MAZE SOLVING ROBOT

A PROJECT REPORT

Submitted by

BL.EN.U4AIE21015	B Sai Abhishek
BL.EN.U4AIE21017	B Ruchith balaji
BL.EN.U4AIE21038	Chillakuru Hari

BACHELOR OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENCE

for the course

21AIE213-Robotics Operating System

Guided and evaluated by

Mr Rajesh Sir,

Dept. of CSE,

AMRITA SCHOOL OF COMPUTING, BANGALORE

AMRITA VISHWA VIDYAPEETHAM

BANGALORE 560 035

June – 2023

TABLE OF CONTENTS

Acknowledgement:

We offer our sincere pranams at the lotus feet of Universal guru, MATA AMRITANANDA MAYI DEVI who showered her blessings throughout this project. We are very thankful to Br. Viswamrita Chaitanya Swamiji, Director, Amrita School of Engineering, Bangalore, for his valuable support. We are indebted to thank Dr. Sriram Devanathan, Principal, Amrita Vishwa Vidyapeetham, Bangalore for engraving a path for us to utilize the available resources to the fullest and there by widen our perspective of education and growth through it.

We deeply express our sincere thanks to Dr. Nippun, Chairperson, Department of Computer Science and Engineering for giving us the opportunity and encouraging us to use our abilities to the fullest in doing the project. It is our privilege to express our sincerest regards to our project guide, Mr Rajesh, Lecturer, Department of Computer Science and Engineering, for the valuable inputs, guidance, encouragement, wholehearted cooperation and constructive criticism throughout the duration of our project.

LIST OF FIGURES

CHAPTER - 1

PROBLEM STATEMENT

Problem Statement:

Designing and implementing a maze-solving robot using ROS2 (Robot Operating System 2) and Gazebo simulation environment, with the aim of efficiently navigating complex mazes autonomously. The problem encompasses the following challenges:

Maze Representation: Developing a mechanism to accurately represent a maze in a digital format that can be easily interpreted by the robot. This includes handling different maze configurations, such as varying sizes, walls, and dead-ends.

Perception and Localization: Integrating sensors and algorithms to enable the robot to perceive its environment and accurately determine its position within the maze. This involves sensor fusion, mapping, and localization techniques to overcome uncertainties and noise.

Path Planning and Decision Making: Developing intelligent algorithms to compute optimal paths from the robot's current location to the maze's goal. The solution should consider factors such as path length, obstacle avoidance, and efficiency.

Robot Control and Actuation: Implementing a control system that enables the robot to move and interact with its environment. This requires integrating ROS2 frameworks for controlling the robot's actuators, such as motors and wheels, ensuring precise movement and coordination.

Simulation and Testing: Utilizing Gazebo, a physics-based simulation environment, to test and validate the robot's performance in various maze scenarios. This includes evaluating the robot's ability to navigate different maze configurations, handling unexpected situations, and optimizing performance.

Scalability and Robustness: Designing the system to be scalable, allowing it to handle mazes of different sizes and complexities. Moreover, ensuring the robot's ability to handle real-world scenarios by accounting for noise, sensor inaccuracies, and possible hardware failures.

The objective of this project is to develop an efficient and reliable maze-solving robot that can navigate complex mazes autonomously using ROS2 and Gazebo. By addressing the challenges mentioned above, the project aims to advance the field of robotics by providing a robust framework for autonomous navigation in intricate environments.

The successful completion of this project will result in a maze-solving robot that can autonomously navigate complex mazes using ROS2 and Gazebo simulation. This system will provide a valuable tool for various applications, such as warehouse automation, search and rescue operations, and educational purposes. Furthermore, it will contribute to the advancement of robotics by demonstrating the capabilities of ROS2 and Gazebo in solving real-world challenges in autonomous navigation.

