AERO - Simulation

Part 1 – Qualitative PID Control

Objectives

- Aero pitch control.
- Proportional-integral-derivative (PID) compensator.
- Manual PID tuning.

Theoretical background

1.1 Proportional Compensation

A proportional compensator drives the plant based on the difference between the current position of the system and the desired position. This difference is magnified by the *proportional gain* Kp which can be found either experimentally or calculated based on system requirements such as rise time. Greater proportional gain will result in a system with a shorter *rise time*, that is, the time needed for the system to reach the desired position. However since the system is constantly accelerating toward the set point, large proportional gains will generally lead to a system with large overshoot, and which oscillate for a long time.

1.2 Derivative Compensation

To deal with the overshoot and oscillation caused by a proportional compensator, many systems implement a derivative compensator in parallel. This compensator drives the plant based on the rate of change of the position (or velocity) of the system. As with proportional control, this derivative is magnified by the *derivative gain Kd.* The derivative compensator effectively acts as added damping in underdamped systems. To improve the stability of systems with derivative compensation, filtering is often added to prevent spikes in the derivative component of the compensation due to signal noise.

1.3 Integral Compensation

In many cases, the combination of proportional and derivative gain will result in a system which does not settle sufficiently close to the setpoint. In this case, an integral compensator may be added. This compensator drives the system based on the integral of the error over time magnified by the *integral gain Ki*. This component of the controller increases the longer the system remains far from the setpoint.

1.4 PID Control

The complete proportional, integral, and derivative control can be expressed mathematically as follows

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt}. \tag{1.1}$$

The corresponding block diagram is given in Figure 1.1. The control action is a sum of three terms referred to as proportional (P), integral (I) and derivative (D) control gain. The controller Equation 1.1 can also be described by the transfer function

$$C(s) = k_p + \frac{k_i}{s} + k_d s. \tag{1.2}$$

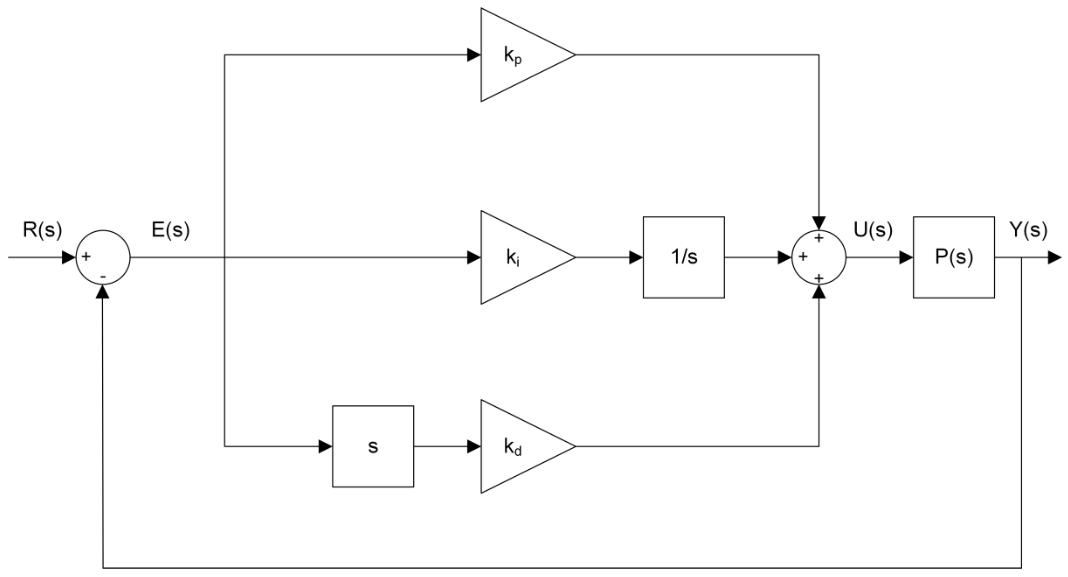

Figure 1.1: Block diagram of PID control

Preparation questions

Q1. Propose a method to identify the transfer function of a linear system from its unit step response!

Q2. Given a second order system and the plot of response to a unit step, provide the equations allowing to obtain the transfer function?

