1 Lab Scope and Aim

Flying robots or Micro Aerial Vehicles (MAV)s are becoming increasingly common and span a huge range of sizes and shapes and they are widely used in different applications such as power-line monitoring, bridge inspection, urban structure coverage, forest fire inspection, etc. The aim of this lab is to introduce the students to the concepts of unmanned aircraft modeling, control, as well as into the basics of motion planning. In particular, the robot that you will use in this lab is the crazyflie 2.0 and you will be developing your control and motion algorithms within Simulink framework. This lab assignment involves (a) an intro to the CrazyFlie 2.0 (b) template code to use on the robot.

Prerequisites: The students are suggested to have a good background in the following areas:
- *Automatic Control Systems*
- *Advanced Mathematics*
- *Complete Lab1 and Lab2 Tasks*
- *Complete knowledge of coordinate frames*
- *Simulink and Programming in General*

Lab Supplementary material: This lab is accompanied with the supplementary material to support the assignments found in Canvas and google drive link. Check this document for the details.

Note1: Description of crazyflie found in `https://wiki.bitcraze.io/projects:crazyflie2:index` **and** `https://wiki.bitcraze.io/projects:crazyflie2:userguide:index`
Note2: Provide plots demonstrating reference and actual trajectories.
Note3: Use the provided "Lab3_Crazyflie_Virtual.slx" simulink file.
Note4: In this lab you need to download the virtual machine provided in the drive link presented in Section 3.1 (same as lab 2)
Note5: It is expected that the Crazyflie moves slowly from among different poses in the simulation

2 Simulink

The lab implementation is structured around Simulink. Simulink, developed by MathWorks, is a graphical programming environment for modeling, simulating and analyzing dynamical systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. More information regarding simulink can be found at `https://se.mathworks.com/products/simulink/getting-started.html`

3 CrazyFlie

The Crazyflie Nano Quadcopter is a miniature quadcopter as depicted in Figure 3. It weights about 27 gr and is 9 cm motor-to-motor. The flight time is up to 7 min with standard 170 mAh Li-Po battery. It has IMU with 3 axis gyro, 3 axis accelerometer, 3 axis magnetometer, and high precision pressure sensor. The framework is already installed and you should not upgrade the on-board framework. More information regarding crazyflie can be find in `https://www.bitcraze.io/crazyflie-2/`.

Figure 1: Crazyflie 2.0.

3.1 Virtual Machine with simulated Crazyflie

In this part you will download and configure a new Virtual Machine file that includes the simulated Crazyflie executable file.
To use the new virtual machine file follow the steps below:

1. Download the modified Virtual Machine from here (same as lab 2).
2. Run the virtual machine using the Virtual Machine Player from Lab 2
3. The first time you start the virtual environment you will get a notification whether you moved or copied the file. Select the option "I copied it" as shown in Figure 2.

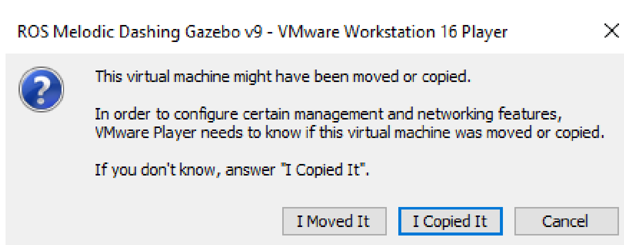

Figure 2: Virtual Machine first time message

3.2 Start simulated Crazyflie

After completing the steps provided in subsections 3.1 you are now ready to start working with the simulated Crazyflie in the virtual machine. Initially, you need to select the executable icon with the name "Gazebo Crazyflie" as shown in Figure 3. This will start the Gazebo simulated Crazyflie. When the simulation is up and running, start the Simulink file provided in Canvas, confirm that your connection with ROS network has been established (same step as in Lab2), since it is expected that the IP has changed compared to the other Virtual Machine from Lab2. Now you are ready to move to the Assignments section.

Figure 3: Crazyflie executable file to start the simulation

4 Assignments

Task 1: Hovering Create a controller for the Crazyflie to hover at 0.5 m altitude from the initial position of Crazyflie.

Task 2: Pose A to Pose B Create a controller of your choice, which can track arbitrary position and yaw references. The controller shall be stable and converge to the position reference, $\|p - p_{ref}\|$ to within 0.1 m and 0.1 radians. Where p is the position of the Crazyflie and p_{ref} is the reference point.

Task 3: Following a trajectory In this task you will define different trajectories and validate your controller with each of them. The trajectory followed by the crazyflie and reference path should be plotted for evaluation.

- Create a trajectory of square reference at constant altitude 0.5 and the Crazyflie should follow the reference trajectory looking in a fixed direction.
- Create a trajectory of square reference at constant altitude 0.5 and the UAV should follow the reference trajectory looking at the next waypoint.
- Create a trajectory of circular reference of 1 meter diameter with constant altitude 0.5 and the Crazyflie heading should be towards the center of circular path while it follows the reference trajectory.
- Create a circular reference of 1 meter diameter, tilted 45 degrees in space as depicted in Figure 4, while looking in a fixed direction. Modify the previous controller as necessary to track the path.

Task 4: Swarm of Crazyflies In this task you should develop your Crazyflie to follow another Crazyflie that will navigate inside the flying arena (Another crazyflie cannot be included in the simulation so we assume a virtual one). More specifically, you will have to keep constant distance to the leader and follow its trajectory. Your Crazyflie will start from random initial position. You will have access to all states of the leader Crazyflie, but there will be no information regarding the trajectory that will follow.

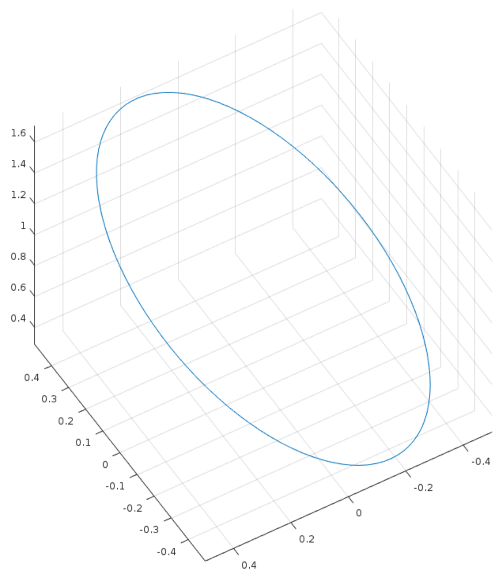

Figure 4: A circular path tilted 45 degrees in space with a diameter of 1 meter.