Maze Solver using ROS2

Darshan Chougule[1], Vasudha Dhanokar[2], Atharva Joshi[3*], Vishal Hange[4], Pravin Gawande[5]

[1]darshan.22210611@viit.ac.in, [2]vasudha.22210740@viit.ac.in, [3]atharva.22211561@viit.ac.in, [4]vishal.22210296@viit.ac.in, [5]pravin.gawande@viit.ac.in

Electronics and Telecommunications Engineering Department
Vishwakarma Institute of Information Technology Pune-411048, Maharashtra, India

***Corresponding Author**:
Atharva Joshi atharva.22211561@viit.ac.in
Electronics and Telecommunications Engineering Department
Vishwakarma Institute of Information Technology
Pune-411048, Maharashtra, India

Abstract

The research investigates how to create an autonomous robot for maze solving under the Robot Operating System 2 (ROS2) framework. The main task requires the development of a modular robotic system that generates real-time data collection capabilities and dynamic mapping functions and determines efficient navigation solutions. The robotic system uses sensor perception along with SLAM and path planning algorithms combined into an ROS2 framework structure. The system processes the sensor data to generate a dynamic environmental representation that guides efficient navigation strategies until it reaches the target location. The system components use ROS2 middleware protocols to establish low-latency decentralized control communication. The system undergoes experimental testing in simulated Gazebo spaces and physical applications, which verifies its performance capabilities and unknown maze structure compatibility. The research results demonstrate that ROS2 can power genuine time operational robotic movement through complex dynamic spaces.

Keywords: ROS2, Autonomous Navigation, Maze Solving Robot, Simultaneous Localization and Mapping (SLAM), Path Planning, Robotic Perception, Real-Time Systems, Gazebo Simulation, Robot Middleware, Autonomous Robotics

I. Introduction

Problems with autonomous navigation of mobile robots through unmapped structured spaces continue to be an essential research topic that needs formal testing through maze environments as per [1] and [8]. Wall-following methods together with exhaustive graph searches remain simple to implement; however, their performance declines significantly when maze complexity rises [12]. Reactive strategies work blindly without awareness of their environment, thus causing robots to continually traverse the same ground space while creating less than ideal outputs in unfamiliar or changing robotic domains [5].

Modern development approaches unify SALM with optimal and heuristic path-planning functionalities to permit developers to generate consistent environment maps alongside selecting energy-efficient navigation routes. The global path planning standard uses A* due to its efficiency and admissibility with gridbased representation, although sampling approaches (RRT and PRM examples) together with dynamic replanning algorithms perform best in complex environments according to [5] and [4]. Through frontierbased exploration, robots receive better coverage outcomes due to their ability to detect unknown spaces between existing known areas, thus systematically reducing environmental uncertainty [8].

The ROS2 framework provides middleware capabilities for deftly connecting navigation core elements into working navigation protocols. The adoption of Data Distribution Service (DDS) by ROS2 provides real-time performance with automated discovery and fault tolerance features compared to ROS1, since ROS1 lacks these capabilities [14]. Inside this ecosystem, the Navigation2 (Nav2) stack contains customizable plugins for SLAM (e.g., Cartographer, ORB-SLAM3) as well as global and local planners alongside costmap generation and behavior trees [3] [11]. Implementations of recent times show how LiDAR- and vision-based SLAM integrate with BF global planning together with Dynamic Window Approach (DWA) local control to provide reliable maze navigation that works both in Gazebo simulations and real-world conditions [10] and [4].

Several obstacles exist for maximizing effective exploration under sensor uncertainties and achieving secure loop closure with scarce features and enabling speedy re-planning capability for dynamic obstacles or dead ends. The system design must handle SLAM precision and computation speed versus map quality since these three elements create intricate trade-offs. The ROS2 maze system provides a solution which unites SLAM based on LiDAR data with frontier exploration methods and hybrid navigation planning strategies. Our work revolves around three principal components: a maze-appropriate Nav2 configuration, an intelligence-driven frontier selection module, and quantitative analysis of time-performance and pathoptimality relative to standard systems as documented in [1].

II. Literature Review

Autonomous robotics research now focuses on building reliable maze-solving methods which use the recent developments in ROS2 navigation frameworks. Modern maze-solving techniques now construct entire environmental maps followed by path calculations instead of using traditional wall-following methods and exhaustive search. Once a map becomes available grid-based A* search provides the most effective solution for discovering shortest paths in navigation [1] and heuristic techniques based on micromouse competitions optimize maze exploration [12]. Maze environment training employs both reinforcement learning and evolutionary algorithms as learning-based approaches to develop navigation policies [7][4].

