3D Control

The main objective of this project was to control a drone’s flight and meet the desired specification. The controller will be tested on 5 different testing scenarios as listed below:

1. 1\_Intro
2. 2\_AltitudeControl
3. 3\_PositionControl
4. 4\_Nonidiealities
5. 5\_TrajectoryFollow

Overview:

We are provided with lots of supporting files for this project. Our main goal is to complete the logic in QuadControl.Cpp file. Later we had to tune the controller parameters to pass the given tests. Tuning was the hardest part of this project.

1\_Intro:

In this scenario at first the thrust on drone was first set and in we were supposed to create the thrust such that it helps drone hover. Our task was updating the mass value in QuadControlParams.txt file. Below is the image after updating the mass value:

Graphical user interface

Description automatically generated with medium confidence

2\_AltitudeControl:

In this we were supposed the design now the actual controller in QuadControl.cpp file. Before developing the control logic, we were supposed to generate appropriate motor commands. This was done in GenerateMotorCommands() function. Below is the code for same:

Calendar

Description automatically generated with medium confidence

Now we designed the control logic for attitude control in AttitudeControl() function. However, before we did that, we first completed the logic of Body Rate Control and Roll Pitch Control. Below is the code and simulation image for Attitude Control:

Text

Description automatically generated

Graphical user interface

Description automatically generated

3\_PositionControl:

In this section we implemented the logic for position control that controlled lateral position i.e., the x and y position of drone. The code was implemented in LateralPositionControl() function. Below is the code snippet and simulation:

Text

Description automatically generated

Chart

Description automatically generated with medium confidence

Also after this the final step was to implement YawControl() function which helped in controlling the yaw angle. This was one of the easiest to implement controller. Below is the code snippet:

Text

Description automatically generated

4\_Nonidealities:

In this section there were 3 drones with different nonidealities. Our task was to tune the controller to deal with these nonidealities and properly control the flight of the drone. Below is the simulation of the same:

Graphical user interface

Description automatically generated

5\_TrajectoryFollow:

In this scenario we were supposed make the drone fly in predetermined trajectory. The trajectory was a shape of figure 8. We had to again tune the controller gains to achieve this feat. Below is the simulation:

A picture containing text, athletic game

Description automatically generated