The Final Project

due Sunday, June 16th, 2024[1]

The final project is based on the problems and solutions studied in the course. It should include an experimental component, implemented in Python or in C++. It can be an extension of one of the exercises given during the semester—see examples below. The work should be reported in a brief document typed (not handwritten) in English or Hebrew, the main body of which will comprise at most six pages in 12pt font. You can add as many appendices as you wish for more experiments, more details of any sort algorithmic or other, more figures etc. The main body of the report should include a clear statement of the problem, the solution techniques, and the main observations, results and conclusions drawn from the work.

The main points and desired properties of your project:

- Compact.
- Individual—each student conducts her/his own project.
- Carefully crafted in form and content.
- Includes implementation and experiments.
- Implementation in Python or C++.
- A limited form of code sharing is allowed, but only after approval by the course's team. Such code will be put in a repository accessible by all the students in the course.
- The program should be runnable by the course's team; include instructions how to operate it.
- Comprehensive, well-designed set of scenarios for the experiments, which will help to find and demonstrate interesting properties of the problem and the solution.
- Main body of the report: up to six pages, 12pt font.
- Submission deadline: Sunday, June 16th, 2024. You may ask **in advance** for a later deadline. Also, if you need the final grade in the course earlier, coordinate an earlier submission with the course's team.
- Submission instructions will appear on the course website.

Below you will find examples of possible projects. However, you are encouraged to come up with your own ideas for projects.

[1] You may ask **in advance** for a later deadline.

Example 1

Extend Exercise 4.2 to work with an arbitrary, but small, number of disc robot, say up to five discs. Compare two different methods for producing paths of short total length, for instance: (i) PRM+HGraphs, (ii) dRRT+HGraphs, (ii) RRT*, or (iv) dRRT*.

Example 2

Extend Exercise 4.2 to work with an arbitrary, but small, number of discs. Design an algorithm that will balance between clearance and total path length. Experiment with different types of combinations of these two objectives. Account for two types of clearance: (i) between each robot and the obstacles, and (ii) between pairs of robots.

Example 3

Extend Exercise 4.2 to work with an arbitrary, but small, number of discs. Apply LBT-RRT (see reference below) for finding paths of small total length. Experiment with different epsilon values, for epsilon as defined in the paper.

O. Salzman, D. Halperin: Asymptotically near-optimal RRT for fast, high-quality, motion planning. IEEE Transactions on Robotics 32(3): 473-483 (2016).

Example 4

Extend Exercise 4.3 to find a valid coordinated motion for the two discs maintaining a distance of at least $\Delta > 0$ between the robots throughout their motion, when such a motion exists. Implement the solution and provide experimental evidence of its performance.

Additional topics

Throughout the lessons, and in particular from the 8th lesson on, various additional topics were/will be suggested. In particular related to “combinatorial filetring”, tight motion planning, mutli-arm optimized coordination using the Jacobi system, and more.