Out: Oct 30 **Due:** Nov 13th, 11:59pm.

16.30/31: Feedback Control Systems

Fall 2015

Laboratory Exercise #3

In this laboratory exercise you will implement the LQR feedback controllers that you computed in PSET3 - initially in simulation using the Simulink model that is part of the toolbox and then on the actual drone in order to use this controller to actually fly it.

Please make sure to follow all safety procedures before you fly the drone. In particular, please make sure to wear safety glasses and make sure your drone is fitted with the wheels.

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 your toolbox is functional to be used with the drone. This might include rerunning BuildUtils.sh and DronesetMACaddress.sh if you updated the entire toolbox by downloading the github-repo during PSET2.
2. Open MATLAB navigate to MIT_ROSMAT/trunk/matlab and run startup.m.
3. Open the file controllers/controller_fullstate/controller_lqr.slx. It contains a template simulink block "ControllerLQR" for the LQR fullstate feedback controller that you will implement. Use this block as your controller block in sim_quadrotor.slx.
4. **Briefly comment on how the structure of this controller differs from the controller used in LAB2. In particular, briefly address the fact that the dynamic model used in PSET/LAB2 used system input variables different from those in PSET/LAB3.**
5. Now run the original version of the m-file LQRcontrol.m from PSET 3 (i.e., with the original values for bounds and cost weights). It generates the LQR fullstate feedback controller matrix $K_{lqr\_toMotorcmd}$.
6. Complete the "ControllerLQR" simulink block such that it implements an LQR fullstate feedback controller that uses the matrix $K_{lqr\_toMotorcmd}$. In the "ControllerLQR" block, please navigate to "FullstateController" to implement the missing steps.
(Side note: The other blocks are essentially just for infrastructural purpose. They have quick descriptions inside and we will have a look at them in a recitation for those who are interested.)
7. Simulate the drone's behavior with this new controller.

In your report, show your implementation of the "FullstateController" block. Generate the following plots and add them with comments to 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 shows the four motor commands send to the drone with respect to time, in one figure.**

In your report, for each of the three plots, briefly discuss (in one short paragraph of at most five sentences) whether the data is what you expected and why. Also, evaluate the control performance: Tell us if you are satisfied with the control performance and if you anticipate issues with the real drone.

8. Generate embedded C code from this LQR feedback 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 in pages 19-20, and carefully watch the DesigningControllers video.

9. Fly your drone with the LQR feedback control system. For this step, carefully read the "Flying" section of the GettingStarted document in pages 21-25, and carefully watch the FlyingAndAnlyzingData video. Once the drone is in stable hover, hit the 'y' key to provide an altitude reference that is 0.6m higher (you can use the 'e' key, if you would like to exit the experiment).

10. 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 in page 26, and carefully watch the FlyingAndAnalyzingData video.

    Use the FlightAnalyzer.m file to generate the following plots and add them with comments to 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.**

    In your report, for each of the two plots, briefly discuss (in one short paragraph of at most five sentences) whether the data is what you expected and why. Tell us if you are not satisfied with the control performance and list reasons that could explain it - you might be able to link them to LQR cost weights!

11. Repeat steps 5-10 with a new LQR controller that has faster altitude dynamics. To this end, first try the LQR parameters for faster altitude dynamics that you came up with in PSET3. It is very well possible that these parameters decreased your control system's robustness significantly and, as a result, the real drone became unstable. You need iterate your choice of LQR parameters then. As mentioned in PSET3 tweaking the cost weights on z and ρ only is promising.

12. **Please tell us how long it took you to complete this lab exercise.**

You should upload one .pdf report through the stellar website.

If you run into trouble, please utilize the Piazza website. 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.