**Motion Planning Spring ‘24 HW 1**

2019-17973 Seung Seok Oh

**Problem 1**

The shortest path found using the A\* algorithm is shown in Figure 1. The center of the rectangular grid is the node, and the shortest distance from the starting point (green dot) to the goal (blue x) is shown as a red line. The heuristic function of the A\* algorithm is the Euclidean distance from that point to the goal.

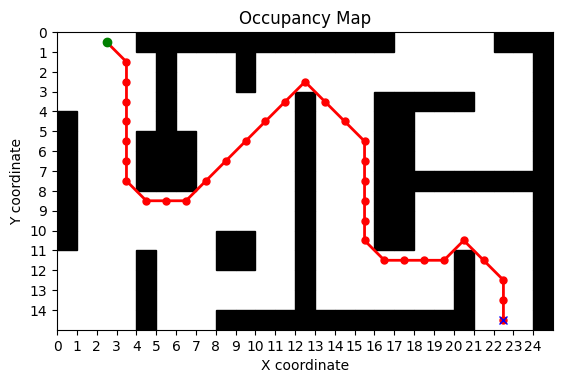


Figure 1 Shortest path calculated with A\* algorithm

**Problem 2-(a)**

Figure 2 shows the path found by performing the RRT algorithm on a given map of 688 pixels by 477 pixels. The nodes and edges generated by the RRT algorithm are colored in black, and the final path is colored in blue. We can see that the final path to the goal is a straight line because the algorithm terminates if the randomly generated point can connect to the existing nodes without colliding with obstacles and can lead to the goal at once. Since each node is randomly generated, naturally, different paths were generated for each attempt and the time required to navigate varied.

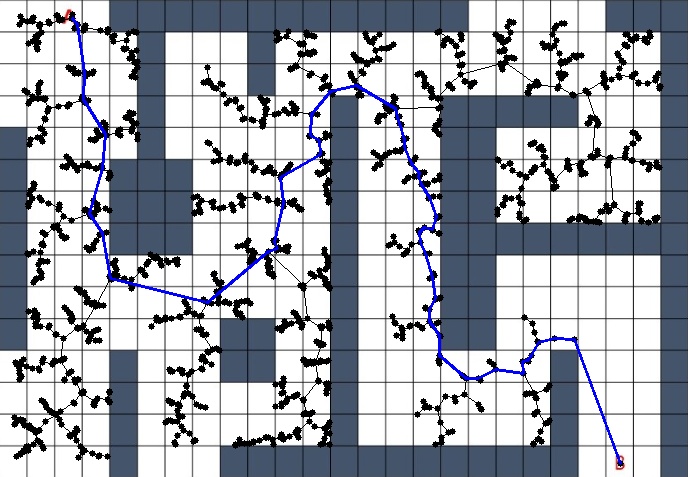


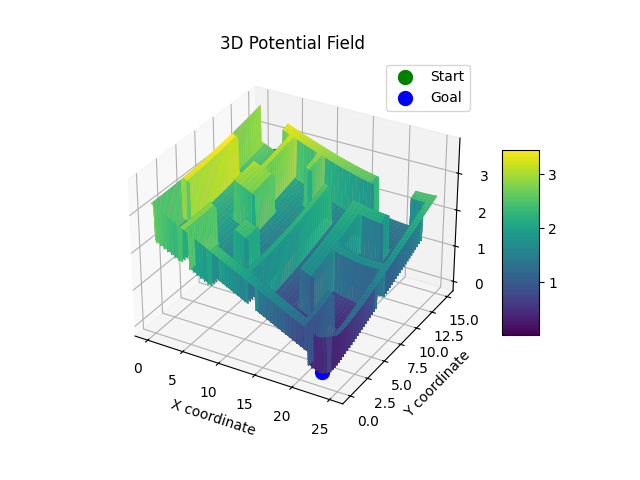
Figure 2 Path generated with RRT algorithm

**Problem 2-(b)**

The potential fucntion used in the Potenial Field Method is calculated as the sum of the attractive potential, which moves the object closer to the goal, and the repulsive potential, which moves the object away from obstacles(walls). The formulas for each are as follows:

,

The 3d visualization of the configured Potential field is shown in Figure 3.



