**PES UNIVERSITY, Bangalore**

(Established under Karnataka Act No. 16 of 2013)

**Department of Computer Science & Engineering**

**Mobile and Autonomous Robots (UE22CS343BB7)**

**6th Semester**

**Mini-Project**

**Project Title:** **Autonomous Maze Solving TurtleBot3 Simulation using ROS and Gazebo**

**Team Details:**

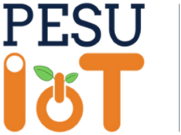
**1] VINOD NARASAPPA MELMARI [PES1UG22CS699]**

**2]VIVEKANANDA K [PES1UG22CS708]**

**3]VEERESH AMARAGATTI [PES1UG23CS847]**

**4]ANSHUL SURENDRA GOWDA [PES2UG22CS084]**





**Professor Name:**

**Dr. Ashok Kumar Patil &**

**Dr. Adithya Balasubramanyam**

**Project Description:**

This project simulates an autonomous robot (TurtleBot3) that navigates and solves a maze using the Robot Operating System (ROS) in the Gazebo simulator. The robot uses LIDAR sensors to perceive the environment and employs a right-hand wall-following algorithm to navigate the maze effectively. The objective is to simulate real-world robot navigation strategies in a virtual environment.

**Project Objectives:**

 Simulate a mobile robot navigating through a maze autonomously.

 Utilize ROS to integrate sensor data and control logic.

 Implement a basic maze-solving strategy (right-hand rule).

 Use Gazebo for realistic robot simulation.

 Demonstrate autonomous decision-making in dynamic environments

**PES UNIVERSITY, Bangalore**

(Established under Karnataka Act No. 16 of 2013)

**Department of Computer Science & Engineering**

**Methods and Materials:**

**1. System Design:**

* The system comprises a virtual TurtleBot3 robot placed in a maze world within Gazebo.
* LIDAR sensors mounted on the robot provide 360-degree distance data.
* ROS handles communication between sensor data and control logic.

**2. Algorithm/Model Development:**

* Right-Hand Wall Following Algorithm: The robot follows the wall on its right side to explore the maze and find the exit.
* Obstacle detection is performed using sensor data from the LIDAR.

**3. Implementation Steps:**

1. Clone the GitHub repository.
2. Build the ROS workspace using catkin\_make.
3. Launch the simulation using the maze\_solver.launch file.
4. Observe the robot solving the maze autonomously in the Gazebo environment.

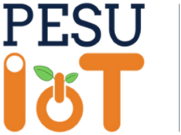
**4. Hardware Components:**

*Not applicable* (Simulation only)

**5. Software Tools:**

* ROS Humble
* Gazebo
* Python
* RViz
* Ubuntu 22.04
* TurtleBot3 Packages





**Project Outcome:**

* 1. Output results

 The TurtleBot3 robot successfully navigates and exits the maze using a right-hand wall-following strategy.

 Sensor data visualization in RViz shows obstacle detection in real-time

* 1. Simulation video link (drive link)

https://drive.google.com/file/d/1m-9-AguhVt6CGbmoUoWXR2uceenhTs-F/view?usp=sharing

* 1. GitHub link (Source code)

**References:**

** Official TurtleBot3 Documentation**

** ROS Tutorials: http://wiki.ros.org/ROS/Tutorials**

** Gazebo Documentation:** [**http://gazebosim.org/**](http://gazebosim.org/)