Q3. Given the following open loop transfer function : $G(s) = 1/(s^2 + 5s + 10)$ use Matlab and find a PID controller which will provide the following performance for an input step function:

1. Overshoot =0

2. Steady state error=0

3. Settling time <0.25 sec.

If you didn't find the transfer function, explain which requirements couldn't be satisfied? why?

2. In-Lab Exercises

2.1 Proportional Control

1. Open q_aero_qualitative_PID_Control.mdl. As shown in Figure 2.1 This QUARC® model implements basic command and unity feedback for the Quanser Aero. A control signal of 0 fed into the Motor Voltage Gain will command the Quanser Aero to remain at "Level" flight, that is, with +17V applied to both motors. The command signal is added and subtracted from the "Level" voltage for motor 0 and 1 respectively.

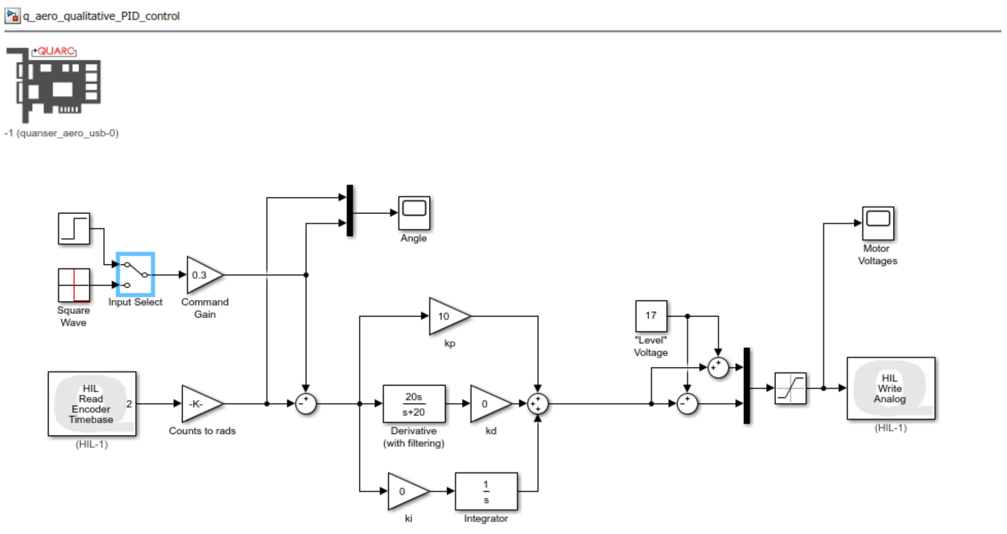

Figure 2.1: Unity-feedback block diagram

2. Build and run the QUARC® controller. The response should look similarly as shown in Figure 2.2.

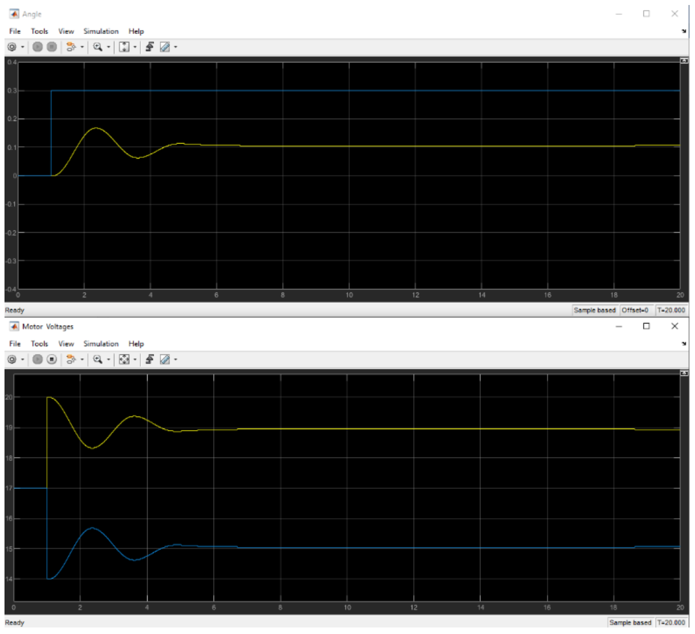
Angle
File Tools View Simulation Help
Ready
Sample based Offset=0 T=20.000
Motor Voltages
File Tools View Simulation Help
Ready
Sample based T=20.000

Figure 2.2: Quanser Aero Proportional gain feedback step response.

