ASEN 3728 Aircraft Dynamics
Programming Homework 2

Due date listed on Gradescope.

In this assignment, you will implement a controller for a quadrotor in the inertial x-z plane. You will follow the linear control design process outlined in lecture to first choose control gains for the linearized equations of motion and then optionally modify the controller to improve performance under wind.

The quadrotor dynamics, including physical parameters such as mass, gravity, moment of inertia, aerodynamic coefficients, and control limits are implemented in the `quadrotorDynamics` function. For the two-dimensional motion, the state vector is $\mathbf{x} = [x_E, z_E, u^E, w^E, \theta, q]^T$.

The linearized equations of motion are

$$\begin{pmatrix} \Delta \dot{x}_E \\ \Delta \dot{z}_E \\ \Delta \dot{u} \\ \Delta \dot{w} \\ \Delta \dot{\theta} \\ \Delta \dot{q} \end{pmatrix} = \begin{pmatrix} \Delta u \\ \Delta w \\ -g\Delta\theta \\ \Delta Z_c/m \\ \Delta q \\ \Delta M_c/I_y \end{pmatrix}$$

You will implement a linear control law for the pitch moment with the following architecture:

$$\Delta M_c = -k_1 \Delta q - k_2 \Delta\theta + k_3 \Delta u^E + k_3 k_4 (\Delta x_E - x_{E,r}).$$

All files are available by cloning the git repository at `https://github.com/zsunberg/Aircraft-Dynamics-Materials` and navigating to the `assignments/P2` directory. A zip file is also available at `https://github.com/zsunberg/Aircraft-Dynamics-Materials/raw/main/zips/assignments/P2.zip`. It is possible that there will be bugfixes to the assignment after it is released. These will be announced on Canvas.

The gains k_5 and k_6 for the linear control law for Z_c have already been set to acceptable values in the template files. These gains were selected using the pole assignment strategy. You may modify these values to better suite your controllers needs, but you are not required to do so.

Task 1. Choose k_1 and k_2 using the pole assignment strategy with the pitch rate and pitch angle dynamics. Choose a target rise time and damping ratio and calculate k_1 and k_2 accordingly. In your submission, you do NOT need to show the derivation of the equations for the gains, however, you must include your chosen rise time and damping ratio and include the code used to calculate the gains, along with printing their values.

Task 2. Choose k_4 using the pole assignment strategy by assuming that the inner loop controller is much faster than the outer loop controller as shown in class. Choose a target time constant for the outer loop that is 5-10x slower than the inner loop. In your submission, you do NOT need to show the derivation of the equations for the gain, however, you must include your chosen time constant and include the code used to calculate the gain, along with printing its value.

Task 3. Choose k_3 by plotting a root locus and choosing an appropriate value. The deliverables for this task are the root locus plot and the printed value of k_3.

Task 4. Print the gain matrix K and the closed-loop dynamics matrix, A^{cl} for the state vector $\mathbf{x} = [\Delta x_E, \Delta u^E, \Delta\theta, \Delta q]^T$. Then, use the `initial` function to plot the response of the closed loop system. The code in `TEMPLATE_report.m` gives the C and D matrices to create two plots, one that outputs the state values and one that outputs the control values. Verify that the gains that you chose produce acceptable performance (the origin is reached in a reasonable time). You may wish to revise the gains to improve performance. The deliverables for this task are the K and A^{cl} matrix along with the two plots.

Task 5. Implement this control law in the `quadrotorLinearControls` function and use the `simulateQuadrotor` and `displayTrajectory` functions to visualize the motion of the quadrotor. The deliverable for this task is the plot produced by `displayTrajectory`. Optionally, you can animate the simulation with `animateQuadrotor`, but this is not a deliverable.

Task 6. Run the `evaluate` function on your control law. This will give you a score based on how fast the quadrotor reaches the goal and stays within the vicinity of the goal. Full credit is earned if the quadrotor reaches the goal region within 20 seconds. The score then decreases from 100 to 0 as your goal time increases from 20 seconds to 40 seconds. The `evaluate` function will produce a `submission.json` to certify your score. You will upload this file to Gradescope to receive credit for this assignment.

Task 7. (Optional) Optionally, you may include wind in the simulation to boost your resulting score. Increasing the average wind speed will increase the the score of control laws that can reach the target within the required time. Modifications may include changing the control gains of the linear control law or changing the structure of the control law itself. Experiment with different average wind speeds to see how robust of a controller you can design.

Deliverables

In order to use the template files, rename them by removing `TEMPLATE_`. To produce the report with plots, using the Matlab command `publish('report.m', 'pdf')` is highly recommended. Submit the following files to Gradescope:

- `submission.json` (make sure that the Gradescope autograder runs successfully when you submit!)
- `report.pdf` containing required output from the tasks.
- `quadrotorLinearControls.m`
- `quadrotorControls.m` (if you choose to do Task 7)