CHAPTER - 2

IMPLEMENTATION

`Models:

a) Robot Model:

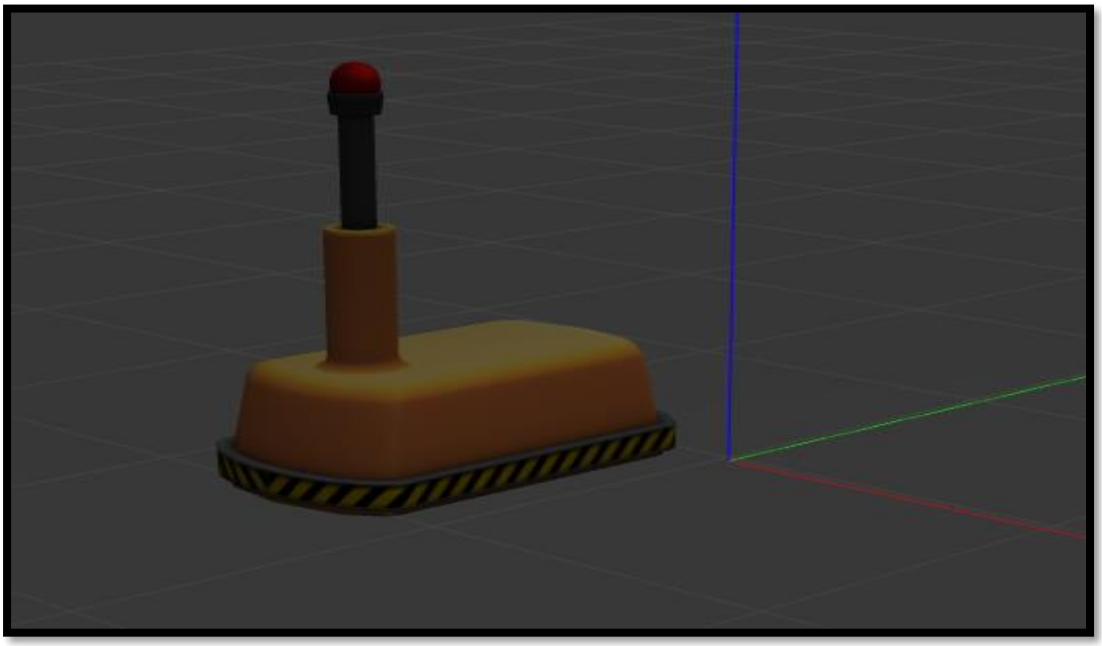

Fig-1 Robot Model

b) Maze-Models:

Maze-1:

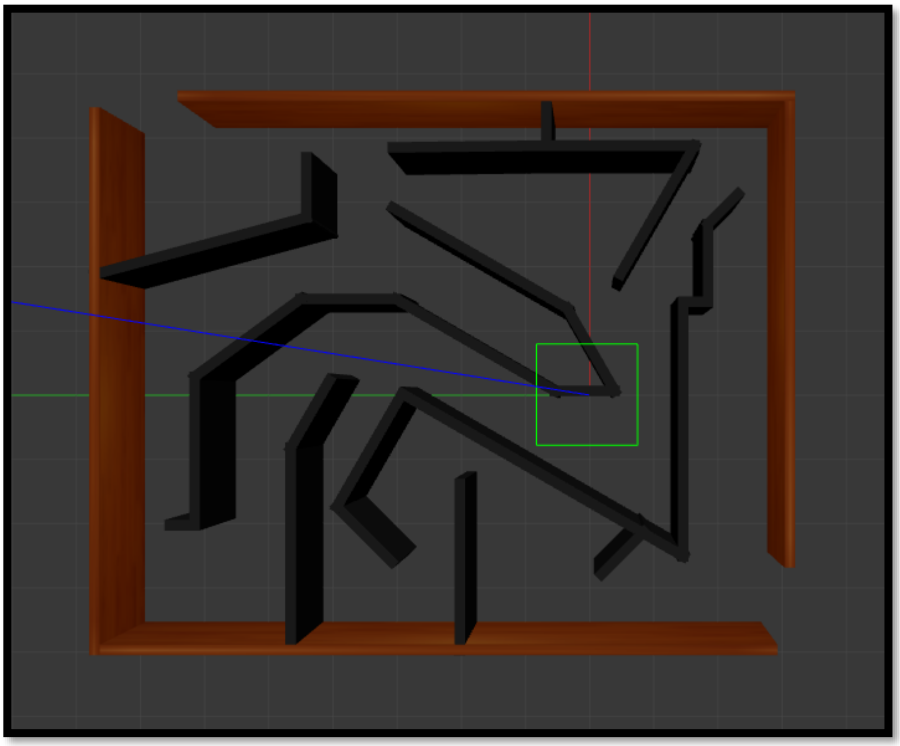

Fig-2.1 Maze Model-1

Maze-2:

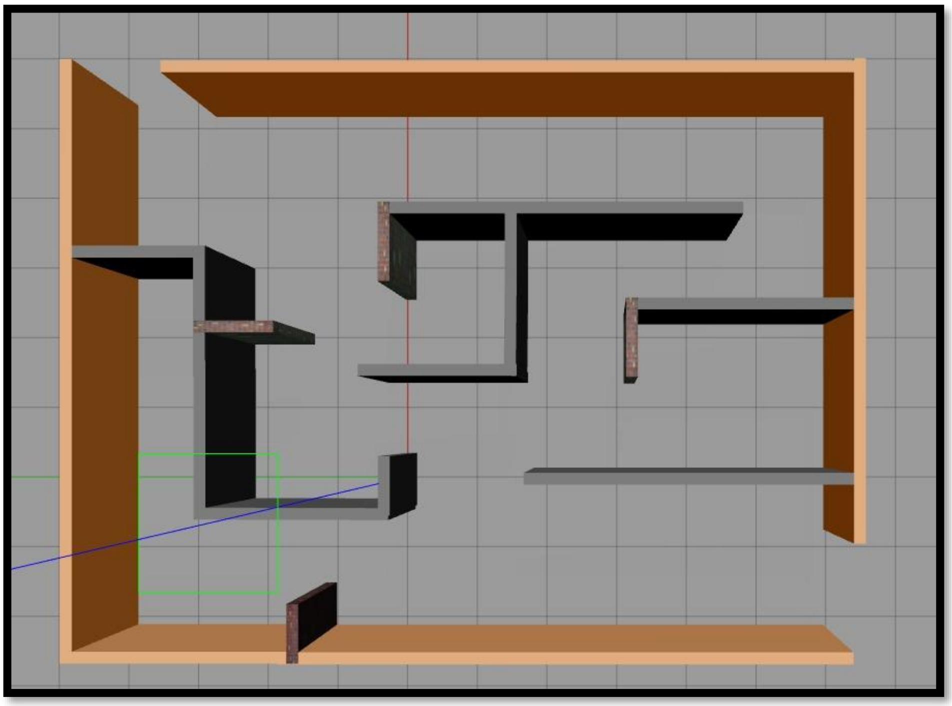

Fig-2.2 Maze Model-2

Nodes:

Spawn Demo Node:

The Spawn Demo Node is responsible for initializing the maze environment in Gazebo simulation. It loads the maze model, which includes the walls, pathways, and the goal position. The node utilizes the ROS2 interface to communicate with Gazebo, sending the necessary commands to spawn the maze in the simulation environment. It also subscribes to relevant topics to receive information about the robot's position and status within the maze. The Spawn Demo Node plays a crucial role in setting up the initial conditions for the maze-solving robot's operation.

Robot Controller Node:

The Robot Controller Node is responsible for controlling the movement of the maze-solving robot within the maze. It utilizes the ROS2 framework to receive input commands and sensor data from the robot's hardware, such as wheel encoders or a camera. Based on the received information, the Robot Controller Node implements path planning algorithms, such as Dijkstra's algorithm or A* search, to determine the optimal route from the robot's current position to the goal.

It then converts the planned path into appropriate control commands to actuate the robot's motors and wheels, enabling it to move along the designated path. The Robot Controller Node also incorporates collision avoidance .

Robot Estimator Node:

The Robot Estimator Node plays a vital role in estimating and maintaining the robot's position and orientation within the maze. It integrates sensor data, such as odometry information from wheel encoders and IMU (Inertial Measurement Unit) readings, to estimate the robot's state. The node employs sensor fusion techniques, such as Extended Kalman Filters (EKF) or Particle Filters, to combine the sensor data and generate a more accurate estimation of the robot's position and orientation. This estimation is crucial for the Robot Controller Node to plan optimal paths and control the robot's movement accurately. The Robot Estimator Node continuously updates its estimates based on the incoming sensor data, ensuring the robot's position and orientation remain reliable throughout the maze-solving process.