3. Vary the proportional gain between 10 and 100. Run the QUARC® model with various proportional gains. What effects does increasing the proportional gain have on the system response?

4. Given the specification that the controlled system has a peak time of less than 1.1 seconds, is it practical to use purely proportional control? Why or why not?

2.2 Derivative Control

1. Because the angle data from the encoder is a discrete signal, some filtering is added to the derivative to smooth out discontinuities when the encoder count changes.

$$F(s) = \frac{20s}{s + 20}$$

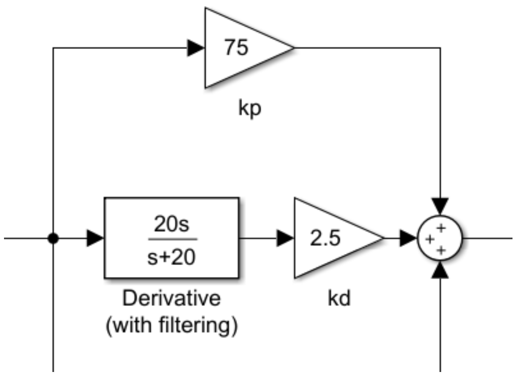

Figure 2.3: PD gains in feedback loop

2. Set the proportional gain to 75 and vary the derivative gain between 0 and 100. Run the QUARC® model with various derivative gains. What effects does increasing the derivative gain have on the system response? How does the system response differ when the derivative gain is very small (e.g. <1)
3. Change the "Input Select" switch to select the square wave input and update the run time to inf (for continuous run).
4. It is desired that the PD compensated system has a peak pitch which does not exceed ±0.35 radians when commanded with a 0.3 radian square wave. What is the minimum value of k_d which produces a system with acceptable overshoot?

2.3 Integral Control

1. The response should now approach the commanded value quickly and settle in a short time. However even with a large proportional gain, the final position of the Aero still differs from the command value. To combat this steady-state error, add an integrator with an integral gain Ki in parallel with the other two gains as shown in Figure 2.5.

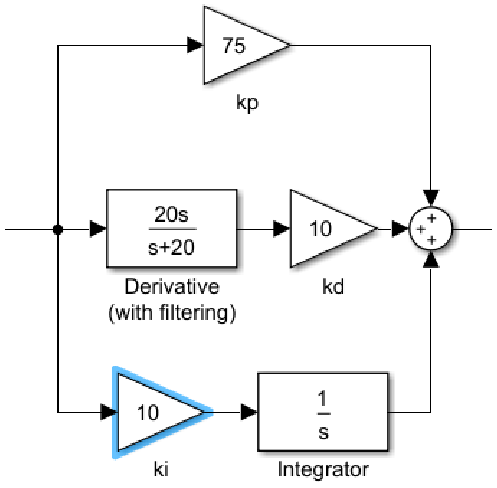

Figure 2.4: PID gains in feedback loop

2. Keeping the proportional and derivative gain at the values identified in previous step vary the integral gain between 0 and 100. Run the QUARC® model with various integral gains. What effects does increasing the integral gain have on the system response? What happens when the integral gain is increased too much?
3. Given the specification that the system settles within 3% of the command value. For the 0.3 radian square wave this means that the steady state value must be in the range from 0.291 to 0.309 radians. What integral gain results in a system which meets this requirement?

2.4 Response Tuning

1. If your response does not match the overshoot (<0.35 radians peak pitch) and peak time (<1.1 seconds) specifications from previous steps, try tuning your control gains until your response satisfies all three requirements (including steady-state error). Plot the resulting system response overlaid on the command waveform and linear representations of the peak overshoot limits. Record the final PID gains and comment on how you modified your controller to arrive at those results.

   Stop the QUARC® controller.