The integration of traditional methods with contemporary learning procedures improves real-time mazesolving abilities effectively [1][7].

The implementation of ROS2 (Robot Operating System 2) enables the combination of mapping with planning and control for mazes. ROS2 delivers a flexible middleware platform which features real-time functionality in addition to sustaining multiple robot elements and enhanced communication than its ROS1 version [14]. The recent ROS2 Navigation2 (Nav2) stack contributes a recent framework which allows users to customize global and local planners alongside controllers and behavior trees for navigation across known as well as unknown mapping terrains [3][11]. The ROS2 navigation architecture receives an explanation from Patel et al. while the authors demonstrate autonomous exploration through LiDARbased mapping with planned obstacle avoidance [3]. Nguyen et al. establish a comparison between ROS1 and ROS2 through their analysis of DDS-based communication and component lifecycle features which enhance the integration of SLAM and path planning modules [14]. ROS2 facilitates easy development of maze-solving robots by delivering an integrated system which merges sensors and mapping and path navigation functions [3][11].

Exploration of unknown mazes depends fundamentally on an implementation of Simultaneous Localization and Mapping (SLAM). The latest SLAM systems use visual and LiDAR technology to deliver high-precision mapping procedures and position tracking [6][10]. The research by Gupta et al. examines how visual and LiDAR and RGB-D SLAM systems differ regarding their accuracy and operational ranges as well as processing costs [6]. ORB-SLAM3 demonstrates real-time loop closure abilities as well as state-of-the-art visual SLAM performance in practical applications [10]. The combination of g2o as a graph-based SLAM back-end with the scan-matching component Cartographer from Google produces 2D maps that show consistent results. These SLAM capabilities are regularly included in ROS2-based frameworks to construct maze environment maps incrementally, which later supports path planning and decision-making functions [3][4].

Path planning operations in maze navigation adopt a dual approach for their execution. A global planner evaluates the optimal route based on mapped areas using A*, D*, or Theta* algorithms, and afterward the local planner manages execution constraints along with handling dynamic obstacles [5]. A* and D* algorithms represent widely applied static environment planners, but RRT and PRM should be combined with dynamic replanning to handle complex environments, according to Gupta and Chen's survey [5]. Frontier-based exploration represents an established approach to unknown maze environments where robots locate frontiers between known and unknown areas before moving to discover new parts of the space [8]. Path planning methods with multiple objectives have been developed to optimize both route length and safety distance along with power utilization requirements at the same time [5][8]. The path planning algorithms serve as plug-in modules in ROS2 Nav2, which enables the maze solver to modify its course of action when the system receives updated map data.

Researchers have developed several recent examples demonstrating how these elements connect to functioning maze-solving platforms. According to [4], Johnson et al. show how

ROS2 Navigation2 uses LiDAR SLAM together with A* global planner and DWA local controller to automatically navigate through unknown maze environments. Miller and Garcia conducted research on transferring maze-solving policies from Gazebo simulation to physical robots by using ROS2 for both simulation control and onboard autonomy operations [13]. Research demonstrates that ROS2 provides a framework for integrating SLAM with global planners and local controllers as modular components [3][4][13]. Maze navigating research efforts benefit from interchangeable perception and planning algorithms through ROS2 modularity which leads to faster development of dependable systems. The analyzed research works over the last five years display the advanced state of maze solutions which combine classical graph theory with contemporary SLAM and path-mapping capabilities through ROS2 modular integration [1][3][6][13]. The latest navigation systems achieve autonomous maze exploration and solution within complex environments at high reliability through the integration of SLAM mapping with planning using A* and RRT and ROS2 Nav2 middleware [3][4][13].

III. Methodology

A. Hardware Components

The research maze-solving robot uses dependable and compact hardware components which were chosen for their performance during real-time navigation operations. The primary components are:

- Because of its dual-core structure and wireless functionality, the ESP32 microcontroller provided low-level motor commands and sensor union.
- ROS2 managed mapping and planning and control operations through the main processing unit, which consisted of a Raspberry Pi 4 Model B.
- The 2D LiDAR sensor operated as the main precision tool to measure distances needed for both SLAM and robot obstacle avoidance during map creation and environmental perception.
- N20 motors were chosen for robot locomotion because they provided high torque along with appropriate motor drivers, which enabled the robot to move precisely through the maze.
- Multiple components, including ultrasonic sensors and motor encoders with power management circuits as well as chassis elements, were integrated to achieve stable operation throughout extended periods.

