International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 04 April 2025
Page No: 1617-1624
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0230

Intelligent Autonomous Mobile Robot with Integrated Visual & Sensor-Based Navigation

Samita Bhandari[1], Darsh Jobanputra[2], Aarya Mourya[3], Dev Raval[4]
[1]Assistant professor, Dept. of ECS, Shree L.R. Tiwari College of Engg., Mumbai, Maharashtra, India.
[2,3,4]UG Scholar, Dept. of ECS, Shree L.R. Tiwari College of Engg., Mumbai, Maharashtra, India.
Emails: *samitabhandari@slrtce.in[1], darshj.tech@gmail.com[2], aarya.a.maurya@gmail.com[3], devraval2603@gmail.com[4]*

Abstract
The rapid advancements in robotics and artificial intelligence have paved the way for the development of sophisticated Autonomous Mobile Robots (AMRs) capable of navigating and interacting with dynamic environments. In this research, an intelligent autonomous mobile robot equipped with integrated visual and sensor-based navigation is designed and developed. The robot incorporates key functionalities, including teleoperation, autonomous navigation, SLAM (Simultaneous Localization and Mapping) using LiDAR and cameras, object detection and classification, obstacle avoidance, object following, hand gesture control, and advanced path planning algorithms. Control algorithms will ensure precise movement and stability. This study focuses on creating a versatile AMR that can adapt to complex and dynamic environments, leveraging state-of-the-art technologies for high precision and reliability.

Keywords: *Autonomous Mobile Robot, Navigation, SLAM, Object Detection, Path Planning*

1. Introduction
The field of Autonomous Mobile Robots (AMRs) has emerged as an important domain in robotics, based on the developments in artificial intelligence and sensor technology. AMRs are meant to navigate and execute tasks in dynamic environments independently, without any human control. One of the biggest challenges in this domain is to equip robots with the ability to make correct decisions from real-time inputs from their environment, which means combining visual and sensor-based navigation systems for optimal autonomy. As AMRs develop, they are gaining more and more ability to execute complex tasks with high accuracy, and hence becoming an integral component of modern automation. This project suggests designing an intelligent AMR with a set of sensors like LiDAR, IMU, and Cameras so that the robot can move and sense its environment correctly. These sensors offer the data required for tasks such as obstacle detection, localization, and path planning. Its modular architecture allows it to be built for other purposes. Its robust computational system relies on cutting-edge microcontrollers and single-board computers for the rapid processing of complex data. With the ongoing advancements in robotics, these technologies allow autonomous mobile robots to operate more efficiently than human workers. The possible use cases of the robot include robotics education, warehouse automation, healthcare, surveillance, and environmental monitoring. The use of both visual and sensor-based navigation means the robot is able to work in the real world, being flexible and performing its task with ease.

2. Literature Review
Autonomous Mobile Robots (AMRs) are at the forefront of robotics research, particularly due to their ability to navigate and interact with dynamic environments. They have been widely used in industries such as logistics, healthcare, manufacturing, and surveillance. However, for these robots to operate autonomously and efficiently, they require integrated technologies for navigation, perception, and control. Below is an overview of the critical components of AMRs based on recent

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 04 April 2025
Page No: 1617-1624
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0230

research:

2.1.Multi-Sensor Integration

Sensor systems are a vital component for enabling AMRs to perceive and interact with their environment. LiDAR (Light Detection and Ranging) is widely used for mapping and obstacle detection due to its precision in measuring distances [1]. Cameras complement LiDAR by providing visual data, which is essential for object detection, classification and camera-based navigation techniques—such as lane detection and line following, which utilize visual cues to guide the robot along predefined paths. IMUs (Inertial Measurement Units) are used to provide orientation data, improving the robot's localization accuracy. Research by Thakur, A., et al. (2023) combined LiDAR and cameras to create more accurate maps of environments and enhance obstacle detection, demonstrating the growing importance of sensor fusion in AMR development [2].

