MECH 550
Project 3
Peiguang Wang, Sichao Zhang

1. A succinct statement of the problem that you solved.
1) We implement core codes of random tree planner based on the algorithm of RRT.
2) We build two different environments (called Fence and Maze). In these two environments, we test point robot and square robot to see their performances, and the process is visualized.
3) Then we use benchmarking to compare several other planners with RTP in computation time, path length, number of states and so on. From the figures obtained, we observe the differences and make the conclusion.

2. A short description of the robots (their geometry) and configuration spaces you tested in exercise 2.
1) The point robot. A point has two parameters, its x-axis coordinate(x) and y-axis coordinate(y). x and y defines its location in the workspace. Its configuration space is the workspace but except the obstacles.
2) The square robot. A square has four parameters, the x-axis(x) and y-axis(y) coordinates of its center, the length of its edge Side Length and the rotation angle θ with respect to its center. x and y defines its location in the workspace, Side Length defines its size and θ defines its rotation state. Configuration space for a square robot is a 3 dimension spaces, since it has 3 DOFs (x, y, θ).

3. Images of your environments and a description of the start-goal queries you tested in exercise 2.
We build two different environments. One is called fence, and the other is called maze.
Environment 1 (Fence):
Figure 1 shows the first environment fence.

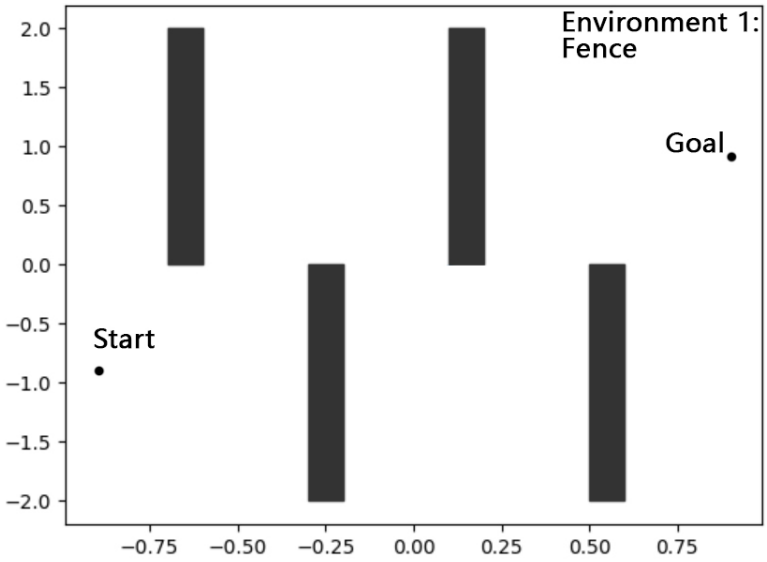

Figure 1. Environment 1 Fence

The coordinate of the start point is (-0.9,-0.9), and the goal point is (0.9,0.9).

Environment 2 (Maze):
The second environment we create is called maze, as is shown in Figure 2.

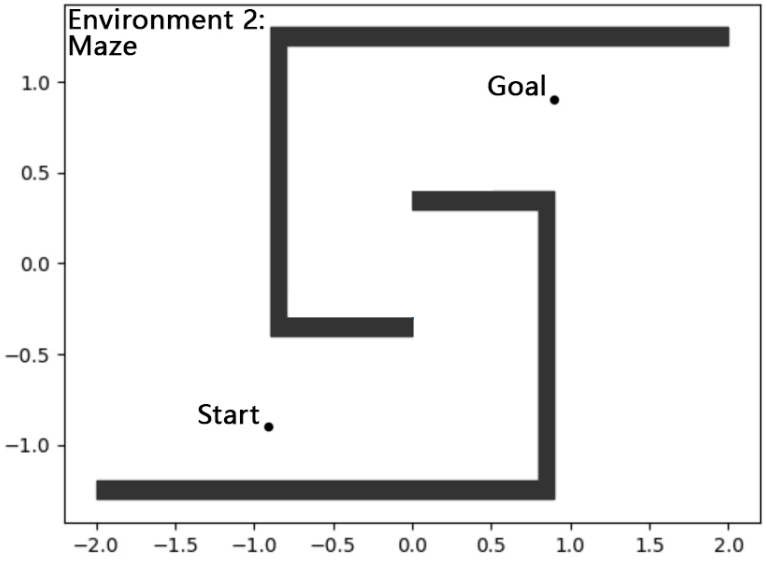

Figure 2. Failure of the Square Robot to reach the goal in time in Environment 1

The coordinate of the start point is (-0.9,-0.9), and the goal point is (0.9,0.9). Also we test point and square robot in this environment.