**Figure 3.** Visualization of the potential field in 3d

However, the potential field method is prone to local minima, and in fact, at a local minima between the walls not far from the starting point, the gradient became zero and we were unable to proceed further. I tried several times by adjusting the parameters of the equation, but the results were similar.

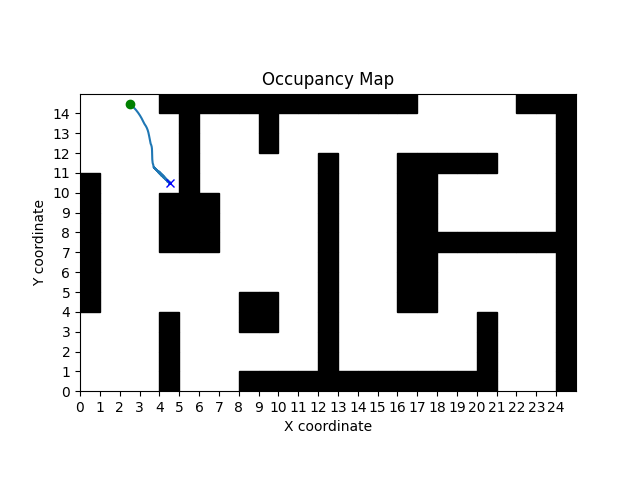
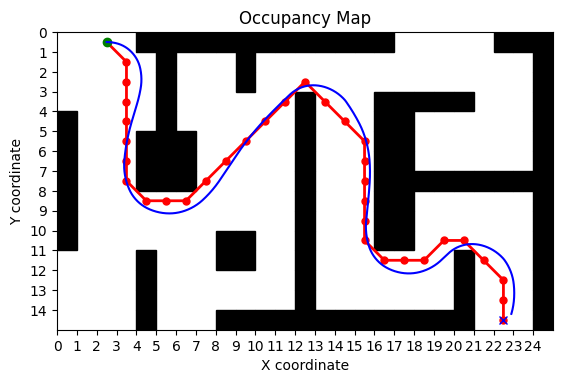


Figure 4. Path generated with Potential Field Method

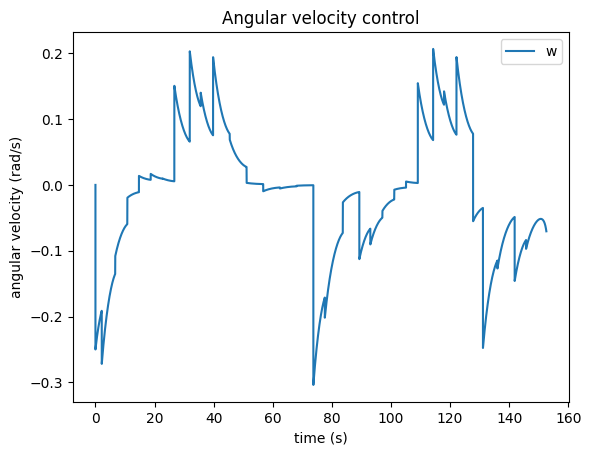
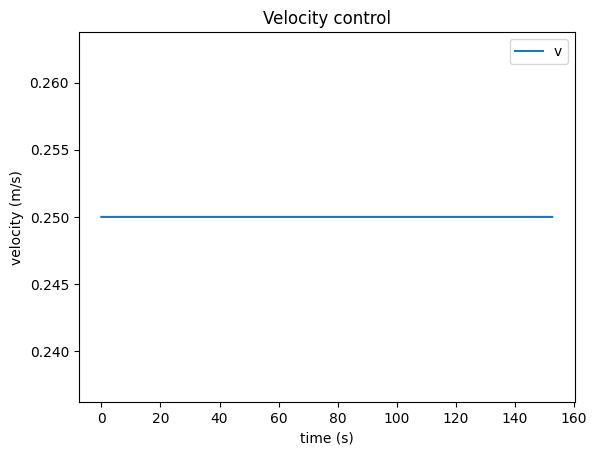
**Problem 3**