2.2.Simultaneous Localization and Mapping (Slam)

SLAM allows robots to explore and map unknown environments while keeping track of their location. Dissanayake et al. (2001) introduced probabilistic SLAM, which allowed robots to localize themselves in 2D environments [3]. More recent studies have improved SLAM for complex, 3D environments by integrating data from multiple sensors. The fusion of LiDAR and visual SLAM techniques has been shown to improve mapping accuracy in dynamic environments, as seen in the work by Qian, J., et al. (2019) [4]. These advancements are crucial for AMRs operating in real-world, unpredictable conditions where environmental features change frequently.

2.3.Object Detection and Classification

Object detection is a critical capability for AMRs, as it enables the robot to identify and interact with objects within its environment. Recent advancements in deep learning have improved the accuracy of object detection in dynamic settings. Aulia, U., et al. (2024) showed that Convolutional Neural Networks (CNNs) could successfully classify and detect objects in cluttered environments, improving the robot's ability to safely navigate and avoid obstacles [5]. Real-time object detection has become a key focus, allowing AMRs to avoid collisions and perform tasks such as object manipulation and tracking [6].

2.4.Path Planning Algorithms

Path planning ensures that an AMR can find the most efficient route while avoiding obstacles. Classical algorithms, such as A* and Dijkstra's, are effective in static environments but are less efficient in dynamic, real-time conditions. To address these limitations, researchers have focused on more advanced algorithms, such as Rapidly-exploring Random Trees (RRT) [7], which are better suited to environments where obstacles change frequently. Additionally, the Dynamic Window Approach (DWA) allows AMRs to avoid collisions by dynamically adjusting their path in response to immediate obstacles [9]. These methods are essential for ensuring safe and efficient navigation in unpredictable settings.

2.5.Control Algorithms for Precision and Stability

Control algorithms play a key role in ensuring that AMRs move with precision and stability. Proportional-Integral-Derivative (PID) controllers have been widely used due to their simplicity and effectiveness in stabilizing robot movement. More sophisticated methods, such as Model Predictive Control (MPC), have gained popularity for their ability to account for both current and future states of the system [10]. These advanced control strategies allow AMRs to maintain stability and navigate complex environments with high precision.

2.6.Challenges and Future Directions

Despite the progress in AMR research, several challenges remain, particularly in the areas of multi-sensor fusion, real-time decision-making, and adaptive navigation in dynamic environments. Reinforcement learning and AI-driven techniques are being explored to enhance decision-making and adaptability. These technologies will improve the robot's ability to react to unforeseen changes in its environment, ensuring more robust and autonomous operations [12]. Future research will also likely focus on integrating new sensor technologies such as radar and ultrasonic sensors to enhance AMRs' capabilities, particularly in low-visibility or challenging environments.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 04 April 2025
Page No: 1617-1624
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0230

3. Proposed System

The proposed system is an Autonomous Mobile Robot (AMR) designed to achieve high-precision navigation in dynamic environments through multi-sensor fusion, adaptive decision-making, and split-compute architecture. Unlike conventional platforms reliant on single-sensor perception or static algorithms, this AMR integrates a 360° 2D LiDAR, monocular vision, and ROS2 middleware to balance computational efficiency with real-time responsiveness, addressing challenges like dynamic obstacle avoidance and human-robot interaction [2]. The System Objectives are:-

- Teleoperation
- Autonomous Navigation
- Object Detection and Classification
- Dynamic Obstacle Avoidance
- Usage of Path Planning algorithms
- Implementation of Control Algorithms

3.1.System Architecture

The proposed system uses a split-compute ROS 2 architecture which separates perception, planning, and control tasks among dedicated hardware layers. This promotes scalability and improves real-time performance, using the ESP32-WROOM-32E for low-level control activities and the RaspberryPi-4B for advanced processing. The architecture modules are:- (Figure 1)

