perception.
- **Motor Driver (L293D)**: Dual H-bridge for DC motors.
- **Differential Drive Chassis**: Two-wheel drive with encoders.

B. Software Stack

Fig. 1 illustrates the software pipeline: ROS2 Jazzy, Gazebo Harmonic, Navigation2 stack, SLAM Toolbox, and RViz2.

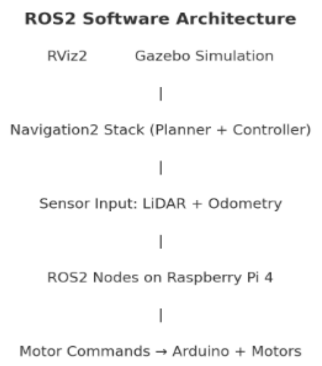


Fig.1. Software Architecture Flowchart

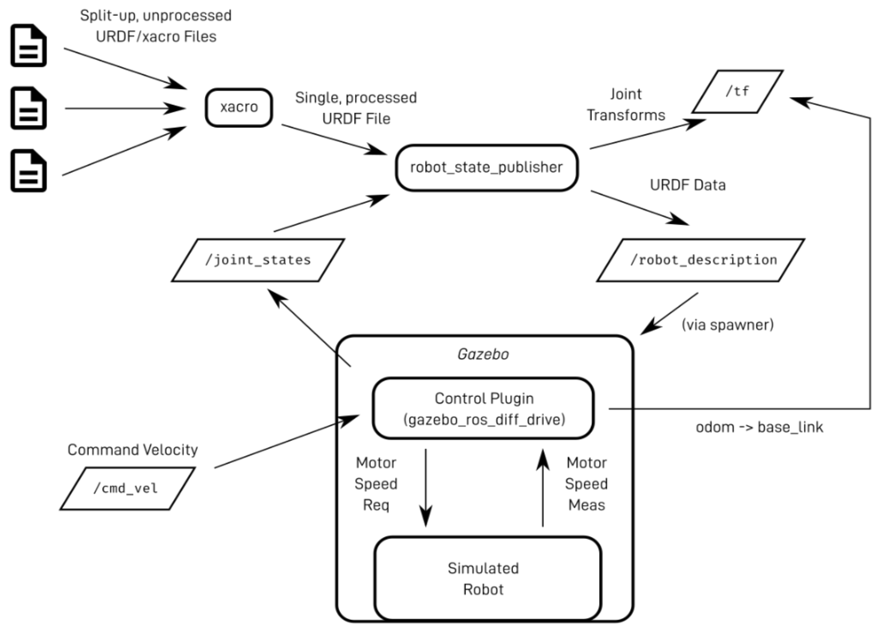
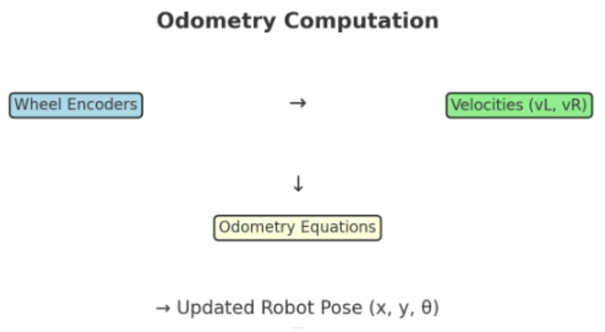


Fig 2. Block Diagram of Software Simulation Setup

IV. Kinematic Modeling and Odometry



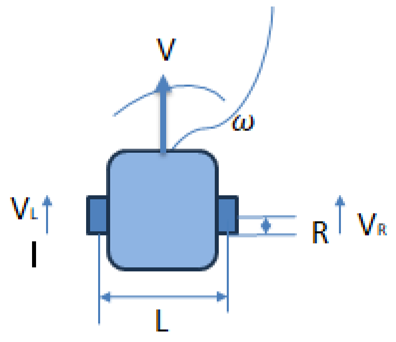


Fig. 3a – Odometry Flow Diagram Fig. 3b shows the odometry flow.

