

1 Presentation

The Quanser Aero Experiment experiment can be configured as a conventional dual-rotor helicopter, as shown in Figure 1.1. The front rotor that is horizontal to the ground predominantly affects the motion about the pitch axis while the back or tail rotor mainly affects the motion about the yaw axis (about the shaft).



Figure 1.1: Quanser Aero Experiment

The tail rotor in helicopters is also known as the anti-torque rotor because it is used to reduce the torque that the main rotor generates about the yaw. Without this, the helicopter would be difficult to stabilize about the yaw axis. Because the rotors on the Quanser Aero Experiment are the same size and equidistant from each other, the tail rotor also generates a torque about the pitch axis. As a result, both the front and back/tail rotors generate a torque on each other.

Note: The dynamic coupling between the pitch and yaw axes is **only seen when using the low-efficiency rotors**. It is not witnessed with the high-efficiency rotors provided. See the Quanser Aero User Manual for more information about the two different types of rotors supplied.

Topics covered:

- Derive linear equations of motion for the 2 DOF Helicopter configuration.
- Find the transfer function and state-space representation models.
- Identify the viscous damping coefficients about the pitch and yaw axes experimentally.
- Estimate the various torque thrust constants experimentally.
- Design a de-coupled PD control to control the pitch and yaw axes.
- Simulate the closed-loop system and implement on the Quanser Aero Experiment .
- Design state-feedback control using LQR optimization.
- Simulate the closed-loop system and implement on the Quanser Aero Experiment .
- Design and implement an Kalman-based LQG observer and controller for the Aero 1 DOF attitude configuration.

2 Modeling

2.1 Background

2.1.1 Equations of Motion

The free-body diagram of the Quanser Aero Experiment is illustrated in Figure 2.1.

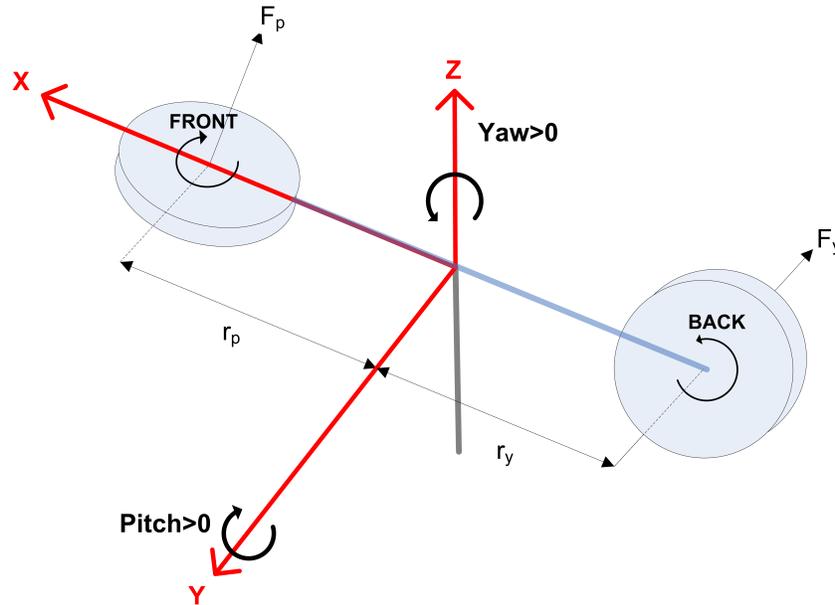


Figure 2.1: Simple free-body diagram of Quanser Aero Experiment

The following conventions are used for the modeling:

1. The helicopter is horizontal and parallel with the ground when the pitch angle is zero, i.e. $\theta = 0$.
2. The pitch angle increases positively, $\theta(t) > 0$, when the front rotor is moved upwards and the body rotates **counter-clockwise (CCW)** about the Y axis.
3. The yaw angle increases positively, $\psi(t) > 0$, when the body rotates **counter-clockwise (CCW)** about the Z axis.
4. Pitch increases, $\dot{\theta} > 0$, when the front rotor voltage is positive $V_p > 0$.
5. Yaw increases, $\dot{\psi} > 0$, when the back (or tail) rotor voltage is positive, $V_y > 0$.

When voltage is applied to the pitch motor, V_p , the speed of rotation results in a force, F_p , that acts normal to the body at a distance r_p from the pitch axis. The rotation of the propeller generates a torque about the pitch rotor motor shaft which is in turn seen about the yaw axis. Thus rotating the pitch propeller does not only cause motion about the pitch axis but also about the yaw axis. As described earlier, that's why conventional helicopters include a tail, or anti-torque, rotor to compensate for the torque generated about the yaw axis by the large, main rotor.

Similarly, the yaw motor causes a force F_y that acts on the body at a distance r_y from the yaw axis as well as a torque about the pitch axis.

We can develop a simple linear model that takes this coupling into account, and represents the motions of the Quanser Aero about the horizontal, i.e. when the body is parallel with the ground. The equations of motion are:

$$J_p \ddot{\theta} + D_p \dot{\theta} + K_{sp} \theta = \tau_p, \quad (2.1)$$

$$J_y \ddot{\psi} + D_y \dot{\psi} = \tau_y \quad (2.2)$$

where the torques acting on the pitch and yaw axes are

$$\begin{aligned}\tau_p &= K_{pp}V_p + K_{py}V_y, \text{ and} \\ \tau_y &= K_{yp}V_p + K_{yy}V_y.\end{aligned}$$

The parameters used in the EOMs above are:

- J_p is the total moment of inertia about the pitch axis,
- D_p is the damping about the pitch axis,
- K_{sp} is the stiffness about the pitch axis,
- K_{pp} is torque thrust gain from the pitch rotor,
- K_{py} is the cross-torque thrust gain acting on the pitch from the yaw rotor,
- V_p is the voltage applied to the pitch rotor, and
- V_y is the voltage applied to the yaw rotor motor.

Similarly, the total moment of inertia and damping about the yaw axis is J_y and D_y , respectively, K_{yy} is the torque-thrust gain from the yaw rotor, and K_{yp} is the cross torque-thrust gain acting on the yaw axis from the pitch rotor. Some of these model parameters are given in the Quanser Aero Experiment User Manual.

Remark the coupling between the pitch and yaw rotors. The total torque acting on each axis is generated from both rotors. Thus the total torque acting on the pitch equals $\tau_p = K_{pp}V_p + K_{py}V_y$ and the total torque on the yaw is $\tau_y = K_{yp}V_p + K_{yy}V_y$.

The total moment of inertia acting about the pitch and yaw axes are

$$\begin{aligned}J_p &= J_{body} + 2J_{prop} \\ J_y &= J_{body} + 2J_{prop} + J_{yoke}\end{aligned}$$

Expressing the rotor as a single-point mass, the inertia acting about the pitch or yaw axis from a single rotor is $J_{prop} = m_{prop}r_{prop}^2$. Modeling the helicopter body as a cylinder rotating about its center, the inertia is $J_{body} = m_{body}L_{body}^2/12$. Finally the forked yoke that rotates about the yaw axis can be approximated as cylinder rotating about its center as well and expressed as $J_{yoke} = m_{yoke}r_{fork}^2/2$. Evaluating the moment of inertia using the parameters listed in the Quanser Aero Experiment User Manual gives:

$$\begin{aligned}J_p &= 0.0219 \text{ kg}\cdot\text{m}^2 \\ J_y &= 0.0220 \text{ kg}\cdot\text{m}^2\end{aligned}$$

which are closed to the moment of inertia values listed in the User Manual that were derived from the CAD model.

2.1.2 Transfer Function Model

Taking the Laplace transform of the equations of motion given in Equation 2.2

$$J_p \left(\Theta(s)s^2 - \theta(0^-)s - \dot{\theta}(0^-) \right) + D_p \left(\Theta(s)s - \theta(0^-) \right) + K_{sp}\Theta(s) = K_{pp}V_p(s) + K_{py}V_y(s)$$

and

$$J_y \left(\Psi(s)s^2 - \psi(0^-)s - \dot{\psi}(0^-) \right) + D_y \left(\Psi(s)s - \psi(0^-) \right) = K_{yp}V_p(s) + K_{yy}V_y(s)$$

Because this is a MIMO system with two outputs and two inputs, the system is represented as a set of four transfer functions: $\Theta(s)/V_p(s)$ and $\Theta(s)/V_y(s)$ for pitch and $\Psi(s)/V_p(s)$ and $\Psi(s)/V_y(s)$ for yaw. Using this and assuming all

the initial conditions are zero, i.e. $\theta(0^-) = 0$, $\dot{\theta}(0^-) = 0$, $\psi(0^-) = 0$, and $\dot{\psi}(0^-) = 0$, we obtain the following transfer functions describing the system motions relative to the different inputs:

$$\frac{\Theta(s)}{V_p(s)} = \frac{K_{pp}}{J_p s^2 + D_p s + K_{sp}} \quad \text{and} \quad \frac{\Psi(s)}{V_p(s)} = \frac{K_{yp}}{J_y s^2 + D_y s} \quad (2.3)$$

and

$$\frac{\Theta(s)}{V_y(s)} = \frac{K_{py}}{J_p s^2 + D_p s + K_{sp}} \quad \text{and} \quad \frac{\Psi(s)}{V_y(s)} = \frac{K_{yy}}{J_y s^2 + D_y s}. \quad (2.4)$$

2.1.3 Linear State-Space Representation

Given the linear state-space equations: $\dot{x} = Ax + Bu$ and $y = Cx + Du$, we define the state for the Quanser Aero Experiment as

$$x^T = [\theta(t), \psi(t), \dot{\theta}(t), \dot{\psi}(t)], \quad (2.5)$$

the output vector as

$$y^T = [\theta(t), \psi(t)]$$

and the control variables as

$$u^T = [V_p(t) \ V_y(t)]$$

where θ and ψ are the pitch and yaw angles, respectively, and V_p and V_y are the motor voltages applied to the pitch and yaw rotors (i.e. the main and tail rotors). Using the equations of motion in Equation 2.2, the state-space matrices are

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -K_{sp}/J_p & 0 & -D_p/J_p & 0 \\ 0 & 0 & 0 & -D_y/J_y \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ K_{pp}/J_p & K_{py}/J_p \\ K_{yp}/J_y & K_{yy}/J_y \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \text{and} \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

2.1.4 First-Order Response

The step response of a first-order transfer function

$$Y(s) = \frac{K}{\tau s + 1} U(s) \quad (2.6)$$

where K is the DC or steady-state gain and τ is the time constant is illustrated in Figure 2.2. This is for a system with $K = 1$ and $\tau = 0.05$.

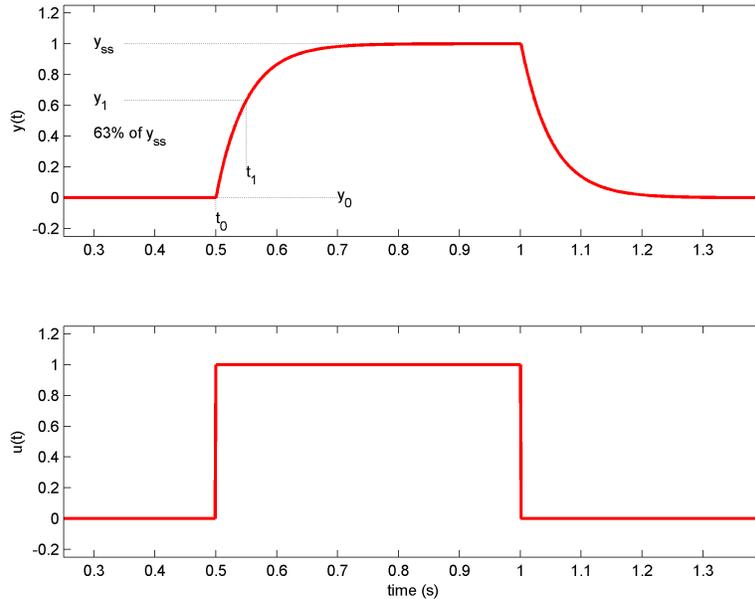


Figure 2.2: First-order step response

To obtain the time constant from the response, find the time it takes to reach $1 - e^{-1}$ or 63% of its final steady-state value:

$$y(t_1) = y_1 = (1 - e^{-1})(y_{ss} - y_0)$$

The time constant is $\tau = t_1 - t_0$, where t_0 is the start time of the step and t_1 is the time it takes to reach 63% of the final value, as illustrated in Figure 2.2.