3.2.Sensor Data Acquisition

This module collects real-time environmental data using multiple sensors, including the YDLIDAR-X4 2D LiDAR for 360° range-finding, the MPU6050 IMU for tracking motion, and the Lenovo 300 FHD camera for visual input. The LiDAR records high-resolution Point Clouds for spatial mapping, with the IMU delivering accelerometer and gyroscope data for odometry correction. The monocular camera supplies RGB images for object detection and gesture recognition. Raw sensor data is published to specialized ROS2 topics (e.g., /scan, /imu/data, /image_raw) for further processing [3]. (Figure 2)

3.3.Teleoperation Module

This module enables keyboard control via *ROS2* for initial testing, calibration, and human-in-the-loop validation in edge-case scenarios (e.g., cluttered sites & narrow passages) [8].

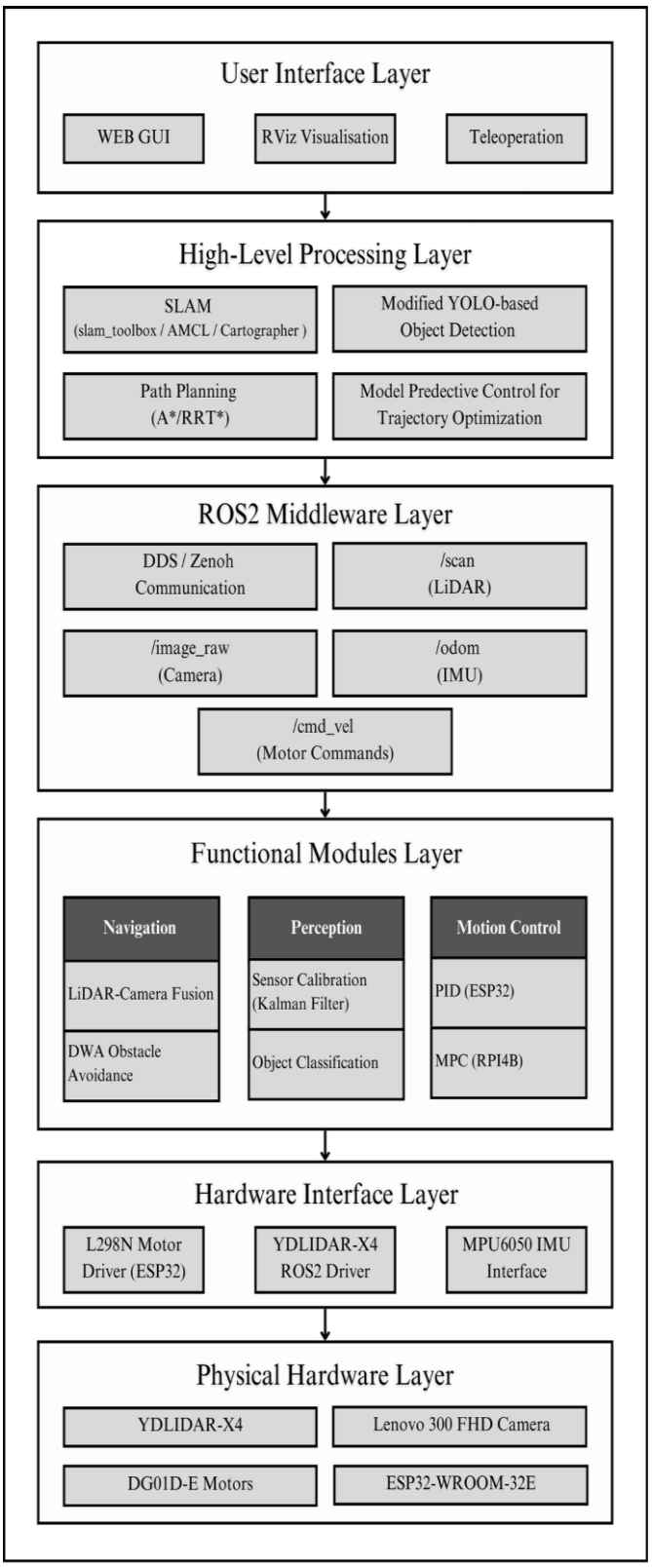

Figure 1 Proposed AMR System Architecture Block Diagram

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 04 April 2025
Page No: 1617-1624
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0230

