Trajectory Planning and Control of a Non-holonomic Platform

1 Overview

This assignment is dedicated to the trajectory planning and control of a non-holonomic autonomous system, that follows a given path while avoiding the obstacles encountered along the track. The chosen mathematical model for planning and control is implemented in MATLAB and the simulations are analyzed.

2 Background

Non-holonomic systems are mechanical systems where the velocities (magnitude and/or direction) and other derivatives of the position are constrained [1]. The trajectory planning and control of such systems involves finding a collision-free motion between an initial (start) and a final (goal) position within a specified environment [2]. Since the 1950s, many algorithms have been proposed by researchers to seek the optimal path of mobile robots. These algorithms are generally classified into three categories; classical, bionic and artificial intelligence based algorithms. The classical algorithms include cell decomposition method, sampling based method (SBM), graph search algorithm, artificial potential field and dynamic window approach. Among the bionic algorithms are genetic algorithm, ant colony optimization algorithm, particle swarm algorithm, firefly algorithm, bacterial foraging algorithm, cuckoo search and artificial bee colony algorithm. Artificial intelligence based algorithms mainly include fuzzy control and neural network algorithms [3].

In this assignment, the Probabilistic Roadmap (PRM) method which is a type of sampling based algorithm and the Hybrid A* path planner algorithm are used to find and plan the trajectory and their performances are compared. Furthermore, the tracking control is achieved through the Pure Pursuit Controller for the paths planned by both the algorithms [4].

3 Methodology

The main steps in developing a control and trajectory planning system for a non-holonomic platform are:

1. Environment setup
2. Implementing path planning algorithm for trajectory planning
3. Trajectory smoothing
4. Kinematic modelling
5. Non-holonomic path following control

These steps are explained in detail below:

3.1 Environment setup

The first step involves the generation of a map containing obstacles and specifying the start and end (goal) points. In MATLAB this is done by using the "binaryOccupancyMap" command which creates a 2-D occupancy map object.

3.2 Implementing path planning algorithms for trajectory planning

The PRM and Hybrid A* algorithms are implemented in MATLAB for planning the path in the map containing obstacles generated previously.

3.3 Trajectory smoothing

If necessary, the path generated should be smoothed to make it suitable for a non-holonomic robot motion. This can be done using the" optimizePath" command.

3.4 Kinematic modelling

For developing the control scheme, the kinematic model of the non-holonomic robot is defined first.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega$$

Where θ is the angular position, v is the linear velocity and ω is the angular velocity. In MATLAB the kinematics model is defined by the 'differentialDriveKinematics' command which creates a differential-drive vehicle model to simulate simplified non-holonomic robot dynamics.

3.5 Non-holonomic path following control

In this assignment, the Pure Pursuit Algorithm, which is a robust and reliable technique, is employed for controlling the mobile platform motion along the path. The algorithm determines the waypoints and traverses the mobile platform through these waypoints. The lookahead distance is optimized based on average displacement error and average angle error values for generating the smooth motion with less computational time. The proposed Pure Pursuit algorithm is easier to implement and velocity tuning also requires less effort. MATLAB has the "controllerPurePursuit" System object for this.

4 Simulations and Results

After running the simulations, the following path with the start at [3, 3] and goal at [32, 38] position, is generated by the PRM method:

4.1 PRM

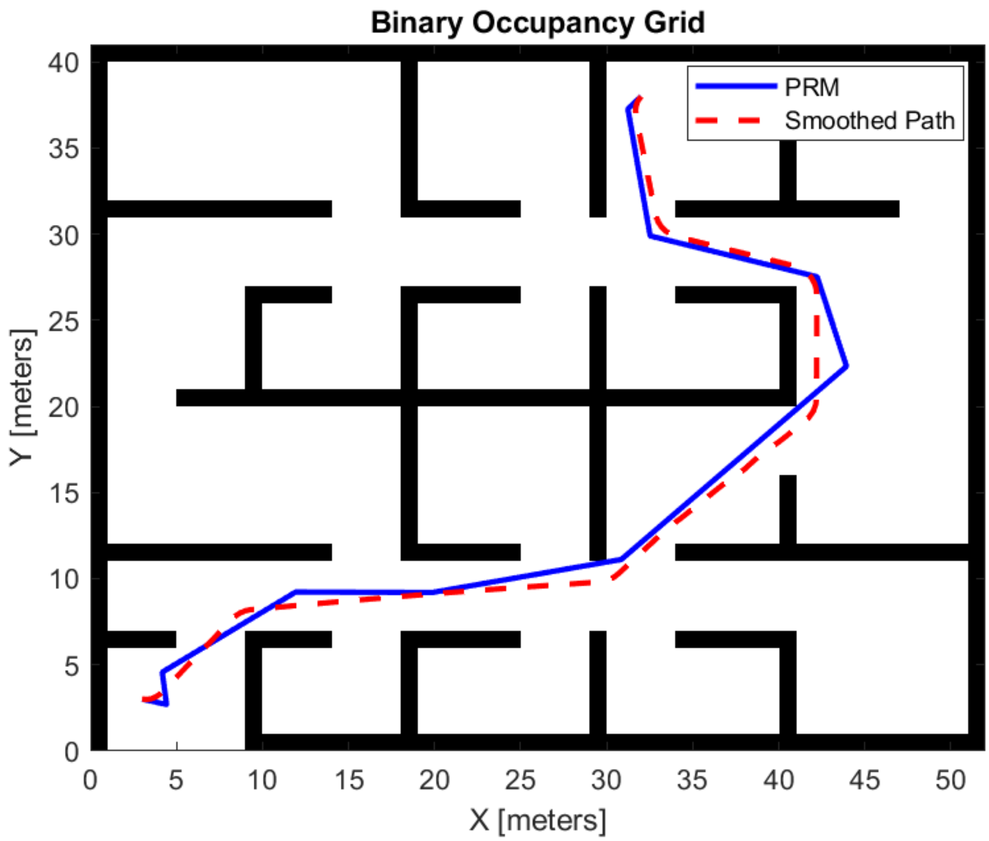

Figure 1 Path generated by PRM

Notice that although the obstacles are avoided and the goal position is reached, the path generated by PRM is not very smooth so it was further optimized. The controller tracking is shown below:

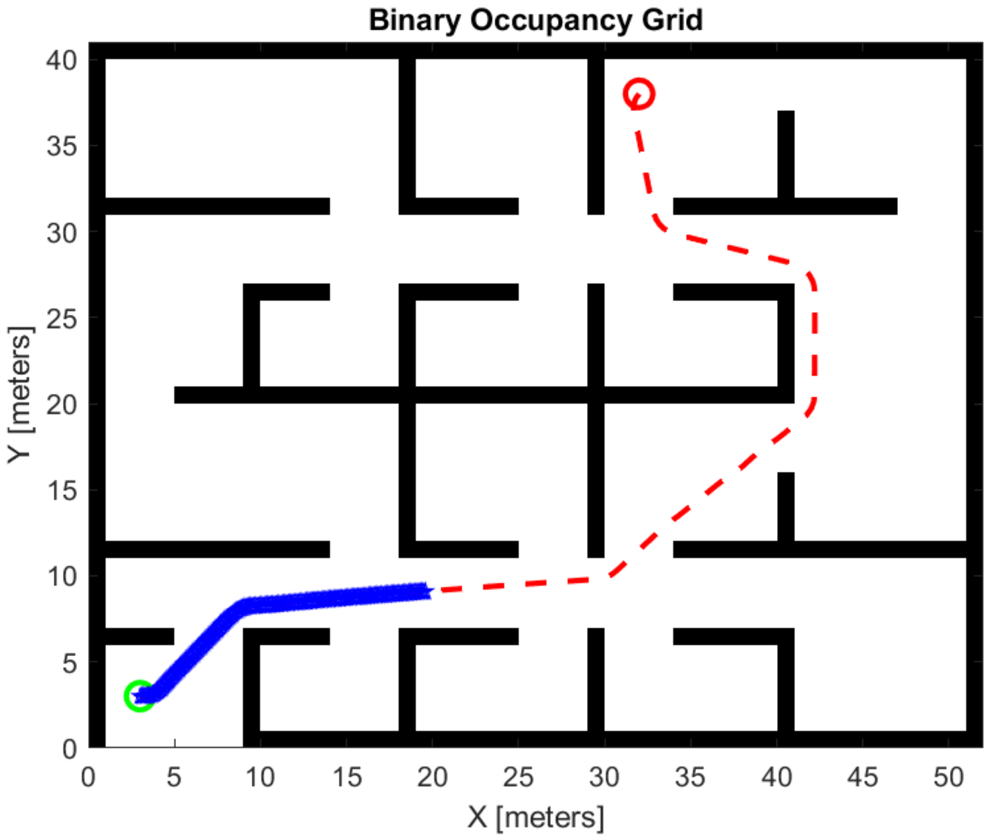

Figure 2 Path tracking by the robot (midway)

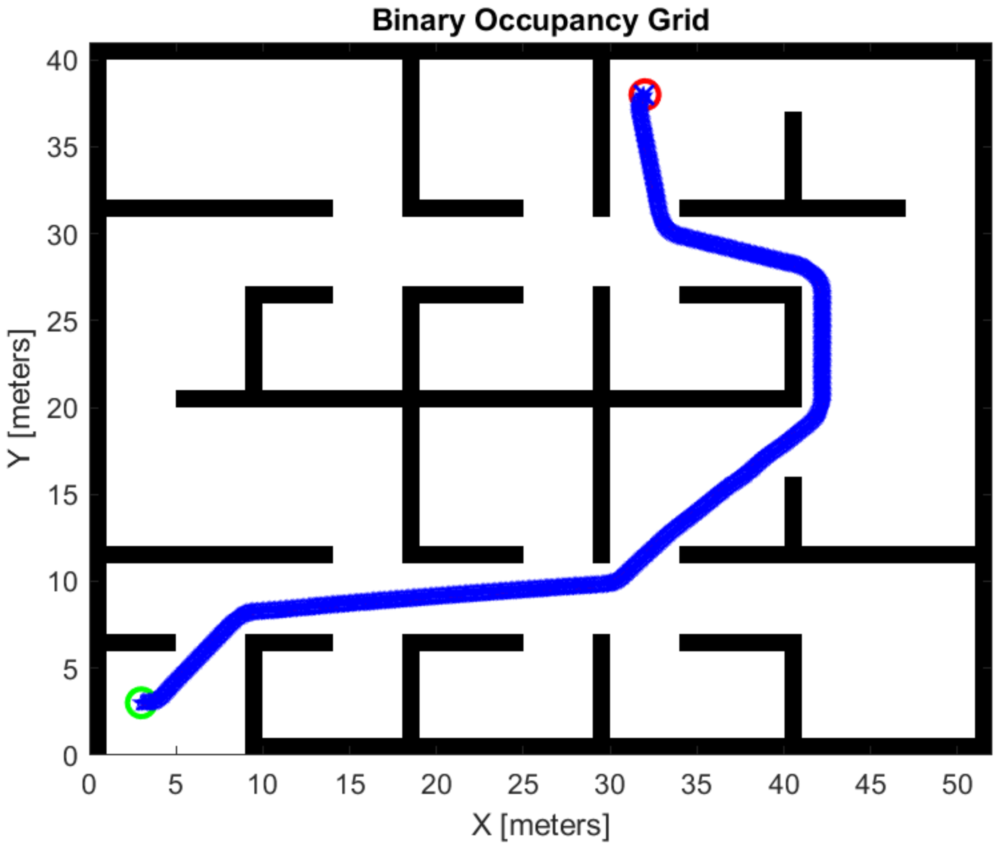

Figure 3 Path tracking by the robot (complete)

The time elapsed to traverse the path is calculated by the MATLAB “tic” and “toc” commands and it comes out as 16.96 seconds. The path length is 69.5831.

