Quanser Aero 2 Lab Procedure
Qualitative PID Control

Setup

1. Make sure the Aero 2 has been tested as instructed in the Quick Start Guide.
2. Launch MATLAB and browse to the working directory that includes the Simulink models for this lab.
3. Configure the Aero 2 in the 1 DOF pitch-only system:
    a. Unlock the pitch axis and lock the yaw axis.
    b. Both rotors are horizontal.
    c. Adjust weights on rotors so the Aero 2 body sits level.
4. Connect the USB cable to your PC/laptop.
5. Connect the power and turn the power switch ON. The Aero base LED should be red.

Proportional Control

1. Open the q_aero2_qualitative_PID_Control.mdl via MATLAB. This should look similar to Figure 1. This model implements basic commands and unity feedback for the Aero 2. The inputted signal (either step or square wave) goes through a gain of 0.4 and is added or removed from a *Trim Voltage* of 10V that is applied to the thrusters. In this lab, we will implement a PID controller to learn how to achieve a desired steady-state given requirements.

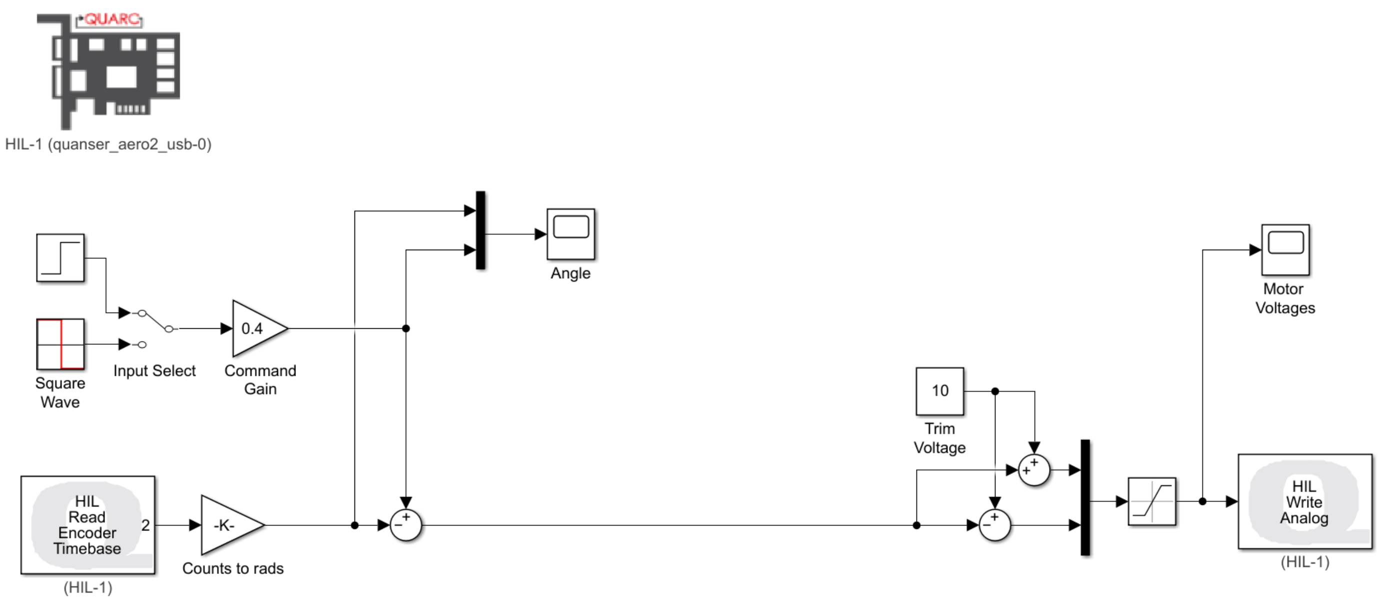
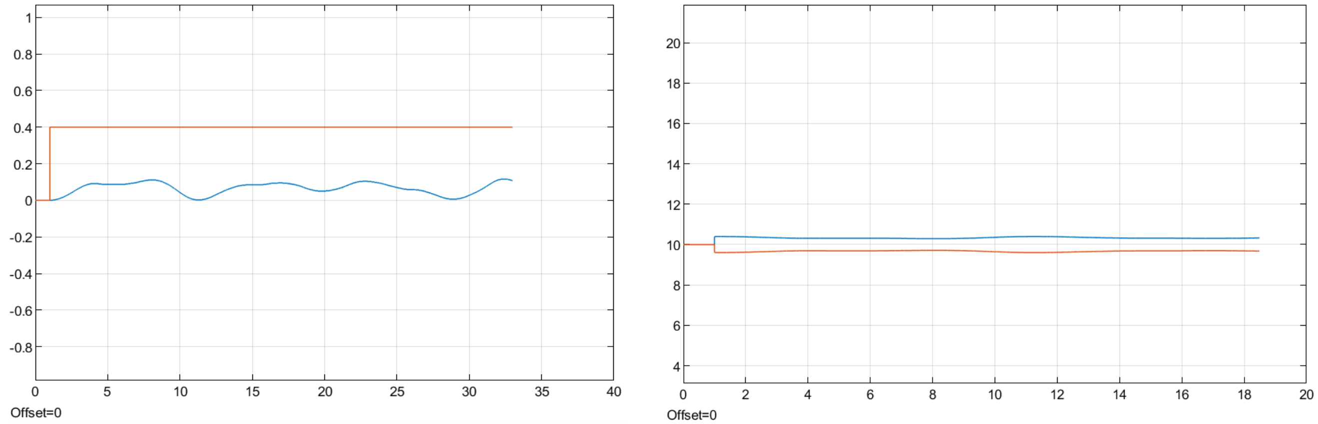


