AE353: Design Problem 01
(Control of a 2D Quad-Rotor)

T. Bretl

This paper describes the first design problem that you will complete in AE353 (Spring 2020). It asks you to design, implement, and test a controller for a 2D quad-rotor. It also asks you to submit a report that describes both your method of approach and your results.

I. Nomenclature

q_1	=	horizontal position of robot
q_2	=	vertical position of robot
q_3	=	angle of robot in radians with respect to the horizontal
f_R	=	thrust of right-hand rotor
f_L	=	thrust of left-hand rotor
m	=	mass
J	=	moment of inertia
r	=	spar length
g	=	acceleration of gravity
$f_{\max}$	=	maximum thrust

II. Goal

The code provided in `DesignProblem01.m` simulates the motion of a 2D quadrotor (a “bi-rotor”) in MATLAB. For simplicity, we will call this 2D quadrotor “the robot.” This robot has two actuators, a right and left rotor, which can each apply a net thrust. This robot also has sensors—an inertial measurement unit (IMU), a motion capture system, or GPS, for example—that can measure its position, orientation, linear velocity, and angular velocity. The robot starts with some random position and orientation. The goal is to achieve a stable hover at some particular position and orientation that you get to choose.

III. Model

The motion of the robot is governed by the ordinary differential equations

$$\begin{aligned} m\ddot{q}_1 &= -(f_R + f_L)\sin q_3 \\ m\ddot{q}_2 &= (f_R + f_L)\cos q_3 - mg \\ J\ddot{q}_3 &= r(f_R - f_L) \end{aligned}$$

where q_1 is the horizontal position, q_2 is the vertical position, q_3 is the angle of the robot from the horizontal (in radians), m is the mass, J is the moment of inertia, g is the acceleration of gravity, r is the spar length (i.e., the distance between the center of mass and the center of each rotor), f_R is the thrust applied by the right rotor, and f_L is the thrust applied by the left rotor. Note that rotors cannot produce negative thrust, so f_R and f_L are bounded below by 0 and are bounded above by some $f_{\max}$. All parameters can be accessed within your controller code (`Controller.m`) in the structure `parameters` (e.g., `parameters.m` is the mass of the robot).

IV. Tasks

A. Do all of these things

- Choose the position and orientation that you want to achieve.

- Linearize the model about your chosen position and orientation and express the result in state-space form.
- Consider the application of zero input (remembering that "zero input" is *not* necessarily the same as "zero thrust in both rotors"). Determine if the resulting system is asymptotically stable. Implement a controller that applies zero input and plot the results as evidence to confirm (or deny) your prediction.
- Consider the application of state feedback. Determine if the resulting system is asymptotically stable. Implement a controller that applies state feedback and plot the results as evidence to confirm (or deny) your prediction.

B. Do one of these things (your choice)

- Carefully examine the differences between the results of your linear model and of the nonlinear simulation.
- Carefully examine how the initial conditions affect the resulting motion.
- Design, implement, and test a method of disturbance rejection that allows the robot to achieve hover at the desired position and orientation with zero steady-state error, despite an unknown added mass. You can simulate the motion of the robot with an unknown added mass (chosen at random) with the optional parameter `'disturbance'`.

C. Write a report

The report must be a PDF document that was generated using LaTeX that conforms to the guidelines for "Preparation of Papers for AIAA Technical Conferences" (`https://go.aerospace.illinois.edu/aiaa-latex-template`). The author must be listed as "Anonymous" (with no affiliations). The report must be exactly four pages. It must have an abstract (one short paragraph that summarizes key results) and five sections: Nomenclature (a list of all symbols used in your report), Goal, Model, Control Design, and Results. You may never have used LaTeX before. That's fantastic! Help will be provided, and you will learn a lot.

V. Deliverables

A. Submit a first draft by 11:59PM on Wednesday, February 5, 2020

This draft must be a PDF document that follows the guidelines provided in Section IV.C and that includes, at minimum, a rough version of your "Goal" and "Model" sections. It must be submitted at the following URL:

`https://forms.illinois.edu/sec/286241`

You may submit as many times as you want—only the last submission will be recorded.

B. Submit a second draft by 11:59PM on Wednesday, February 12, 2020

This draft must be a PDF document that follows the guidelines provided in Section IV.C and that includes, at minimum, a rough version of your "Control Design" and "Results" sections. It must be submitted at the following URL:

`https://forms.illinois.edu/sec/7837038`

You may submit as many times as you want—only the last submission will be recorded.

C. Submit a final report by 11:59PM on Friday, February 21, 2020

This report must be a PDF document that follows the guidelines provided in Section IV.C. You must submit three things in addition to the report:

- The LaTeX source files that you used to produce the document (in a ZIP folder with all figures, references, etc., in addition to the `.tex` file).
- The MATLAB code that you used to produce all of the figures, tables, and other results that are included in the document (in a ZIP folder). Your code must have a MATLAB script called `GenerateResults.m`. It should be possible for any of your peers to reproduce *everything* in your report by executing this script.
- The implementation of your controller as a single file `Controller.m`. If one of your peers runs the simulator with this controller, they should see behavior that is consistent with claims made in the report.

Your report must be submitted at the following URL:

`https://forms.illinois.edu/sec/1969565`

You may submit as many times as you want—only the last submission will be recorded.

VI. Evaluation

Your work will be evaluated based on meeting intermediate deadlines (10%), on peer reviews both of your code (20%) and of your report (50%), and on your peer reviews of other reports (20%). Reviews will focus entirely on content this time, as you learn how to use LaTeX. Reviewers will be encouraged to ignore issues with style, grammar, or any other aspect of your presentation, as long as there is no barrier to understanding your work.

Peer review will be "double-blind." You won't know who reviewed your report, and your reviewers won't know whose report they reviewed. To enable a double-blind review process, it is **very important** that your final report be completely anonymous. To repeat, **DO NOT INCLUDE YOUR NAME** or anything else that would identify you in any of the materials you submit—not in your PDF, in your two ZIP folders, or in your MATLAB script. Other details of peer review will be posted to piazza and discussed in class.

VII. On-Time Submission

Draft reports and final reports must be submitted on time or they will receive zero credit. I am so serious about this that I will even give you extra credit if you submit your final report early:

> **If the last submission of your final report occurs by 11:59PM on Wednesday, February 19, 2020—so, 48 hours before the final deadline—then you will receive 5% extra credit.**