For a differential drive robot, linear velocity V and angular velocity ω relate to wheel velocities v_L, v_R as:

$$v_L = \frac{2V - \omega L}{2R}, \quad v_R = \frac{2V + \omega L}{2R}$$

where R is wheel radius, L is wheelbase. Odometry updates robot pose (x,y,θ) via:

$$x_{t+\Delta t} = x_t + V\cos(\theta_t)\Delta t,\ y_{t+\Delta t} = y_t + V\sin(\theta_t)\Delta t,\ \theta_{t+\Delta t} = \theta_t + \omega\Delta t.$$

V. Gazebo Simulation and SLAM

A. Environment Setup

A virtual world in Gazebo with static and dynamic obstacles was configured. URDF files describe robot geometry and sensor mounts. SLAM Toolbox performs 2D mapping.

B. Navigation2 Configuration

Global and local costmaps are tuned for obstacle inflation radius and sensor update rates. Behavior trees define recovery strategies.

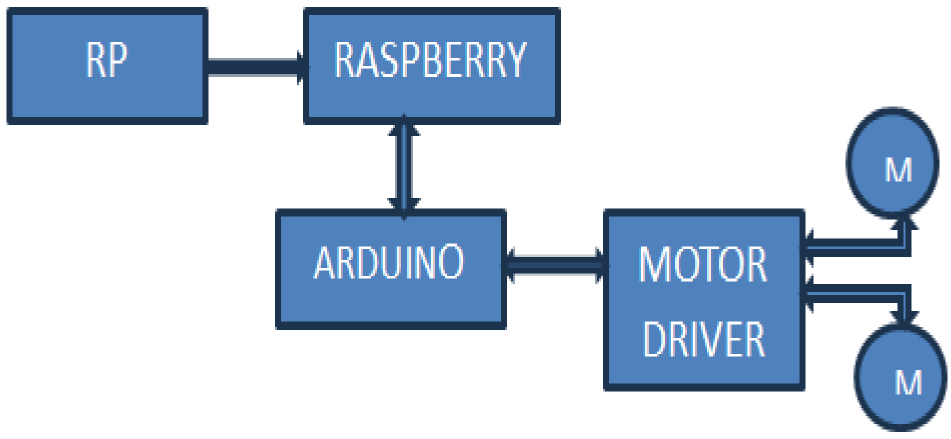


Fig. 4a. Block diagram of hardware

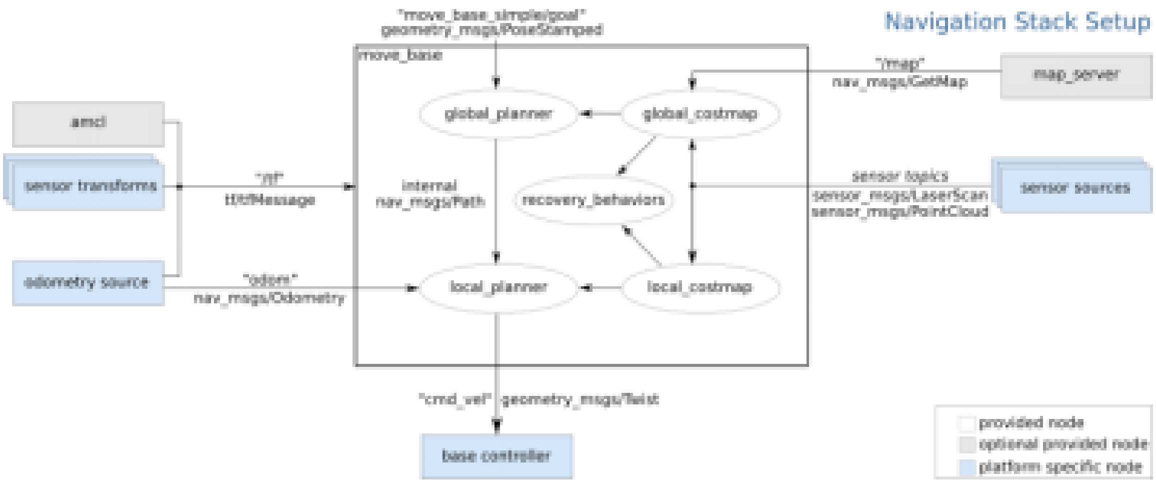


Fig 4b. Navigation Stack Setup for Autonomous Navigation Robot

VI. Results

Multiple simulation trials measuring path efficiency and collision rates were conducted. Table I summarizes performance metrics over ten runs.

