PART 7: Demonstration

This part is to be performed in the μASTA arenas. Detailed instructions on how to use the workstations surrounding these arenas can be found here:

`http://rsewiki.elektro.dtu.dk/index.php/UASTA`

The following three exercises can be completed by adapting the Simulink model from part 6. To do so, uncomment and wire the ROS communication blocks (red area), and comment the blocks related to the simulated drone. This Simulink model can be run either directly on the workstations, or on your own laptop with an ethernet connection to the workstation. In either case, make sure that the ROS network configuration is setup correctly.

Exercise 7.1: Navigating a 3D maze

Complete this exercise in the μ ASTA far from the pool. In this cage, you need to use the CrazyFlies that mount the loco positioning deck. The goal of this exercise is to be able to navigate the 3D maze. Use either the greedy best-first or A* search algorithm to obtain a path which the drone can follow. The path must be scaled to match the real maze. An offset is also needed. How the scaling and offset is applied can be seen in the bottom of the `map_3d.m` script. Adjust the scalings and offsets as needed. The code provided in exercise 4 can be used, but it must be upgraded from 2D to 3D if this is not already done. Use way-points to control the drone.

The starting position is marked E, F, G or H and it can be seen on the floor of the maze. The goal is to reach A, B, C or D, also marked on the floor of the maze. A map of the maze is contained in the file called `wall.txt`.

On the demonstration day a new maze will be made. The map of the demonstration day maze will be provided. The format of the map will be the same as with the `wall.txt` file.

Exercise 7.2: Porting the position controller to the real system

Complete this exercise in the μASTA close from the pool using the Optitrack 3D motion tracking system. Make sure that the CF you are using carries reflective markers. Usage instructions for the Optitrack setup can be found here:

`http://rsewiki.elektro.dtu.dk/index.php/ASTA_motion_capture`

This exercise can be completed either with your own position controller or with the default one. The real drone's attitude controller accepts roll and pitch commands as in the simulation (these commands just need to be converted from radiant to degrees). However, it tracks yaw rate commands instead of yaw, and thrust commands are expressed as PWM commands, which are integer values ranging from 0 to 60'000. These values essentially corresponds to the average voltage applied to the motors.

Task: Identify the relationship between the thrust commands calculated by the position controller and the PWM commands accepted by the real drone's attitude controller. Fitting a first order polynomial (i.e., $y = ax + b$) on a data-set consisting of PWM commands and vertical accelerations should do the trick. You can gather this dataset by flying the drone manually in attitude mode using a joystick or a gamepad. Then, implement this relationship in the Simulink model and try to get a CF to:

1. hover stably on point (1, 1, 1) for 10 s with a maximum error of 10 cm, and
2. move from setpoint (1, 1, 1) to (2, 1, 1) while ensuring that the overshoot is lower than 30 cm.

Report: For each experiment, show:

1. a x-y plot,
2. a plot of x, y, and z versus time (with references), and
3. a plot of the x, y, and z error versus time.

Exercise 7.3: Aggressively navigating through hoops

This exercise can be completed either with you own position controller (if it accepts velocity and acceleration setpoints) or with the default one. Also, it requires the thrust-PWM relationship from exercise 7.2.

Task: Modify the code in `uas_trajectory.m` to navigate through the hoops standing in the μASTA close to the pool.
Report: Show the x-y plot of the reference trajectory superimposed with the achieved one.