The coordination and interaction between these three nodes (Spawn Demo Node, Robot Controller Node, and Robot Estimator Node) form the backbone of the maze-solving robot system. Together, they enable the robot to navigate the maze autonomously, control its movement along the planned path, and estimate its position accurately within the maze environment.

BLOCK DIAGRAM:

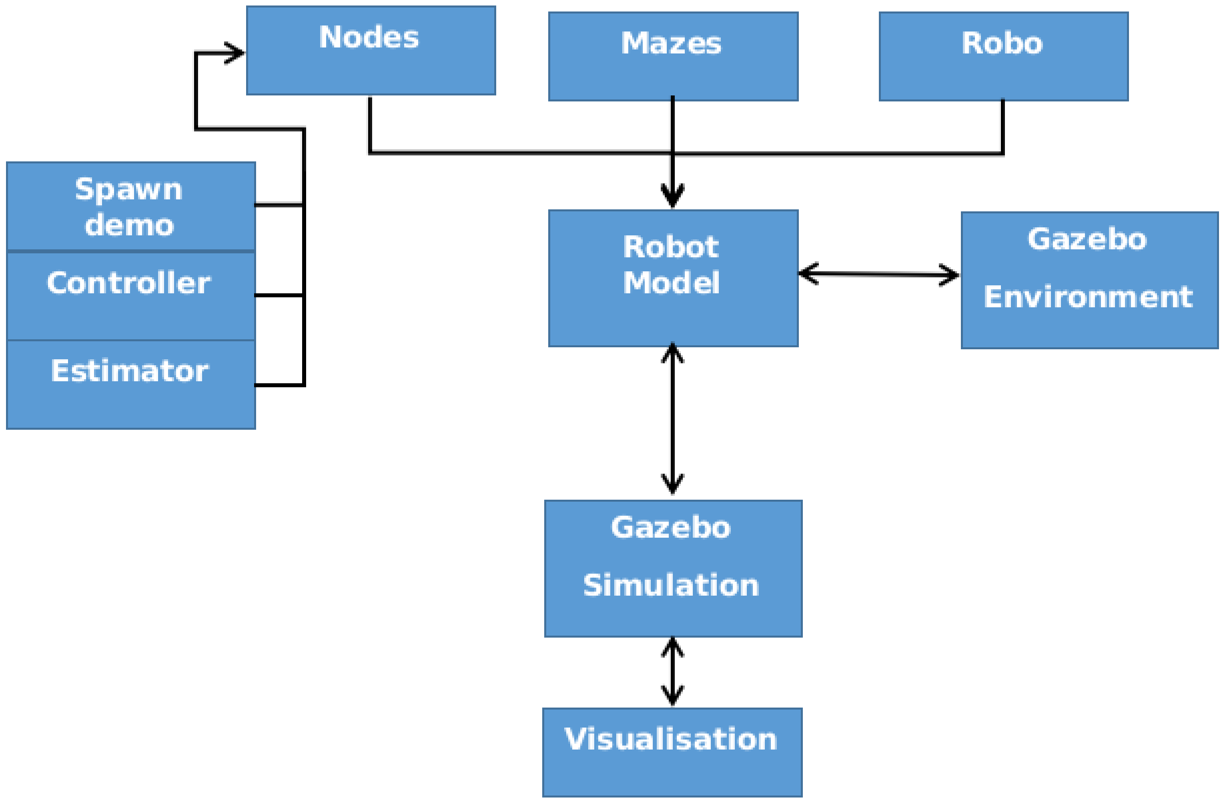

Fig-3 Block Diagram

FLOW CHART:

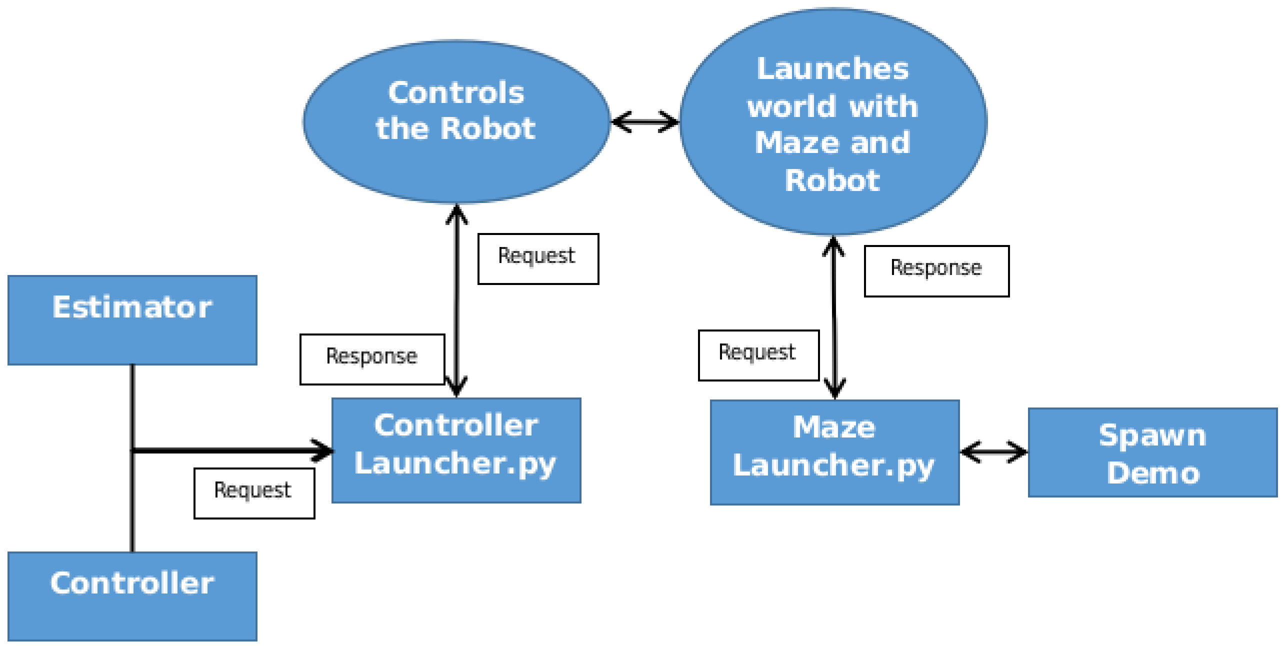

Fig-4 Flow Chart

CHAPTER - 3

RESULTS

Maze-1:

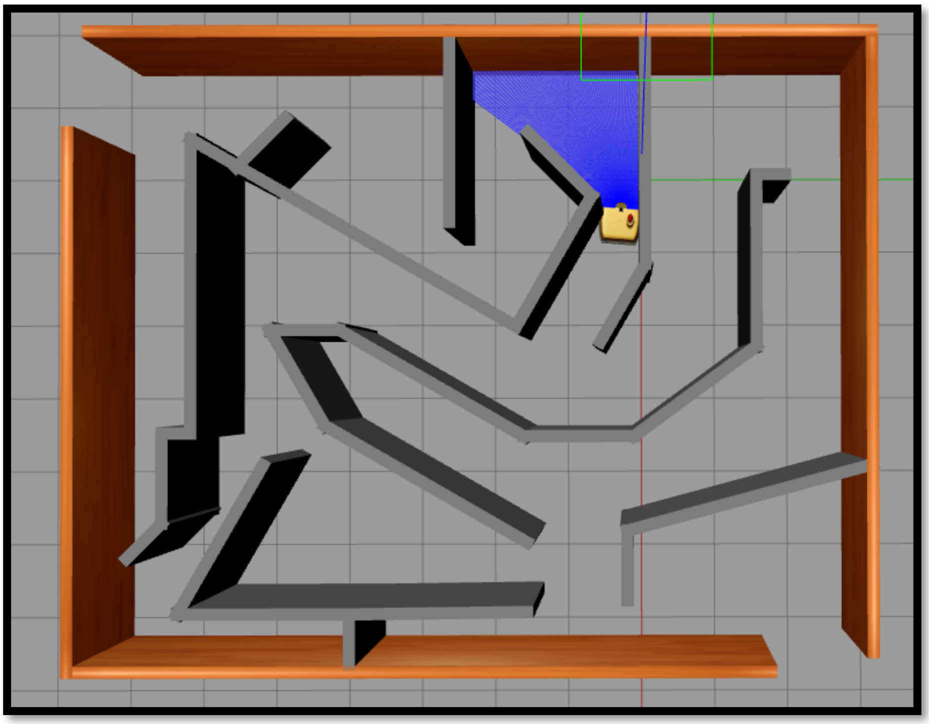

Fig-5.1 Simulation Maze 1

Maze-2:

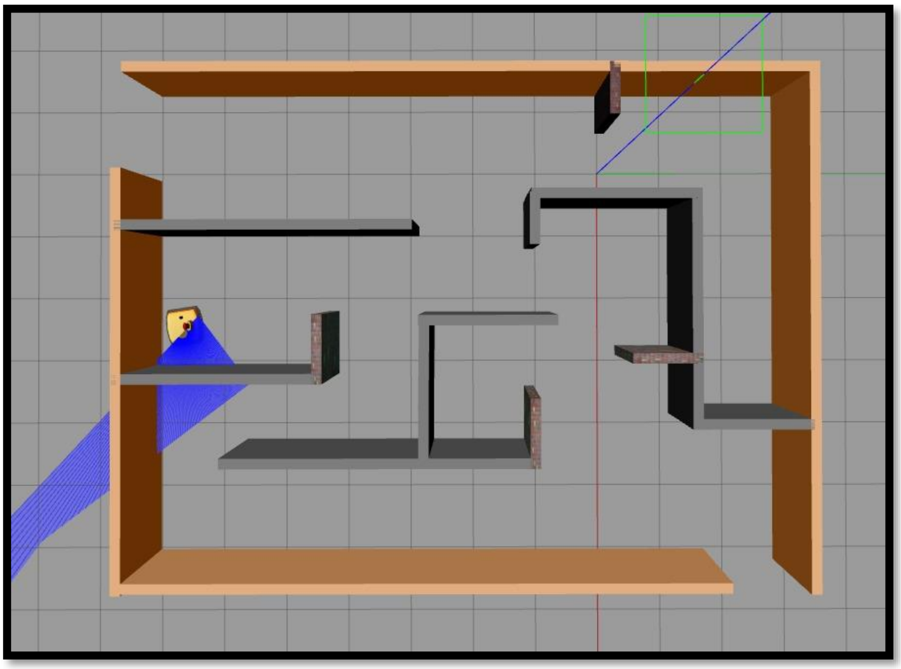

Fig-5.2 Simulation Maze 2

CHAPTER - 4

CONCLUSION

In conclusion, the integration of ROS (Robot Operating System) and Gazebo simulation environment provides a powerful platform for developing a maze-solving robot. By leveraging the capabilities of ROS2, the maze-solving robot can achieve autonomy, efficient navigation, and intelligent decision-making in complex maze environments. Gazebo simulation allows for realistic testing and validation of the robot's performance before deploying it in real-world scenarios.

Through the development of various nodes, such as the Spawn Demo Node, Robot Controller Node, and Robot Estimator Node, the maze-solving robot can efficiently tackle the challenges of maze navigation. The Spawn Demo Node sets up the maze environment in Gazebo, initializing the walls, pathways, and goal position. The Robot Controller Node utilizes path planning algorithms and collision avoidance mechanisms to control the robot's movement and navigate it through the maze. The Robot Estimator Node accurately estimates the robot's position and orientation, enhancing the overall reliability of the maze-solving process.

By addressing the challenges of maze representation, perception and localization, path planning, robot control, simulation, scalability, and robustness, the maze-solving robot system demonstrates its ability to navigate intricate mazes autonomously. Additionally, considerations such as human interaction, error handling, optimization, and documentation ensure usability, reliability, and knowledge sharing within the robotics community.

CHAPTER - 5

REFERENCES

[1] Partition-tolerant and byzantine-tolerant decision-making for distributed robotic systems with iota and ROS 2
https://ieeexplore.ieee.org/abstract/document/10073943/

[2] Distributed maze solving by cooperative robotic platforms.

https://search.proquest.com/openview/174b48b102f385d60d3a77bc52f48197/1?pq-origsite=gscholar&cbl=18750