1 Introduction

In this lab, you will design and implement three PID controllers in Aviones for longitudinal motion. The PID gains will be developed by examining the appropriate root locus plots.

2 Assignment

1. In the main aviones directory, open the file longitudinal_modes.txt and add the following modes:
   - Nothing (0)
   - Pitch with elevator (1)
   - Speed with Throttle (2)
   - Altitude with pitch (3)
   Run Aviones and open "View -> Control Window". Under Autopilot Control Modes: Longitudinal you should be able to select each of these modes.

2. In autopilot_main.cpp, replace the statement “if (ab.longitudinal_mode == 1)” with a case statement to handle the three autopilot modes that you will design in this lab. For case=1 use the function pitchAttitudeHold, for case=2 insert a new function speedHold, and for case=3 insert a new function altitudeHold. Create the appropriate header functions and insert template functions into autopilot_loops.cpp. The input to altitudeHold should be ab.altitude and the input to speedHold should be ab.airspeed.

3. Modify the function print errors, and the associated Matlab routine to add the following plots:
   a. Pitch command (ab.pitch) and Pitch (states.theta) vs. time.
   b. Speed command (ab.airspeed) and airspeed (states.airspeed) vs. time.
   c. Altitude command (ab.altitude) and altitude (h=-states.z) vs. time.
   d. elevator vs. time.
   e. throttle vs. time.

4. (pitchAttitudeHold). The objective is to implement and tune a PID controller that regulates the pitch angle \( \theta \) by using the elevator \( \delta_e \) as input. Since the pitch rate \( q \) is available from the rate gyro sensor, it is not necessary to differentiate the pitch signal. The structure of the PID controller is shown in Figure 1
   Implement this controller in the function pitchAttitudeHold. The elevator typically saturates at about 45 degrees. Write the function, so that you can tune the PID controller using the "View -> Autopilot Registers."

5. (speedHold). Throttle can be used to control the airspeed. The feedback structure is shown in Figure 2.
   Implement a PI controller for speedHold and tune the gains to get adequate performance.
6. (altitudeHold). The traditional method of autopilot design is called successive loop closure. The heading hold function is designed using this idea. Figure 3 shows the basic idea. The pitchAttitudeHold function is used as the inner loop to the altitudeHold command.

Implement a PI controller for altitudeHold and tune the gains to get suitable performance. The output of altitudeHold should be the desired pitch angle which is then fed to the pitchAttitudeHold function.

When the pitchAttitudeHold loop is enabled, you should also enable the speedHold loop to maintain airspeed. Otherwise the UAV pitches up but looses speed instead of increasing altitude.

3 Lab Write-up

Turn in the following items to the TAs.

1. A print out of the three autopilot functions.

2. A brief description of the PID gain tuning process that you used and the final PID gains that were obtained.

3. Plots showing step responses for each of the autopilot modes.