To find the time constant from first-order response from an impulse (or a short step) as depicted in in Figure 2.3, find the time it takes for the response to reach e^{-1} or 37% of its final steady-state value.

In this case we need to find

$$y(t_1) = y_1 = e^{-1}(y_0 - y_{ss})$$

and the time constant is $\tau = t_1 - t_0$.

2.1.5 Second Order Response

The free-oscillatory equation of motion of a second-order system described by

$$J\ddot{\alpha} + D\dot{\alpha} + K\alpha = 0 \quad (2.7)$$

is shown in Figure 2.4. Assuming the initial conditions $\alpha(0^-) = \alpha_0$ and $\dot{\alpha}(0^-) = 0$, the Laplace transform of Equation 2.7 is

$$\alpha(s) = \frac{\alpha_0/J}{s^2 + D/Js + K/J} \quad (2.8)$$

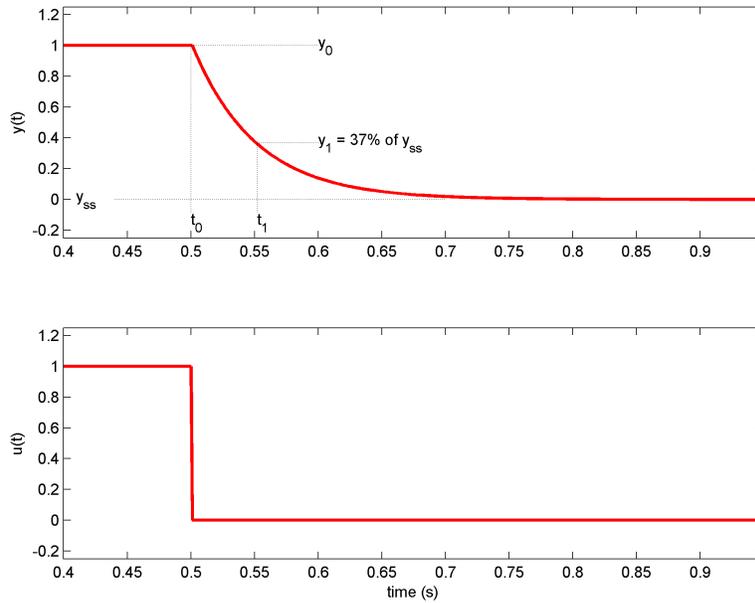


Figure 2.3: First-order impulse decaying response

The prototype second-order equation is defined

$$s^2 + 2\zeta\omega_n s + \omega_n^2,$$

where ζ is the damping ratio and ω_n is the natural frequency. Equating the characteristic equation in Equation 2.8 to this gives

$$\omega_n^2 = \frac{K}{J}$$

and

$$2\zeta\omega_n = \frac{D}{J}$$

Finding the Natural Frequency

The period of the oscillations in a system response can be found using the equation

$$T_{osc} = \frac{t_n - t_1}{n - 1} \quad (2.9)$$

where t_n is the time of the n^{th} oscillation, t_1 is the time of the first peak, and n is the number of oscillations considered. From this, the damped natural frequency (in radians per second) is

$$\omega_d = \frac{2\pi}{T_{osc}} \quad (2.10)$$

and the undamped natural frequency is

$$\omega_n = \frac{\omega_d}{\sqrt{1 - \zeta^2}}. \quad (2.11)$$

Finding the Damping Ratio

The damping ratio of a second-order system can be found from its response. For a typical second-order underdamped system, the subsidence ratio (i.e. decrement ratio) is defined as

$$\delta = \frac{1}{n} \ln \frac{O_1}{O_n} \quad (2.12)$$

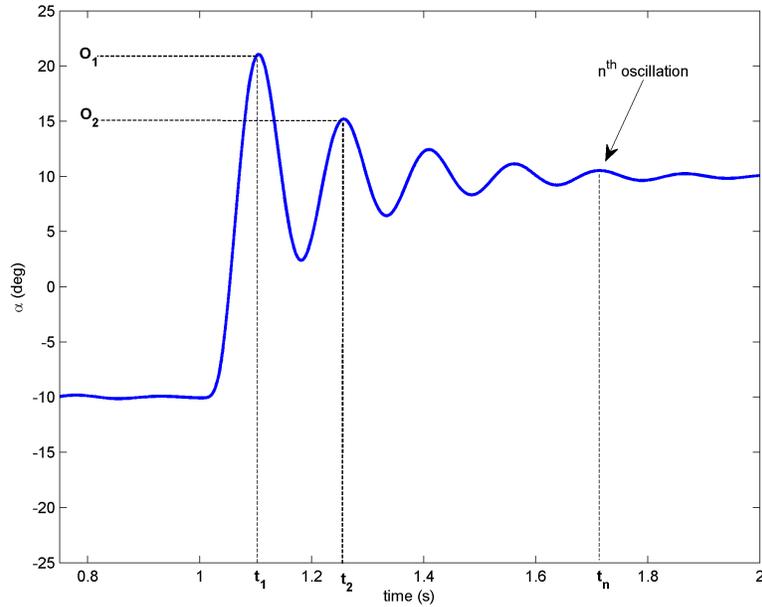


Figure 2.4: Free Oscillation Response

where O_1 is the peak of the first oscillation and O_n is the peak of the n^{th} oscillation. Note that $O_1 > O_n$, as this is a decaying response. The damping ratio can then be found using

$$\zeta = \frac{1}{\sqrt{1 + \frac{2\pi^2}{\delta^2}}} \quad (2.13)$$

2.1.6 Estimating the Viscous Damping Coefficients

The viscous damping coefficients acting about the pitch and yaw axes, D_p and D_y in Equation 2.2, can be found from the free-oscillation response. The free-oscillation response about the pitch and about the yaw are different, however.

Pitch Axis

By locking the yaw axis (using the Allen key supplied), this allows us to focus on the 1 DOF pitch-only system. Apply a short step voltage to mimic an impulse and get the free-oscillation response of the pitch. Remark that the impulse response is second-order free-oscillation response. The resulting 1 DOF pitch-only equations of motion is

$$J_p \ddot{\theta}(t) + D_p \dot{\theta}(t) + K_{sp} \theta(t) = 0. \quad (2.14)$$

Taking its Laplace transform gives

$$J_p(\Theta(s)s^2 - \theta(0^-)s - \dot{\theta}(0^-)) + D_p(\Theta(s) - \theta(0^-)) + K_{sp}\Theta_p(s) = 0.$$

Assuming the initial velocity is zero, $\dot{\theta}(0^-) = 0$, and solving for position we get

$$\Theta(s) = \frac{J_p}{J_p s^2 + D_p s + K_{sp}} \theta(0^-) = \frac{J_p/D_p}{s^2 + D_p/J_p + K_{sp}/J_p} \theta(0^-).$$

The pitch free-oscillation transfer function matches the prototype second-order transfer function in Equation 2.8. Based on the measured damping ratio and natural frequency of the response, the friction (or stiffness) of the system is

$$K_{sp} = J_p \omega_n^2 \quad (2.15)$$

and the viscous damping is

$$D_p = 2\zeta\omega_n J_p. \quad (2.16)$$

Yaw Axis

The 1 DOF yaw-only equations of motion is

$$J_y \ddot{\psi} + D_y \dot{\psi} = 0.$$

In terms of angular rate, the equation becomes

$$J_y \dot{\omega}_y(t) + D_y \omega_y(t) = 0 \quad (2.17)$$

where $\omega_y(t) = \dot{\psi}(t)$. Taking its Laplace transform

$$J_p(\Omega_y(s)s - \omega_y(0^-)) + D_y \Omega_y(s) = 0$$

and solving for the speed we get

$$\Omega_y(s) = \frac{J_y}{J_y s + D_y} \omega_y(0^-) = \frac{J_y/D_y}{J_y/D_y s + 1} \omega_y(0^-).$$

The yaw free-oscillation transfer function matches the prototype first-order transfer function in Equation 2.6. Based on the measured time constant of the response, its damping can be found with

$$D_y = \frac{J_y}{\tau}. \quad (2.18)$$

2.1.7 Estimating the Thrust Parameters

By locking the yaw axis (using the Allen key supplied), this allows us to focus on the 1 DOF pitch-only system, i.e. eliminating any motions introduced in the yaw axis when applying a voltage to the pitch rotor. The equations of motion for the 1 DOF actuated system is

$$J_p \ddot{\theta} + D_p \dot{\theta} + K_{sp} \theta = K_{pp} V_p.$$

In terms of angular rate and not excluding the stiffness estimate for this analysis, we get the pitch equation

$$J_p \dot{\omega}_p + D_p \omega_p = K_{pp} V_p \quad (2.19)$$

where $\omega_p = \dot{\theta}$ is the angular rate of the pitch axis. Solving for the thrust gain we get

$$K_{pp} = \frac{J_p \dot{\omega}_p + D_p \omega_p}{V_p} \quad (2.20)$$

Remark that this is the thrust torque gain parameter. To force thrust gain would be K_{pp}/r_p , where r_p is the distance between the helicopter pivot and the center of the pitch rotor.

Similarly, to find the thrust gain acting on the yaw axis only system, lock the pitch axis (using the Allen Key), and apply a voltage to the tail rotor. This system is represented by

$$J_y \ddot{\psi} + D_y \dot{\psi} = K_{yy} V_y$$

or

$$J_y \dot{\omega}_y + D_y \omega_y = K_{yy} V_y \quad (2.21)$$

where $\omega_y = \dot{\psi}$ is the angular rate of the yaw axis. The yaw torque thrust gain is

$$K_{yy} = \frac{J_y \dot{\omega}_y + D_y \omega_y}{V_y}. \quad (2.22)$$

The cross-torque thrust parameters, K_{py} and K_{yp} in Equation 2.2, represent the coupling between the axes. To find the cross-torque acting on the pitch axis from a torque applied to the tail rotor, unlock both pitch and yaw axes such that it is free to move in 2 DOF, apply a voltage to the tail rotor, and examine the response of the pitch. The equations representing these dynamic, when $V_p = 0$, are

$$J_p \ddot{\theta} + D_p \dot{\theta} = K_{py} V_y.$$

Putting this in terms of angular rate, $\omega_p = \dot{\theta}$, and solving for the gain we get

$$K_{py} = \frac{J_p \dot{\omega}_p + D_p \omega_p}{V_y} \quad (2.23)$$

Similarly, to identify the cross-torque gain parameter that is generated about the yaw axis from a pitch torque (i.e. voltage applied to the front rotor), we have the equation

$$J_y \ddot{\psi} + D_y \dot{\psi} = K_{yp} V_p$$

and the gain can be found using

$$K_{yp} = \frac{J_y \dot{\omega}_y + D_y \omega_y}{V_p}. \quad (2.24)$$

2.2 In-Lab Experiments

2.2.1 Estimating Viscous Damping Coefficients

Example lab results of finding the pitch and yaw damping coefficients are given in this section. Note that results may vary between different Quanser Aero Experiment systems.

