Date of current version Monday 20, 2020.

Digital Object Identifier PLACEHOLDER

EEET 4075 Mechatronic System Design 2 - Practical Report 2

MICHAEL J. DUKE[1]
[1]University of South Australia, Mawson Lakes, SA 5095 Australia (e-mail: dukmj002@mymail.unisa.edu.au)

This work was supported in part by the University of South Australia

I. INTRODUCTION

PATH planning, open loop, closed loop controllers advantages and disadvantages.

II. METHODOLOGY

Open loop controllers advantages and disadvantages

Closed loop controllers advantages adn disadvantages

The steps required to navigate are create an obstacle map, create a valid path map, path through map.

FindFinalPath - This function loads the file 'map' which contains the obstacles to avoid, the map is inflated by the radius of the robot so that the geometry of the robot does not need to be considered when determining collision, and it can be modeled simply as a point representing its centre. Then using the matrix of waypoint coordinates it determines the path from each waypoint to the next using the FindLocalPath function explained later and concatinates that to a single matrix of coordinates representing the final path. FindLocalPath - This function begins by creating a probibilistic roadmap (PRM) with the inflated map created in the FindFinalPath function. It then attempts to find a path between two coordinates, also passed from FindFinalPath, and if one cannot be found, it increases the number of nodes, regenerates the roadmap, and tries again. This process repeats until a feasible path can be found, this path is then returned to the FindFinalPath function to be concatinated.

III. RESULTS AND DISCUSSION

PRM is not a path planning algorithm in and of itself, it simply creates a map of interconnected nodes that can then be used by a path planning algorithm to find a path. The actual path planning algorithm used by the findpath() function is an A* algorithm, which, put simply, tests what would be the shortest possible path from start to finish, a straight line, until an obstacle or the finish is encountered. If the finish has not been encountered it then tests the paths using nodes on either side of the previous path until, once again, an obstacle or the

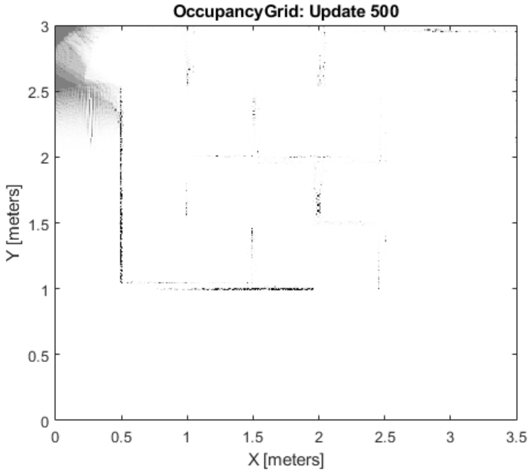

FIGURE 1: Occupancy map generated in simulation.

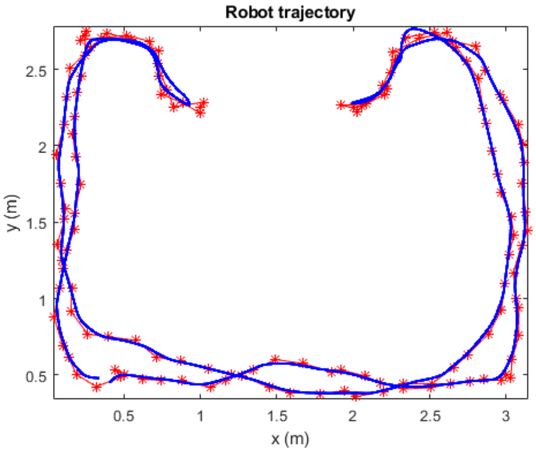

FIGURE 2: Simulated robot trajectory from path planning.

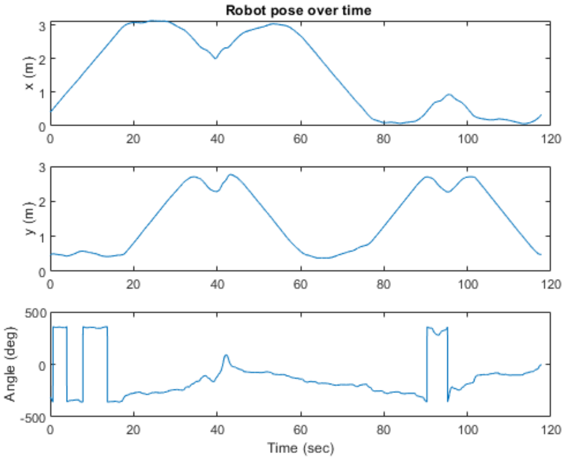

FIGURE 3: Simulated robot pose during figure 2 trajectory.

finish is encountered. This process repeats until the finish is actually encountered.

IV. CONCLUSION