

MEAM 620 Project 1 Phase 4

Spring 2020

Report due: TBA

1 Introduction

This document details instructions for using the hardware and software as well as the assignment and submission procedures.

For this assignment, you will work as a team to implement the controller and trajectory generator you developed in the previous phases on a Crazyflie 2.0.

You will need to sign up for lab time slots online, the instructions for which would be posted on Piazza. There will be two groups in the lab at the same time. Each group has its own Crazyflie and is provided with a desktop computer from which all code will be run.

You are encouraged to bring your own laptop so all group members can fully participate. You may also want to bring USB sticks to transfer files more easily onto the desktop.

2 Safety

Due to its small size, the only serious risk is facial injuries. Here are the safety rules.

- Only one (1) student total at any one time is allowed in the VICON area.
- Students in the VICON area must wear safety goggles.
- In case a quad is out of control, protect your face!

3 Lab 1: Controller and Waypoint Navigation

Your task for this phase is to implement a controller and trajectory generator of your choosing and get them to work with a real CrazyFlie quadrotor. As a group, you will need to complete the tasks below and demonstrate to a TA that you were able to fly the trajectories. The environment you will be using is located in the `~/meam620-20/proj1_1/code/` folder on the desktop provided. You are welcome to create additional files as needed to organize your scripts - just keep all of your work in your folder.

You will be running the waypoints given in `~/Desktop/points.json`

3.1 What to bring

You must bring the following scripts from the previous phases:

- `se3_control.py`
Note: will almost certainly require gain tuning.
- `waypoint_traj.py`
Be prepared: you will likely have to adjust the velocity profiles of your trajectory generator on the fly!

3.2 Lab Steps

Your lab session will be following these steps:

1. Safety instructions, familiarize yourself with hardware. Go through the Startup Procedure in section 4.
2. Tune your controller until the robot hovers stable, using `~/Desktop/hover_points.json`. Copy and rename it to `~/Desktop/points.json` in order to use it. Contact the TA so they can confirm you passed this step.
3. Create custom waypoint trajectories to test step responses or other features of your controller. **Do not delete any of the premade points JSON files!** The coordinates of the points in the JSON are relative to where the quad takes off from. The units are in meters, so be conservative! An empty points list is simply a hover. For reference, **towards the front is X, towards the left is Y, up is Z.**
4. Fly simple test waypoints in `~/Desktop/box_points.json`, and ensure you are confidently flying this trajectory.
5. After each run, save the desired point trajectory to a file to be included in your report, as well as the rosbags.
6. At the end of the lab, make sure you have copied your rosbags, as well as updated gains for `se3_control.py` and `waypoint_traj.py` onto a storage device.
7. Clean up!
 - Make sure the Crazyflie is back on the charger.
 - Delete `se3_control.py` and `waypoint_traj.py`, as well as any rosbags you generated in `~/data`.
 - Delete all custom trajectory waypoints EXCEPT for `~/Desktop/box_points.json` and `~/Desktop/hover_points.json`.
 - Delete the current set of points with `rm ~/Desktop/points.json`.
8. Before you leave, check in with a TA.

4 Startup Procedure

1. Make sure that the **VICON system** is on and running, and tracking the quad you intend to fly.
2. Check that the battery indicator of the CrazyFlie is blinking (not solid) red, and that the VICON tracks the quad.
3. Place the quad near the middle of the VICON area (Safety goggles!). Place the blue lights facing the Desktop computers.
4. Make sure the other quads are hidden from the VICON.
5. Place your code `se3_control.py` and `waypoint_traj.py` into the directory `~/meam620-20/proj1_1/code/`.
6. Open a new terminal and navigate to `~/Desktop`. You can use the command
7. Symlink the trajectory points you want to fly to `~/Desktop/points.json`. Use the command `ln -s hover_points.json points.json`.
8. In the terminal of the desktop, run `mav_gui`. This will bring up a GUI where you can control the Crazyflie. Take note of the Motors OFF button! It's very important!