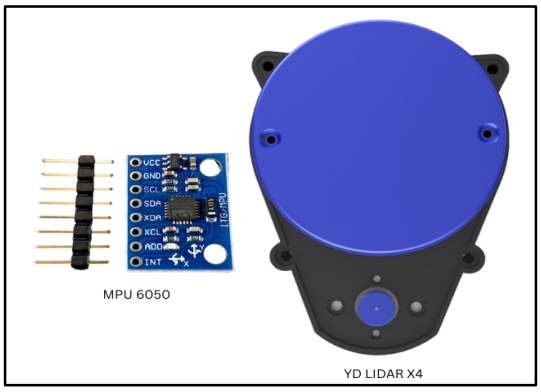

Figure 2 MPU-6050 & YD LIDAR X4

3.4.Sensor Data Preprocessing

Raw sensor data are calibrated and synchronized to be compatible with downstream modules. The ESP32-WROOM-32E microcontroller filters IMU noise and publishes corrected odometry data to the /odom topic. On the Raspberry Pi 4B, LiDAR scans and camera feeds are temporally aligned using ROS2 message_filters, whereas camera images are resized and normalized for maximum computation efficiency. Noise-free, synchronized data for perception tasks are achieved at this step [2]. (Figure 3,4)

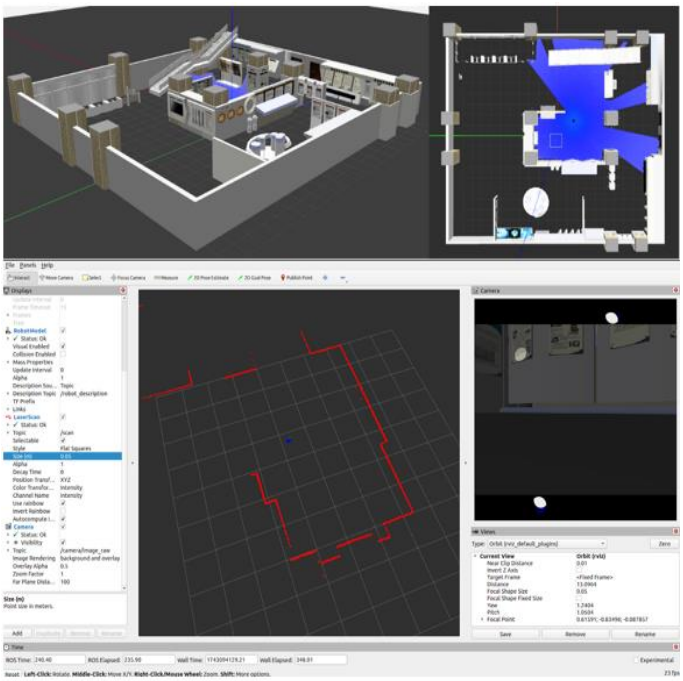

Figure 3 Gazebo Simulation (/scan & /camera_raw)