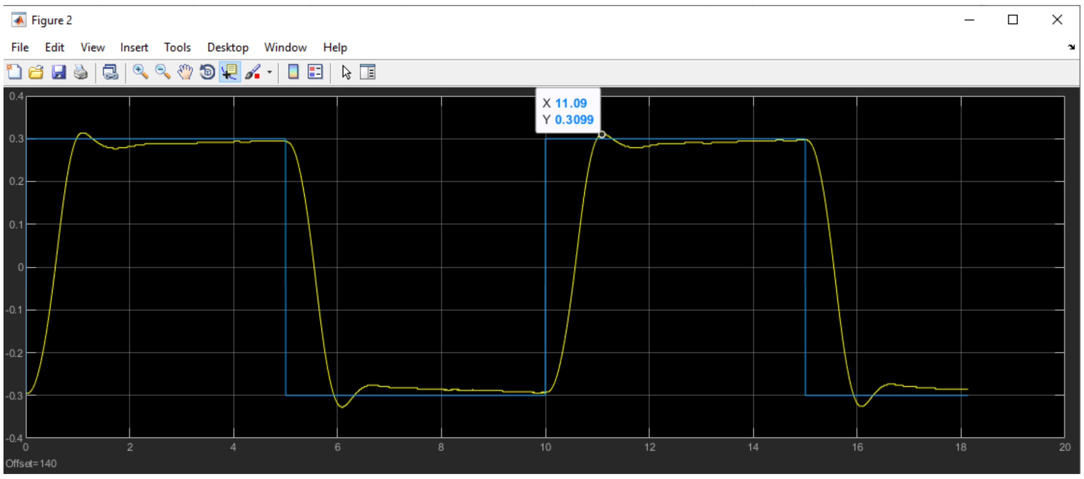

Figure 2.5: Tuned step response

Part 2 – Aero Simulation

Objectives

- Automation tasks.
- State machines
- Manual PID tuning in complex configurations (parallel, cascade)

Theoretical background

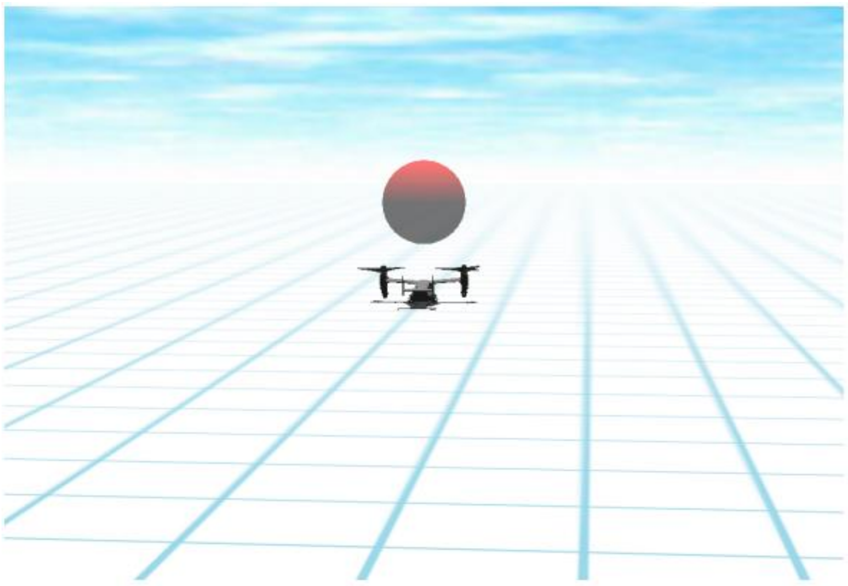

Figure 1.1: Aero vtol simulation

The scenario that is presented in this challenge is that of a 2DOF flight control task. The Quanser Aero is represented by the V-22 Osprey VTOL vehicle, where the roll of the Osprey relates to the pitch of the Quanser Aero, and the thrust of the vehicle's propellers is proportional to the speed of the propellers on the Quanser Aero. Your task is to control the altitude and horizontal position of the simulated vehicle via the pitch and thrust of the Quanser Aero. The simplest solution will be a nested PID control scheme however more complex solutions such as state-space modeling, gain scheduling, or state-machine control are also applicable and may result in a more successful controller. When the simulation begins, the Osprey begins on the ground at (0, 0, 0) and the first waypoint is highlighted with a red sphere with a diameter of 10 m. In order to proceed, the controller must maintain the position such that the error in both the X and Z position is less than ±5 m for $\sim 5s$. When the error is within acceptable limits the waypoint indicator will gradually change from red to green. After 5 seconds the next waypoint will be activated. The target waypoints may be changed in the main model to make the path easier or harder to traverse. Further challenge is added by increasing the maximum wind speed. This will cause a horizontal crosswind to be randomly applied every 5 seconds, perturbing the controller. The flight dynamics subsystem provides the required inputs to the system: The desired and actual position in both the X and Z planes and the measured pitch of the Quanser Aero.

