Sim Lab 6. Final demonstration and competition

`Update 2022-11-29: clarified error in scoring section`

This is an unusual lab: you will not be working *during* the lab, but will work on your own time to complete this. There is no in-person Lab 6.

6.1 Introduction

This is the final experimental section for this class. In labs 1 through 5, you have implemented a feedback control strategy to stabilize a quadcopter in hover, including performing state estimation and identifying the vehicle's physical parameters. Each group was given the same hardware – now we will see which group is able to get the best performance out of the system.

The goal is to have the vehicle take off, and land softly at a certain fixed point in the flight space. During the flight, the vehicle must also achieve a minimum height.

The vehicle cannot measure its horizontal position with the provided sensors – this means that the vehicle also cannot estimate its horizontal position (it is *unobservable*). We know that the horizontal position is a pure integrator of velocity, meaning that it has an eigenvalue at zero, which in turn means that we cannot stabilize the horizontal position. This means that the vehicle's estimate of its position will inevitably become worse as time goes on; you'll have to take this into account for the control.

There will be no synchronous lab session for this. You and your team will prepare your solution, and evaluate your own performance. You will submit your deliverables (see Sec-

tion 6.5), and you will be scored on your ability to accomplish the goal.

6.2 Experimental procedure

The basic goal is to take off, fly to the minimum height of 1.5m, and then land softly close to the target at a position (1,0,0)m in the space, at the latest after 25s. The experiment will be run as follows:

1. You will start the competition simulator program.
2. The program will automatically send some radio commands (see below) for the duration of the flight.
3. At the end of the flight, the program reports:
   (a) Whether the vehicle cleared the minimum height during the flight.
   (b) The distance to the target position, as a distance.
   (c) The maximum proper acceleration norm experienced by the vehicle during its flight, as a measure for whether it achieved e.g. a soft landing (and didn't crash on the way).

This experiment is done completely autonomously, there is no opportunity for you to push any buttons through the GUI.

Note – the "Print Info" button is also disabled. To get the system to print information, you can call that function from within your main loop, e.g. triggered at a certain time step.

6.3 Evaluation program button sequence

For your initial development, you may simply use the standard simulator program with the usual optional button inputs etc.; note though, that you *have to* use the evaluation program during scored flights.

The evaluation program is pretty simple, and does the following, where t is the time since the program started, in seconds. The "Arm" button is sent throughout.

- $0 < t \leq 1s$: Program transmits the following buttons: Yellow

- $1 < t \leq 2s$: Program transmits the following buttons: (none)
- $2 < t \leq 3s$: Program transmits the following buttons: Red, Blue
- $3 < t \leq 23s$: Program transmits the following buttons: Red
- $23 < t \leq 24s$: Program transmits the following buttons: Red, Green
- $t > 24s$: Program transmits the following buttons: (none)
- $t = 26s$: Program terminates.

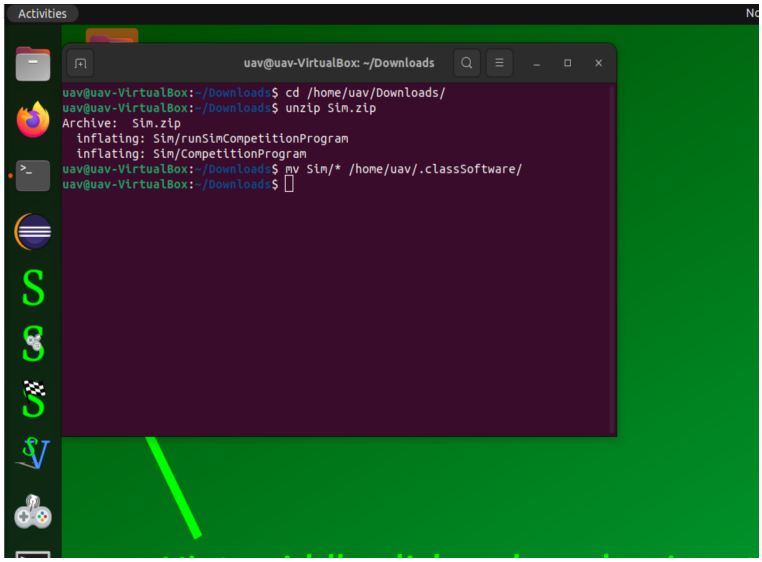
You may use the Yellow, Blue, and Green button signals for anything you like (e.g. use Yellow to reset your state estimator, use Blue to perform a takeoff, and use Green to perform a soft landing). The vehicle must also still be armed to fly.

6.3.1 Getting the evaluation program

From your virtual machine, go to `https://tinyurl.com/uavvm` (remember that you have to log in thru Calnet), and download the `sim.zip` file. Then, open a terminal window, and enter the following commands (easiest is to copy & paste them into the terminal):

- `cd /home/uav/Downloads/`
- `unzip Sim.zip`
- `mv Siunzip Sim.zipm/* /home/uav/.classSoftware/`

The output should look like the below:



You can now run the competition program by clicking the button with the green 'S' with a checkered flag over it (seventh from the top).

6.4 Scoring

The grade for this lab is broken down as follows:

1. Final displacement [70%]: your score will be based on your lowest total displacement legal flight, relative to the best performance of your peers. This displacement is measured on the floor, from the target position to the center of the vehicle where the vehicle first touched the ground.
2. Smoothness [30%]: this score reflects e.g. how soft your landing was, and is the peak proper acceleration norm experienced during the flight. Lower is better.

We will apply a monotonic weighting to the above scores, and add them together to produce the final score. A flight that does not achieve the minimum height is scored zero on both scores.

6.5 Deliverables

You must submit as a single `zip` file the content of the `source` folder that contains your user code. We may use this to verify your claimed scores below. You must submit by **12:30 (just after midday) on Thursday, 1 December**.

Then you will run the simulation program 5 times in a row, and for each flight keep track of the reported "Distance to target", the "Roughness", and the "Max height reached". You will report these in a table, with a row for each run, and then extra rows reporting the maximum over all 5 runs, the minimum over all the runs, and the average over all the runs. These five runs have to be done consecutively, you're not allowed to run it many times and cherry-pick the runs that work well. If we evaluate your code, we must get performance similar to your reported results.

Prof. Mueller's code achieves a final position error of around 0.3m, with a roughness of around 17.

This lab is a *competition*, so your performance will be compared to that of your classmates. Grading will be very gentle, but the winners will receive eternal fame & glory in addition to the best score for the lab. Winners will be per section (ME136 and ME236U separately). There is a separate, similar, competition for the hardware groups.