AR523: Robot Manipulators

Assignment 1: Path planning of a robotic arm with obstacle avoidance

Overview:

- The goal of this assignment is to implement sampling-based motion planning for a robotic manipulator in simulation. In particular, you will implement Rapidly-exploring Random Trees (RRT) and its bidirectional variant BiRRT, and use them to plan collision-free paths.
- The simulation environment is PyBullet (Fig. 1), which provides realistic collision checking and kinematics utilities. You may refer the following for learning PyBullet: PyBullet Official Documentation or Workshop on Pybullet by IIT-BHU robotics club

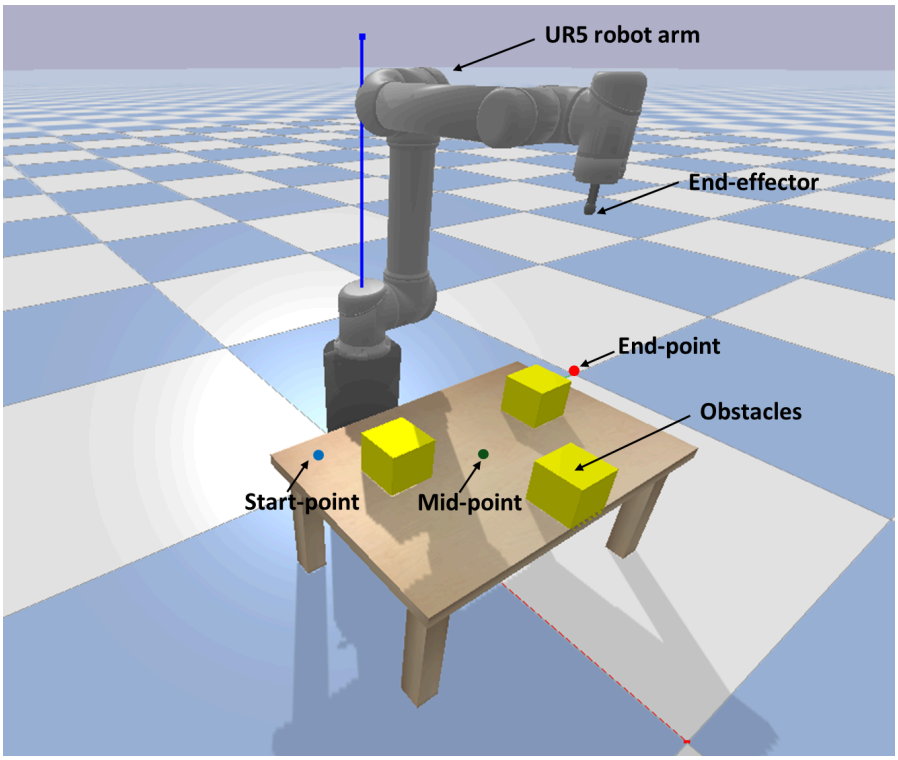

Fig 1: PyBullet simulation environment

- Please start the assignment on time. A viva-based interaction will be conducted for evaluation. Refer to lecture notes for theoretical background.

We hope that this assignment deepens your understanding of motion planning for manipulators!

Exercises:

The starter code and specific tasks to be carried out are provided in the form of a Github repository at this link. Please follow the description provided. If you find any kind of error in basic code, please correct yourself

The assignment has 6 exercises

- Part 1: RRT in Joint Space
- Part 2: BiRRT in Joint space
- Part 3: RRT in Task space
- Part 4: BiRRT in Task space
- Part 5: Analysis and Comparison
- Part 6: Evaluation on unseen environment

For **parts 1-4**, you are expected to deliver the following:

- Build the tree and visualize it.
- Find and visualize the path
- Control the robot to move to the target objects following the found path. Also, visualize the trajectory being followed by the robot's end-effector.

For **part 5**, prepare a report comparing part 1-4. You can use any metrics to compare based on your understanding. Some metrics are provided below for reference.

- Computation time to find a path.
- Path length/smoothness.
- Success rate over multiple runs
- Smoothness of end-effector trajectory

For **Part 6**, your planner will be tested on a different environment, where obstacle positions will be varied. The goal is to evaluate adaptability and robustness of your implementation.

Extra Credits:

- Implement a path smoothing algorithm on the found path with RRT/BiRRT
- Implement RRT* (optimal variant of RRT) and compare with RRT/BiRRT
- Implement and control a robot using a trajectory planning algorithm with velocity/acceleration limits

Instructions:

- This assignment is to be done individually or in pairs (working in pairs is recommended and will be rewarded). The choice is yours.
- We will create a google group for discussion. Please post your queries and also help others by responding to their queries (This will be rewarded too!)
- Please submit it on google form as a single zip file named *<A1_StudentID>.zip* or *<A1_StudentID1_StudentID2>.zip*. The zip file should contain a full code, running instructions, and analysis in the PDF file.
- The **submission date** is 5:00 pm IST on Monday, 27th October, 2025. Late submission will incur a daily 10% score adjustment for up to two days.
- This assignment will carry ~5-15% **weightage** overall. As mentioned, the instructor will provide the final weightage after closure of all assignments.
- Your first submission within the prescribed time period is final. Please do not ask for replacing the files later (e.g., on account of any incorrect upload). Such tampering beyond the submission time is not fair to other students in class who submit on time.
- The assignment must be done only from your own original efforts. Do not use any existing implementation/report from online sources. Do not violate the academic honesty code.
- Please write your implementation without use of AI code generators (e.g., GPT). Taking assistance from such tools will take away the opportunity to deeply understand the working of the algorithms. Hence, it is important to do this exercise on your own. A standard AI code generated solution will be part of automated code similarity checking, Submissions that resemble similarity beyond acceptance to standard AI-generated code will be flagged and treated as external assistance beyond your own effort.

Acknowledgement:
Thanks John Rebeiro (TA) for his help in preparing this assignment!