Preparation questions

The control architecture for the Aero simulation is displayed in figure 1.2, each PID controller block has an input r which represent the desired command and an input y which is the current state. You may consider the internal block diagram for the PID blocks as a simplified PID model described in previous section.

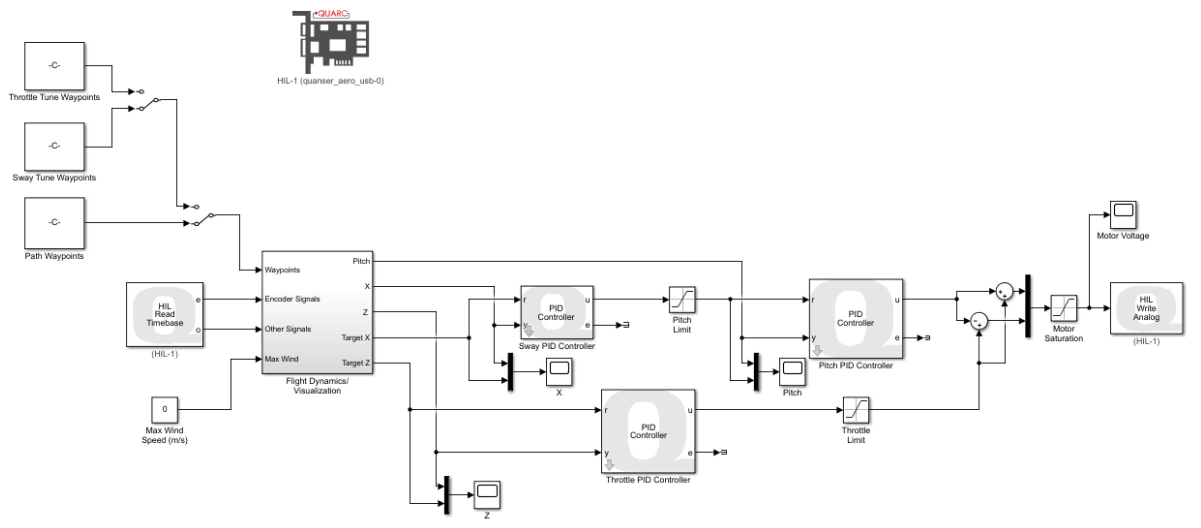

Figure 1.2: Aero model: q_aero_vtol_simulation.mdl

Q1 what are the state variables for the Aero simulation (based on the Flight dynamics block output).

Q2 Considering the above control architecture what are the inputs and outputs of each PID block.

Q3 Create a simplified block diagram of the above model. Stating the state variables, command variables and indicating which controller is in parallel and which in a cascade configuration.

Cascade configuration: [Designing Cascade Control System with PI Controllers - MATLAB & Simulink (mathworks.com)]

Bonus Questions:

Q4 Considering a full quadcopter configuration (3d) what are the state variable of its motion.

Q5 Using the above control configuration for a full quadcopter how many controllers are required to stabilize its motion (in 3d) and in what configuration. Create a simplified block diagram of the proposed model. What assumptions are required for such a design?

2. In-Lab Exercises

2.1 Throttle PID Controller

1. Open q_aero_vtol_simulation.mdl As shown in Figure 1.2 . Double clock the Pitch PID Controller block and update the PID values as found in section 1.
2. Verify that the manual Switches are configured for the Throttle Tune Waypoints. Compile and run the simulation

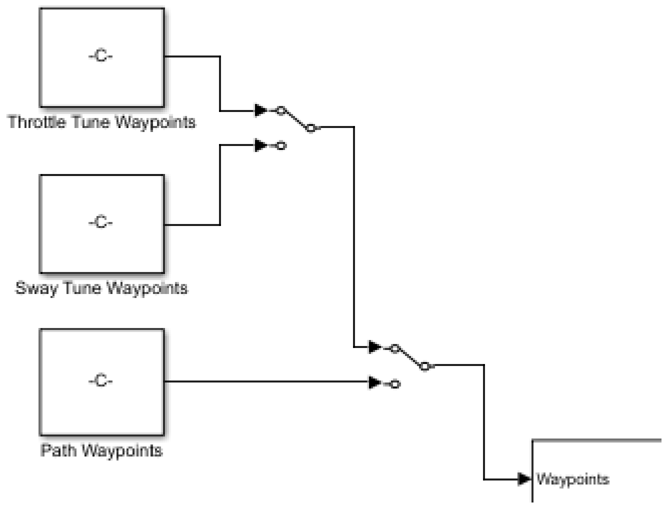

Figure 1.3: Aero simulation - Manual switch configuration.