Finding Pitch Damping:

1. Lock the yaw axis to enable motions about the pitch axis only.
2. Run the *q_aero_free_osc_response_pitch* QUARC controller to apply a -20 V impulse to first bring the helicopter front rotor down and then examine its free-response as it oscillates upwards. The Simulink diagram used to generate the QUARC controller is shown in Figure 2.5.
3. See the sample response obtained in Figure 2.6

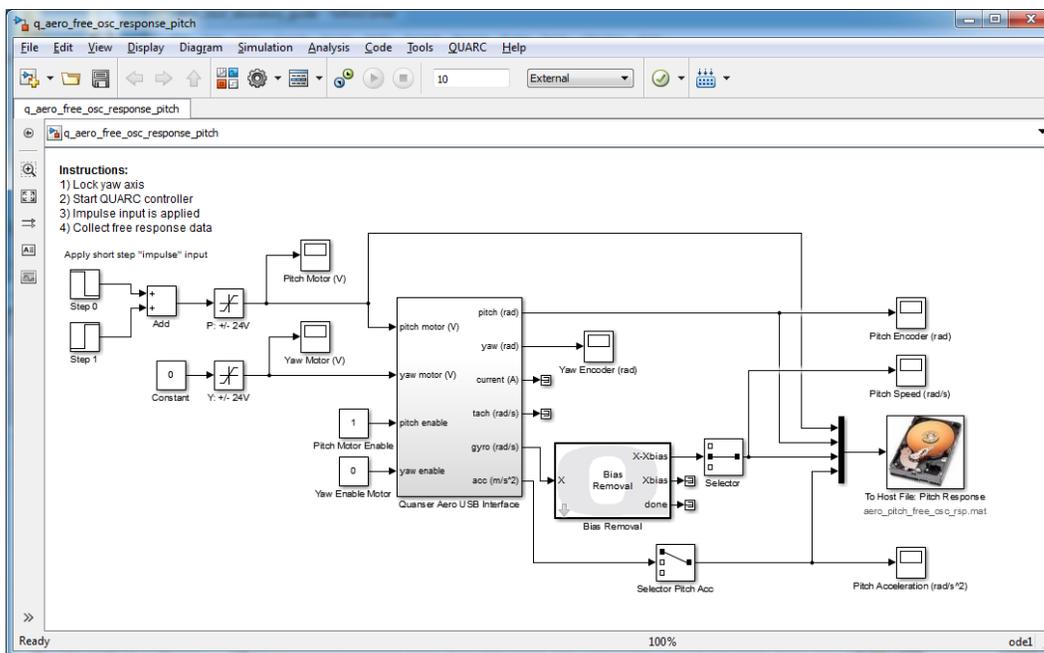


Figure 2.5: Model used to acquire free-oscillation response about pitch

The natural frequency and damping ratio can be measured from the response in Figure 2.6 using Equation 2.11 and Equation 2.13. The peak time and overshoot for the first and fourth oscillations, i.e. $n = 1$ and $n = 4$, are:

$$\begin{aligned} O_1 &= 0.644 \text{ deg} \\ t_1 &= 5.23 \text{ s} \\ O_n &= 0.187 \text{ deg} \\ t_n &= 19.75 \text{ s}. \end{aligned}$$

Using Equation 2.12 with the measured oscillation peaks we obtain the following the subsidence ratio

$$\delta = \frac{1}{3} \ln \frac{0.644}{0.187} = 0.3091$$

and the results damping ratio, using Equation 2.13, is

$$\zeta = 1/\sqrt{(1 + 2\pi^2/0.309)} = 0.124$$

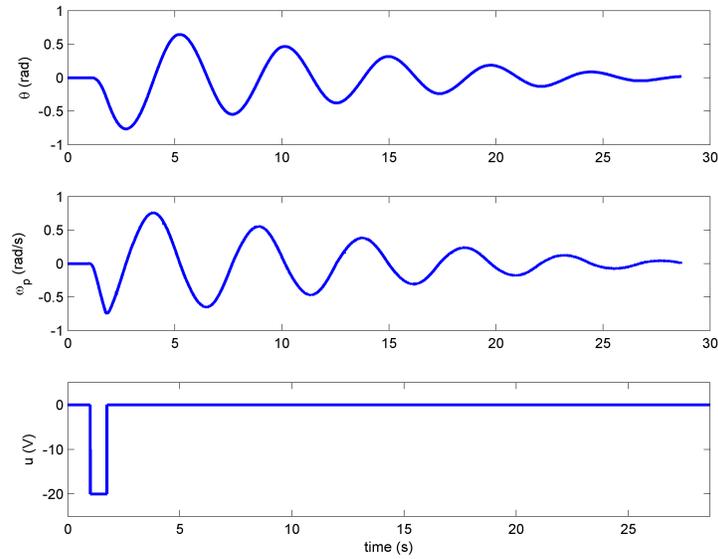


Figure 2.6: Free-oscillation response about pitch

The oscillation period and damped natural frequency are

$$T_{osc} = \frac{19.75 - 5.23}{4 - 1} = 4.84 \text{ s.}$$

and

$$\omega_d = \frac{2\pi}{T_{ocs}} = 1.30 \text{ rad/s}$$

which results in the natural frequency

$$\omega_n = 1.30 / \sqrt{(1 - 0.124^2)} = 1.31 \text{ rad/s.}$$

Using equations Equation 2.16 and Equation 2.15, we find that the stiffness and viscous damping coefficient about the pitch axis are

$$K_{sp} = J_p \omega_n^2 = 0.0219(1.31^2) = 0.0375 \text{ (N-m/rad)}$$

and

$$D_p = 2\zeta\omega_n J_p = 2(0.124)(1.31)(0.0219) = 0.00711 \text{ V-s/rad.}$$

Finding Yaw Damping:

1. Lock the pitch axis
2. Run the `q_aero_free_osc_response_yaw` QUARC controller to apply an impulse to the tail rotor and measure the corresponding yaw response.
3. See the sample response obtained in Figure 2.7 after .

To find the time constant, examine the decaying response that starts at an initial maximum speed of $\omega_0 = 0.56 \text{ rad/s}$ at time $t_0 = 2.06 \text{ s}$ and eventually settles down to 0 rad/s . The response is similar to the decaying free-oscillation response discussed in Section 2.1.4. 37% of the final value between ω_0 and 0 rad/s is

$$\omega_1 = (1 - 0.63)\omega_0 = 0.207 \text{ rad/s.}$$

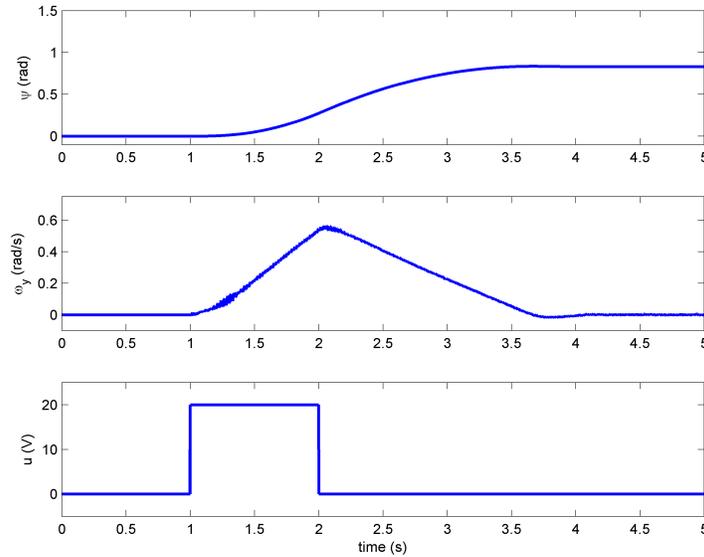


Figure 2.7: Free-oscillation response about pitch

which occurs at $t_1 = 3.06$ s. The time constant is therefore

$$\tau = t_1 - t_0 = 1.00 \text{ s}$$

and using Equation 2.16 the damping is

$$D_y = \frac{J_t}{\tau} = 0.0220 \text{ V-s/rad.}$$

2.2.2 Estimating the Thrust Gain Parameters

In this section, the torque thrust gains given in Equation 2.2 are estimated. Each gain is found individually by locking either the pitch yaw axis. Note that results may vary between different Quanser Aero Experiment systems.

Pitch thrust gain identification:

1. Lock the yaw axis.
2. Run the `q_aero_step_response_pitch` QUARC controller to apply a step voltage 20V to the main rotor, shown in Figure 2.8
3. Sample response is given in Figure 2.9.

Using Equation 2.20 and the step response in Figure 2.9 we find that the main thrust gain acting on the pitch axis is

$$K_{pp} = \frac{J_p \frac{\Delta\Omega_p}{\Delta t} + D_p \Delta\omega_p}{V_p} = 0.0011 \text{ N-m/V}$$

where $\Delta\omega_p = 0.926 - 0$ rad/s is the change in angular rate of the pitch axis, which occurs between $\Delta t = 2.30 - 1$ s, giving an acceleration of $0.926/1.30 = 0.712$ rad/s².

Yaw thrust gain identification:

1. Lock the pitch axis.

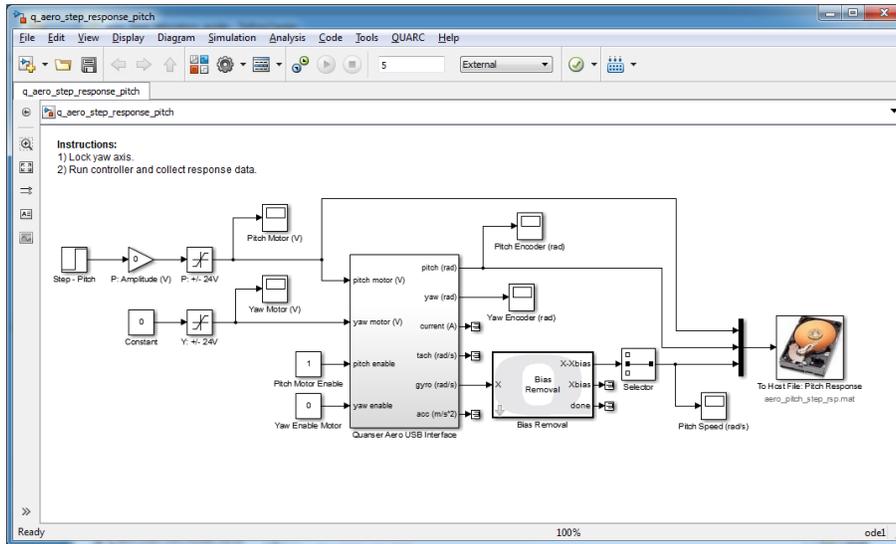


Figure 2.8: Pitch Step Response

2. Run the *q_aero_step_response_yaw* QUARC controller to apply a step voltage 20V to the tail rotor.
3. Sample response is given in Figure 2.10.

Using Equation 2.22, we find that the yaw thrust gain is

$$K_{yy} = \frac{J_y \frac{\Delta\Omega_y}{\Delta t} + D_y \Delta\omega_y}{V_y} = 0.0022 \text{ N-m/V}$$

where $\Delta\omega = 1.57 - 0 \text{ rad/s}$ is the change in angular rate of the yaw axis between $\Delta t = 5 - 1 \text{ s}$.

2.2.3 Estimating the Cross-Thrust Gain Parameters

In this section the cross-torque gain parameters are estimated. Each axis is allowed to move freely and the response of each axis is examined when applying a torque to the other axis.

Identifying the cross-torque gain parameter K_{py} :

1. Unlock both the pitch and yaw axes to enable the full 2 DOF motion.
2. Run the *q_aero_step_response_pitch_from_yaw* QUARC controller to apply a step voltage 20V to the tail rotor and observe a response in the pitch.
3. Sample response is given in Figure 2.11.