To generate a trajectory that can follow the path generated by the A\* algorithm while satisfying the given dynamics, we used the pure pursuit method using the concept of lookahead. The pure pursuit method is a method that sets a lookahead point that is currently in front of the car on a given path and sets the control to follow that point. The closer the lookahead, the more accurately the path is followed, and the farther the lookahead, the smoother the actual path. In this problem, we can control both linear velocity and angular velocity, and using pure pursuit, we can control the car sufficiently with angular velocity in [-1,1] for a given constant linear velocity. (For convenience, I set the physical length of a compartment to 1 meter.)

The actual path obtained by controlling the car with the pure pursuit method is shown in Figure 5, and the car’s control values at each hour are shown in Figure 6.



**Figure 5.** Trajectory of the car by using pure-pursuit method



**Figure 6.** Control values (Linear velocity and angular velocity) of each timestamp

**Problem 4**

To avoid two cars traveling in opposite directions, I used the pure pursuit method from Problem 3, and when the cars got close, we used the potential field method from Problem 2(b). The path generated by this method is shown in Figure 7.

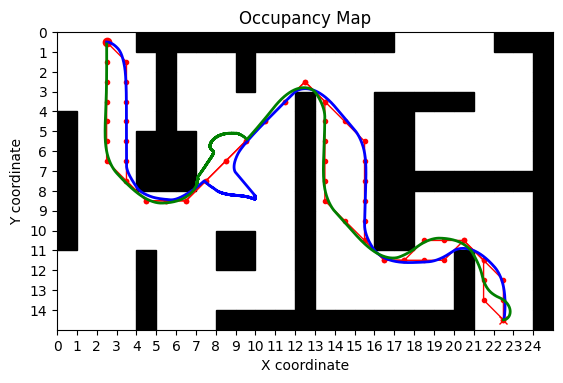
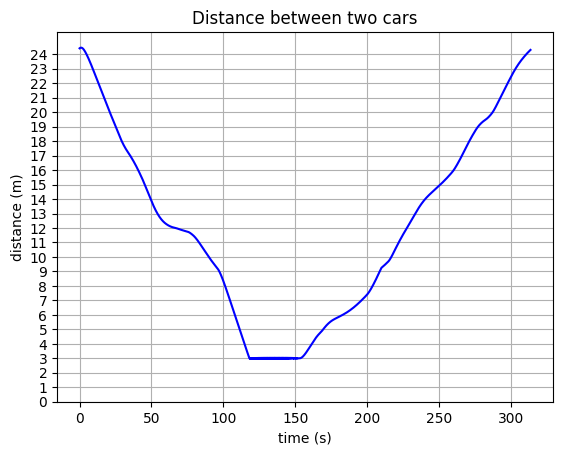


Figure 7. Trajectories of two cars avoiding each other

The two cars each follow a path generated by the A\* algorithm, and when they come within a distance of 3 of each other, they move in the direction of the gradient of the generated potential field. The potential field is calculated by adding the repulsive potential from the other car to the one used in problem 2(b). If the distance between the two cars is greater than 3, we can see that it finds the node of the a\* path closest to its position and controls it using the pure pursuit method again.



**Figure 8.** Distance between two cars at each timestamp

The limitation of this method is that it may fall into a local minima in the region controlled by the potential field method, or if the linear velocity becomes large, the radius of rotation increases, causing it to hover in place. I conducted some parameter tuning to avoid this problem, but it is expected to be difficult to use in general situations. Also, in Figure 8, which shows the distance between the two cars as a function of time, we can see that the gap between the two cars was kept constant at 3, which can be interpreted as the robot was controlled alternately by the potential field method and the pure pursuit method. In a real-world environment, this could lead to sharp turns or jerking of the car.