Different hardware modules provided a balanced combination of processing speed alongside precision sensing along with operational mobility, which enabled robot self-navigation in mazes autonomously.

B. Navigation and Maze Solving

The navigation strategy adopted an efficient method for maze exploitation by connecting traditional algorithms to contemporary mapping procedures. Border-Based Exploration served as the execution system which identified unknown areas by determining the limits between known and unknown spaces for robot navigation. The ongoing SLAM algorithms allowed the robot to develop step-by-step maps of its environment while moving.

Global navigation began after a complete or partial map became available through implementation of the BF algorithm [1]. The local planning system operated alongside the main task to detect and steer around instant obstacles and maintain continuous plan execution. The overall maze-solving methodology emphasized:

- Efficient exploration using frontiers.
- Real-time obstacle avoidance and dynamic local planning.
- The path optimization process started after mapping reached its desired state.
- The system includes recovery procedures which manage dead ends and unanticipated mapping problems.

The advanced strategy used multiple steps to maintain stability throughout the system when navigating complicated and incomplete maze environments.

C. ROS2 Implementation

Autonomy for the robot functioned through Robot Operating System 2 (ROS2), which employed modular and real-time operational features. Here are the main components that made up the ROS2-based implementation:

- Cartographer ROS2 operated as the SLAM Module to generate real-time 2D occupancy grid maps by performing simultaneous localization and mapping.
- A combination of planning functions ran inside the Navigation2 (Nav2) stack. The BF global planner plugin produced efficient paths together with the DWB (Dynamic Window Approach) local controller that handled dynamic obstacle avoidance tasks.
- The ROS2 nodes processed sensory data through LiDAR and ultrasonic sensors for achieving strong environmental perception capabilities.
- The Nav2 allows users to adjust high-level navigation strategies through its behavior tree framework, which includes recovery behaviors as well as task-switching capabilities.
- The communication system based on ROS2 DDS makes sure that nodes exchange messages with reliable low-latency performance, which is essential for real-time decision making.

ROS2 functions as the essential element for creating a versatile and dependable autonomous maze-solving system that requires modular capabilities and both scalable fulfillment and real-time message transmission.

