Homework/Project 5: Path Planning

Due Date and Place: Please check the syllabus and Carmen for submission deadline (online submission in Carmen)

Philosophy: Learning by doing, also called experiential learning, is a proven and effective method of teaching. It is expected that students taking the course will master path planning methods like RRT, RRT*, and Hybrid A*.

Related Unit: Unit 7

Homework/Project Aims: The aims are to apply the methods learned in Unit 7 on path planning. The path planning methods used were explained in the Unit 7 lectures.

Project Background: First, you will be examining the roles of turning radius and maximum connection distances on the RRT path planning algorithm. Then you will replace the given grid map with a parking lot grid map and create three different paths for moving a car to a parking space using RRT, RRT*, and Hybrid A* path planning algorithms.

Format: Prepare your report using PowerPoint. You should have a cover page and a final page. Please use the provided template. Convert your PowerPoint report to pdf before submitting it online. Cut and paste any Simulink diagrams and Matlab m files into your report. Also, add your Matlab and Simulink files with an explanation of how to use them in a readme.txt file. Submit two files in Carmen: your pdf report and a zip file containing your Matlab/Simulink and readme.txt files.

Software: Matlab will be used in this homework assignment.

Homework/Project Statement:

The given HW live script "**ECE5553_HW_5.mlx**" includes an example MATLAB path planing live script: "Plan Mobile Robot Paths using RRT".
https://www.mathworks.com/help/nav/ug/plan-mobile-robot-paths-using-rrt.html

Students are expected to modify the provided script to achieve the tasks given in the homework/project.

Problem

[PART 1]: *(20 points)*
In the given script, the minimum turn radius for the vehicle is assigned while defining the state space for the vehicle. For different minimum turning radius values (0.5, 1, 1.5, x, 3) repeat the simulation, where x is a user-defined minimum turning radius value. You can choose 'x' such that $1.5 < x \leq 2$.

- For each minimum turning radius: Show the occupancy map. Plot the search tree from the solnInfo. Interpolate and overlay the final path. Hint: see the second figure in the example.
- Check the length of the found path, the number of iterations to get the solution, and the generated number of nodes for the solution. Hint: These can be accessed by checking: "pthObj.pathLength", "solnInfo.NumIterations", "solnInfo.NumNodes". Present these simulation results as a table.
- Compare the results and comment on how the minimum turn radius affects the RRT solution.

Note: For some of the minimum turning radius selections, there might not be a solution. If you see this result, just comment on why RRT could not find a solution.

[PART 2]: *(20 points)*

Set the minimum turning radius to 1m. For different maximum connection distances (0.5, 1, 1.5, 2), repeat the simulation.

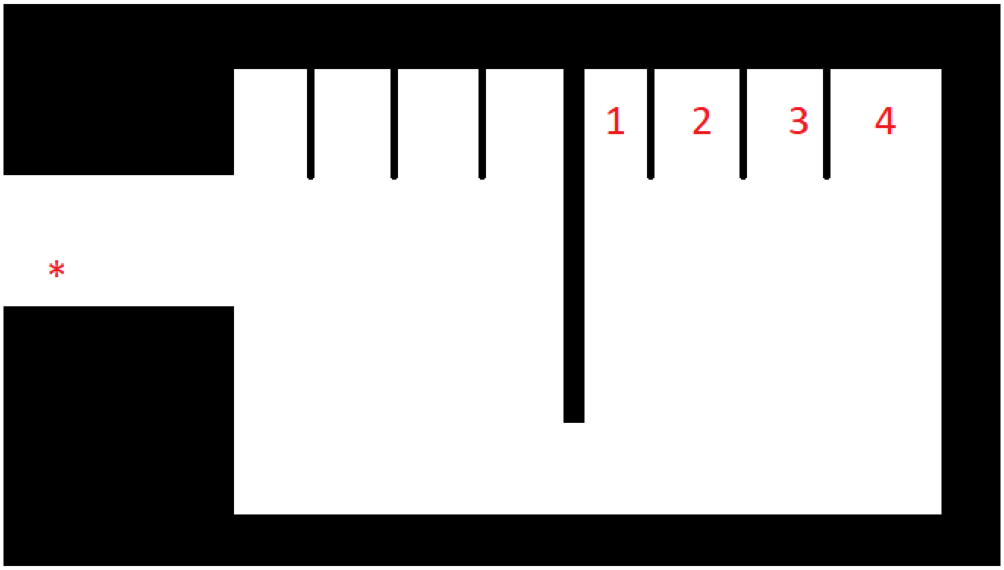
- For each maximum connection distance: Show the occupancy map. Plot the search tree from the solnInfo. Interpolate and overlay the final path. Hint: see the second figure in the example.
- Check the length of the found path, the number of iterations to get the solution, and the generated number of nodes for the solution.
- Comment on how different maximum connection distances affect the RRT solution. (You can experiment with your choice of maximum connection distance values to further analyze your thoughts).

