Michigan Technological University

MEEM 4707: Autonomous system

Spring, 2024

Lab - 7

By

Colton Kriecher and Amanda West

Name	Logic Build	Coding	Report Writing	Total
A	50%	50%	50%	150%
B	50%	50%	50%	150%

Problem 1

Write a piece of code in the python node (You can use wall_follow.py script). Your code should determine the command velocities and publish it on the topic "/cmd_vel."

1) Demonstrate the path planning and path following on a real robot. **Capture the trajectories in the Gazebo and real-world (using dead-reckoning) and include them in your lab report.**

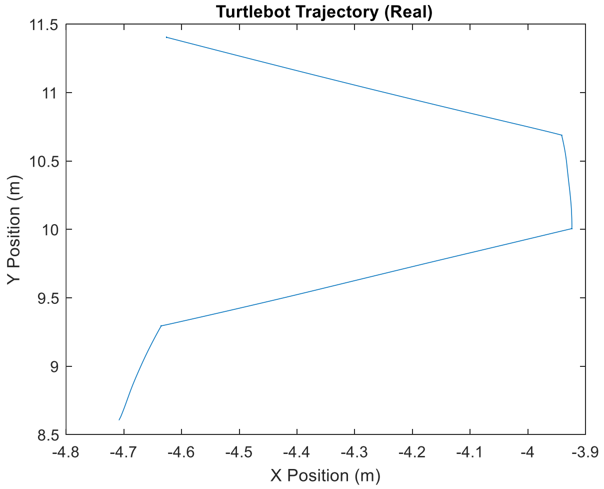

Figure 1: Real World TurtleBot Path

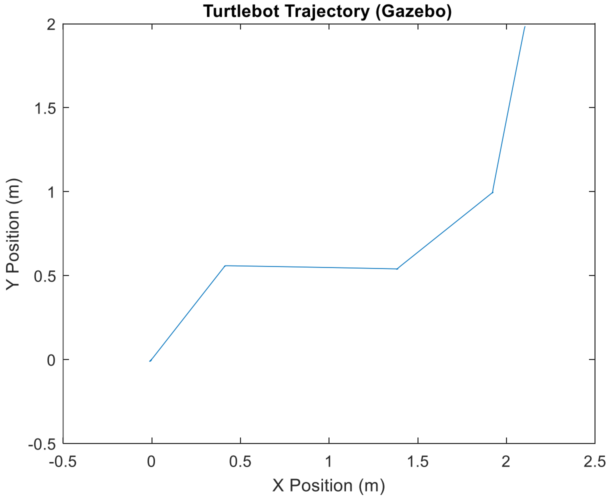

Figure 2: Simulated TurtleBot Path

2) Report the total travel distance and travel time taken to reach the goal.

The total distance traveled is 3.414 m in 35.798 seconds.

Problem 2

1) Discuss how did the robot find the optimum path?

   The robot found the optimum path by using the potential field method. The code calculates the attractive potential, or the potential between the current and goal position, and the repulsive potential, or the potential of the robot to avoid potential obstacles. The cal_potential_field computes a field combining both repulsive and attractive potential. Finally, the potential field planning initializes the start and goal coordinates and taking into consideration the potential field, calculates the optimum path.

2) What can be done to improve the performance further? Explain your idea briefly.

   We can use a visibility graph instead of the potential field method. A visibility graph connects all the visible points in the environment. The shortest path can then be calculated using these points.

Objective

The purpose of this assignment is to write a script creating a path planning algorithm for the TurtleBot using the results from a potential field method exercise.

Approach to achieve the Objective

We planned to look at one of the paths from the pre-lab and use the distance between the stars and a set angular and linear velocity to recreate that path in a simulated environment. We planned to determine the time it takes for each movement of the path, including turning and linear movements. We then will use each of those times in an if-else statement to trigger the linear or angular velocity.

Challenges faced and countermeasures taken

What problems did you face?

- We had to adjust the angles in the code to follow the path that was created with the potential field method.
- We were dividing the linear distance by the angular velocity when we were trying to determine the time it takes to move forward one leg when we needed to divide by the linear velocity.

The difference in strategy: Pre-lab vs. Lab strategy

The purpose of the prelab was to write a script that would determine the optimum path based on the grid size. Using the 0.5-meter grid path from the pre-lab, we looked at the time it takes to turn and move forward. We began with an initial delay of 1 second, and then added onto that time to turn 45 degrees towards the obstacle. By changing the 45 degrees to radians and dividing it a set angular velocity, we can determine how long the TurtleBot is going to turn for. We took this same strategy to move forward by multiplying the grid we are using by the distance between two stars and then multiplying that by how many stars are in that leg. We then divided that by the linear speed to determine the time it takes to move the first leg. We continued to do this with different angles and distances until we reached the goal position. We took the time it takes for each turn and linear movement and used an if else statement to determine when to turn and move forward.

Observations and Learnings

- The potential field method can be used to determine the optimum path between a start and goal position.
- You can change the grid for the path to determine how effective the path is.
- By determining the time it takes for each individual movement, you can create a path with if-else statements.