

MFET 442 Programming Robots with ROS

Spring 2023

Lab Assignment 10 (BONUS ASSIGNMENT)
Due Date: Thursday, March 30, 2023 (11:59 pm ET)

1 Introduction

This lab assignment is a bonus assignment. The assignment aims to provide insights into popular path-planning approaches, such as probabilistic road maps and graph search methods like Dijkstra's algorithm.

2 Objectives

1. Manually generating graph for planning
2. Implement probabilistic road maps on a given map

3 Assignment

1. Dijkstra's algorithm: Shortest path search on manually generated graph
 - Download the given occupancy map *knoy_demo_track.jpg*.
 - Figure 1 shows the robot path's start and two goals. The coordinates of the points are $q_{start} = (485, 504)$, $q_{goal1} = (60, 277)$ and $q_{goal2} = (125, 125)$. The coordinates are given along the axis shown in the figure.
 - Manually select at-least 20 vertices on the image and obtain the coordinates of each.
 - Create a script called *manual_plan.py*.
 - Add a method to this script that creates a graph from the given vertices, which has edge weights as the Euclidean distance between the vertices. Only add edges that are valid (do not pass through walls or gray regions).
 - Add another method that implements Dijkstra's algorithm to find the shortest path on a graph (You are free to use any library/ChatGPT to help you write Dijkstra's).
 - Save the graph, and the shortest path to each of the goals in a pickle file (use protocol version 3) named *manual_path.pkl*.

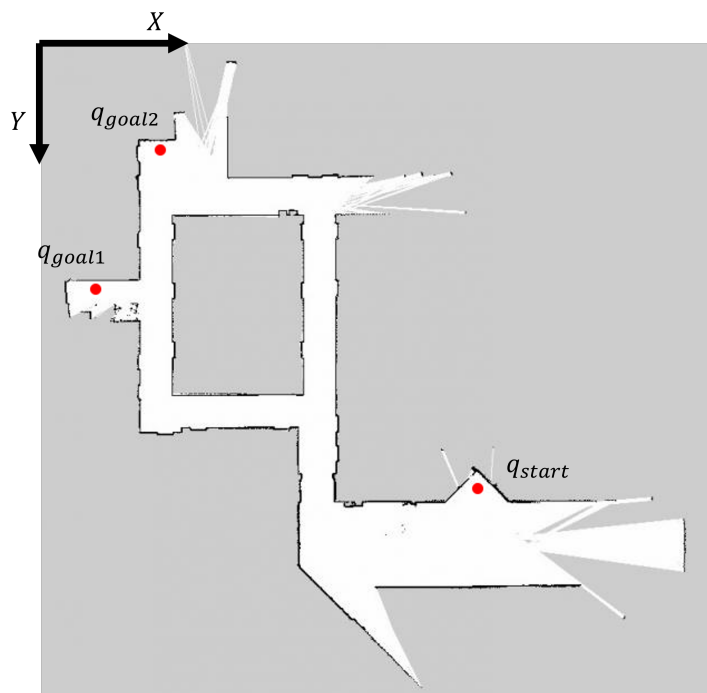


Figure 1: Knoy Demo Track Map

2. Probabilistic Road Maps: Implement a basic probabilistic road map to achieve the same functionality as Problem 1.
 - Create a script named *prm.py* with a class named PRM.

- Add a method in `PRM` to take an occupancy map image and generate the graph by randomly sampling the nodes on the map. [Be sure to eliminate invalid nodes and edges from the graph]
- Add another method to compute the shortest path given a graph using Dijkstra's algorithm.
- Add appropriate comments in your code to explain how it works.
Refer to <https://web.ics.purdue.edu/~rvoyles/codingStyle.html> for coding conventions.
- Save the graph and the path to each of the goals in a pickle file (use protocol version 3) named *prm_path.pkl*.

4 Submission Instructions

- Zip the script files (*manual_plan.py*, *prm.py*) and the pickle files (*manual_path.pkl*, *prm_path.pkl*) into a zip file named *lab10-[teamName].zip*. Submit only this .zip file.
- All the elements of your submission must be your own work. They will be checked for plagiarism. If you used code from online sources (ChatGPT) cite them at the beginning of your script.
- You can submit your assignment any number of times before the deadline. Only the final submission will be evaluated.
- Submissions not following these guidelines will not be evaluated.
- The total points for the assignment is **20 pts**, with **10 pts** for each problem.
- Late submission rules: Late submissions are not accepted.