EC4.403: Robotics: Planning and Navigation **Spring 2022**

Assignment-1: RRT

Released: Jan 27^{th}, 2022 *Deadline: Feb 18^{th}, 2022*

Instructions

- Submit your code files and a report in the GitHub classroom repository. Do not upload the simulation videos to the assignment repository.
- Provide brief explanations on the approach followed, kinematic relations used and screenshots of simulations generated.
- Provide the OneDrive/Google Drive links to the generated simulation videos in the report or in the repository README.
- Starter code in Python has been provided in the assignment repository, but if you are not comfortable with Python, feel free to use MATLAB.
- Plagiarism check will be done on all submissions, so please do not copy. If found, you would be given a straight zero for the assignment.

Question 1: Non-holonomic RRT for differential drive

The goal of this task is to implement the RRT path planning algorithm for a non-holonomic robot. The robot is to navigate a two dimensional space, avoiding collisions with obstacles, travelling from its initial location to a goal location. Given exact localization information (robot's initial position, obstacle location, goal location), your task is to implement a path planning decision maker to drive the robot from its initial position to the desired location using RRT. You can show the robot as a point in the simulations instead of showing the robot body. You are required to test your algorithm on the following:

- Test environment:

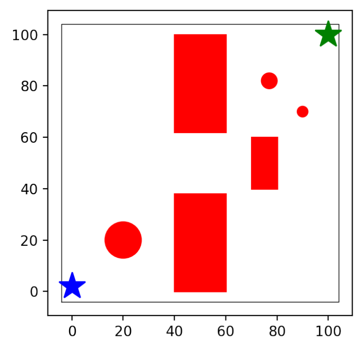

Figure 1: Test environment: Blue star denotes start position, green star denotes goal position. Obstacles are shown in red.

- Custom environments: Generate at least 1 custom environment, choosing the obstacle positions, number of obstacles, robot start and goal positions by yourself. Submit the simulations in this custom environment.

You must submit **two** videos for each of the above cases, (1) showing the evolution of your RRT tree and the generated platform centre trajectory, and (2) showing the trajectories of the left and right wheels along with the platform centre trajectory.

You can find a few examples of expected videos in the following Drive link:

https://drive.google.com/drive/folders/1oPWwCtq0A8YvwIqwf9acPjedvucg0TJ1?usp=sharing

Follow the instructions in the starter code to save the snapshots of the robot trajectory at every step, and then stitch the snapshots together using the script **mkmovie.sh** to generate the simulation videos. Refer to the repository README on GitHub classroom for instructions to use the script.

Question 2: Holonomic RRT for quadrotor UAV

The goal of this task is to implement the RRT path planning algorithm for a quadrotor UAV, based on what was taught in the class. The quadrotor is to navigate a three dimensional space, avoiding collisions with obstacles, travelling from its initial location to a goal location. Given exact localization information (robot's initial position, obstacle location,goal location), your task is to implement a path planning decision maker to drive the robot from its initial position to the desired location using RRT. You can show the robot as a point in the simulations instead of showing the robot body. You are required to sample pitch (θ), roll(ϕ) and thrust T assuming a fixed yaw(ψ) to derive acceleration command, and use acceleration control (tune suitable constants) for finding the quadrotor position.

You are required to test your algorithm on the following:

- Test environment:

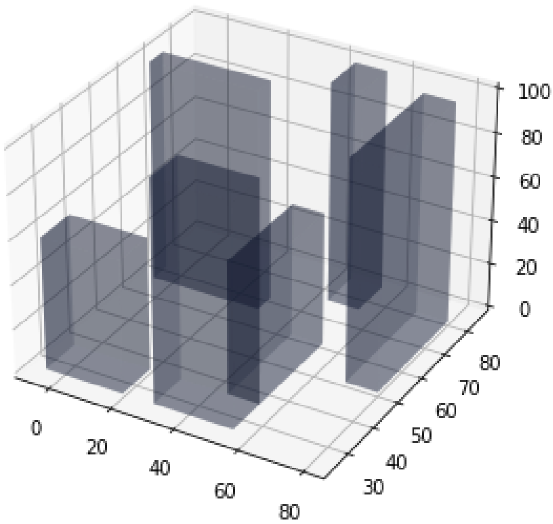

Figure 2: Test environment: UAV is required to go from (0,0,0) to (100,100,100).

- Custom environments: Generate at least 1 custom environment, choosing the obstacle positions, number of obstacles, robot start and goal positions by yourself. Submit the simulations in this custom environment.

Submit a video simulations for each of the above cases for the quadrotor trajectory, and the plots of θ, ϕ and T.

Deliverables:

- Code for both questions. You can use Python or MATLAB. Please ensure the code is well written, and we can ask you to explain certain snippets of it during the evaluations.
- For Q1: Links to two simulation videos for each environment.
- For Q2: Graphs of ϕ, θ and thrust T, along with the link to the simulation video for the UAV motion.
- A report summarising your understanding of the algorithms and explaining the results.

Feel free to reach out to the TA for any queries. All the best!