LAB 5: PATH PLANNING

Due: Thursday, April 5th 11:55pm EST

The objective of this lab is to implement and test path planning capabilities for Cozmo, specifically the RRT algorithm. To facilitate easier debugging when coding your RRT, we have broken this lab into two parts, a simulated portion that will have an autograder, and a lab demonstration with Cozmo. Cozmo's goal during this demonstration will be to find a specific cube face and navigate to it while accounting for obstacles.

Part 1 – Simulation

For the simulation component, we have provided the following files to assist in development and debugging:

`util.py`: contains general methods you may find of use, as well as the definition of the Node class.

`cmap.py`: representation of the map and c-space. Features include start location, goal location, obstacles, and path storage. Map configuration is loaded from json file supplied at object creation.

`gui.py`: the visualizer

`autograder.py`: used to grade the RRT solution in simulation

`rrt.py`: implementation of RRT path planning

Step 1: Complete the `node_generator` method in `rrt.py`. This method should return a randomly generated node, uniformly distributed within the map boundaries. Make sure to check that the node is in free-space. Additionally, implement your code such that with a 5% chance the goal location is returned instead of a random sample.

Step 2: Complete the `step_from_to` method in `rrt.py`. This method takes as input two nodes and a limit parameter. If the distance between the nodes is less than the limit, return the second node. Else, return a new node along the same line but limit distance away from the first node.

Step 3: Complete the `RRT` method in `rrt.py`. Complete the main loop of the algorithm by generating random nodes and assembling them into a tree in accordance with the algorithm. Code for goal detection, tracking parents, and generating the path between the start and end nodes is already provided within cmap.py.

Step 4: Once the above steps are complete, validate that your RRT algorithm works. We have provided several maps for testing purposes, located in the maps folder. You can run the algorithm on one map with a graphical visualizer by executing rrt.py (you can change the map in the main method at the

bottom of the file), or you can run the algorithm on multiple maps at once without the visualizer by executing `python autograder.py gradecase1.json`.

Step 5: RRT should now be working but will be returning very convoluted paths. Improve performance by implementing path smoothing by completing the `get_smooth_path` method in `cmap.py`.

compute_smooth_path

Part 2 – Cozmo

For the Cozmo portion you will complete the `CozmoPlanning` method in `rrt.py`. This method is executed by the `RobotThread` and should contain all your Cozmo behavior. Your implementation should identify a target cube and use RRT to find a path to a specific face, follow the path found by RRT, and replan (reset and run RRT again) to avoid any obstacle cubes that are added during navigation. As the behavior is running the map should be updated to reflect any new cubes that Cozmo sees. New cubes may become visible (or be added), but existing cubes will not move. Note that the `detect_cube_and_update_cmap` method has been provided.

To test your code with the robot, run `python3 rrt.py -robot`. Cozmo's starting position should be the same as the goal position from the previous lab, (6, 10, 0) ~ (x, y, theta).

Note that if the target cube is not visible from Cozmo's starting location, Cozmo should navigate to the arena center to look for it and navigate to it once it is seen.

Grading: A portion of your grade will come from the autograder running your code on new maps in simulation. The remainder of your grade will be based on the in-class demo. The exact point breakdown is as follows:

RRT implementation, autograded with 6 maps, 10 points per solved map	60 pts
The robot follows the path found by RRT	15 pts
The robot identifies a target cube and navigates to a specific face	10 pts
The robot replans to avoid obstacle cubes	10 pts
The map is updated to reflect newly seen cubes	5 pts

Submission: Create a zip file that includes both `rrt.py` and `cmap.py`. Make sure you enter the names of both partners in a comment at the top of the files. Make sure the files remain compatible with the autograder. Name the zip file `lastname1firstname1_lastname2firstname2.zip`. Only one partner should upload the file to T-Square. If you relied significantly on any external resources to complete the lab, please reference these in the submission comments