Out: Sept 27 **Due:** Oct 5, 11:59pm.

16.30/31: Feedback Control Systems

Fall 2015

Laboratory Exercise #1

This laboratory exercise is designed to familiarize you with the Parrot Rolling Spider mini-drone platform and its Matlab/Simulink Toolbox. It also allows us instructors to make sure the platform is working correctly on your computer. Before starting this laboratory exercise, please carefully read the GettingStarted document and watch the three videos "FlashingTheDrone", "Designing-Controllers", "FlyingAndAnalyzingFlightData".

The GettingStarted document provides all details on how you can upload the custom firmware on your drone, how you can design/simulate control systems with the Matlab toolbox, how you can generate and cross-compile embedded C code and upload the compiled binaries to your drone, and finally and how you can conduct an experiment and download the data into Matlab for analysis. The videos demonstrate all steps involved. A brief description of the videos are provided below:

1. **FlashingTheDrone:** This video shows you, step by step, how you can upload and run the custom firmware on the Parrot Rolling Spider mini-drone.
2. **DesigningControllers:** This video shows you how you can design and simulate control systems using Matlab/Simulink, how you can generate embedded C code of your control system in the Simulink block diagrams form, and how you can compile and upload the embedded C code to your drone.
3. **FlyingAndAnalyzingFlightData:** This video shows you how you can fly your drone, how you can download the flight data, and how you can analyze this data using Matlab.

Please consider looking at the GettingStarted document while watching the videos.

First, please make sure to follow all safety procedures. For this laboratory exercise, please execute all of the following steps, and turn in a report and data as described in bold font below:

1. First make sure that the Matlab/Simulink toolbox is fully installed. You can utilize the virtual machine for this purpose. To install the virtual machine, take a look at the "Equipping your ubuntu" section of the GettingStarted document on pages 11-12.
2. Upload the custom firmware to your drone. To do so, carefully read all steps for "Flashing the drone" part of the GettingStarting document on pages 13-15, and carefully watch the FlashingTheDrone video. Execute all these steps to flash your drone.
3. Make sure that you can connect to your drone and disconnect from your drone. For this step, carefully read the at the "Connecting to the drone" section of the GettingStarted document on page 16.
4. Simulate the PID controller available in the toolbox using the simulation model in "sim_quadrotor.slx," using the reference values that we provide in the same Simulink block diagram. The PID controller is the default controller in the simulink model and can also be found in the following folder of the Matlab toolbox:
   "trunk/Matlab/Simulation/controllers"
   Carefully read the "Simulation and Control Design" section of the GettingStarted document on page 17, and carefully watch the DesigningControllers video. You do not have to change the controller parameters (as demonstrated in the video for illustration).

In your report, explain what reference values we put in the quadcopter. Generate the following plots and add into your report:

- **One plot that describes the position of the drone. This plot should show three variables, namely the x, y, z coordinates of the drone, with respect to time, in one figure.**
- **One plot that describes the orientation of the drone. This plot should show three variables, namely roll, pitch, and yaw angles of the drone, with respect to time, in one figure.**
- **One plot that describes the velocities of the drone in the body-attached frame. This plot should show three variables, namely the instantaneous velocities of the drone in the body-attached frame, with respect to time, in one figure.**
- **One plot that describes the angular velocities of the drone in the body-attached frame. This plot should show three variables, namely the instantaneous angular velocities of the drone in the body-attached frame, with respect to time, in one figure.**

5. Generate embedded C code from this PID control system, and compile/upload the embedded C code to your drone. For this step, carefully read the "Embedded code generation" section of the GettingStarted document on pages 18-19, and carefully watch the DesigningControllers video.

6. Fly your drone with the PID control system. For this step, carefully read the "Flying" section of the GettingStarted document on pages 20-24, and carefully watch the FlyingAndAnlyzingData video. When the drone is flying, do not provide any reference commands through the DroneKeyboardPilot software (you can use the 'e' key, if you would like to exit the experiment); let the drone settle to the hover position and wait for the experiment to end (the experiment will end automatically after 20 seconds if not aborted by manually hitting 'e').

7. Download the data of the this experiment from your drone into the Matlab environment. For this step, carefully read the "Data Analysis" section of the GettingStarted document on page 25, and carefully watch the FlyingAndAnalyzingData video.

   Use the FlightDataAnalyzer.m file to generate the following plots and add into your report:

   - **One plot that shows the time evolution of IMU, altitude, and motor commands. This plot is the first figure generated by FlightAnalyzer.m.**
   - **One plot that shows the time evolution of altitude. This plot is the second figure generated by FlightAnalyzer.m.**
   - **One plot that shows the time evolution of positions and velocities in the x and y directions. This plot is the third figure generated by FlightAnalyzer.m.**
   - **One plot that shows the time evolution of optical flow and velocities in x and y directions. This plot is the fourth figure generated by FlightAnalyzer.m.**

   In your report, for each of the four plots, briefly discuss (in one short paragraph of at most five sentences) whether the data is what you expected and why.

   Also, rename the RSdata.mat data you downloaded into a new file entitled "LastName_experiment1.mat", and upload this file.

8. Again, fly your drone with the PID control system. For this step, carefully read the “Flying” section of the GettingStarted document on pages 20-24, and carefully watch the FlyingAndAnlyzingData video. A few seconds after the drone reaches the hover position, press the y key in the DroneKeyboardPilot software to have the drone fly to an altitude that is 0.6 meters higher.

9. Download the data associated with this experiment from your drone into the Matlab environment. For this step, carefully read the “Data Analysis” section of the GettingStarted document on page 25, and carefully watch the FlyingAndAnalyzingData video.

   Use the FlightDataAnalyzer.m file to generate the following plots and add into your report:

   - **One plot that shows the time evolution of IMU, altitude, and motor commands. This plot is the first figure generated by FlightAnalyzer.m.**
   - **One plot that shows the time evolution of altitude. This plot is the second figure generated by FlightAnalyzer.m.**
   - **One plot that shows the time evolution of positions and velocities in the x and y directions. This plot is the third figure generated by FlightAnalyzer.m.**
   - **One plot that shows the time evolution of optical flow and velocities in x and y directions. This plot is the fourth figure generated by FlightAnalyzer.m.**

   In your report, for each of the four plots, briefly discuss (in one short paragraph of at most five sentences) whether the data is what you expected and why.

   Also, rename the RSdata.mat data you downloaded into a new file entitled “LastName_experiment2.mat”, and upload this file.

10. Feel free to fly your drone around using the keys ‘a,w,s,d’ to go left, forward, backward, and right respectively; the keys ‘j’ and ‘l’ to turn clockwise and counter clockwise, respectively; and the keys ‘i’ and ‘m’ to go up and do down, respectively.

You should upload one .pdf report and two .mat data files through the stellar website. More specifically, please upload a report that *(i)* has your name on it, *(ii)* explains what reference values are used in the simulation in file “sim_quadrotor.slx,” *(iii)* the three sets of plots described in steps 4, 7, and 9, *(iv)* briefly describe what you are seeing in each plot and tell us whether the results are as you expected them to be, and *(v)* how much time it took you to complete this laboratory assignment. Please also upload two .mat files that contain the data described in steps 7 and 9. The files should be named “LastName_experiment1.dat” and “LastName_experiment2.dat.”

If you run into trouble, please utilize the forum. Please do not send individual emails to the instructors. If you use Piazza, your fellow students can answer your questions as well; hence, you get a quicker response. Furthermore, your fellow students will also benefit from the answers.