Table I Simulation Performance Metrics

Metric	Mean	Std. Dev.
Path length (m)	12.5	0.8
Completion time (s)	45.2	3.4
Collision count	0	0

RViz2 visualizations (Fig.5) confirm accurate map generation and path following.

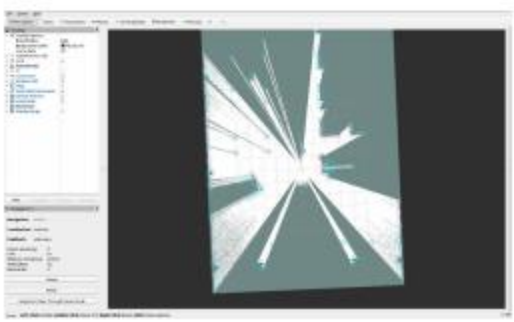


Fig 5a. Visual Representation of SLAM Toolbox for map creation
Fig 5b. Visual representation of RViz Software

VII. Discussion

GAMYA demonstrates that low-cost hardware can achieve robust autonomous navigation with ROS2. Calibration of odometry and costmap parameters was critical. Dynamic obstacle avoidance leveraged LiDAR data at 5 Hz with negligible latency.

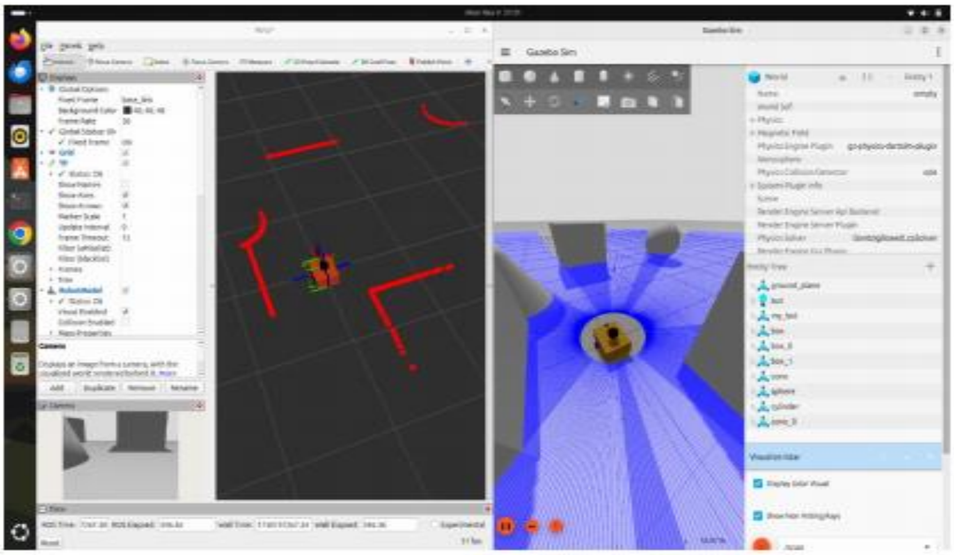


Fig 6. Simulation of ‘Gamya’ in GAZEBO & RViz

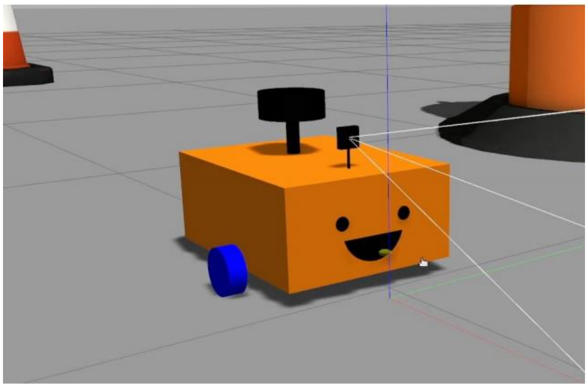


Fig. 7a. URDF model of GAMYA

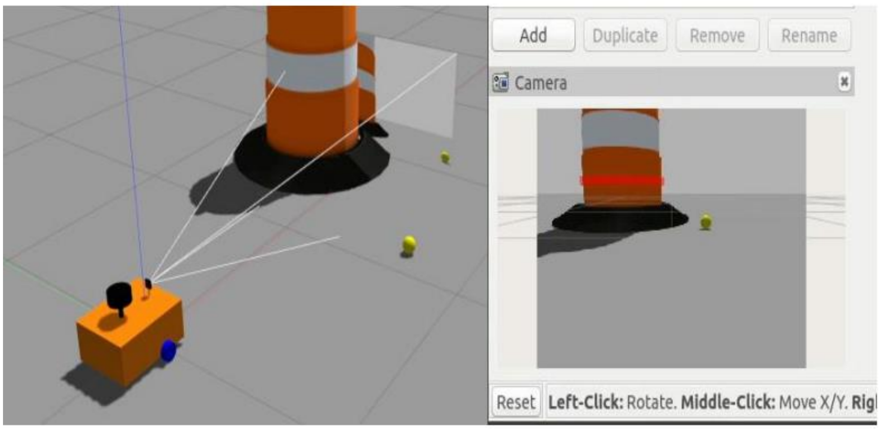


Fig.7b.CAMERA visual publishing through node

VIII. Conclusion and Future Work

This paper presents the GAMYA, an affordable ROS2-based autonomous robot validated in simulation. Future work includes real-world deployment, integration of RGB-D cameras for 3D mapping, and machine learning for adaptive path planning.

References

[1] M. Quigley et al., *Programming Robots with ROS*, O'Reilly, 2015.
[2] S. Macenski et al., "Robot Operating System 2: Design, Architecture, and Uses in the Wild," *Science Robotics*, vol. 6, no. 58, 2021.
[3] R. Arulampalam et al., "A Tutorial on Particle Filters...," *IEEE Trans. Sig. Proc.*, vol. 50, no. 2, 2002.
[4] T. Ohno & Y. Okada, "Autonomous Navigation Using ROS and 2D LiDAR," *J. Robot. Auton. Syst.*, vol. 132, 2022.
[5] D. Holz et al., "Towards Autonomous Navigation...," *Intl. J. Robot. Res.*, vol. 29, no. 14, 2010.
