ELEC 6910P: Introduction to Aerial Robotics
Project 1: Phase 3

Assigned: 10-11-2016 Due: 10-28-2015

1 Project Work

In phase 1 and phase 2, a controller and a trajectory generator have been implemented based on a quadrotor simulator. It is time for you to test your work on real quadrotors. In phase 3, you will be divided into 6 groups and every group will be provided with a quadrotor, which is equiped with a Pixhawk autopilot and a Odroid XU4 mini PC. Every group needs to implement your work on Odroid and achieves your quadrotor with the ability of hovering and following a designed path stably. In the phase, your task includes:

1.1 Controller

This controller has a little difference with that one in phase 1. Thanks to Pixhawk, your qaudrotor has already had a good performance in attitude control. The phase 3 controller only needs to transfer desired and true states to attitudes and thrusts of quadrotor, with assumption that quadrotors can reach the desired attitudes immediately. Despite of this difference, this controller also needs to control quadrotor to hover and follow designed paths.

1.2 Hovering and Trajectory Tracking

Three assignments are required for phase 3. They includes:

1. **assemble**. TAs will teach you how to assemble a small quadrotor, which will be used by your group till the end of this lecture. So you not only need to assemble it, but also have to maintain it.

Figure 1: the quadrotor for phase 3

2. **hover**. Your quadrotor should be able to hover at a hovering point.

3. **track**. Your quadrotor should be able to track a pre-generated trajectory, you can generate the trajectory offline and hard-code the polynomial coefficients into the C++ file. You will get bonus if you can generate the trajectory online. It's very difficult, but if you used the second method of minimum snap algorithm in the previous project, it might be much easier to be implemented on C++.

2 Sturcture of ROS Code

You need to finish your phase 3 on ROS. If you are not familiar with ROS, we recommend you to briefly go through the tutorial in **ros.org**. The main code structure for quadrotor has been implemented. You just need to fill three functions. All the details can be found on `trajectory.h` and `control_function.h`.

2.1 Description for Some ROS Packages

We provide you two folders, the one named **code_structure** is the entire work space used in the mini-computer on the drone, if you are interested in its structure, you can have a look on it. But you may not be able to build it since your laptop may not have all the dependencies. And the folder named **your_work** contains the files that you need to work with. You can directly write the code on the mini-computer when you are in the lab session, and there is no necessity to worry about the compilation and building issue, since we have installed all the dependencies in the mini-pc.

1. `pos_cmd_generator` is used to manage quadrotor modes, which include MANUAL, HOVER and AUTO status.

2. `so3_controller` provides control basis and the interface with autopilot for your controller.

3. `mavlink_message` is the interface between ROS and Pixhawk autopilot.

4. `mocap_optitrack` and `pos_vel_mocap` are the interfaces between ROS and our Optitrack System and provide odometry measurements of quadrotors(as feedback of the system).

3 Submission

When you complete the task you could submit your code and documents on Canvas before **10-28-2016 23:59:59**. The title of your submission should be "`proj1phase3_YOUR-NAMES`".

Your submission should contain:

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

   (a) Figures plotted by rviz. You should also record a small video to support your demonstration of hovering and trajectory tracking.

   (b) Analysis of your result and some discussion. (For example, parameter studies).

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

2. Files `trajectory.h`, `control_function.h`, as well as any other c++ files you need to run your code.

You will be graded on successful completion of the code and how quickly and accurately your quadrotor follows the generated path. And, before you submit the project, please demo your quadrotor to TAs.