Observe how a positive torque about the yaw axis yields (i.e. positive voltage to yaw/tail rotor) results in a positive torque about the pitch axis. Using Equation 2.23, the cross-thrust gain for the pitch from a yaw voltage is

$$K_{py} = \frac{J_p \frac{\Delta\Omega_p}{\Delta t} + D_p \Delta\omega_p}{V_y} = 0.0021 \text{ N-m/V}$$

where $\Delta\omega_p = 0.86 - 0 \text{ rad/s}$ is the change in angular rate of the pitch axis that occurs in 1 second, $\Delta t = 1.83 - 1 = 0.83 \text{ s}$.

Identifying the cross-torque gain parameter K_{yp} :

1. Unlock both the pitch and yaw axes to enable the full 2 DOF motion.

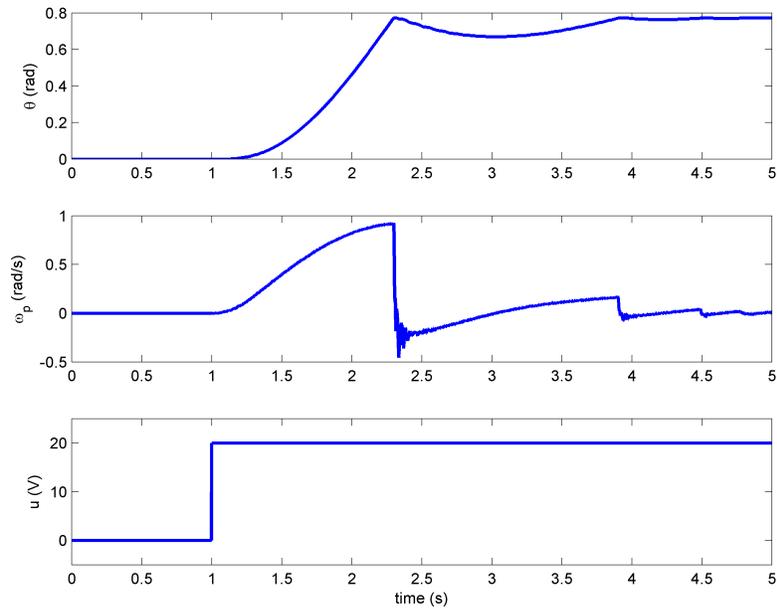


Figure 2.9: Pitch Step Response

2. Run the `q_aero_step_response_yaw_from_pitch` QUARC controller to apply a step voltage 20V to the main rotor and examine the yaw response.
3. Sample response is given in Figure 2.12.

In this case, applying a positive torque about the pitch generates a *negative* torque about the yaw axis. Using Equation 2.24 and the response in Figure Figure 2.12, the cross-thrust gain of the yaw due to a voltage in the pitch rotor is

$$K_{yp} = \frac{J_y \frac{\Delta\Omega_y}{\Delta t} + D_y \Delta\omega_p}{V_p} = -0.0027 \text{ N-m/V}$$

where $\Delta\omega_y = -1.95 - 0$ rad/s is the change in angular rate of the pitch axis over $\Delta t = 5 - 1 = 4$ s. The gain value is negative, again, because a positive pitch torque results in a negative torque about the yaw.

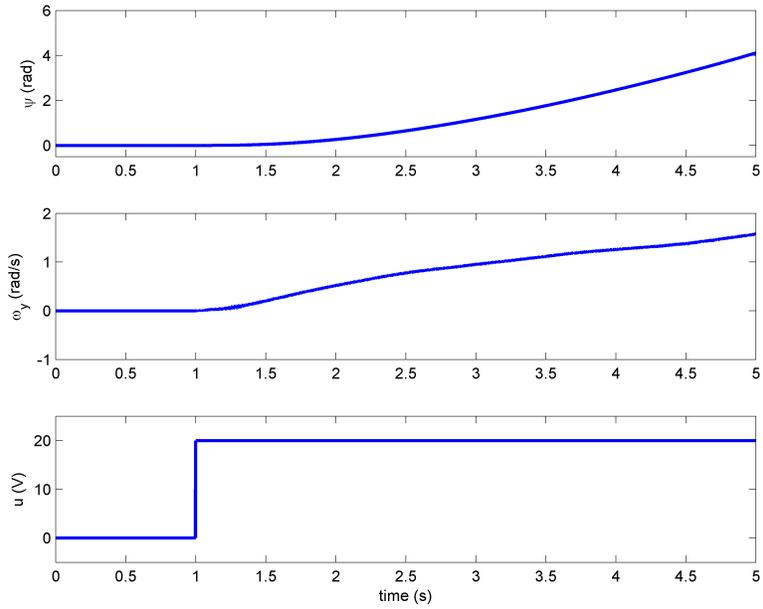


Figure 2.10: Yaw Step Response

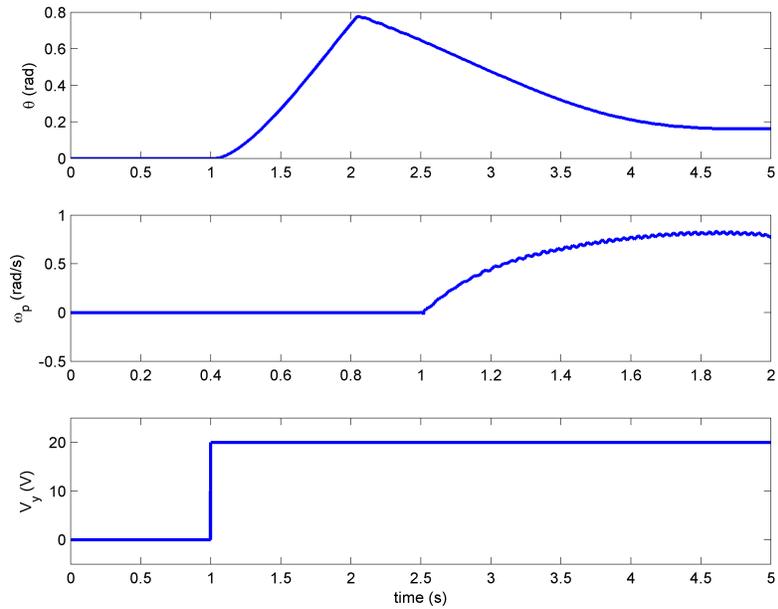


Figure 2.11: Pitch Step Response from Yaw Voltage

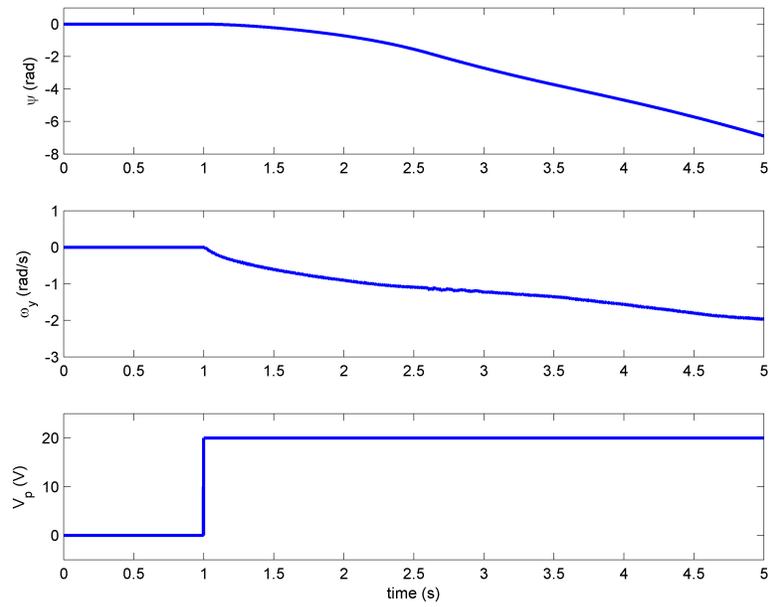


Figure 2.12: Yaw Step Response from Pitch Voltage

3 PD Control

3.1 Background

This section presents a PID-based controller to stabilize the pitch and yaw axes to desired angles. A variation of the classic PD control will be used: the proportional-velocity control illustrated in Figure 3.1. Unlike the standard PD, only the negative velocity is fed back (i.e. not the velocity of the *error*) and a low-pass filter will be used in-line with the derivative term to suppress measurement noise. The combination of a second-order low-pass filter and the derivative term results in a high-pass filter, $H(s)$, which will be used instead of a direct derivative.

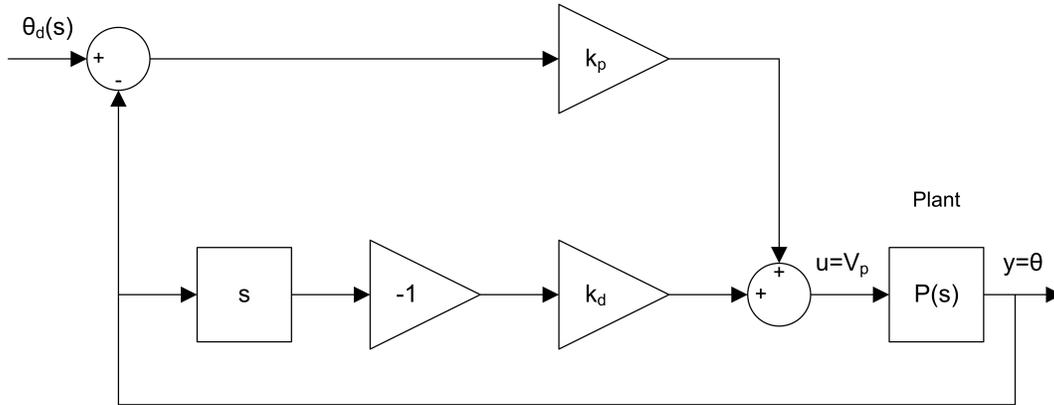


Figure 3.1: Proportional-Velocity (PV) control

The proportional-velocity (PV) control has the following structure

$$u = k_p (r(t) - y(t)) - k_d \dot{y}(t), \quad (3.1)$$

where k_p is the proportional gain, k_d is the derivative (velocity) gain, $r = \theta_d(t)$ is the reference pitch angle, $y = \theta(t)$ is the measured pitch angle, and $u = V_p(t)$ is the control input (applied motor voltage to the pitch rotor). Similarly, for the PV control of the yaw axis, these would be $r = \psi_d(t)$, $y = \psi(t)$, and $u = V_y(t)$.

The closed-loop transfer function for the pitch-only axis is denoted $Y(s)/R(s) = \Theta_d(s)/\Theta(s)$. Assuming all the initial conditions are zero, i.e. $\theta(0^-) = 0$ and $\dot{\theta}(0^-) = 0$, and taking the Laplace transform of Equation 3.1 yields

$$U(s) = k_p(R(s) - Y(s)) - k_d s Y(s),$$

which can be substituted into the $\Theta(s)/V_p(s)$ transfer function model given in Equation 2.3 to get

$$Y(s) = \frac{K}{s(\tau s + 1)} (k_p(R(s) - Y(s)) - k_d s Y(s)).$$

The steady-state gain and time constant of the pitch transfer function are

$$K = \frac{K_{pp}}{D_p} \text{ and } \tau = \frac{J_p}{D_p}$$

Solving for $Y(s)/R(s)$, we obtain the closed-loop transfer function

$$\frac{Y(s)}{R(s)} = \frac{K k_p / \tau}{s^2 + (1 + K k_d) / \tau s + K k_p / \tau}. \quad (3.2)$$

Using the standard second-order prototype transfer function

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (3.3)$$

we can express the PV control gains based on a the required natural frequency, ω_n , and damping ratio, ζ , with the equations

$$k_p = \frac{\tau\omega_n^2}{K} \quad (3.4)$$

and

$$k_d = \frac{2\tau\zeta\omega_n - 1}{K}. \quad (3.5)$$

Peak Time and Overshoot Equations

We can use the following expressions to obtain the required ω_n and ζ from the peak time and overshoot specifications. In a second-order system, the amount of overshoot depends solely on the damping ratio parameter

$$PO = 100e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \quad (3.6)$$

The peak time depends on both the damping ratio and natural frequency of the system and it can be derived as:

$$t_p = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}} \quad (3.7)$$

Generally speaking, the damping ratio affects the shape of the response while the natural frequency affects the speed of the response.