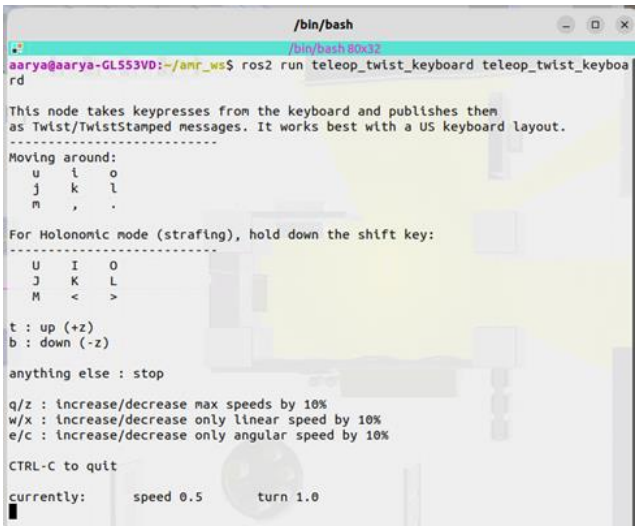

Figure 4 Teleop_Twist_Keyboard

3.5.SLAM Module

This module uses LiDAR-based Simultaneous Localization and Mapping (SLAM) to create 2D occupancy grid maps of the environment along with estimating the robot pose. The point cloud information from the YDLIDAR-X4 is combined with MPU6050 IMU data for odometry correction, improving pose estimation accuracy in dynamic or rough terrains. The system prevents the use of a single localization method, providing flexibility to incorporate probabilistic, filter-based, or optimization-based SLAM methods as needed [3]. (Figure 4)

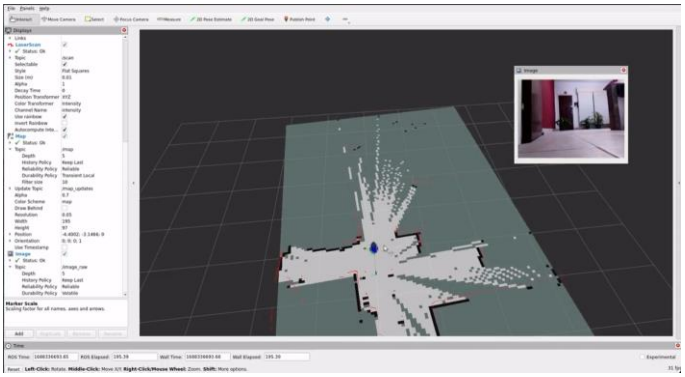

Figure 5 Simultaneous Localization & Mapping

3.6.Obstacle Detection and Classification Module

Real-time obstacle detection is enabled by a light, modified *YOLO* model running on the Raspberry Pi

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 04 April 2025
Page No: 1617-1624
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0230

4B. The Lenovo 300 *FHD* monocular webcam provides 1080p images to the model, which detects and classifies obstacles with bounding boxes [6]. Detected objects are published on the /detected_objects topic, guiding navigation choices while favoring low-latency inference over computationally heavy models such as ResNet-based CNNs [5]. (Figure 6)

Figure 6 Object Detection & Classification

3.7.Path Planning Module

Global and local path planning are separated to achieve efficiency and flexibility. The global planner computes optimal paths based on graph-based methods (e.g., A*) on pre-constructed maps, and the local planner adaptively corrects trajectories in real time based on sampling-based techniques (e.g., RRT*) to fit around obstacles that are sensed by LiDAR or vision [7]. (Figure 7)

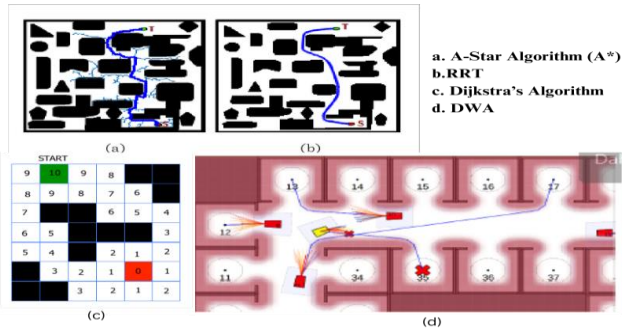

Figure 7 Algorithm-(A*, RRT, Dijkstra's, DWA)

3.8.Obstacle Avoidance Module

Using current LiDAR scans and camera-detected obstacles, the Dynamic Window Approach (DWA) considers potential velocity profiles. Then, the robot can select safe paths with the help of Costmaps which dynamically update obstacle positions by merging LiDAR point clouds and object detection results. In cluttered, or uncertain environments, this reactive strategy lowers the risk of collisions [9].