9. Make sure the *Robot* name in the GUI matches the number of your Crazyflie. For Crazyflie with the number '03', the Robot name should be 'crazy03' and so on.
10. When you are ready to fly, alert the TA.
11. You are now ready to fly! Open a new terminal, while still being able to see the GUI. Click *Motors ON* and then *Take Off*. Then click *Land*, and then finally *Motors OFF*. This is the safest way to land the quadrotor.
12. To run your controller, click *Motors ON* and then *Take Off*. When you have noted that the Crazyflie is hovering stably, then you may run `run_control mav_name:=crazy03` with the number corresponding to the correct quadrotor in the terminal to run your controller. It will run whatever waypoints are in `~/Desktop/points.json`. Then click *Land*, and then finally *Motors OFF*.
VERY IMPORTANT: If the quadrotor is unstable, hit *land* then *motors OFF* **immediately**. If the quadrotor is *wildly* out of control, you may also hit *Motors OFF*, but it may damage the quad as it drops out of the air.
13. Each time the student controller is run, data will be stored in a *rosbag* with the VICON data and commands, in the folder `~/data`. Ensure after every flight that the data is in fact being created. We will provide a script for you to process it after this lab.

5 After a Crash

Check the following things:

- The battery is centered and the quad balance
- All props are straight, level, and spin freely
- None of the motor holders are broken
- If it no longer flies right contact the TA for repairs.

6 Information about the CrazyFlie 2.0

(Disclaimer: the following section is partially cut-and-paste from Bitcraze’s website)

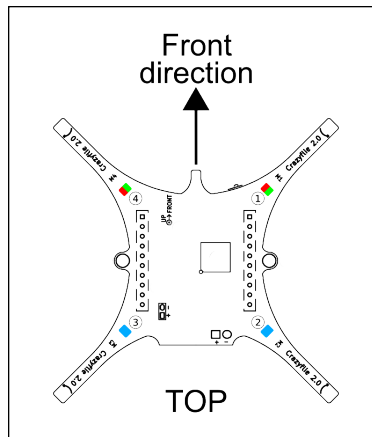


Figure 1: The meaning of the lights:

- power on and all is good: The blue LEDs (2 and 3) are fully lit and the front right LED (1) is blinking red twice every second.
- power on and all is good but **sensors** are not yet calibrated: The blue LEDs (2 and 3) are fully lit and the front right LED (1) is blinking red with 2 seconds interval. Put the Crazyflie 2.0 on a level surface and keep it absolutely still to calibrate.
- radio connected: The front left LED (4) is flickering in red and/or green.
- battery low: The front right LED (1) is fully lit in red. It’s time to land and re-charge the battery.
- charging: The back left blue LED (3) is blinking while the right back blue LED (4) is lit.
- self test fail: The right front LED (1) is repeatedly blinking five short red pulses with a longer pause between groups.

First things first: On the bottom of each CrazyFlie, you can find a marker that should read “01” through “07”. Make sure you are flying the robot you intend to, not the one of the other team!

The *back* of the CrazyFlie is where the battery cables are hanging, the *front* is where the little nose protrudes from the board. The CrazyFlie is charged through a micro-usb cable, and (in our situation) controlled via 2.4GHz RF signal.

To switch the CrazyFlie on, push the microscopic button on the *front* of the board, somewhat hidden by the battery. This will initiate the startup sequence:

- run self tests - the Crazyflie 2.0 checks that the hardware is OK
- calibrate sensors - the Crazyflie 2.0 reads its sensors to get base values. It must be absolutely still to do this, so it is best to put it on a level surface for a second.
- ready to fly!

The lights of the CrazyFlie are labeled M1-M4, see Fig. 1 for their meaning. The most important one is the battery indicator: blinking red is good, SOLID RED IS BAD. If the battery is empty you will be given a different quad to fly.

7 Using the VICON

There is a separate PC dedicated to the VICON. Unless it is already running, start the “VICON Tracker” software, and click to the “Objects” tab (we already defined tracker objects for you).

Scroll down until you find the crazyflie you want to track and mark it. They are named e.g. “meam04”. If you now put the quad into the VICON area you should see it show up on the VICON Tracker software, with a little coordinate system attached. If you don’t see the coordinate system, the tracking is not working.

Any coordinates you will be given refer to the VICON’s reference frame. The $[0,0,0]$ spot is marked on the ground in the middle of the area, with the x-axis pointing away from you (towards the elevators), the y-axis to the left (towards the hallway), the z-axis up.

8 Differences Between Simulator and Real Environment

- The Crazyflies accept attitude commands, i.e. you can directly command the angle, and the on-board controller will close the loop for you. Any moments you command will be ignored.
- You will have to re-tune your gains because the mapping between commanded thrust and actually observed force is not perfect, and in general the attitude controller is much less responsive than what you developed in the simulator.
- The good news is: you will only have to tune the position gains.
- You may have to adjust the mass slightly up or down to make the quad rotor hold altitude at exactly the desired position.