Decoupled PD control design:

Note that the dynamic coupling between the axes is not considered in this control design. The PD compensator is designed separately for each axis and is based on a model that uses only the main thrust torques (i.e. $K_{pp}V_p$ and $K_{yy}V_y$) and not the cross-coupling torques. This is known as a *de-coupled controller*. As a result, simultaneously controlling the pitch and yaw axes to a desired reference command may yield unexpected motions. For example, large overshoot can be seen in the pitch axis as the yaw is tracking a reference command signal.

3.2 In-Lab: PD Control Design and Simulation

In this lab we will design PV gains according to a set of specifications and simulated the closed-loop response to ensure it matches those specifications. The PV control that is implemented in described in Section 3.1.

Desired closed-loop response specifications:

1. Steady-state error: pitch $e_{ss} \leq 2$ deg.
2. Peak time: $t_p \leq 2$ s.
3. Percent Overshoot: $PO \leq 7.5\%$.
4. No actuator saturation: $|V_y| \leq 24V$ and $|V_p| \leq 24V$.

The closed-loop response is the simulated in Simulink® using the `s_aero_2dof_pid_control` shown in Figure 3.2.

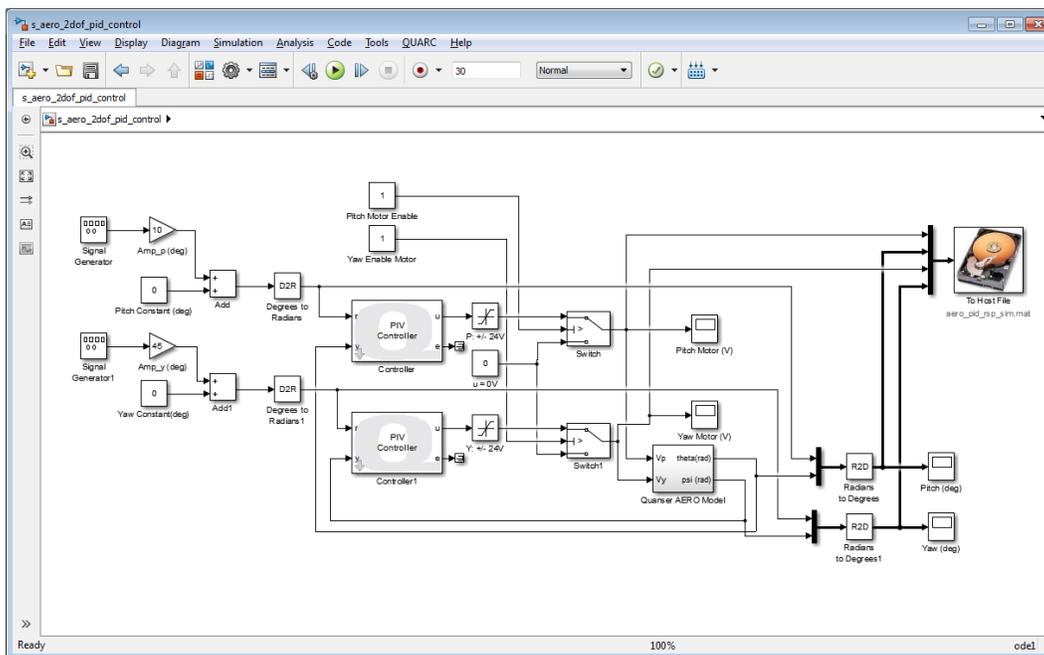


Figure 3.2: Simulink model used to simulate PV response

This model uses the QUARC PID block to implement the PV control for the pitch and yaw axes. The *Quanser AERO Model* subsystem contains the four transfer functions given in Section 2.1.2. Remark that this is the full system model that includes the coupling between the axes. In Section 3.1, the PV gains are designed for a single-input, single-output system. Because the design does not include the coupled dynamics, we will *over-design* the peak time to ensure the specification above is met. To do this, let's design a set of PV control gains for the peak time $t_p = 1.75$ and the original percent overshoot of $PO = 7.5\%$. Using Equation 3.6 and Equation 3.7, we would need a damping ratio and natural frequency of:

$$\zeta = -\log\left(\frac{PO}{100}\right) \frac{1}{\sqrt{\log\left(\frac{PO}{100}\right)^2 + \pi^2}} = 0.636$$

and

$$\omega_n = \frac{\pi}{t_p \sqrt{1 - \zeta^2}} = 2.33 \text{ rad/s.}$$

Using Equation 3.4 and Equation 3.5, we can find what PV gains are required to meet the natural frequency and damping ratio above for the pitch

$$k_{p,p} = 108 \text{ V/rad and } k_{d,p} = 52.4 \text{ V-s/rad} \quad (3.8)$$

and for the yaw

$$k_{p,y} = 54.1 \text{ V/rad and } k_{d,y} = 19.6 \text{ V-s/rad.} \quad (3.9)$$

Running the closed-loop PV control simulation:

1. Open the Matlab scripts *quanser_aero_parameters.m* and *aero_pd_design.m*. As shown below, *quanser_aero_pv_design.m* loads the model parameters that are set in *quanser_aero_parameters.m* script. They are loaded with default damping coefficient and thrust gains that were estimated in Section 2.2. You may want to update these for value you found. The DC gain and time constant for the pitch and yaw axes, according to the transfer functions given in Section 2.1.2, are used to design the PV gains for each axis individually.

```
% Load model parameters
quanser_aero_parameters;
%
% Peak time and overshoot specifications
PO = 7.5;
tp = 1.75;
% Damping ratio from overshoot specification.
zeta = -log(PO/100) * sqrt( 1 / ( ( log(PO/100) )^2 + pi^2 ) );
% Natural frequency from specifications (rad/s)
wn = pi / ( tp * sqrt(1-zeta^2) );
%
%% Pitch PV Design
% Set DC gain and time constant for pitch axis
K = Kpp/Dp;
tau = Jp/Dp;
% Proportional gain (V/rad)
kp_p = tau*wn^2/K;
% Derivative/velocity gain (V-s/rad)
kd_p = (2*tau*zeta*wn-1)/K;
%
%% Yaw PV Design
% Set DC gain and time constant for yaw axis
K = Kyy/Dy;
tau = Jy/Dy;
% Proportional gain (V/rad)
kp_y = tau*wn^2/K;
% Deriveative/velocity gain (V-s/rad)
kd_y = (2*tau*zeta*wn-1)/K;
```

2. Run the script to generate PV control gains for each axis. Note that the gain depends on the model parameters set in *quanser_aero_parameters.m* and the peak time and overshoot specifications set: t_p and PO . Using the standard model parameters and the specifications above, the following control gains are generated:

```
kp_p =
```

```
107.7148
```

```
kd_p =
```

```
52.4365
```

```
kp_y =
```

54.1163

$kd_y =$

19.5924

3. Run the *s_aero_2dof_pid_control* Simulink diagram to simulated the closed-loop response of the Quanser Aero Experiment using the transfer function model in Section 2.1.2.
4. See the samples response given in Figure 3.3.

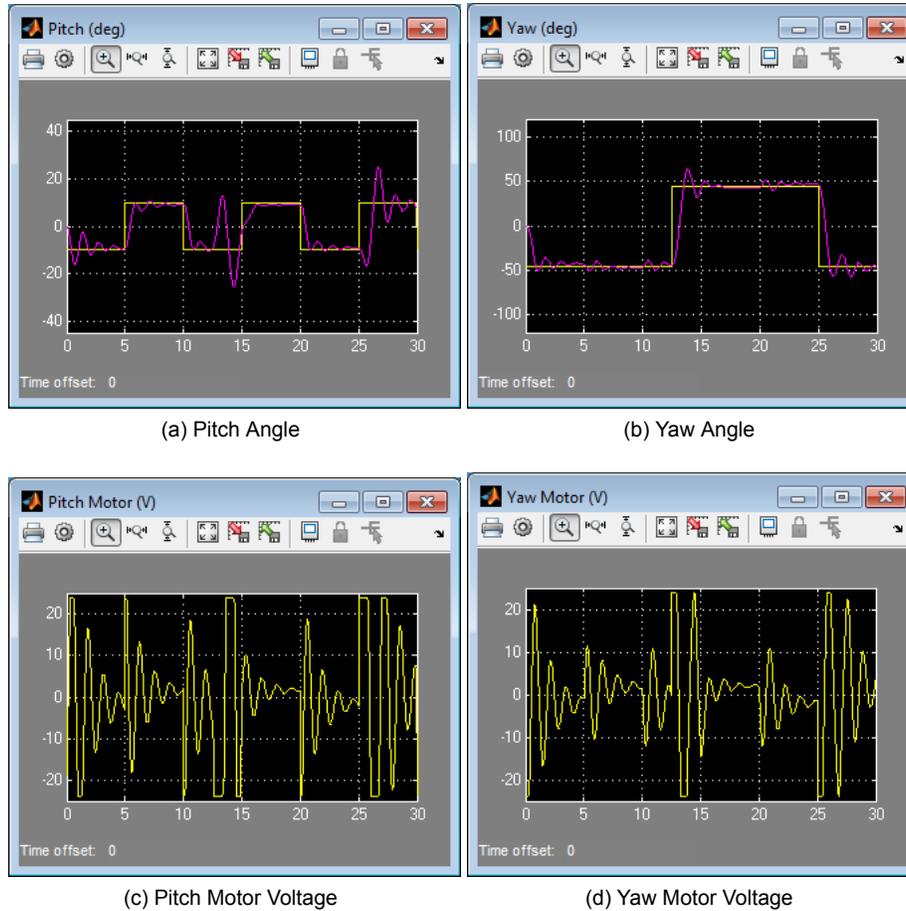


Figure 3.3: Sample response in scopes when running PV simulation

Response Analysis

The obtained simulated response is shown in Figure 3.4. Because the transfer function model used in Simulink takes into account the dynamic coupling between the axes, the pitch is affected by a change in the yaw and vice versa. For example, as the yaw goes from 45 to -45 deg at the 12.5 sec mark the pitch tends to increase. Similarly, when the reference command in the pitch changes it affects the yaw axis.

The peak time, overshoot, and steady-state error of the simulated response are:

1. Steady-state error: $e_{ss} = 0 \leq 2$ deg for both pitch and yaw.
2. Peak time: pitch $t_{p,p} = 7.17 - 5 = 2.17 > 2$ s, yaw $t_{p,y} = 13.9 - 12.5 = 1.4 \leq 2$ s.
3. Percent Overshoot: pitch $PO_p = (10.4 - 12.5)/20 = 2\% \leq 7.5\%$, yaw $PO_y = (63.8 - 45)/90 = 20.9\% > 7.5\%$.

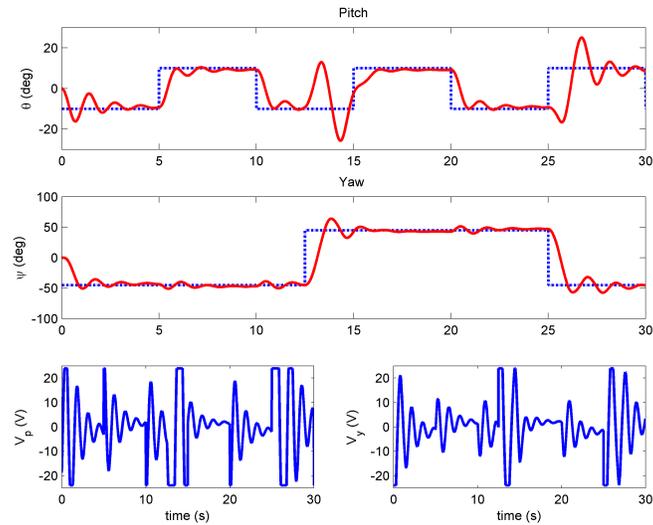


Figure 3.4: Sample PV simulated closed-loop response

4. Actuator saturation: $|V_y| \leq 24V$ and $|V_p| \leq 24V$.

The pitch peak time and yaw overshoot do not satisfy *the control design specifications* given above. However, this is mainly due to the dynamic coupling effect. For instance, the overshoot in the yaw is computed at time 12.5 s when the yaw goes from -45 to 45 deg. At the same time, the pitch attempts to maintain its setpoint of -10 deg which results in a negative voltage being applied to pitch motor. From the modeling lab in 2.2.3, we know that applying a negative voltage to the pitch rotor would cause a positive torque about the yaw.