3.9.Control Module

Motor control is divided between hardware layers for accuracy and stability. The ESP32-WROOM-32E runs PID controllers to control wheel velocities through PWM signals to the L298N Dual-H Bridge motor driver, with encoder feedback from motors, for closed-loop control. The Raspberry Pi 4B, in turn, operates high-level Model Predictive Control (MPC) to plan trajectories, publishing velocity commands to the /cmd_vel topic [10].

3.10. Communication Interface

Inter-module communication is handled by middleware Robot Operating System 2 (ROS2) based on protocols like DDS or Zenoh with minimal latency. Sensor readings and motor commands are transmitted in real time from ESP32-WROOM-32E to Raspberry Pi 4B with the help of micro-ROS. External devices like keyboards (for teleoperation) are interfaced through standard ROS2 interfaces like teleop_twist_keyboard [8].

3.11. Reporting and Analytics Module

ROS2 inspection tools monitor system performance metrics including navigation latency, path error, and obstacle collision rate. Visualization tools like RViz, Foxglove and rqt_plot offer actionable data such that iterative parameter adjustment of SLAM, planning, and control can be realized [11]. (Figure 8)

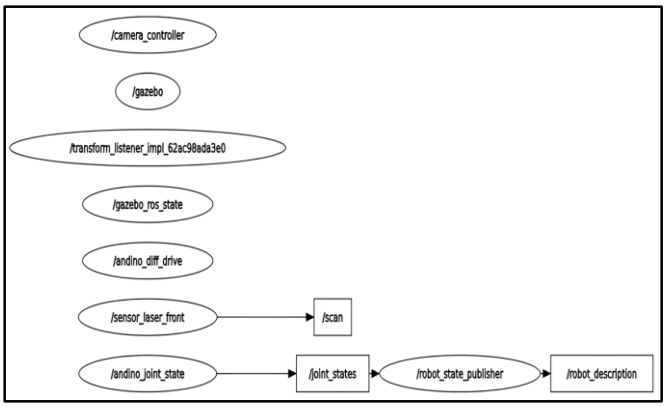

Figure 8 rqt_graph

4. Methodology

4.1.System Design

The first step is to design the hardware and software architecture of the robot simultaneously in iterative

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 04 April 2025
Page No: 1617-1624
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0230

simulations. A custom chassis is designed using CAD tools like Autodesk Fusion to offer mechanical stability, heat sink, and optimal placement of sensors (LiDAR, IMU, monocular camera). Raspberry Pi 4B 8GB RAM for high-level processing and the ESP32-WROOM-32E Dev Kit for real-time processing, are selected for their low-cost and high-performance features. The ROS2 framework is used to develop modular communication interfaces between sensors, algorithms, and actuators to offer scalability for future additions.

4.2.Sensor Integration & Data Fusion

In the second step, sensors are calibrated and synchronized to allow robust perception. YDLIDAR-X4 LiDAR and MPU6050 IMU are synchronized in time using ROS 2's package message_filters and geometric transformations to match the field-of-view of the camera with LiDAR scans. Raw data are denoised by median filtering LiDAR outliers and Gaussian blur for camera streams. Kalman filter-based fusion of LiDAR odometry with IMU measurements improves localization in dynamic environments [2].

4.3.Algorithm Development

In the third step, navigation algorithms are created and tested in simulation. SLAM is performed with LiDAR and a custom dataset is used to train a lightweight, modified YOLO algorithm for real-time object detection. Path planning uses A* for global path planning and RRT* for local obstacle avoidance [7], with Dynamic Window Approach (DWA) dynamically adjusting trajectories [9]. PID and MPC algorithms are co-designed such that PID gains are tuned on the ESP32 for motor control, and MPC optimizes the trajectories on the Raspberry Pi. These algorithms are tested in Gazebo, with different obstacles and lighting conditions.

4.4.System Integration & Testing

In the next step, the physical prototype is tested in limited real-world settings, e.g., indoor spaces with obstacles. Sensor fusion pipelines are tested against motion capture system ground-truth, obstacle avoidance against static and moving obstacles. ROS2 bag files record navigation metrics (e.g., path deviation, collision rates) for iterative tuning. Fail-safes, e.g., teleoperation override, are added to account for edge cases during field tests.

