Aero 2 Lab Procedure
Rotor Speed 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. Make sure the Quanser Aero is configured as the 1 DOF pitch-only system:
    a. Unlock the pitch axis and lock the yaw axis.
    b. Both rotors are horizontal.
    c. Mount weight on each rotor.
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.

Control Design

Design a PI speed control for the Aero 2 rotor according to the following peak time and percent overshoot specifications:

$$t_p = 0.15 \text{ s}$$
$$PO = 2.5\% \tag{1}$$

For best results. use the steady-state gain, K. and time constant, τ, model parameters that were obtained from conducing the *Rotor Modelling* lab. Otherwise use these default values.

$$K = 19.0 \text{ rad/s/V}$$
$$\tau = 0.165 \text{ s} \tag{2}$$

1. Evaluate the natural frequency and damping ratio needed to meet the desired peak time and percent overshoot given in Equation 1.
2. Show how the PI gain equations were found.
3. Calculate the PI gains needed for the system response to have the peak time and percent overshoot specifications given in Equation 1.

Rotor Speed Control Simulation

In this section, the PI rotor speed control is simulated using the rotor transfer function model found in the *Rotor Step Response Modelling* lab. The *s_aero2_rotor_pi* Simulink model shown in Figure 1 uses the Simulink *2 DOF PID Controller* block to implement the PI controller outlined in the Application Guide. The rotor speed reference and simulated response are displayed in the *Rotor (rad/s)* scope and the control input/motor rotor voltage is shown in the *Motor (V)* scope.

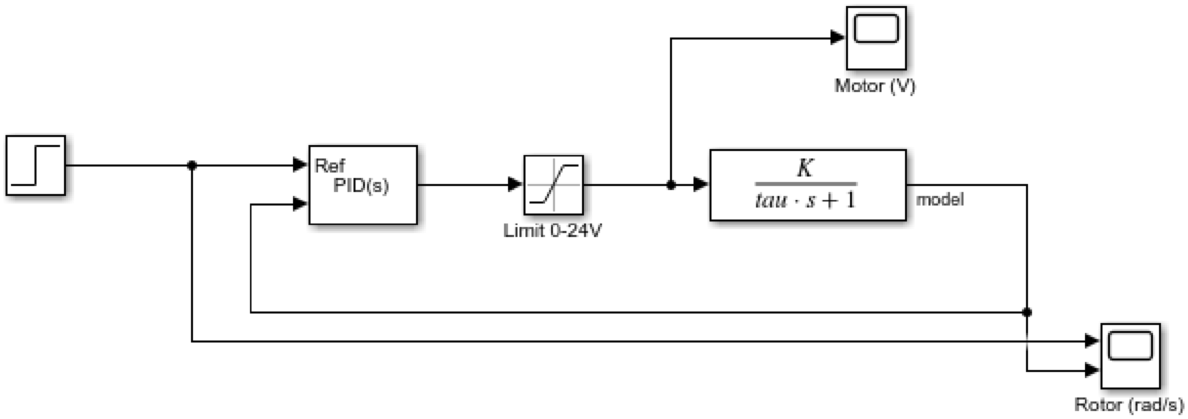

Figure 1 – Simulates the closed-loop rotor PI control speed response.

Follow these steps to simulate the rotor speed control response:

1. Open the *aero2_rotor_speed_control_student.mlx* MATLAB Live Script.

2. Enter the steady-state gain, K, and time constant, τ, model parameters found in the *Rotor Modelling* lab, or use the default ones given in Equation 2, in the Live Script as MATLAB variables *K* and *tau*.

3. Enter the PI gain equations, *kp* and *ki,* in the Live Script that calculate the gains based on the desired peak time, *tp*, and percent overshoot, *PO*, specifications. Currently, the PI gains are set to default values.

4. Open the *s_aero2_rotor_pi* Simulink model.

5. Run the *s_aero2_rotor_pi* Simulink model to simulate the closed-loop response of the PI control. The response using the default PI gains $k_p = 0.1$ V/(rad/s) and $k_i = 5$ V/rad is shown in Figure 2. Note that the response with the designed PI gains will be different.

