Robotics Assignment

Feziwe M Shongwe
2135313

Bonisiwe Mdletshe
1920451

Karabo Maila
2332030

Puseletso Nakana
2341162

I. Introduction

The goal of this project was to deploy a TurtleBot to monitor an environment for a suspicious object. As honours students we were tasked to explore the environment, build a map, and navigate to specified locations using a combination of SLAM (Simultaneous Localization and Mapping) and motion planning algorithms. The system was implemented using ROS (Robot Operating System), with various nodes for control, localization, and navigation.

II. Algorithms and Implementation

A. Exploration and Mapping

The exploration and mapping phase was carried out using the Gmapping algorithm, a laser-based SLAM algorithm that allowed the robot to create a map of the environment while simultaneously tracking its location. The TurtleBot was equipped with odometry data and a laser range-finder mounted horizontally.To visualize and navigate the environment, we used RViz, which was equipped with several features like costmaps to represent the robot's environment, highlighting obstacles,free spaces and for planning long-distance paths.Particle cloud to visualize the particles used by the AMCL algorithm for localization.This process was time-consuming, and we often made mistakes, requiring us to restart the whole map generation process. Although the final map did not capture the environment perfectly, we used Digital Image Processing techniques to enhance it.

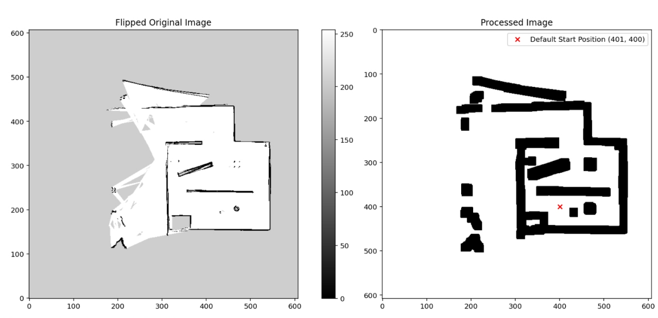

Fig. 1. Gmapping Image map and Processed image map

B. Occupancy Map

To process the image, as shown in fig. 1, and create an occupancy array map, we utilized several tools: OpenCV for reading and thresholding the image, NumPy for manipulating and processing the image array, and Scikit-image for performing binary erosion. The processed image was saved for path planning purposes, with obstacles represented by black areas and free space represented by white areas.

For path planning, an occupancy map was generated using the saved processed image file. Each cell in the array represented a specific area of the environment, where a value of 0 indicated free space and a value of 1 indicated an obstacle. To navigate using the occupancy map, coordinates needed to be transformed from world coordinates to pixel coordinates. This transformation was essential to determine if there was an obstacle at the goal position on the occupancy map and to find the shortest path using the PRM algorithm.

C. Path Planning with PRM

In implementing the TurtleBot navigation system, several key algorithms and methods were utilized. The PID controller facilitated precise control of velocities, enabling the robot to maneuver towards specified goals while mitigating positional errors. Meanwhile, the Probabilistic Roadmap Method (PRM) employed random point generation and collision checking to construct a roadmap of feasible paths in the environment. Dijkstra's algorithm was then applied to find the shortest path within this roadmap, guiding the robot from its current position to the goal. Throughout this process, transformations between world coordinates and pixel coordinates, including quaternion transformations for orientation adjustments, were crucial for mapping and navigation tasks. These transformations facilitated obstacle detection, accurate localization, and path planning

In conclusion, the integrated PID control and PRM path planning ensured effective navigation of the TurtleBot, achieving reliable performance for surveillance.