4. Summarize your experience in implementing the planner and testing in exercise 2. How would you characterize the performance of your planner in these instances? What do the solution paths look like?

Experience: We spent much time in reading the OMPL code. We reviewed the code of RRT planner and then write our Random Tree Planner based on *RRT.h* and *RRT.cpp*. To test the planner, we carefully read the *MyRigidBodyPlanning.cpp* in *OMPL_HandsOn_Solution* directory for reference.

The performance of RTP planner has two characteristics: First, it is a random solution. The planner does not solve the problem every time in 5 seconds time limit. Even if it find solutions, the solutions vary in different attempts. Then, solution path is not very smooth: the path sometimes have abrupt turns. While RRT planner always have a smooth planning path.

Here are some figures of the solution paths. Figure 3 shows a point robot and a square robot finding its path using RTP planner in environment 1.

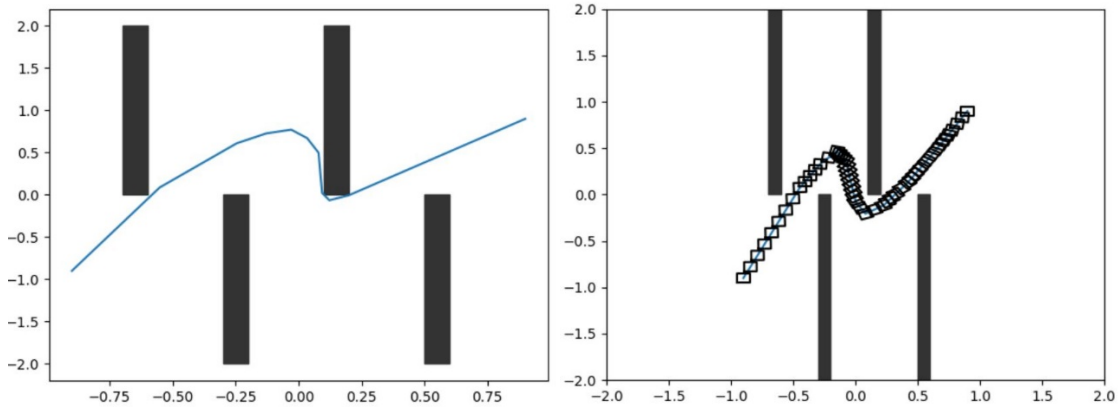

Figure 3. Paths of Point and Square Robot performed in Environment 1

With some possibility, the robot fails to reach the goal in time. Figure 4 shows one of these situations.

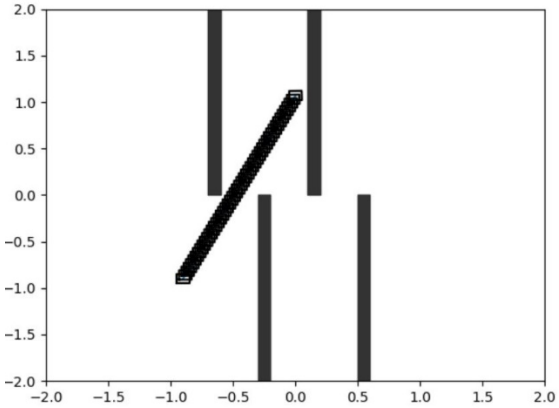

Figure 4. Failure of the Square Robot to reach the goal in time in Environment 1

Figure 5 shows a point robot and a square robot finding tis path using RTP planner in environment 2.