[PART 3]: (20 points)

Create your own occupancy map from an image. For this part, a parking lot image is provided for you: **"plot.png"**.

Tips:

1. To create your own occupancy, you can check the example at https://www.mathworks.com/help/nav/ref/binaryoccupancymap.html#buvr49e.
2. To adjust the dimensions of the parking lot, set the resolution to 20.
3. MAP = binaryOccupancyMap(P, RES) creates a binaryOccupancyMap object from the matrix, P, with RES specified in cells per meter.



- Park the car located at ' * ' to your choice of numbered parking lots. For this part, set the minimum turning radius to 5m. Set the maximum connection distance to 2m. This task requires you to initialize the start and goal points and the initial directions of the vehicle.
- Show the occupancy map for the parking task. Plot the search tree from the solnInfo. Interpolate and overlay the final path.
- Check the length of the found path, the number of iterations to get the solution, and the generated number of nodes for the solution.
- Measure the time elapsed for planning the path. Tip: Type "help tic" to the command window.

[PART 4]: *(20 points)*

Use the same settings used in Part 3.

- Change your path planner to "plannerRRTStar". Tip: type "help plannerRRTStar" in the command window to familiarize yourself with the RRT* path planner.
- Move and park the car located at ' * ' in your choice of numbered parking lots (Choose the same parking lot you chose in Part 3). For this part, set the minimum turning radius to 5m. Set the maximum connection distance to 2m. This task requires you to initialize the start and goal points and the initial direction of the vehicle.
- For the parking task, show the occupancy map. Plot the search tree from the solnInfo. Interpolate and overlay the final path.
- Check the length of the found path, the number of iterations to get the solution, and the generated number of nodes for the solution.
- Compare the performance of the RRT solution with RRT*.
- Measure the time elapsed for planning the path. Tip: Type "help tic" in the command window.

[PART 5]: *(20 points)*

Use the same occupancy map used in Part 3 and Part 4.

- Change your path planner to "plannerHybridAStar". Tip: type "help plannerRRTStar" in the command window to get familiarized with the HybridAStar path planner.

- Move and park the car located at ' * ' at your choice of numbered parking lot (Choose the same parking lot you chose in Part 3 & 4). For this part, set the minimum turning radius to 5 m. Set the MotionPrimitiveLength to 2m.
- Sample Code: plannerHybridAStar(stateValidator,'MinTurningRadius',5,'MotionPrimitiveLength',5);
- For the parking task, show the occupancy map. Plot the A* search branches and overlay the final path. Use the show function to display the path.
- Tip: Check the example at https://www.mathworks.com/help/nav/ref/plannerhybridastar.html#mw_d2c3defd-2480-484e-8e38-4e40171630a4
- Repeat the simulation for MotionPrimitiveLength = 5m.
- Measure the time elapsed for planning the path. Tip: Type "help tic" to the command window.
- Comment on the performance of the planner.