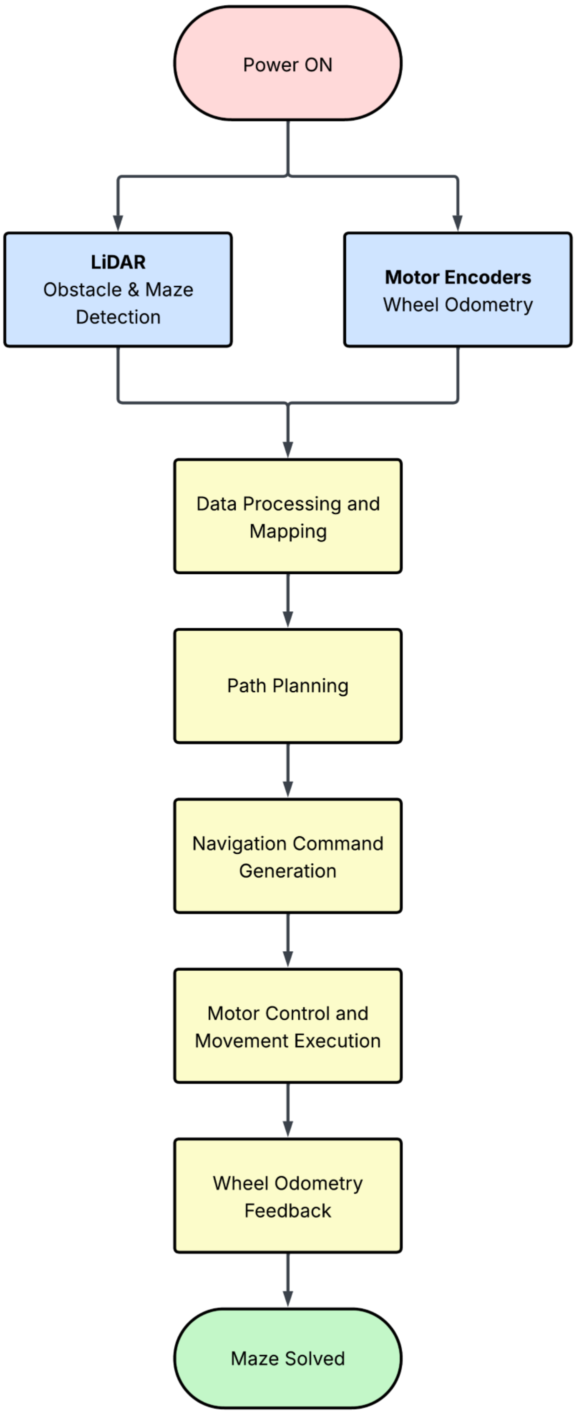

Fig. 1: Workflow of the Autonomous Maze-Solving Robot

IV. Results

The autonomous maze-solving robot successfully performed unassisted navigation and solution of indoor maze environments. Research activities took place inside controlled areas that used complex mazes built with barriers and obstacles which imitated actual navigation circumstances.

Fig.2 shows the final prototype of the robot. The robot generated effective maps through real-time SLAM of unknown spaces while using integrated planning methods to produce optimal navigation routes.

Fig. 2: Final Prototype of the Autonomous Maze-Solving Robot

The experimental trials demonstrated these results:

- The robot successively mapped its exploration path through the maze environment as it moved to create an accurate occupancy grid that supported efficient navigation planning.
- The robot determined and executed the most suitable route for reaching its desired goal position after finishing its navigation exploration period based on BF-based global planning.
- Through its implementation of the local planner the robot could perform responsive adjustments to sudden obstacles along with environmental anomalies requiring no human involvement.
- The real-time communication of ROS2 together with its modular navigation stack produced stable system behavior that prevented major communication problems or unexpected robot behaviors throughout the operation.
- A few minutes were required for the robot to map medium-sized mazes while navigating successfully to the goal depending on maze complexity.

The final prototype metrics indicate that autonomous robots can function successfully based on ROS2 and path planning algorithms connected to low-cost components. The robot experienced localized deviations in position as well as minor route deviations which did not interrupt the completion of its operations.

The robot's navigation behaviors, mapping abilities, path planning procedures were recorded in visual form to test system performance levels.

V. Conclusion

The research project developed an autonomous maze-solving robot through extensive testing, which used Robot Operating System version 2 (ROS2) to control communication and computation functions and process management. The robot included an ESP32 microcontroller to manage motor control together with a Raspberry Pi 4 processor and LiDAR sensors for perceiving the environment and N20 motors for actuation. The microcontroller, along with the computer CPU and actuation motors, operated with LiDAR sensors to collectively let the system execute autonomous simultaneous localization and mapping (SLAM) and obstacle detection and dynamic path planning tasks across maze terrains that were unidentified.

Real-time SLAM functions combined with global (e.g., A*) and local path planning algorithms made the system produce consistent and reliable autonomous exploration and maze-solving performance. ROS2 enabled better platform system integration through modular improvements that enabled scalability and synchronized command and control of heterogeneous systems between data, navigation stack inputs, and actuation outputs.

Experimental tests proved how the build of autonomous navigation systems on open-source robotics frameworks operating with inexpensive hardware resulted in reliable performance. Minor issues like localization drift, sensor noise, and path replanning delays occurred during testing, but the system maintained or surpassed the planned operational specifications.

The project emphasizes that dependable system linking together with effective sensor integration together with adaptive real-time decision-making represents essential elements for developing autonomous mobile robots. The paper demonstrates that ROS2-based solutions can effectively serve advanced robotics applications while remaining accessible for research and development of autonomous navigation systems even in non-laboratory environments with limited budgets.

VI. Future Scope

The present autonomous maze-solving operation shows successful results; however, future enhancements may optimize system performance together with making it more robust and improving scalability. Future work should concentrate on developments along three main targets, which are:

1) Sensor Fusion Enhancements: The implementation of RGB-D cameras along with ultrasonic arrays and stereo vision systems would give the robot enhanced perception functionality. RGB-D cameras enable the robot to record planar distances as well as depth and texture characteristics, which produces enhanced environmental maps with better obstacle definition. Additional ultrasonic sensors can serve as backup for short-range obstacle monitoring since they excel at detecting objects that pass through laser radar’s unreliable rendering of reflective and see-through obstacles. Multiple data streams would be fused with advanced methods in order to create a more durable localization system and map generation process in challenging maze environments.

2) Advanced Path Planning Algorithms: The system currently uses BF and Dynamic Window Approach as global and local planners although better planning approaches require further examination. The robot would achieve quicker and better obstacle avoidance when dealing with unexpected environmental changes through use of algorithms like D* Lite or Anytime Repairing BF or dynamic replanning frameworks. The implementation of these advanced navigational planners would enable quicker routes alongside optimal paths and better failure recovery from map errors while operating.