Figure 1: Unity-feedback block diagram

2. Build and run the QUARC controller. This unity feedback system's response doesn't show us much. It clearly does not do a good job of controlling the Aero 2 and finding a steady state.



(a) Angle (b) Motor Voltages

Figure 2: Aero 2 unity feedback step response

3. Add a *proportional* gain block k_p between the *error sum* blocks as shown in Figure 3.

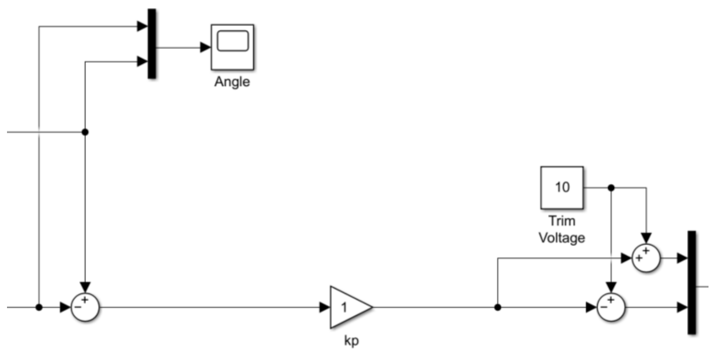


Figure 3: Proportional gain in the feedback loop

4. Vary the proportional gain between 1 and 15. Run the QUARC model with various proportional gains. Take notes on the effect of increasing the proportional gain on the system response. Capture your responses as well.

Derivative Control

5. Add a *transfer function* and derivative *gain* block k_d in parallel with the proportional *gain* as shown in Figure 4.

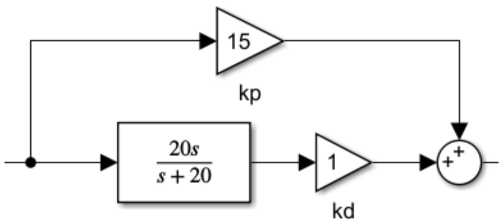


Figure 4: Proportional gain and filtered derivative gain feedback loop

6. The angle data from the encoder is a discrete signal, therefore we will have to add some filtering in order to smooth out discontinuities when the encoder count changes. Set the transfer function for the derivative to

$$F(s) = \frac{20s}{s + 20} \tag{1}$$

7. Set the proportional gain to 15 and vary the derivative gain between 0 and 10.

8. Change the *Input Select* switch to a square wave input.

9. 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. <0.25). Capture your responses.

10. It is desired that the PD compensated system has a peak pitch that does not exceed $\pm$ 0.45 radians when commanded with a 0.4 radian square wave. Note the minimum value of k_d which produces a system with acceptable overshoot.

Integral Control

11. 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 2 still differs from the command value by approximately 0.7 radians. To combat this steady-state error, keeping the proportional and derivative gains at the values identified in the previous step, add an integral gain k_i as well as an integrator in parallel with the other two gains as shown in Figure 5.

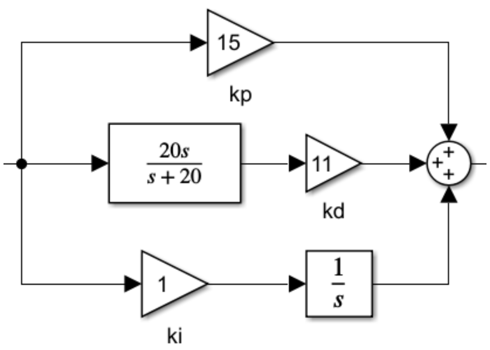


Figure 5: PID gains in a feedback loop

12. Vary the *integral gain* between 0 and 10. Take notes on the effect of increasing this gain on the system response, and capture your responses.

13. Given the specification that the system settles within 3% of the commanded value, this means that the steady-state value must be in the range of 0.388 to 0.412 radians. Note the integral gain that results in a system that meets this requirement.

Response Tuning

14. If your response does not match the overshoot (<0.45 radians peak pitch) and peak time (<2.2 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.

15. Stop and close the model. Power OFF the Aero 2.