3. Manually Tune the PID coefficients of the Throttle PID block until a proper response of the z axis position is achieved. (Settling time <10s, overshoot < 25%, stable motion without crashing). If the aero model loses stability in x axis restart the simulation and continue tuning the parameters.
   Proper range for the PID coefficients: Kp: [1 – 5] Ki: [0 – 1] Kd: [0 – 10].

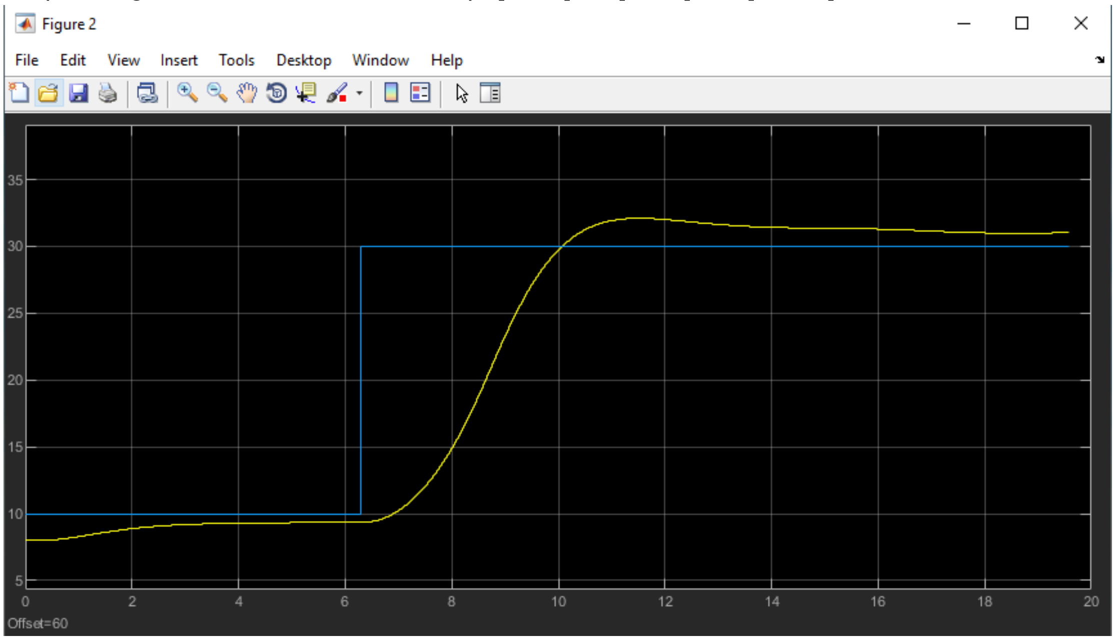

Figure 1.4: Aero simulation - z axis response.

4. Configure the manual switches for the Sway Tune Waypoints.
5. Manually Tune the PID coefficients of the Sway PID Controller until a proper response of the x axis position is achieved. (Settling time <10s, overshoot < 25%, stable motion without crashing). Consider using a PD solution and check if the integral gain is necessary. If the aero model loses stability restart the simulation and continue tuning the parameters.
   Proper range for the PID coefficients: Kp: [0.01 – 0.2] Ki: [0 – 0.01] Kd: [0 – 0.5].
   Considering the response time of the inner loop and the outer loop what can you say about the assumptions for a cascade controller configuration.

