Wall Follower TurtleBot3 using ROS

1. Introduction

This report describes a Python script designed to control a TurtleBot3 robot within a simulated maze environment using Robot Operating System (ROS). The script utilizes laser sensor data to navigate the maze, specifically following the wall on the robot's left side.

2. System Setup

The script existing ROS workspace named `catkin_ws` located under the user directory `/home/muhammad_saqib/`. The script itself resides within the subdirectory `script` inside the ROS package `wall_follower_turtlebot3_using_ros`.

3. Pre-Execution Steps

1. Navigation:

Open a terminal and navigate to the script's directory using the following command:

$ cd /home/muhammad_saqib/catkin_ws/src/wall_follower_turtlebot3_using_ros/script

2. File Permissions:

Grant the script execution permission by running the following command in the same terminal:

$ chmod +x maze_explorer.py

4. Simulation Execution

The maze exploration involves launching three separate ROS nodes across different terminals.

Terminal 1:

1. Start the ROS core by running:

$ roscore

Terminal 2:

1. Launch the desired maze world within Gazebo, the robot simulator, using the following command:

$ roslaunch fira_maze maze_1_world.launch

This command launch file named `maze_1_world.launch` exists within the package `fira_maze`. The specific launch file name and package might vary depending on the maze environment setup.

Terminal 3:

1. Run the maze exploration script using the following command:

$ rosrun fira_maze maze_explorer.py

This command instructs ROS to execute the `maze_explorer.py` script within the `fira_maze` package.

5. Functionality

- Subscribes to the relevant topics (e.g., laser scan data)
- Processes laser data to identify the left wall
- Calculates desired robot motion based on the wall's position
- Publishes velocity commands to control the TurtleBot3