4.2 Hybrid A*

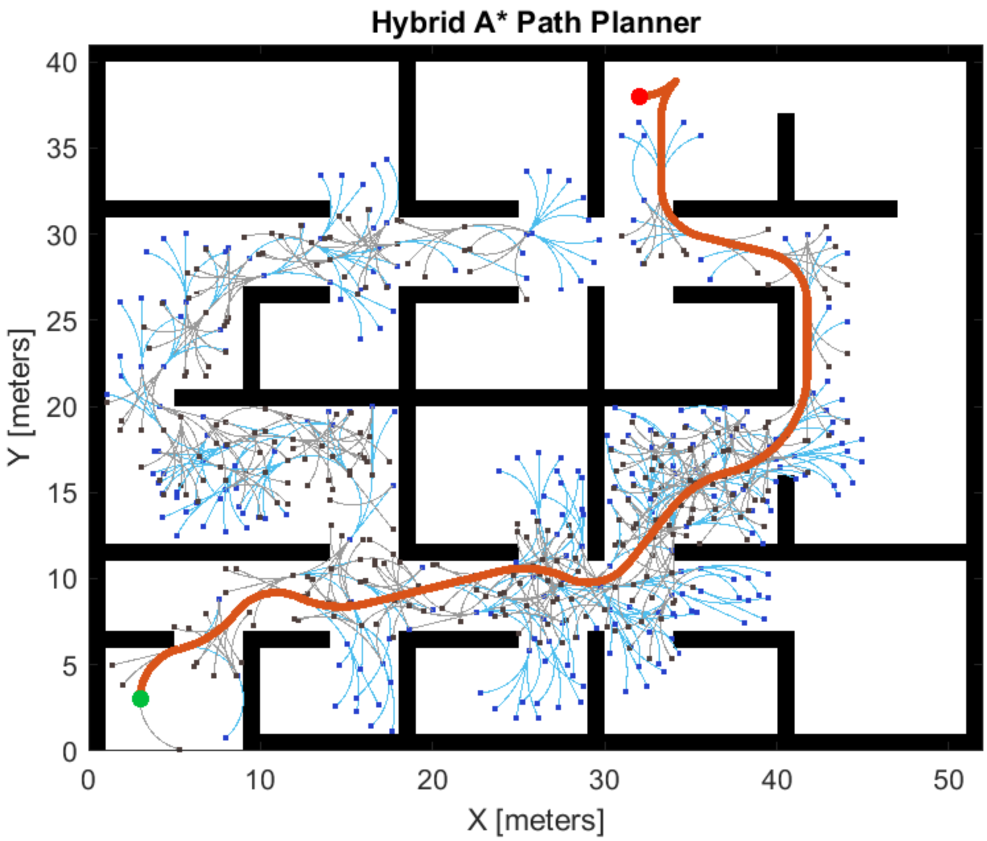

*Figure 4 Path generated by Hybrid A**

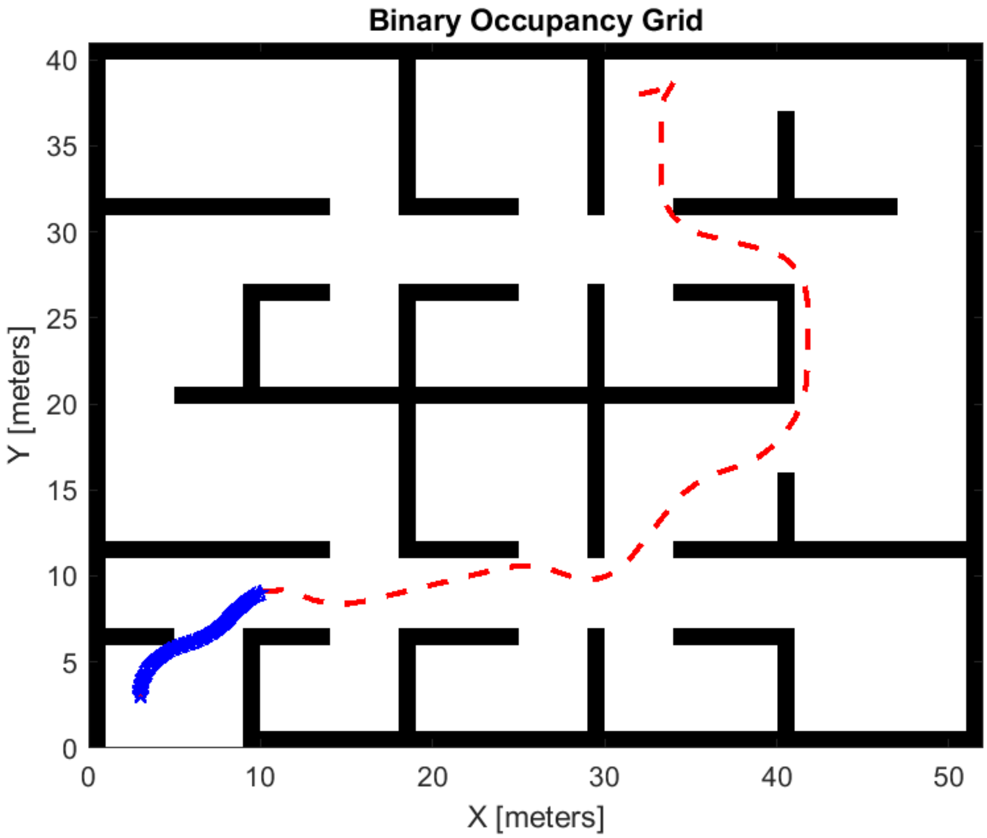

Figure 5 Path tracking by the robot (midway)

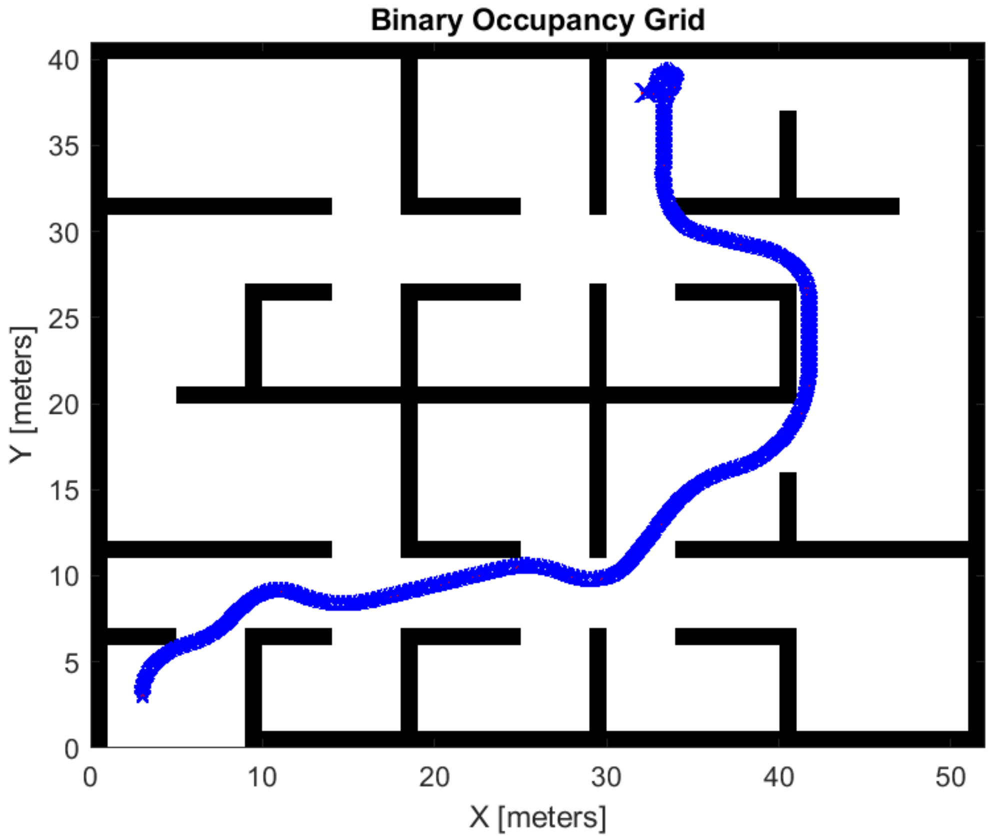

Figure 6 Path tracking by the robot (complete)

The time elapsed is 16.96s. In order to evaluate the path planned, path metrics such as validity, path length, minimum clearance and smoothness are calculated. The validity test result is 1 (true) if the planned path is obstacle free. 0 (false) indicates an invalid path. Values close to 0 for smoothness parameter indicate a smoother path. Straight-line paths return a value of 0. Clearance gives the minimum distance from obstacles. The path metric results for Hybrid A* are summarized in Table 1.