Given the pitch peak time only slightly surpasses our desired specification, we can test these gain on the actual system first and see if they do need to be re-tuned.

Note that although the motor rotors do reach their $\pm 24V$ limit, it's only for a short instant. The Saturation blocks placed in the Simulink diagram ensure that it never goes above this.

3.3 In-Lab: Running PD on System

In this section the PD control Section 3.1 is implemented on the Quanser Aero Experiment using the *q_aero_2dof_pid_control* Simulink diagram shown in Figure 3.5 with QUARC®.

Running the PD on the Quanser Aero Experiment :

1. Unlock both the pitch and yaw axes to enable the full 2 DOF motion.
2. Similarly as done in Section 3.2, generate the PV control gains using the Matlab script *quanser_aero_pv_design.m*:

```
kp_p =
```

```
107.7148
```

```
kd_p =
```

```
52.4365
```

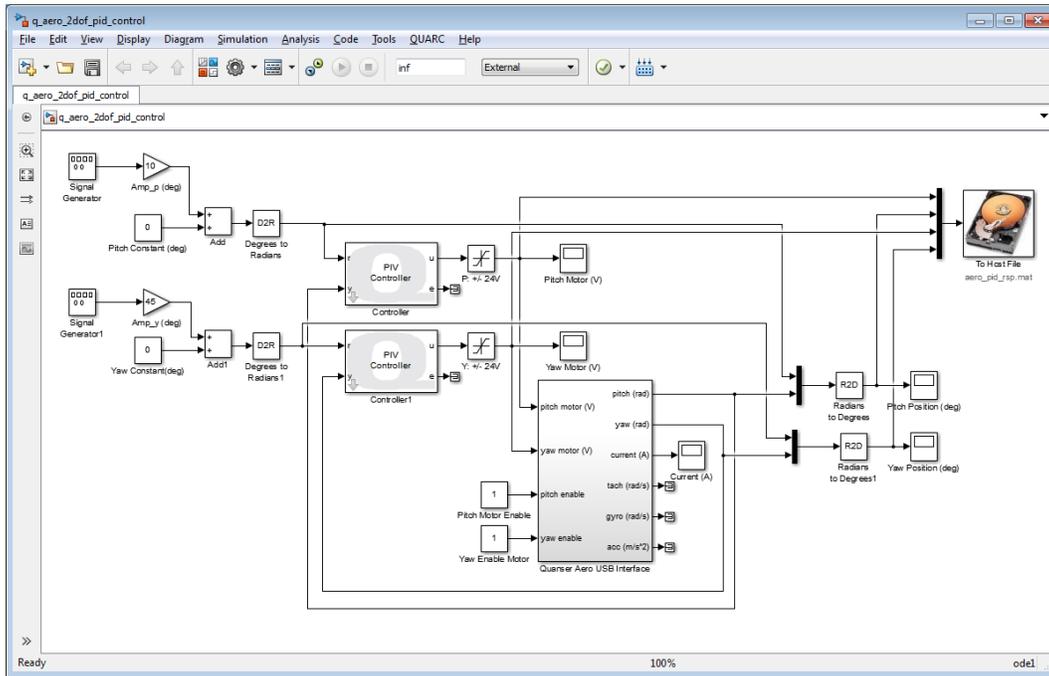


Figure 3.5: Simulink model used to run PV controller on Quanser Aero Experiment system using QUARC

$kp_y =$

54.1163

$kd_y =$

19.5924

3. Enter those gains are set in the *PIV Controller* blocks - for both the pitch and yaw axes.
4. Build and run the *q_aero_2dof_pid_control* QUARC controller to implement the PV controller with the PV gains entered.
5. Examine the obtained closed-loop response and see if it matches the desired specifications given in Section 3.2.
6. If the control specifications are not satisfied, the PV control can be redesigned for a new peak time and overshoot requirements. To do this, adjust the t_p and PO parameters in *quanser_aero_pv_design.m* and run the script to generate the new gains. Set the new gains in the **PIV Controller** blocks (or enter the Matlab parameter names in the blocks so they are updated automatically) and run the QUARC controller again.
7. Sample scope response is given in Figure 3.6.

Response Analysis

The obtained response is shown in Figure 3.7. Similarly as noticed in the simulation, the coupling effect between the axes is seen as the pitch and yaw changes positions.

The peak time, overshoot, and steady-state error of the measured response on the system are:

1. Steady-state error: pitch $e_{ss,p} = |7.56 - 10| = 2.44 > 2 \text{ deg}$, yaw $e_{ss,y} = |53.4 - 60| = 5.5 > 2 \text{ deg}$.
2. Peak time: pitch $t_{p,p} = 5.74 - 5 = 0.7 \leq 2 \text{ s}$, yaw $t_{p,y} = 14.9 - 12.5 = 2.4 > 2 \text{ s}$.

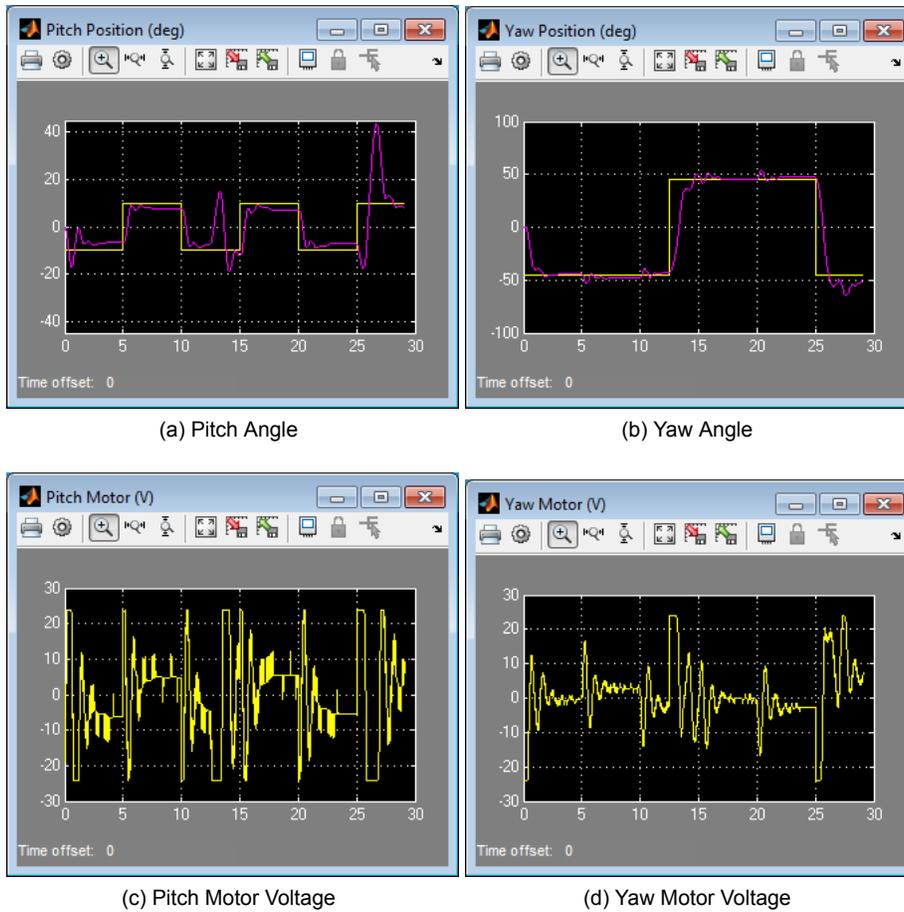


Figure 3.6: Sample response in scopes when running PV on Quanser Aero Experiment

3. Percent Overshoot: pitch $PO_p = 0\% \leq 7.5\%$, yaw $PO_y = (49.8 - 45)/90 = 5.33\% \leq 7.56\%$.
4. Actuator saturation: $|V_y| \leq 24V$ and $|V_p| \leq 24V$.

Probably due to unmodeled friction (e.g. Coulomb friction), the system is slightly more damped than in the simulation which causes there to be less overshoot. However, the additional friction in the actual system causes there to be a larger steady-state error than in the simulation and do not match the control requirements. The error could be minimized by introducing an integrator, increasing the proportional gain, or using a more advanced friction compensation scheme. Finally, the motor voltages peak at the $\pm 24V$ limit, but only for a short instant and Saturation blocks are implemented to prevent over-voltage. These control additions would most likely improve the response time of the yaw such that its peak time meets the specifications.

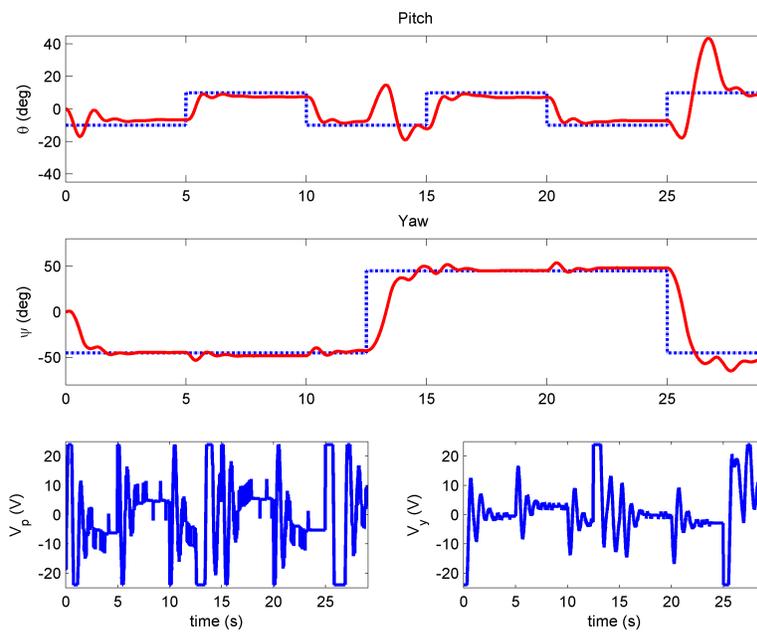


Figure 3.7: Sample PV closed-loop response on Quanser Aero Experiment

4 State-Feedback Control

4.1 Background

In this section a state-feedback controller is designed to regulate the pitch and yaw angles of the Quanser Aero Experiment to desired angles. By using the state-space model given in Section 2.1, we can find a control gain K based on the coupled dynamics to stabilize this system. The control gains are computed using the Linear-Quadratic Regulator (LQR) algorithm. The general state-feedback control is illustrated in Figure 4.1.

