Individual Coursework 2

In this coursework, we aim at modelling, simulating and controlling a quadcopter drone under different conditions. The following materials will be used to guide this exercise:

- [1] A. Gibiansky, Quadcopter Dynamics, Simulation, and Control (pdf, html). An approximated dynamic model of a quadcopter and some potential control strategies are described in the following document. **Please read it carefully before starting the exercise.**
- An errata document (attached on Moodle), fixing a mistake in [1]
- A Matlab code template (`Quadcopter.m`, `Sim_Quadcopter_script.m`) is provided to visualise a quadcopter drone simulation. This template does not include any physics simulation nor any controller, which will need to be implemented as part of this exercise. You are free to modify the code as you wish for implementing the exercises.

The submission of this coursework consists of:

1) A written report (pdf) with a maximum of 5 pages AND a maximum of 2500 words.
2) Organise the Matlab code for each question in separate folders (Q1, Q2, Q3, Q4), and submit everything as a .zip file.

Question 1 [Total 10 marks]

Implement a simulation of a quadcopter, following the non-linear model in [1], using the provided MATLAB visualisation code. The quadcopter should have as input the squared angular velocity Υ_1, Υ_2, Υ_3, Υ_4 of each of the 4 propellers (these variables are defined at the end of page 7 in [1]). Consider the following fixed parameters:

$$m = 0.3\ [\mathrm{Kg}], \qquad I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.4 \end{bmatrix}, \qquad g = 9.8\ \left[\frac{\mathrm{m}}{\mathrm{s}^2}\right], \qquad k_d = 0.2$$

$$k = 1, \qquad L = 0.25[\mathrm{m}], \qquad b = 0.2$$

Where:

- m is the quadcopter mass
- I is the quadcopter rotational inertia matrix
- k_d is the friction constant (assumed equal for all x-y-z directions)
- g is the gravitational acceleration
- L is the length from each propeller to the centre of the quadcopter.
- b, k are propeller constants as defined in [1]

To demonstrate that your simulation is working, test the following simulation scenarios:

- The quadcopter is in equilibrium.
- The quadcopter is in free fall.
- The quadcopter exhibits a change in rotation, at a constant altitude.

The quadcopter sampling time should be 0.1s (for both sensor and input updates), however, the dynamical simulation itself should be updated 10 times faster.

For each scenario verify that the simulation behaviour is consistent with the dynamical model and its specified parameters.

Submit your working code and answer in the report. For your answer: explain how to run your code, present the results of each scenario test, and analyse their results.

Question 2 [Total 10 marks]

Implement a second simulation of the quadcopter, but now using a linearised model approximation with the shape $\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu}$.

Compare the linearised model (implemented in this question) against the non-linear model (implemented in Question 1), by testing both of them in two different non-equilibrium scenarios where:

- The linear model has zero or close to zero error.
- The linear model has a large error.

Submit your working code and answer in the report. For your answer: explain how to run your code, explain the derivation of the linearised model, present the comparative tests, and analyse their results.

Question 3 [Total 20 marks]

Assume the entire state of the quadcopter can be measured by sensors with 100% accuracy. Also assume that the input γ_i to each propeller is limited to +/-1.5. Using the quadcopter simulator from question 1, implement a **full-state feedback controller** that makes the drone perform the following trajectory:

a. Starts at (0,0,0)
b. Moves up to (0,0,5)
c. Stays at (0,0,5) for 5 seconds.
d. Moves to (0,2.5,5)
d. Moves along a circular trajectory with radius 2.5 on a vertical plane with constant x=0, passing through the points (0,0,7.5), (0,-2.5,5), (0,0,2.5) and back to (0,2.5,5).
e. Moves to (2.5,2.5,2.5)
f. Lands at (2.5,2.5,0) safely, by moving down at a constant speed of 0.1 m/s.

Submit your working code and answer in the report. For your answer: explain how to run your code, explain how you implemented the controller, present simulation tests, and analyse their results.

Question 4 [Total 10 marks]

Modify the simulator of question 1 so that the quadcopter measurements come from noisy sensors that do not measure the entire state, and its dynamics are affected by wind disturbances. Choose appropriate sensor and wind models based on a realistic scenario (e. g. based on existing sensors), providing references and justifications as needed.

Re-implement and re-run the controller of Question 3, but now including also a state observer.

Submit your working code and answer in the report. For your answer: explain how to run your code, explain how you implemented the controller, present simulation tests, and analyse their results.