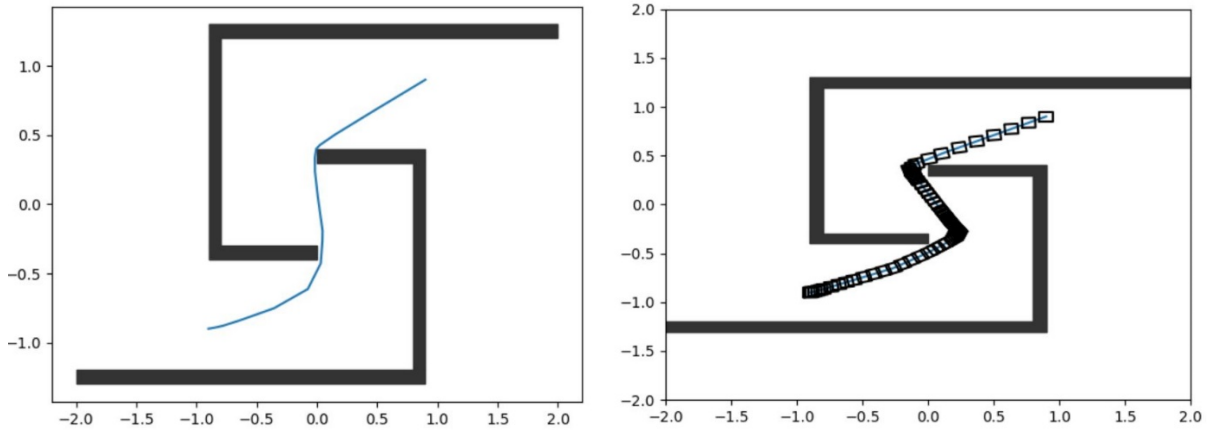

Figure 5. Paths of Point and Square Robot performed in Environment 2

5. Compare and contrast the solutions of your RTP with the PRM, EST, and RRT planners from exercise 3. Elaborate on the performance of your RTP. Conclusions must be presented quantitatively from the benchmark data. Consider the following metrics: computation time, path length, and the number of states sampled (graph states).

Compare RTP with PRM, EST and RRT planners:

(run time limit: 50s; memory limit: 1000M; run count:100)

- Time: RTP spends much more time than the other three planners. Figure 6 shows the computation time of each planner in 3D Cubicles and Twistycool scenarios.

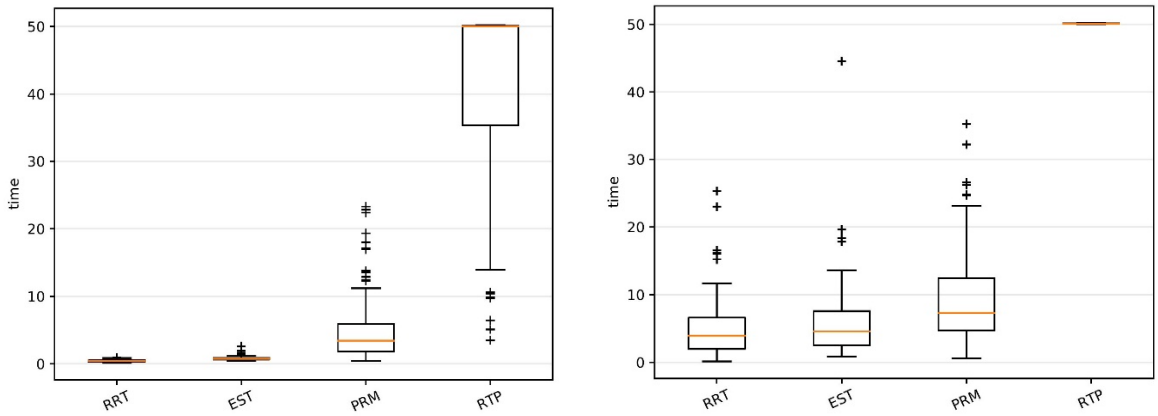

Figure 6. Computation time of four planners in 3D Cubicles and Twistycool scenarios

- Path length: RTP path length is slightly longer than the other three others. Figure 7 shows the path length of each planner.

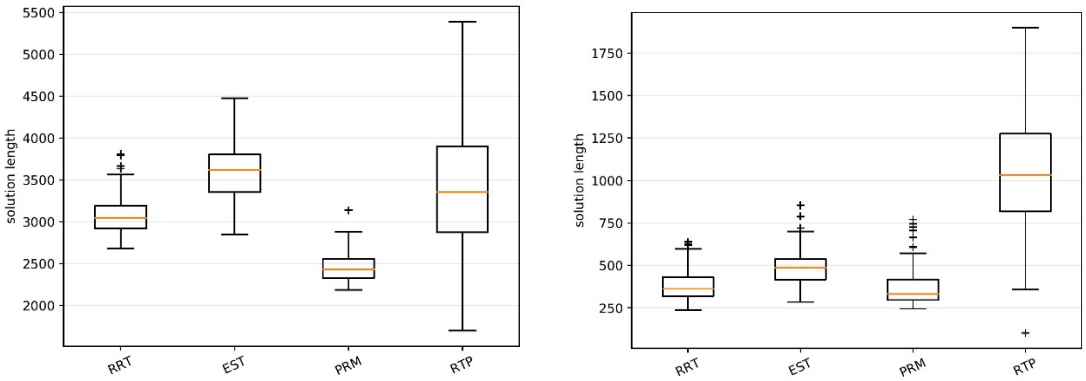

Figure 7. Path length of four planners in 3D Cubicles and Twistycool scenarios

- Graph states: RTP samples much more states than the other three planners. Figure 8 shows the graph of each planner.