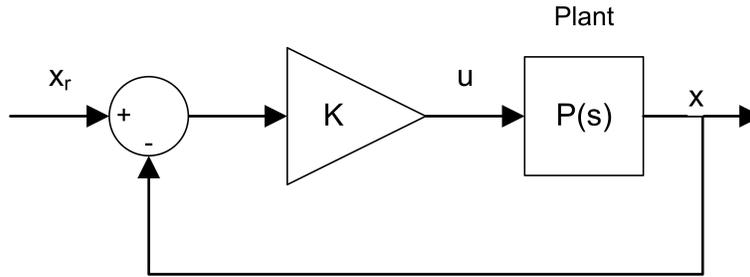


Figure 4.1: Block diagram of balance state-feedback control for rotary pendulum

The state-feedback controller is defined

$$u = K(x_d - x),$$

where x is the state defined in Equation 2.5 in Section 2.1.3,

$$x_d^T = \begin{bmatrix} \theta_d & \psi_d & 0 & 0 \end{bmatrix}$$

is the reference command (or setpoint) state with the desired pitch and yaw angles, θ_d and ψ_d , and

$$u^T = \begin{bmatrix} V_p & V_y \end{bmatrix}$$

is the control input where V_p is the front/pitch motor voltage and V_y is the tail/yaw motor voltage.

Linear Quadratic Regulator (LQR) optimization can be used for finding the control gain parameters of the Quanser Aero Experiment flight control. Given the state-space representation in Section 2.1, the LQR algorithm computes a control law u to minimize the performance criterion or cost function

$$J = \int_0^{\infty} (x_{ref} - x(t))^T Q (x_{ref} - x(t)) + u(t)^T R u(t) dt. \quad (4.1)$$

The design matrices Q and R hold the penalties on the deviations of state variables from their setpoint and the control actions, respectively. When an element of Q is increased, therefore, the cost function increases the penalty associated with any deviations from the desired setpoint of that state variable, and thus the specific control gain will be larger. When the values of the R matrix are increased, a larger penalty is applied to the aggressiveness of the control action, and the control gains are uniformly decreased.

Since there are four states, $Q \in \mathbb{R}^{4 \times 4}$, and two control variables, $R \in \mathbb{R}^{2 \times 2}$. The setpoint, x_d is given above the control strategy used to minimize cost function J is thus given by

$$u = K(x_d - x) = k_{p,\theta}(\theta_d - \theta) + k_{p,\psi}(\psi_d - \psi) - k_{d,\theta}\dot{\theta} - k_{d,\psi}\dot{\psi}. \quad (4.2)$$

Designing an LQR controller:

Various control software already have LQR optimization routines that can be used to generate the state-feedback control gain K . In order for the closed-loop response to satisfy certain time-domain specifications, the closed-loop system is typically simulated using its dynamic model, in software, first. This is an iterative process. By adjusting the weighting matrices Q and R and then running the simulation, we can find a control that satisfies the user's requirements. Further, we must ensure that the control signal u is smooth (i.e. does not chatter) and does not surpass the limits of the actuator.

4.2 In-Lab: LQR Control Design and Simulation

As described in Section 4, LQR is used to find the state-feedback control gain K that will stabilize the Quanser Aero Experiment to the user's desired pitch and yaw angles. Our desired closed-loop response should match the following specifications.

Desired closed-loop response specifications:

1. Steady-state error: pitch $e_{ss} \leq 2$ deg, yaw $e_{ss} \leq 2$ deg.
2. Peak time: $t_p \leq 2$ s.
3. Percent Overshoot: $PO \leq 7.5\%$.
4. No actuator saturation: $|V_p| \leq 24V$ and $|V_y| \leq 24V$.

Matlab[®] will be used to run the LQR optimization routine and generate K and the closed-loop response is the simulated in Simulink[®] using the `s_aero_2dof_lqr_control` shown in Figure 4.2.

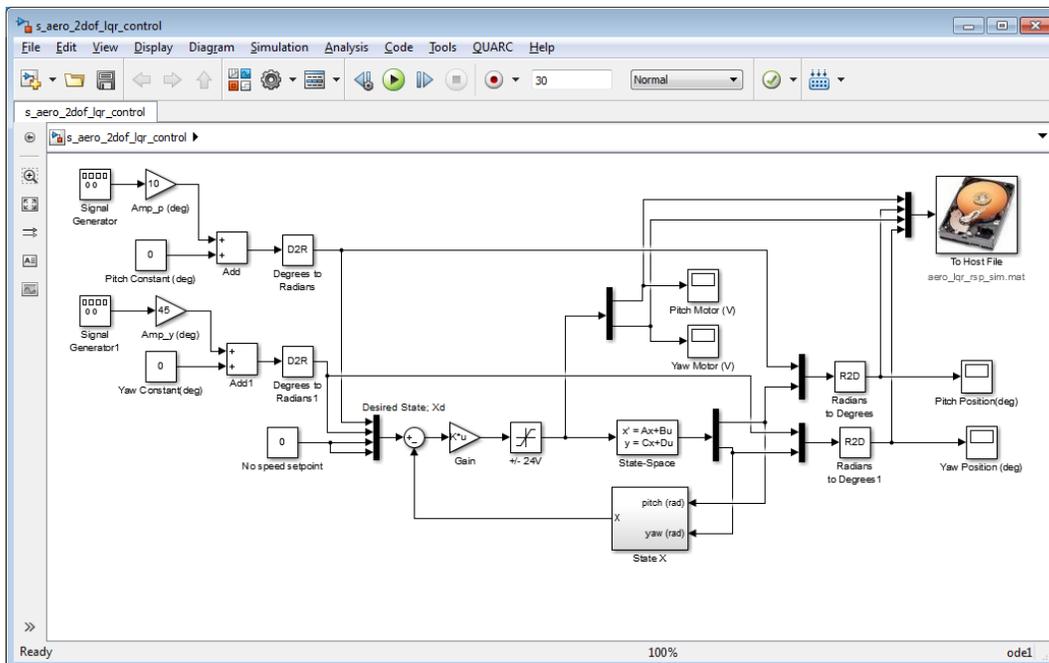


Figure 4.2: Simulink model used to simulate closed-loop state-feedback response

The state-space matrices derived in Section 2.1.3 are entered in the *State-Space* block in Simulink and the control gain is set to the Matlab variable K .

Running the closed-loop state-feedback LQR simulation:

1. Open the Matlab script `quanser_aero_lqr.m` shown below. This loads the model parameters that are set in `quanser_aero_parameters.m` script and the state-space model matrices set in `quanser_aero_state_space.m`. The model parameters in `quanser_aero_parameters.m` are the values that were estimated in Section 2.2. These, of course, can be updated to values reflecting your system more accurately. Once the state-space model is loaded in the workspace, the LQR command is used to compute the control gain K .

```
% Load model parameters
quanser_aero_parameters;
% Load state-space matrices
```

```

quanser_aero_state_space;
%
%% State-Feedback LQR Control Design
Q = diag([200 75 0 0]);
R = 0.01*eye(2,2);
K = lqr(A,B,Q,R)

```

- Run the script to generate control gain K . Note that the gain depends on the model parameters set in *quanser_aero_parameters.m* and the Q and R weighting matrices that are set in the *quanser_aero_lqr.m* script. Use the standard model parameters and the weighting given above, the following control gain is generated:

$K =$

```

98.2088 -103.0645 32.2643 -29.0750
156.3469 66.1643 45.5122 17.1068

```

- Run the *s_aero_2dof_lqr_control* Simulink diagram to simulated the closed-loop response of the Quanser Aero Experiment using the state-space model in Section 2.1.
- See the samples response given in Figure 4.3.

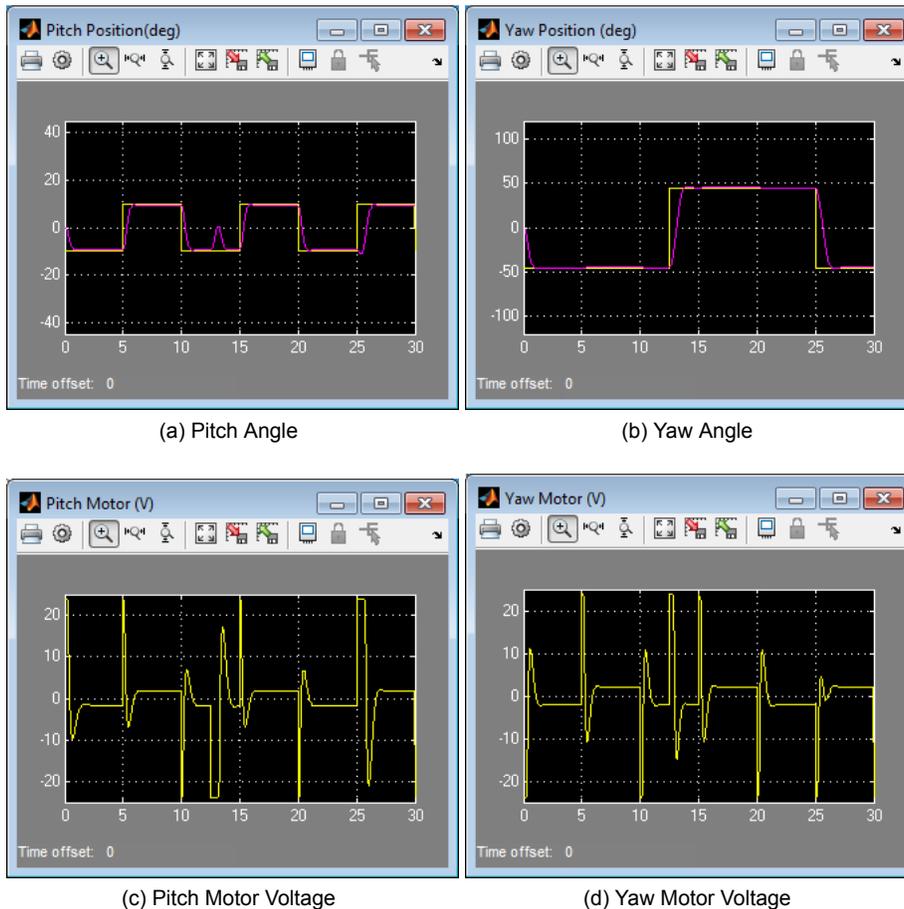


Figure 4.3: Sample response in scopes when simulating LQR control

Response Analysis

Using the weighting matrices

$$Q = \begin{bmatrix} 200 & 0 & 0 & 0 \\ 0 & 75 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad R = \begin{bmatrix} 0.005 & 0 \\ 0 & 0.005 \end{bmatrix}$$

and the state-space matrices (A,B) found previously, we generated the control gain

$$K = \begin{bmatrix} 98.2088 & -103.0645 & 32.2643 & -29.0750 \\ 156.3469 & 66.1643 & 45.5122 & 17.1068 \end{bmatrix}$$

The obtained simulated response is shown in Figure 4.4. Because our model takes into account some of the coupling dynamics, the pitch is affected by a change in the yaw and vice versa. Thus as the yaw goes from 60 to -60 deg at the 12.5 sec mark, the pitch tends to increase slightly. This is compensated by the state-feedback control. Similarly, notice how the changes along the pitch affect the yaw angle.

