# MEAM 620 Project 1 Phase 5 - The Maze

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.

During the second lab session, you are asked to fly through a predefined maze which has been set up in Levine 457. This task will test your controllers, trajectory generators, and path planners in a real world environment. The maze consists of boxes and barriers which you must make the Crazyflie fly around, over, and under, to reach a goal location. During the lab you will record 3 datasets, one from each start to goal.

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 2: The Maze

Your task for this phase is to fly the maze with 3 different start and end points. 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.

### 3.1 What to bring

You must bring the following scripts from the previous phases:

- `se3_control.py`
  Be sure to use your gains from the last lab, not from the project!
- `occupancy_map.py`
- `graph_search.py`
- `world_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.

2. Check with the TA that the VICON system is on and running, and tracking the quad you intend to fly.

3. Check that the battery indicator of the CrazyFlie is blinking (not solid) red, and that the VICON tracks the quad.

4. Place the quad near on one of the unoccupied blue start boxes in the VICON arena

5. Make sure the other quads are hidden from the VICON.

6. Place your code `se3_control.py`, `occupancy_map.py`, `graph_search.py`, and `world_traj.py` into the directory `~/meam620-20/proj1_3/code/`.

7. Set the trajectory speed to very slow in your code to ensure you do not break the quadrotor! Show the TA your trajectory for each map before proceeding. We love that you want to fly fast, but hate when we have to fix the quadrotors when the break, so do not fly fast in this lab. Conditions:

   (a) resolution = 0.125

   (b) margin = 0.3

   (c) average speed: 0.6 m/s

   (d) max speed: 2 m/2

   (e) max acceleration: 3 m/s$^2$

   Each map should take at least 10 seconds the first time you fly to make sure you are flying sufficiently slowly.

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, map:=N*,with N being 1,2, or 3 corresponding to the map you are taking off from. Map 1 starts at the box closest to the computers, Map 2 at the left far box, and Map 3 at the right far box. If your quad is running too fast, STOP! After the end of your trajectory, 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. If the quad is about to hit a barrier, STOP immediately. If you run 'Ctrl+c' in the terminal, it should also turn the motors off.

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.

14. After each run, save and label the rosbags so you know which bag corresponded to which map. You need to get data for all 3 maps!

15. At the end of the lab, make sure you have copied your rosbags, as well as updated files (if applicable) onto a storage device.

16. Clean up!

    - Make sure the Crazyflie is back on the charger.
    - Delete the python files you added as well as any rosbags you generated in `~/data`.

17. Before you leave, check in with a TA.

## 4 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.

# 5 Information about the CrazyFlie 2.0

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

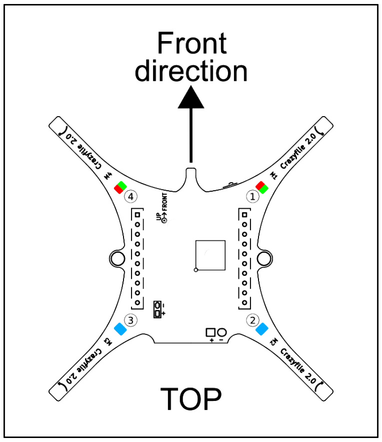

Figure 1: The meaning of the lights:

- 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 read 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.

# 6 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.

# 7 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.