Assignment 2 Report

Group: Sirius
Group member: Yu Miao 44610131
Yaohui Dong 44092849

Question1. Please define the Configuration Space of the problem and describe which method for searching in continuous space do you use.

Generally, a configuration is the parameters that uniquely define the position of each point on the robot. The configuration space is the set of all configurations. It also called C-space. In this assignment, a configuration contains the x and y coordinates of each ASV. That means if the number of ASVs is n, the C-space will be a 2n-dimension space. A configuration is not always valid, it needs to abide by some constraints. The method which our group used in this assignment is called Rapidly-exploring Random Trees (RRT) algorithm. It is an algorithm designed to efficiently search high-dimensional spaces by randomly building a space-filling tree. Firstly, the RRT algorithm performs sampling in the free space. After a valid sampling point is generated, the tree is traversed to find the nearest point from the sampling point. Then, nearest point expands to sampling point with valid steps and stops when there is collision or it reaches the sampling point. After this process, a new point will be added to the tree and the valid steps it has taken will be stored. By continuously iterating until the tree grows to the target area, a valid trajectory will be generated. The basic steps of the RRT algorithm will be displayed below.

```
1. G.init($q_{init}$)
2. For k = 1 to K:
3. $q_{rand}$ = random();
4. $q_{near}$ = searchNearest()
5. $q_{new}$ = expand($q_{near}$, $q_{rand}$)
6. G.add($q_{new}$)
7. end
8. Go to step 2
(G is tree. q is node.)
```

If the solution is not found after 5000 sampling times, the program will empty the tree and start a new attempt.

Question2. If you use sampling-based method, please describe the strategy you apply or develop for each of its four components. Otherwise, please describe the details of your discretization method.

In this assignment, our group uses the sampling-based method and the details about this method will be described below.

1. Generate a random tuple of x, y coordinates to describe to position of the first ASV.
2. Generate random angle (the intersect angle between x-axis and broom) and use this angle and the fixed length of broom (0.05) to calculate x, y coordinates of next ASV.
3. Repeat step two until the x, y coordinates of all ASVs have been calculated. It should be noted that to increase the probability of valid state, a constraint has been enforced when generating angles, that is the angle of the next broom should be smaller and the angle of the last broom is greater than the angle of the first broom – 360 degrees.
4. If the new configuration satisfied the minimum area constraint, boundary constraint, convexity constraint and collision-free constraint, the new configuration will be used to expand the tree.
5. When expanding the tree, the robot has 70% chance to use random state, 10% chance to use goal state and 20 % to use state in the narrow gap of the obstacles. This approach can avoid unnecessary sampling and reduce time to get the solution.

Question3. Which class of scenarios do you think your program will be able to solve? Please explain your answer.

Generally, the program can handle any situation in the workspace, it means the program can give the solution for any initial state, goal state, different size, and location of obstacles in the given workspace. The program also can use for a different number of the ASV. Theoretically, if there exists a path between initial state and goal state, our program can find the solution for the query. The reason why the program can solve these queries will be explained below.

One reason is from one configuration to another configuration, the program uses a special strategy to ensure the configuration can be successfully transformed to another configuration by the movement of the robot and the change in the angle of the broom in accordance with the requirements of the maximum step size.

Another reason is that when the program creates the random configuration, it can cover all valid cases in the workspace. That means if after sampling many times, the tree can expand to the goal state.

Question4. Under what situation do you think your program will fail? Please explain your answer.

Since the sampling is random, the program may fail to get enough valid state to limited time.

This happens especially when handling with narrow gaps. Only valid states in the narrow gaps help robot to pass the gap and most states outside the gap become useless. Since the probability of sampling valid states in the gap is too small, it is time-consuming to find solution.

Another reason is that the method for searching the nearest neighbour from tree is linear search, which means every time to expand the tree, all element in the tree would be traversed. Therefore, when the tree becomes larger, more time is needed for searching.