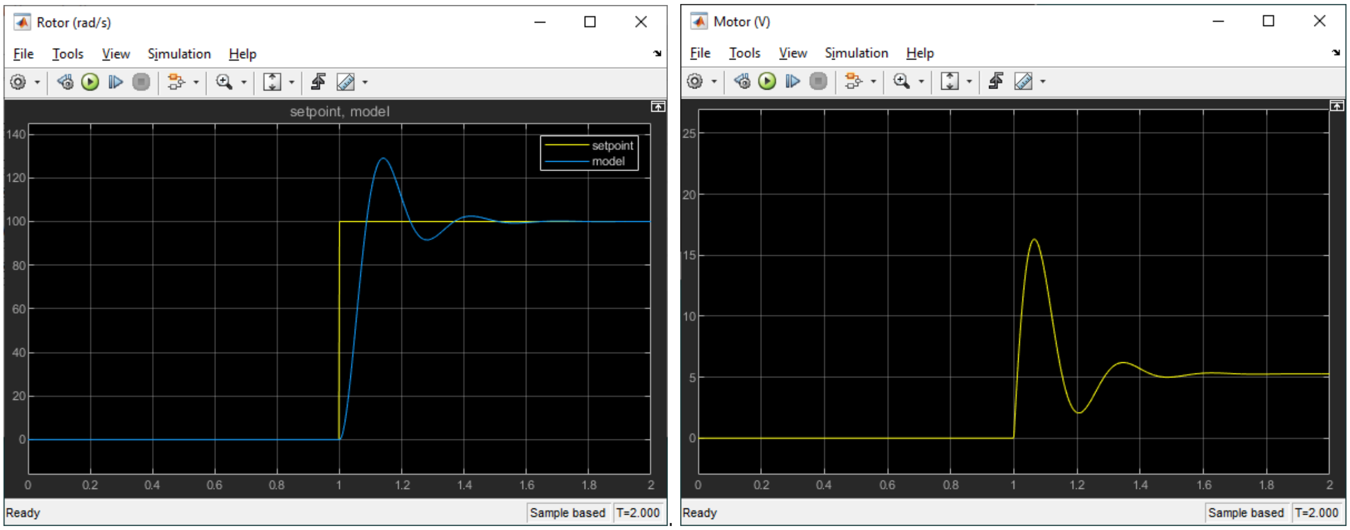

Figure 2 – Sample simulated rotor speed control response using k_p=0.1 and k_i=5.

6. Attach a MATLAB figure showing the response of the rotor speed setpoint and response as well as the input voltage. The motor voltage and rotor angular speed are saved in the MATLAB variables *RotorPISimVm* and *RotorPISimSpeed*. See the example code in the Live Script to create this plot.

7. Does the rotor speed control response satisfy the design specifications in Equation 1 of the Lab Procedure? Can this control be implemented on the Aero 2 rotor?

8. Close the Simulink model

Rotor Speed Control Hardware Implementation

The *q_aero2_rotor_pi* Simulink model shown in Figure 3 is used with QUARC to run the PI rotor speed control on the Aero 2 system. The QUARC *HIL Write Analog* applies the controller output voltage to the front rotor motor and *HIL Read Other Timebase* block measures the rotor propeller angular speed using the digital tachometer. The reference or commanded and measured rotor speed are displayed in the *Rotor Speed (rad/s)* scope and the motor voltage is shown in the *Motor (V)* scope.