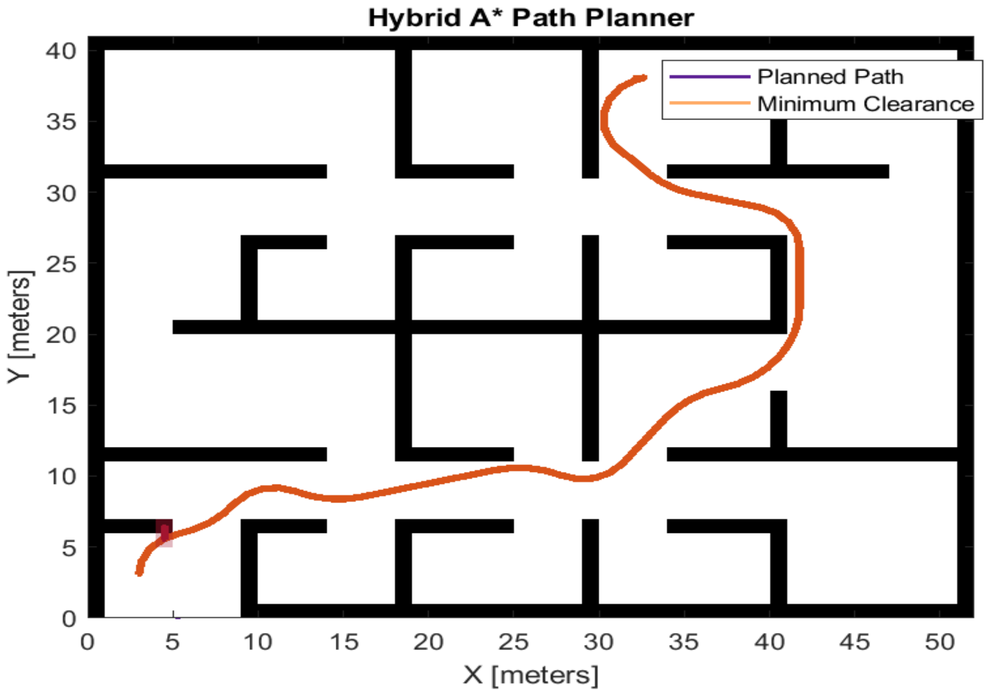

Figure 7 Minimum Clearance of Path from Obstacles

No.	Path Metric	Hybrid A*
1	path length	72.6459
2	isPathValid	1
3	clearance	1
4	smoothness	30.4254

*Table 1 Path Metrics for Hybrid A**

It can be observed that the Hybrid A* algorithm produces a smoother path, but obstacle avoidance distance is low as compared to PRM.

5 Conclusion:

In this assignment, the trajectory planning and control of a non-Holonomic platform is achieved through the Probabilistic Roadmap (PRM) and Hybrid A* algorithms. Simulations demonstrate that the PRM is easier to implement but it doesn't produce smoother paths. The Hybrid A* produces smoother paths but the obstacle avoidance is sometimes not achieved.

6 References

[1] Anthony M. Bloch, Jerrold E. Marsden, Dimitry V.Zenkov, "Nonholonomic dynamics," Notices of the AMS, VOLUME 52, NUMBER 3

[2] Gasparetto, Alessandro & Boscariol, Paolo & Lanzutti, Albano & Vidoni, Renato. (2015),"Path Planning and Trajectory Planning Algorithms: A General Overview," Mechanisms and Machine Science. 29. 3-27. 10.1007/978-3-319-14705-5_1.
[3] Lixing Liu, Xu Wang, Xin Yang, Hongjie Liu, Jianping Li, Pengfei Wang,"Path planning techniques for mobile robots: Review and prospect", Expert Systems with Applications, Volume 227, 2023,120254,ISSN 0957-4174,https://doi.org/10.1016/j.eswa.2023.120254
[4] Sulaiman, Shifa & ap, Sudheer. (2022), "Implementation of pure pursuit algorithm based controller for a mobile platform," 144-151. 10.13180/RANE.2022.23.04.28.