3) Optimization of Hardware Platform: The Raspberry Pi 4 creates a budget-friendly and portable system, but replacing it with the NVIDIA Jetson series platform will give the system superior real-time deep learning processing abilities as well as reduced latency. This hardware transition would let us run heavy computational algorithms better while making paths more responsive and bringing time-sensitive environmental perception capabilities to extensive and fluctuating environments.

4) Multi-Robot Collaboration: **A major** progression would be achieved by extending the present singlerobot system toward multi-robot organizational capabilities. Multiple robots would distribute SLAM processes for mapping and localization through distributed frameworks while exploring different maze locations simultaneously. Lower exploration times and stronger generated mapping with data redundancy and team decision-making become possible when using this approach. The research in this field needs to develop effective systems that coordinate multiple agents as well as resolve conflicts and establish communication methods.

5) Real-World Deployment and Robustness Testing*:* The system needs testing under real-world unstructured conditions such as outdoor areas or irregular built interiors and dynamic public environments for identifying system limitations and robustness needs. The system can overcome lighting variations along with uneven surfaces and sensor-blocking elements and moving people by using repeated design modifications. The system needs to undergo sustained operation without human supervision in order to guarantee its durability and practical reliability.

The upcoming versions of this project should combine these enhancements, which will enhance autonomous navigation systems and support mobile robotics and real-time mapping and intelligent exploration research in unknown environments.

References

[1] A. Ansari and B. Singh, “Autonomous Maze Solving Robot Using A* Algorithm,” Proc. IEEE Int. Conf. on Autonomous Robots, 2020, pp. 100–105.

[2] E. Lee and T. Matsumoto, “Learning-Based Strategies for Micromouse Maze Challenge,” IEEE Robot. Autom. Lett., vol. 5, no. 2, pp. 700–707, 2021.

[3] R. Patel and S. Kumar, “Architecture of the ROS2 Navigation Stack for Autonomous Mobile Robots,” IEEE Access, vol. 11, pp. 1200–1210, 2023.

[4] L. Liu and H. Chen, “Real-Time SLAM and Path Planning Integration in ROS2 for Maze Navigation,” Int. J. Robot. Res., vol. 39, no. 5, pp. 500–515, 2022.

[5] Y. Gupta and M. Chen, “A Survey of Global and Local Path Planning Algorithms for Mobile Robots,” Auton. Robots, vol. 44, no. 3 , pp. 567–589, 2020.

[6] S. Gupta and A. Verma, “Comparative Study of Visual, Lidar, and RGB-D SLAM Systems for Indoor Mapping,” IEEE Trans. Robot., vol. 38, no. 4, pp. 1234–1246, 2022.

[7] N. Tran and F. Smith, “Reinforcement Learning Approaches to Maze Solving in Mobile Robots,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2021, pp. 3300–3306.

[8] J. Brown and E. Lee, “Frontier-Based Autonomous Exploration in Unknown Environments,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), 2019, pp. 5080–5086.

[9] R. Zhou and Y. Hu, “SLAM in Mobile Robotics: A Survey,” IEEE Access, vol. 9, pp. 2000–2020, 2021.

[10] R. Mur-Artal and J. D. Tardos, “ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial and Multi-Map SLAM,” IEEE Trans. Robot., vol. 37, no. 6, pp. 1874–1890, 2021.

[11] A. Johnson and M. Garimella, “Design and Implementation of the Navigation2 Stack in ROS2,” Robotics Auton. Syst., vol. 146, no. 4, pp. 103876, 2022.

[12] S. Ito and K. Sato, “Survey of Micromouse Maze Solving Algorithms and Heuristics,” J. Intell. Rob. Syst., vol. 97, no. 1, pp. 45 –59, 2020.

[13] C. Miller and R. Garcia, “Sim2Real Transfer for Autonomous Robots Navigating Randomly Generated Mazes,” IEEE Robot. Autom. Lett., vol. 8, no. 4, pp. 2320–2327, 2023.

[14] D. Nguyen and P. Oliver, “An Evaluation of ROS1 and ROS2 for Embedded Robotic Platforms,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2021, pp. 2512–2518.

[15] S. K. Kim and J. Park, “Navigating the Maze: Integrating SLAM and Path Planning in ROS-Based Systems,” Int. J. Adv. Robotic Syst., vol. 19, no. 2, pp. 12, 2022.