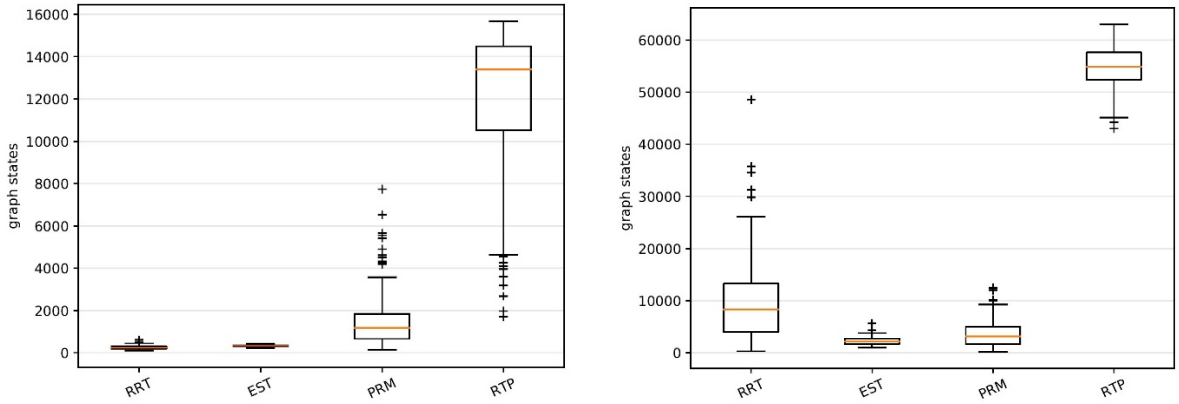

Figure 8. Number of states of four planners in 3D Cubicles and Twistycool scenarios

Table 1 shows the quantitative data (average value) of each planner in 3D Cubicles.

	RRT	EST	PRM	RTP	RTP(solved)
Computation Time	0.41392157	0.79892096	5.1215103	41.31262552	40.23084149
Path Length	3079.30475	3604.315901	2452.285422	3369.929257	3392.464624
Number of States	261.91	332.63	1624.38	11692.97	11338.35955

Table 1. Computation time, path length and states number of each planner in 3D cubicles

Table 2 shows the quantitative data (average value) of each planner in Twistycool.

	RRT	EST	PRM	RTP
Computation Time	5.0772725	5.79126057	9.27228193	50.07802064
Path Length	383.3018826	489.9257361	377.5245979	55175.36
Number of States	10302.81	2242.8	3718.34	1043.701685

Table 2. Computation time, path length and states number of each planner in Twistycool

In Twistycool scenario, we tested the time limit of 5s, 10s, 50s, 300s, 500s and even 1000s, but still we failed to find the solution path by RTP. So the data above only shows the circumstance of 50s time limit rather than other longer time limit. If we can add a planner range for RTP planner, this situation might improve.

We extracted the quantity number using *sqlitebrowser,* and we sorted the result into excel forms. You can find result for Cubicles scenario in *C_sampler_results_TimeLimit50s.xlsx*, and result for *Twistycool* scenario in *T_sampler_results_TimeLimit50s.xlsx.*

6. Rate the difficulty of each exercise on a scale of 1–10 (1 being trivial, 10 being impossible). Give an estimate of how many hours you spent on each exercise, and detail what was the hardest part of the assignment.

Difficulty of each exercise:

Exercise 1: 7
Exercise 2: 6
Exercise 3: 6

Time spent on each exercise:

Exercise 1: 7 hours
Exercise 2: 4 hours
Exercise 3: 8 hours

We spent a large amount of time on reading the codes of OMPL. As to implement the RTP algorithm based on the RRT codes, we had to understand the meanings of functions and variables used in OMPL system. To do that, we searched for references in the OMPL website when meeting unknown functions and variables. I would not say that this process was extremely difficult but it did take us most time.