[6] Mohanraj, M., Suresh, R., & Pradeep, R. (2020). "Design of a Raspberry Pi-Based Autonomous Robot for Indoor Navigation Using Ultrasonic Sensors," *International Journal of Advanced Robotic Systems*, vol. 17(1), pp. 1–9.
[7] Macenski, S., & Tsotsos, K. (2021). "SLAM Toolbox: Real-Time and Lifelong Mapping in ROS2," *Journal of Open Source Software*, vol. 6(63), 2895.
[8] Kretzschmar, H., et al. (2016). "Navigation Planning with Dynamic Obstacle Avoidance for Autonomous Mobile Robots," *IEEE Robotics and Automation Letters*, vol. 1(1), pp. 290–297.
[9] Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic Robotics*, MIT Press.
[10] Fernández, C., & Sampedro, C. (2022). "ROS2-based Design of a Scalable Low-Cost Robotic Platform for Education," *IEEE Access*, vol. 10, pp. 41050–41062.
[11] Bresson, G., Alsayed, Z., Yu, L., & Glaser, S. (2017). *Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving*. IEEE Transactions on Intelligent Vehicles, 2(3), 194–220.
[12] Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., & Burgard, W. (2011). *g2o: A General Framework for Graph Optimization*. IEEE International Conference on Robotics and Automation (ICRA), 3607–3613.
[13] Koenig, N., & Howard, A. (2004). *Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator*. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2149–2154.
[14] Khan, M. Q., & Riaz, Z. (2021). *Survey on Recent Advances in SLAM*. Journal of Navigation and Control Systems, 8(2), 55–67.
[15] Chen, Y., & Liu, L. (2020). *ROS2-Based Robot Design for Teaching Autonomous Navigation*. International Journal of Advanced Robotic Systems, 17(5), 1–9.