Autonomous Surveillance Bot

Clerence Mathonsi (2512711), Gabriel Tidy (2152375), Rotondwa Mavhengani (2114834) and Samuel Oladejo (2441199)
Robotics Group Assignment
University of the Witwatersrand

1 Introduction

In this project, we were challenged by a leading government organization with deploying a robot that will scan its surroundings for suspicious objects. We employed a variety of techniques which are detailed in the sections below, to accomplish this.

2 Mapping

We used teleop to manually navigate and Gmapping to implment SLAM to create a map for the given environment. The map was then saved as a pgm file and converted to an array for compatibility with our motion planning function which computes the most suitable path for the Robot.

3 Motion Planning - Probabilistic Roadmap

Probabilistic Roadmap known as PRM was selected for the motion planning. PRM is a network graph of possible paths in a given map based on free and occupied spaces.

We followed the following steps in the implementation of PRM algorithm:

1. A random node was generated randomly in the configuration space
2. The system checks whether this node lies in free space or not (i.e whether the configuration intersects with an obstacle or not)
3. If the node was in free space, it is added to the graph
4. The newly generated node was then connected to the closest nodes through a straight line
5. The system then checks if the connection between two nodes lies in free space or not (it also checks a fixed width around the connection to account for the width of the robot.
6. If it lies in free space, the connection was added to the graph.
7. Once all the randomly sampled nodes (including the initial and goal nodes) and their paths have been added to the graph, a pre-built path-finding algorithm from networkx was used to determine the shortest path from the initial node to the goal node.

Transformation matrices were used to convert between world coordinates (used to define the initial and goal locations) and map coordinates (used to define obstacle locations and for path planning).

4 Control Mechanism - Proportional-Integral-Derivative

As our control mechanism, we employed a proportional integral derivative controller, which is the industry standard due to its feedback control mechanism. We used gazebo state services to consider the robot's position(x,y) and orientation(x,y,z,w) and the TransformListener function from the tf library to always listen to get the robots' transformation at each time step. We set our initial PID values to: Proportional(P) = 1, Integral(I) = 0, Derivative(D) = 0, and increased P until oscillations occurred. We utilized the euler from quaternion function from the tf.transformations library to convert the orientation values to a single scalar rotation value. To enable the robots movement, we used the Twist package and published desired joint movements to ROS using the */mobile_base/commands/velocity* rostopic.

5 Conclusion

In this project we used SLAM through gmapping, motion planning with PRM and PID control to ensure that a simulated turtlebot reached a user defined goal location while avoiding obstacles in the environment. In the future, we will use the cameras on the robot to detect an object in the environment.