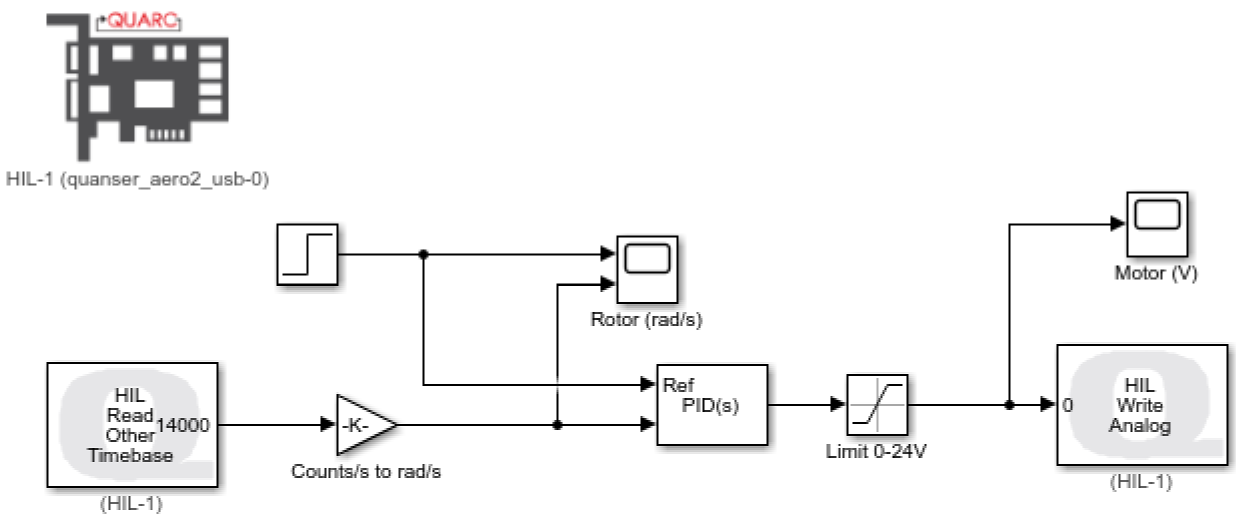

Figure 3 - Simulink/QUARC model used to run the PI control on the Aero 2.

Follow the instructions below to run the rotor speed control on the Aero 2:

1. Open the *aero2_rotor_speed_control_student.mlx* MATLAB Live Script.

2. Open the the *q_aero2_rotor_pi* Simulink model.

3. Make sure the designed PI gains you found and validated in the simulation, kp and ki, have been loaded in MATLAB.

4. Build and run the *q_aero2_rotor_pi* Simulink model in QUARC to run the PI controller on the Aero 2 system. To do this, click on the *Monitor & Tune* button in the Simulink menu under the *Hardware* tab. The model will stop automatically after 5 sec.

5. The response using the default PI gains $k_p = 0.1$ V/(rad/s) and $k_i = 5$ V/rad is shown in Figure 4. Note that the response with the designed PI gains will be different.

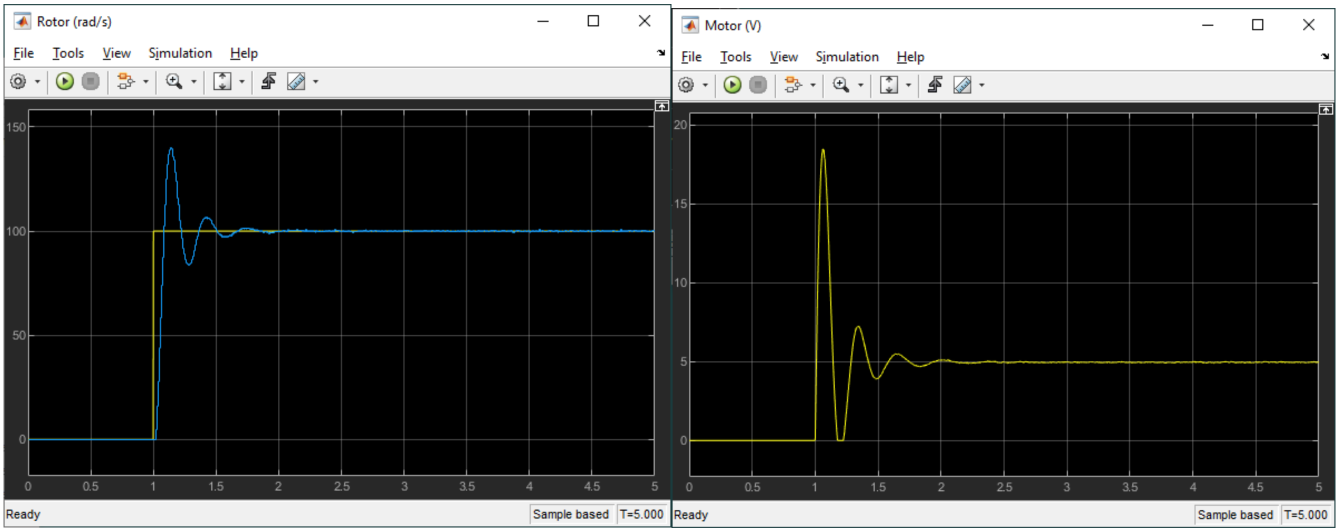

Figure 4 – Sample rotor PI speed control response on Aero 2 using k_p=0.1 and k_i=5

6. Plot the rotor speed response and motor voltage in a MATLAB figure. The motor voltage and rotor angular speed are saved in the MATLAB variables *RotorPIImplVm* and *RotorPIImplSpeed*. See the example code in the Live Script. Attach the response.

7. Is the response on the hardware different than the simulated response? If so, explain why there is a discrepancy.

8. Close the Simulink model

9. Turn off the power on the Aero 2.