ELEC 5660: Introduction to Aerial Robotics
Project 1: Phase 2

Assigned: Feb. 25, 2025 Due: 11:59 PM Mar. 7, 2025

1 Project Work

In phase 1, a controller is implemented and the quadrotor can track a pre-defined trajectory. Phase 2 will focus on trajectory generator. A carefully designed trajectory generator can enable the quadrotor to operate aggressively and precisely. You may use the Matlab in the virtual barn provided by the university.

1.1 Trajectory Generator

A natural way to command the quadrotor is to set waypoints that it needs to pass by. What the trajectory generator needs to do is to generate a trajectory that

1. connects all waypoints (including start and goal points).
2. meets smoothness criterion.

1.2 Your Tasks

1. You have to implement a trajectory generator meeting the requirements above. Two example sets of waypoints are provided in **`test_trajectory.m`**. You can choose either $\mathbf{5}^{th}$ order polynomial trajectory or minimum snap trajectory [1]. The later one is our recommendation and will win you bonus points.
2. You should choose a time allocation strategy and indicate it in your code. For example, you can choose to allocate time to each segment of the trajectory according to the length of the segment. Additionally, You need to design **two** more sets of waypoints (with at least 6 waypoints).

1.3 Example Results

Please refer to Fig.1 for example results. The quadrotor should be able to track the trajectory generated by your code.

2 Structure of Simulator

The simulation code is almost the same but a **`trajectory_generator.m`**. See **`README.txt`** for details.

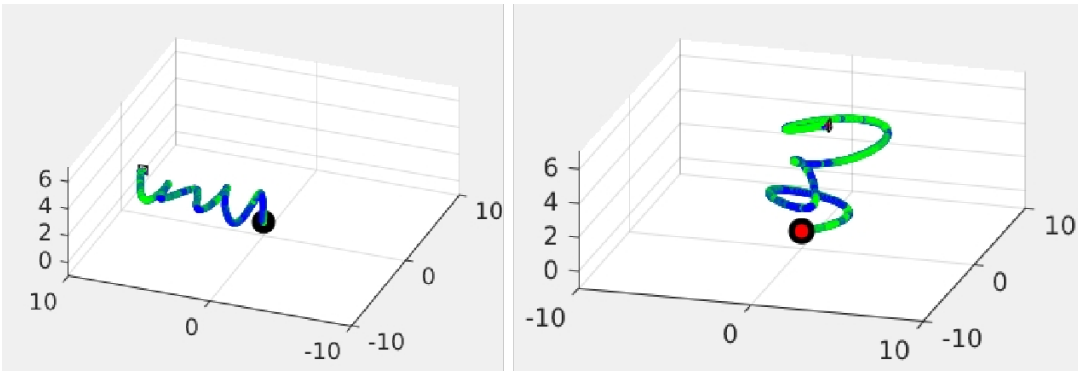

Figure 1: Example results of trajectory generation given the waypoints sets.

3 Tutorial

You can use the naive trajectory generation method in your lecture (only smoothness and connection of waypoints is required), or you can try the optimization-based method (We encourage you generate the trajectory using this method). If you prefer the latter one, you have two ways to implement it.

1. You can use the method in the slides of Lecture 4 or [2] to map the original constrained quadratic program (QP) to an unconstrained QP, and then obtain the closed form solution of the unconstrained QP directly.

2. You can use "quadprog" function in Matlab to solve the constrained QP. This function is originated in Matlab, and you can type in "help quadprog" in the command window for more details.

4 Submission

When you finish the assignment you may submit your code and documents on **canvas** before **Mar. 7, 2025 23:59:00**. The project name for this assignment is titled "**`proj1phase2_YOUR_NAME.zip`**".

Please cite the paper, GitHub repo, or code url if you use or reference the code online. Please keep academic integrity. **Plagiarism** is not tolerated in this course.

Your submission should contain:

1. A **maximum 2-page** document including:

   (a) A brief introduction of the trajectory generation method you used.

   (b) Figures plotted by simulator.

   (c) Statistics about your controller. (For example, RMS error between current state and desired state for position, velocity).

   (d) Analysis of your results. (For example, parameter studies).

   (e) Any other things we should be aware of.

2. Folder **`code`** containing files **`controller.m`**, **`trajectory_generator.m`**, as well as any other Matlab files you need to run your code.

Please do not modify files under the **`readonly`** folder. You will be graded on successful completion of the code and how quickly and accurately your quadrotor follows the generated trajectory. This time we will also test one other sets of waypoints which will not be released.

References

[1] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, May 2011.

[2] A. Bry, C. Richter, A. Bachrach, and N. Roy, "Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 969–1002, 2015.