4.5.Performance Evaluation

In the final step, the robot's performance is measured in terms of metrics such as localization error (RMSE w.r.t. ground truth), task execution time, and obstacle detection accuracy. State-of-the-art baselines vs. platforms (e.g., TurtleBot3, TortoiseBot, etc.) evaluate cost and flexibility. Bottlenecks such as computation delay in high-density environments are detected through result analysis and applied to guide firmware optimizations and hardware developments [13]. (Figure 9)

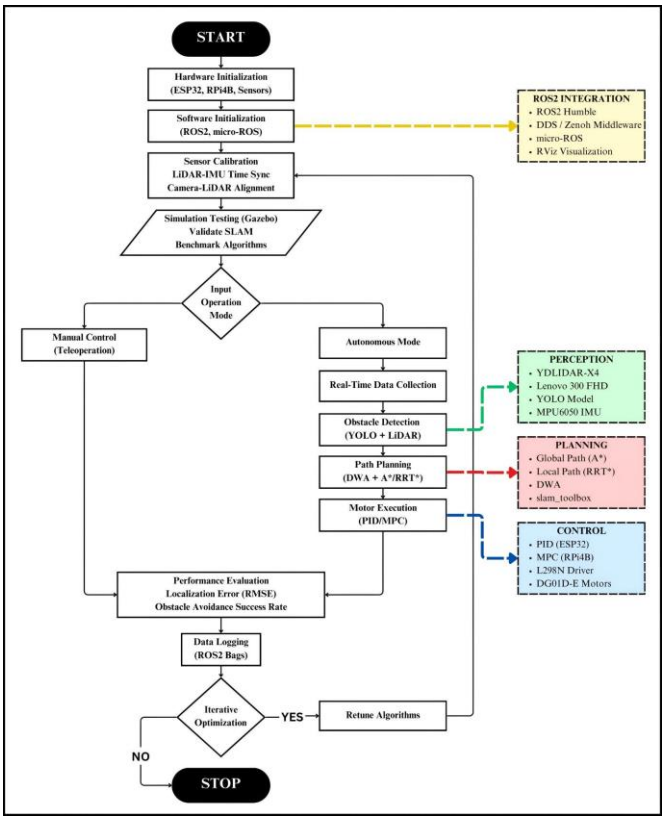

Figure 9 Proposed AMR System Flowchart

5. Future Scope

The proposed system sets a solid theoretical and architectural foundation for a smart AMR, with the next step of physical implementation through integration of modular hardware (ESP32, Raspberry Pi 4B, YDLIDAR-X4, and DG01D-E motors) and the ROS2 software stack. Controlled testing will verify the precision of SLAM, object detection, and path planning, measuring localization error (RMSE),

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 04 April 2025
Page No: 1617-1624
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0230

obstacle avoidance rates, and computation latency. Subsequent work will be in real-world deployment in dynamic environments like cluttered indoor environments and mazes, with challenges like synchronization of sensors, motion on uneven surfaces, and power optimization via hardware-software co-design. Scalability will be experimentally validated through sensor addition (e.g., thermal cameras or depth sensors to improve low-light object detection) or multi-robot coordination software stacks like Open-RMF. Long-term goals potentially include open-sourcing the robot's ROS2 packages, using edge AI accelerators (e.g., NVIDIA Jetson Nano) for real-time processing improvement, and industry partnerships.

