Group Coursework 1

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

[1] A. Gibiansky, Quadcopter Dynamics, Simulation, and Control. 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.**

A Matlab code baseline (`Drone.m`, `quadcopter_script.m`) is provided to visualise a quadcopter drone simulation. This template does not include any simulation of the quadcopter dynamics nor any controller, which will need to be implemented as part of the exercise.

The submission of this coursework consists of:

1) A written report, answering each of the posed questions individually. Explain code and provide evidence of its working condition through plots and/or print outputs as necessary.
2) A Matlab code that implements the simulation of a quadcopter and the requested controllers. This includes modified files `Drone.m`, `quadcopter_script.m`, and any other additional auxiliary scripts that are deemed necessary.

Questions [Total 100 marks]

Question 1 [Total 25 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:

$$m = 0.2\ [\mathrm{Kg}], \qquad I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}, \qquad g = 9.8\ \left[\frac{\mathrm{m}}{\mathrm{s}^2}\right], \qquad k_d = 0.1$$

$$k = 1, \qquad L = 0.2[\mathrm{m}], \qquad b = 0.1$$

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 your simulation is working, implement and run the following experiment:

a. Quadcopter starts at an altitude of 5 metres
b. From 0 to 2 seconds: set a constant and equal value to all the inputs Υ_1, Υ_2, Υ_3, Υ_4 that perfectly compensates the gravitational force, so that the quadcopter stays in the same position (no need to implement a controller).
c. From 2 to 4 seconds: increase the value of all the inputs by 15%, so the quadcopter starts lifting
d. From 4 to 8 seconds set Υ_3 to zero, the drone should start rotating and eventually fall.

Submit the working code **[15 marks]**. For the written answer: explain how the code works **[5 marks]** and provide plots of the simulation that display position and orientation of the quadcopter over time **[5 marks]**.

Question 2 [Total 35 marks]

2a) [15 marks] The non-linear dynamics of the quadcopter in state-space representation are described in [1], in pages 7 and 8. Derive a linearised model approximation with the shape $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$ around any chosen equilibrium point (justify your choice). Then discretise the linear system. You can use Matlab to help with all your calculations in this question (see: `syms`, `jacobian`, `ss`, `c2d`).

2b) [Total 20 marks] Implement a second simulation of the quadcopter, but now using the linear approximation derived in 2a. To demonstrate its different behaviour, use the same experiment from question 1.

Submit the working code **[8 marks]**. For the written answer, explain how the code works **[4 marks]** and compare the quadcopter position and orientation computed with the linear approximation versus the original non-linear model, by plotting both trajectories side-by-side **[4 marks]**. Comment on the results **[4 marks]**.

Question 3 [Total 40 marks]

3a) [Total 30 marks] Assume the entire state of the quadcopter can be measured by "perfect" sensors with 100% accuracy. Using the quadcopter simulator from question 1, implement a feedback controller that makes the drone perform the following trajectory:

a. Starts at (0,0,0)
b. Moves up to (5,5,5)
c. Stays at (5,5,5) for 5 seconds.
d. Moves along a circular trajectory with radius 2.5, with constant altitude z=5, passing through the points (2.5,2.5,5), (0,5,5), (2.5,7.5,5) and back to (5,5,5).
e. Lands at (5,5,0) safely (less than 0.1 m/s linear velocity when hitting the ground).

Submit the working code **[18 marks]**. For the written answer, explain how the code works **[6 marks]** and provide adequate plots of the simulated quadcopter trajectory in position and orientation **[6 marks]**.

3b) [10 marks] Assuming we are now dealing with real sensors, add Gaussian noise to the measurements of the state. Choose reasonable values for the noise magnitude of each state measurement, by identifying examples of sensors that could be utilised to measure them (you can be creative here). Re-run experiment from 3a) and provide plots of trajectories under noisy measurements.