Group Coursework 2

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.**

[2] A. Symington, R. De Nardi, S. Julier and S. Hailes, "Simulating quadrotor UAVs in outdoor scenarios," *IROS* 2014. Another approach to quadcopter simulation. In this exercise, we will be mainly interested in their proposed wind models (Section III A.) for one of the questions.

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. In each of the questions involving Matlab code, provide in the written report 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

1) **[5 marks]** The dynamics of the quadcopter in state-space representation are described in [1] at the start of page 8.

   The following comment is made: *"Note that our inputs are not used in these equations directly. However, as we will see, we can choose values for $\boldsymbol{\tau}$ and T, and then solve for values of $\boldsymbol{\gamma}_i$".*

   a) **[3 marks]** What are the advantages of this strategy?
   b) **[2 marks]** Can you identify a similar strategy (for a different system) described in the lecture slides?

2) **[6 marks]** Page 6/7 of [1] suggests a drone simulation implementation in Matlab. The following code to update the state variables is just an approximation of the non-linear dynamics:

```
omega = omega + dt * omegadot;
thetadot = omega2thetadot(omega, theta);
theta = theta + dt * thetadot;
xdot = xdot + dt * a;
x = x + dt * xdot;
```

   a) **[2 marks]** What is the approximation being done?
   b) **[2 marks]** How can we make sure that this approximation error is very small?
   c) **[2 marks]** What should be done, if we wanted to update the state without using this approximation, i.e an exact solution of the state equations from page 8 in [1]. (you don't need to implement it or solve the equations, just explain how to proceed).

3) **[14 marks]** Considering the state-space quadcopter dynamics at the beginning of page 8 in [1]:
   a) **[6 marks]** Define values for all the fixed model parameters. Justify, based on external references, that these are reasonable for a representing a real quadcopter.
   b) **[8 marks]** Derive a linearised model approximation with the shape $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$ around an equilibrium point. Then discretise it. You can use Matlab to help with all your calculations in this question.

4) **[5 marks]** In page 8 of [1] a PD controller is proposed to stabilise the quadcopter orientation by using as feedback gyroscope measurements of the angular velocities $\dot{\boldsymbol{\phi}}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{\psi}}$.
   a) **[3 marks]** Model an output equation $\boldsymbol{y} = f(\boldsymbol{x}(t))$ based on the gyroscope measurements
   b) **[2 marks]** Suppose that the following additional sensors are added to the quadcopter: an altitude sensor; a GPS sensor to measure horizontal position. Model a new output equation $\boldsymbol{y} = f(\boldsymbol{x}(t))$ based on all available sensors.

5) **[70 marks]** Consider a quadcopter with Wifi communication that allows it to receive a "move" command from a remote controller (further explained below). You may assume that the quadcopter has the parameters defined in question 3)a.

The quadcopter should have the following behaviour:

1: Start on the ground at (0,0,0);

2: Once a "move" command is received, it lifts up to position (0,0,2.5) and stays there until further instructions;

3: Once a second "move" command is received, it does a full circle trajectory at a constant 2.5m altitude with radius 2.5m, ending back at the starting point;

4: Land the drone on the ground safely (less than 0.1 m/s linear velocity when hitting the ground).

a) **[25 marks]** Change the Matlab template code so that it implements both a simulation of the non-linear quadcopter dynamics and a controller to achieve the previously described behaviour. Assume that we can directly measure the quadcopter state with no noise.

b) **[10 marks]** Assume now that the quadcopter does not have direct access to the state. Its measurements are given by all sensors described in question 4 (gyroscope, altitude, GPS), without any noise or errors. Design a new controller under this assumption.

c) **[10 marks]** Add Gaussian noise to the sensor measurements and investigate how much noise is your controller able to handle before starting to fail.

d) **[10 marks]** Add a wind model to your simulation, including a mean wind field from a random direction and turbulent wind. You can use the wind turbulence model described in [2]. Investigate how much wind the controller is able to handle before starting to fail.

e) **[15 marks]** Add a second quadcopter to the simulator that is initially at (0.3,0,0). While the first quadcopter moves with the behaviour previously described, the second one should autonomously move in order to keep a distance of 0.3m to the first robot, with a "follow the leader" behaviour. Assume this second quadcopter only has access to: 1) its own state (without noise); 2) the 3D position of the first quadcopter (without noise).