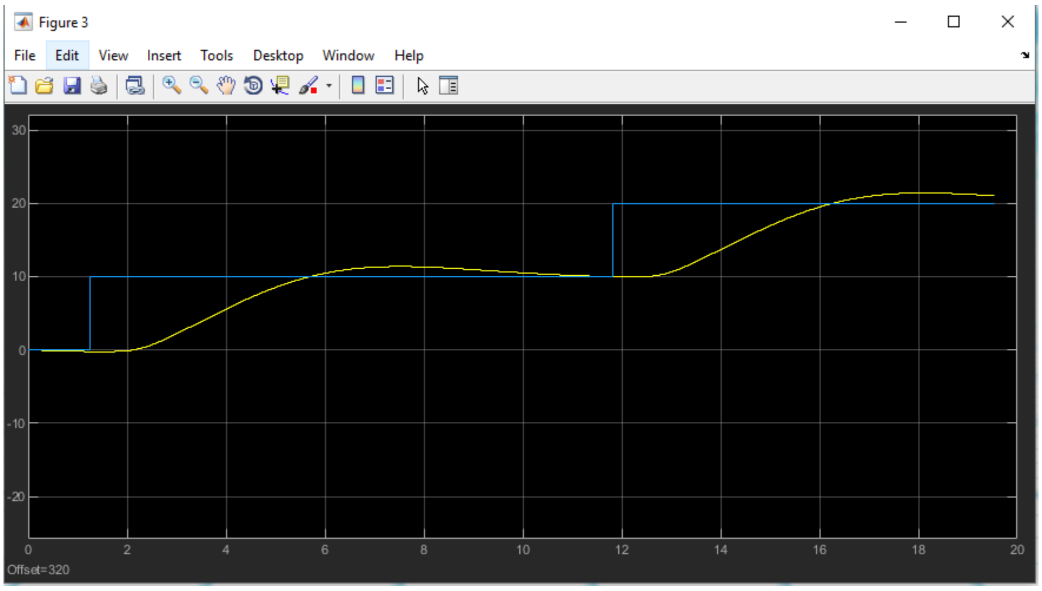

Figure 1.5: Aero simulation - x axis response.

6. Configure the manual switches for the Path Tune Waypoints. And check the simulation for stability for a mixed target position. If the system loses stability / crashes improve the controller's coefficients. Try and notice the influence the x axis command has on the z axis response and wise versa. What can you say about the assumptions for parallel control configuration?

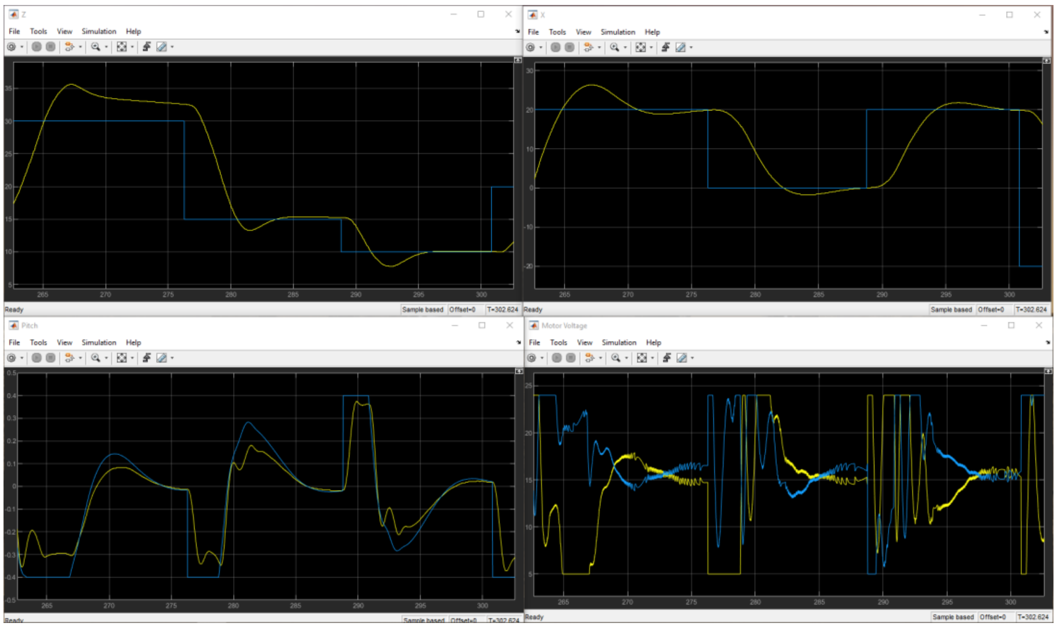
Figure 1.6: Aero simulation – Path waypoints response

Bonus tasks:

7. Add wind perturbance at 10m/s and 25 m/s how the system responses. Did it lose stability? Try to improve the controllers.
8. Change the Path Waypoints and check the stability for various configurations.

Stop the QUARC® controller.