Vision-Based Maze Solving Using LiDAR Navigation and Symbolic Planning for TurtleBot3

Moaaz Alasadi
Deggendorf University
Student ID: 12016599

***Abstract*—Autonomous navigation in unknown indoor environments is a fundamental challenge in mobile robotics. This paper presents a vision-based maze-solving system for a TurtleBot3 robot using LiDAR-based obstacle avoidance and camera-based exit detection. The robot navigates a maze-like environment using reactive control while reasoning about its progress through symbolic state representation. The system is implemented using ROS 2 and evaluated in a Gazebo Classic simulation environment. Experimental results demonstrate reliable wall avoidance, successful exit detection, and correct termination of navigation behavior.**

***Index Terms*—Mobile Robotics, TurtleBot3, LiDAR, Computer Vision, ROS 2, Maze Solving**

I. Introduction

Autonomous navigation is a key challenge in mobile robotics, especially in unknown environments such as indoor mazes. Robots must be capable of avoiding obstacles while making navigation decisions based on sensor data. Maze-solving tasks provide a useful benchmark for evaluating perception, planning, and control capabilities in mobile robots.

In this work, a navigation system is proposed for a TurtleBot3 mobile robot operating in a simulated maze environment. The system combines LiDAR-based obstacle avoidance with vision-based exit detection using camera perception and symbolic reasoning. The robot navigates without a pre-built map and stops automatically once the exit condition is detected.

II. Related Work

Classical navigation strategies such as wall-following and reactive obstacle avoidance are widely used in mobile robotics. LiDAR sensors provide reliable distance measurements for collision avoidance, while vision-based perception enables semantic understanding of the environment. Recent work has shown that combining classical navigation with higher-level perception and reasoning improves robustness.

III. System Overview

The proposed system is implemented on a TurtleBot3 mobile robot model equipped with a 2D LiDAR sensor and an RGB camera. All experiments are conducted in a Gazebo Classic simulation environment using ROS 2 Humble. The system consists of two main functional modules: obstacle avoidance and exit detection.

Reduce to exact page limits Add figure placeholders + captions LiDAR data are used to ensure safe navigation by detecting nearby obstacles, while camera images are processed to determine whether the maze exit is visible to the robot.

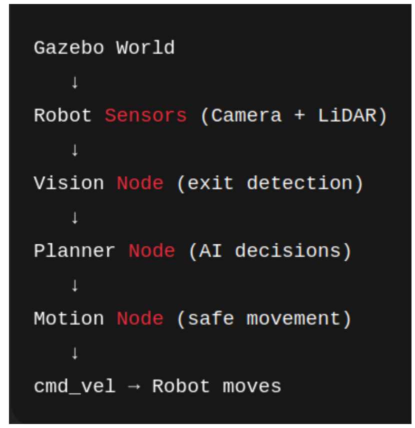

Fig. 1. System Architecture of the proposed vision-based maze-solving

IV. System Architecture

The system follows a modular ROS 2 architecture composed of three primary nodes:

- A vision node for exit detection using camera images
- A symbolic planner node for high-level decision making
- A motion control node for velocity command execution and safety

These nodes communicate through ROS 2 topics, allowing perception, planning, and control to operate independently while remaining coordinated.

V. Methodology

A. LiDAR-Based Navigation

The navigation strategy is based on reactive obstacle avoidance. Distance measurements from the LiDAR sensor are monitored within a forward-facing angular sector. If an obstacle is detected within a predefined safety distance, the robot rotates toward the free space direction. When the path ahead is clear, the robot proceeds forward.

This approach ensures collision-free motion while allowing the robot to navigate narrow maze corridors.

B. Vision-Based Exit Detection

Exit detection is achieved using vision-based processing of camera images to identify a predefined exit symbol. The vision node publishes a symbolic state indicating whether the exit is visible. This information is used by the planner to determine when navigation should terminate.

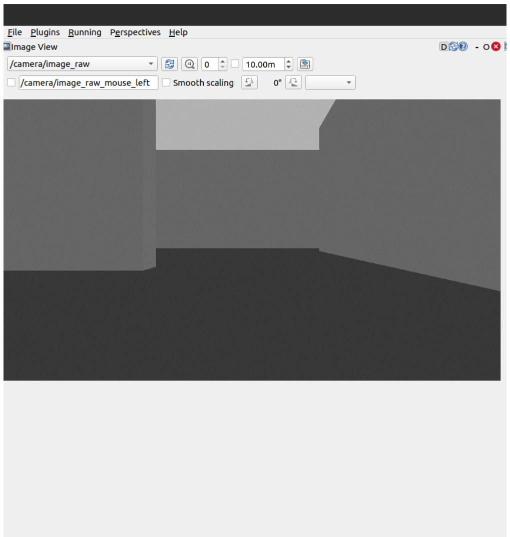

Fig. 2. Robot Camera view showing of the maze environment as display

C. *Symbolic Planning*

A symbolic planning layer reasons over the current maze state and generates discrete motion commands such as forward movement or turning. This abstraction separates high-level decision making from low-level control and improves system clarity and extensibility.

VI. Implementation

The system is implemented using ROS 2 and Python. LiDAR data are received from the `/scan` topic, while camera images are received from the `/camera/image_raw` topic. High-level planner commands are published to an internal topic and translated into velocity commands published to `/cmd_vel`.

The motion control node integrates planner commands with real-time obstacle detection to ensure safe and responsive navigation behavior.

VII. Experimental Results

Experimental evaluation was conducted in a Gazebo Classic simulation environment using ROS 2 Humble. A custom maze world was designed to test navigation performance under constrained conditions.

The robot successfully avoided walls, navigated maze corridors, detected the exit region, and stopped upon reaching the goal, demonstrating the effectiveness of the proposed approach.

VIII. Discussion

The results show that the proposed system can solve maze navigation tasks without relying on global mapping. Symbolic planning simplifies decision making, while reactive obstacle avoidance ensures safe navigation.

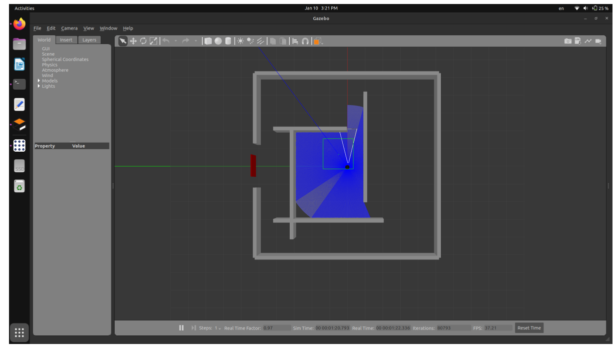

Fig. 3. Custom maze environment implemented in Gazebo for experimental evaluation

IX. Conclusion and Future Work

This paper presented a vision-based maze-solving system for a TurtleBot3 robot using LiDAR-based obstacle avoidance and symbolic planning. The system was implemented in ROS 2 and validated in simulation. Future work will focus on enhancing visual perception robustness, integrating SLAM for mapping, and exploring learning-based navigation methods in more complex environments.

Acknowledgment

The author would like to thank the instructors and laboratory staff for their guidance and support, as well as the open-source ROS and TurtleBot communities for their tools and documentation.

References

[1] Open Robotics, "ROS 2 Documentation," Available: https://docs.ros.org
[2] ROBOTIS, "TurtleBot3 Manual," Available: https://emanual.robotis.com/docs/en/platform/turtlebot3
[3] N. Koenig and A. Howard, "Design and use paradigms for Gazebo," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.