Conclusion

The development of an intelligent autonomous mobile robot (AMR) with integrated visual and sensor-based navigation addresses the complexity of dynamic environment interaction. Using multisensor fusion (LiDAR, monocular camera, and IMU), modular ROS2 architecture, and adaptive techniques such as SLAM, DWA, and Hybrid Path Planning, the system achieves significant autonomy in unstructured indoor environments. The split-compute architecture, featuring a Raspberry Pi 4B 8GB RAM for high-level perception and an ESP32-WROOM-32E for real-time control, offers a cost-effective and scalable solution. This design bridges the gap between industrial systems and academic prototypes, for real-world applications. Future advancements could involve AI-driven adaptive navigation using reinforcement learning. The modularity allows for easy integration of additional sensors like thermal cameras, ultrasonic sensors, etc. for diverse applications. Open-sourcing ROS2 packages for diagnostics and LiDAR-camera fusion promotes community innovation with low-cost robots. This AMR has the potential to transform industries like robotics education, warehouse logistics, etc. by safely navigating dense areas, ultimately enhancing productivity and safety.

References

[1]. Thrun, S. (2002). Probabilistic robotics. Communications of the ACM, 45(3), 52-57. doi: 10.1145/504729.504754.

[2]. Thakur, A., & Rajalakshmi, P. (2023). LiDAR and Camera Raw Data Sensor Fusion in Real-Time for Obstacle Detection. Proceedings of the 2023 IEEE Sensors Applications Symposium (SAS), 1-6. doi: 10.1109/SAS58821.2023.10254075.

[3]. Dissanayake, M. W. M. G., Newman, P., Clark, S., Durrant-Whyte, H.F., & Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. IEEE Transactions on Robotics and Automation, 17(3), 229-241. doi: 10.1109/70.938381.

[4]. Qian, J., Chen, K., Chen, Q., Yang, Y., Zhang, J., & Chen, S. (2022). Robust Visual-Lidar Simultaneous Localization and Mapping System for UAV. IEEE Geoscience and Remote Sensing Letters, 19, 1-5. doi: 10.1109/LGRS.2021.3099166

[5]. Aulia, U., Hasanuddin, I., Dirhamsyah, M., & Nasaruddin, N. (2024). A new CNN-based object detection system for autonomous mobile robots based on real-world vehicle datasets. Heliyon, 10(15), e35247. doi: 10.1016/j.heliyon.2024.e35247

[6]. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788. doi: 10.1109/CVPR.2016.91

[7]. Zammit, C., & van Kampen, E.-J. (2022). Comparison Between A* and RRT Algorithms for 3D UAV Path Planning. Unmanned Systems, 10(2), 129-146. doi: 10.1142/S2301385022500078

[8]. Kawabata, K., Ishikawa, T., Fujii, T., Noguchi, T., Asama, H., & Endo, I. (1998). Teleoperation of autonomous mobile robot under limited feedback information. Field and Service Robotics, 146-151. doi: 10.1007/978-1-4471-1273-0_24

[9]. Qin, H., Shao, S., Wang, T., Li, Y., Wang, N., & Yao, C. (2022). An improved dynamic window approach for mobile robot dynamic

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 04 April 2025
Page No: 1617-1624
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0230

path planning. Proceedings of the 12th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), 348-353. doi: 10.1109/CYBER55403.2022.9907305

[10]. Chen, S., & Chen, H. (2020). MPC-based path tracking with PID speed control for autonomous vehicles. IOP Conference Series: Materials Science and Engineering, 892, 012034. doi: 10.1088/1757-899X/892/1/012034

[11]. Thale, S. P., Prabhu, M. M., Thakur, P. V., & Kadam, P. (2020). ROS-based SLAM implementation for autonomous navigation using Turtlebot. ITM Web of Conferences, 32, 01011. doi: 10.1051/itmconf/20203201011

[12]. Shen, Z., Duan, Y., Cong, Y., Li, W., Du, H., & Zhu, W. (2023). Deep Reinforcement Learning-based Multi-AMR Path Planning Algorithm. Proceedings of the 2023 Chinese Control Conference (CCC), 1984-1987. doi: 10.23919/CCC58697.2023.10240541

[13]. Ferreira, M. A., Moreira, L. C., & Lopes, A. M. (2024). Autonomous navigation system for a differential drive mobile robot. ASTM International Journal of Testing and Evaluation, 52(2), 841-852. doi: 10.1520/JTE20230191