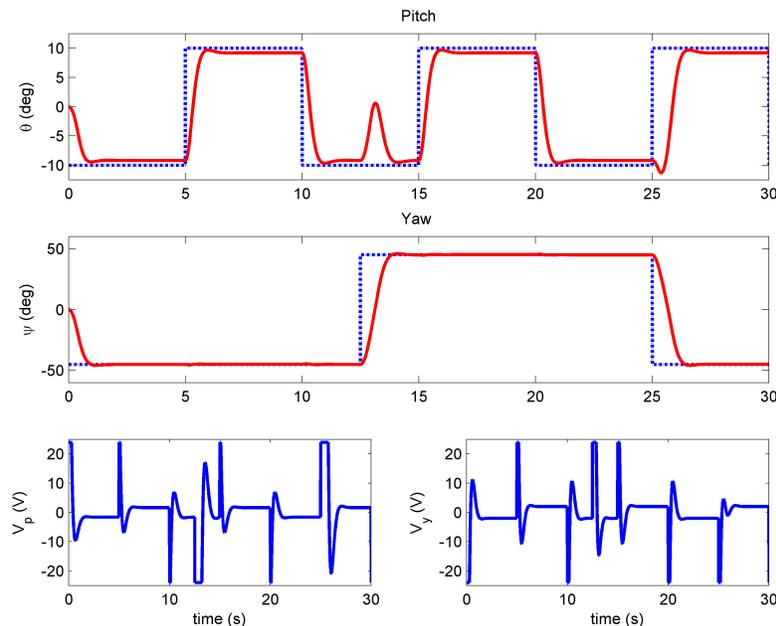


Figure 4.4: Sample LQR simulated closed-loop response

The peak time, overshoot, and steady-state error of the simulated response are:

1. Steady-state error: pitch $e_{ss} = 10 - 9.2 = 0.8 \leq 2$ deg, yaw $e_{ss} = 0 \leq 2$ deg.
2. Peak time: pitch $t_{p,p} = 5.96 - 5 = 0.96 \leq 2$ s, yaw $t_{p,y} = 14.1 - 12.5 = 1.6 \leq 2$ s.
3. Percent Overshoot: pitch $PO = 0\% \leq 7.5\%$, yaw $PO = (45.8 - 45)/90 = 0.89\% \leq 7.5\%$.
4. Actuator saturation: $|V_y| \leq 24V$ and $|V_p| \leq 24V$.

The response matches the desired specifications. Note that although the motor rotors do reach their $\pm 24V$ limit, it's only for a short instant. As noted, Saturation blocks are placed in the Simulink diagram ensure that it never goes above this. Compared to the decoupled PV design in 3.2.

Comparing the simulated PD control response in Figure 3.4 with the state-feedback response in Figure 4.4, notice how the state-feedback control handles the dynamic coupling of the system much more effectively.

4.3 In-Lab: Running LQR on System

In this section the state-feedback control is implemented on the Quanser Aero Experiment using the *q_aero_2dof_lqr_control* Simulink diagram shown in Figure 4.5 with QUARC® .

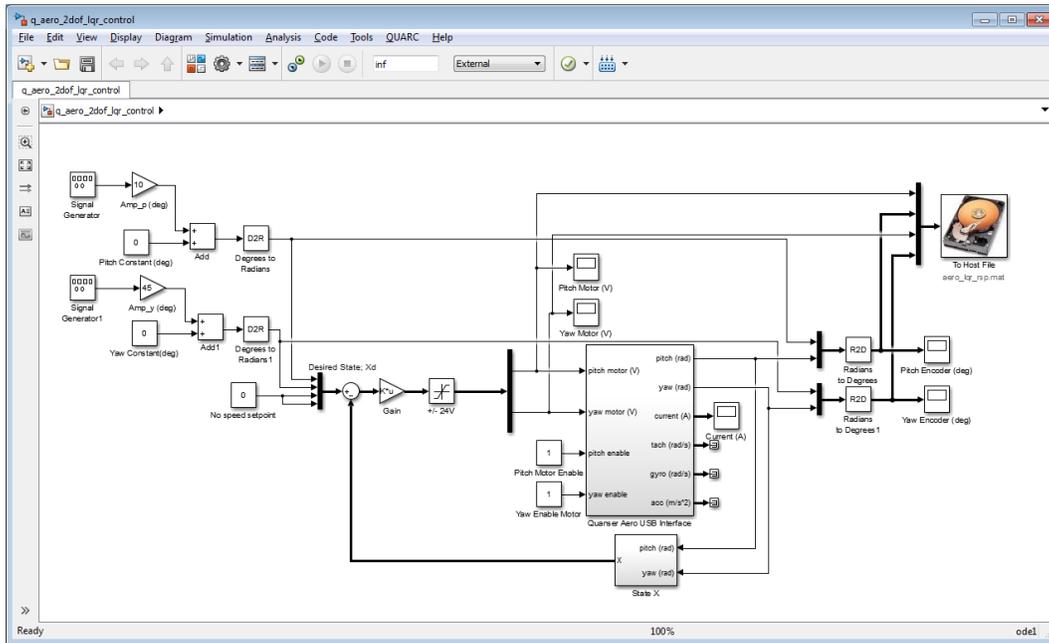


Figure 4.5: Simulink model used with QUARC to run state-feedback control on Quanser Aero Experiment

1. Unlock both the pitch and yaw axes to enable the full 2 DOF motion.
2. Similarly as done in Section 4.2, generate the control gain K using the Matlab script *quanser_aero_lqr.m* and the same Q and R :

$K =$

98.2088	-103.0645	32.2643	-29.0750
156.3469	66.1643	45.5122	17.1068

3. Run the *q_aero_2dof_lqr_control* QUARC controller to implement the state-feedback control with gain K above on the Quanser Aero Experiment .
4. Examine the obtained closed-loop response and see if it matches the desired specifications given in Section 4.2.
5. If it does not match the desired specifications, then you can tune your LQR control according to the guideline given in Section 4.1. To do this, adjust the Q and R weighting matrices in the Matlab script, run the *quanser_aero_lqr.m* script to generate a new gain K , and run the QUARC controller again.
6. Sample scope response is given in Figure 4.6.

Response Analysis The obtained response is shown in Figure 4.7.

The peak time, overshoot, and steady-state error of the simulated response are:

1. Steady-state error: pitch $e_{ss} = |8.09 - 10| = 1.91 \leq 2$ deg, yaw $e_{ss} = |45.26 - 45| = 0.26 \leq 2$ deg.
2. Peak time: pitch $t_{p,p} = 6.19 - 5 = 1.19 \leq 2$ s, yaw $t_{p,y} = 14.5 - 12.5 = 2.0 \leq 2$ s.

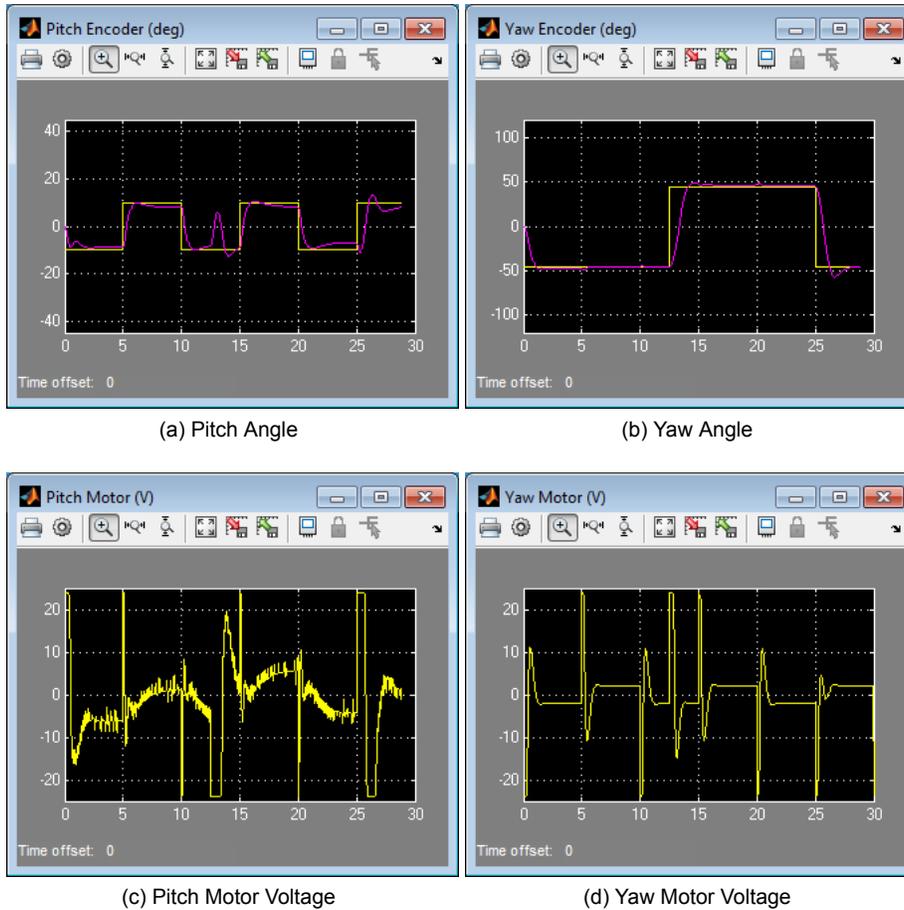


Figure 4.6: Sample response in scopes when running LQR on Quanser Aero Experiment

3. Percent Overshoot: pitch $PO = 0\% \leq 7.5\%$, yaw $PO = (48.25 - 45)/90 = 3.61\% \leq 7.5\%$.
4. Actuator saturation: $|V_y| \leq 24V$ and $|V_p| \leq 24V$.

The response matches the desired specifications. The rotors do reach their limit of $\pm 24V$, but only for a short instant and Saturation blocks are placed in the Simulink diagram to limit the applied voltage to this range. As noticed in the simulation, the coupling effect between the axes is seen as the pitch and yaw changes positions. There is slightly more steady-state error about the pitch axis, which can probably be attributed to unmodeled friction (e.g. Coulomb friction).

As in the simulation, notice that the state-feedback control compensates for the cross-coupling effect of the axes better than the PV controller response shown in Figure 3.7.

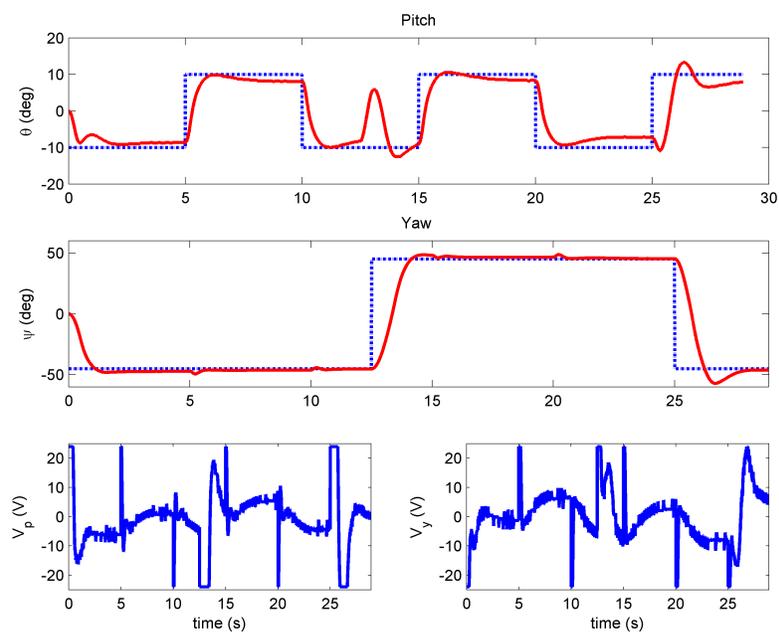


Figure 4.7: Sample response using LQR state-feedback on Quanser Aero Experiment

5 1 DOF Attitude LQG Design

In this section, a Kalman filter is designed that estimates the pitch angle position using the on-board IMU gyroscope and accelerometer sensors for the *Aero 1 DOF attitude* configuration. In this set up, only the front rotor is actuated and the system is constrained to move about the pitch axis (i.e. the yaw axis is locked). Using an IMU board for attitude estimation is common in various aerospace systems as a direct measurements are often not available.

5.1 Background

The Linear-Quadratic Gaussian (LQG) controller is basically a combination of using LQR to find the control gain and Kalman filtering to estimate the state. It also assumes white Gaussian noise acting at the control input and output sensors.

In Section 4.2, LQR was used to find the optimal control gain, K , for the state-feedback control of the Quanser Aero Experiment . When using LQR, we assumed that all the states were being measured directly. In this section, the *IMU acceleration is used to estimate the 1 DOF attitude position*. This position estimate and the rate measurement from the IMU gyroscope are then used in a Kalman filter to estimate state x .

The general block diagram of the Kalman estimator used to control a system is depicted in Figure 5.1.

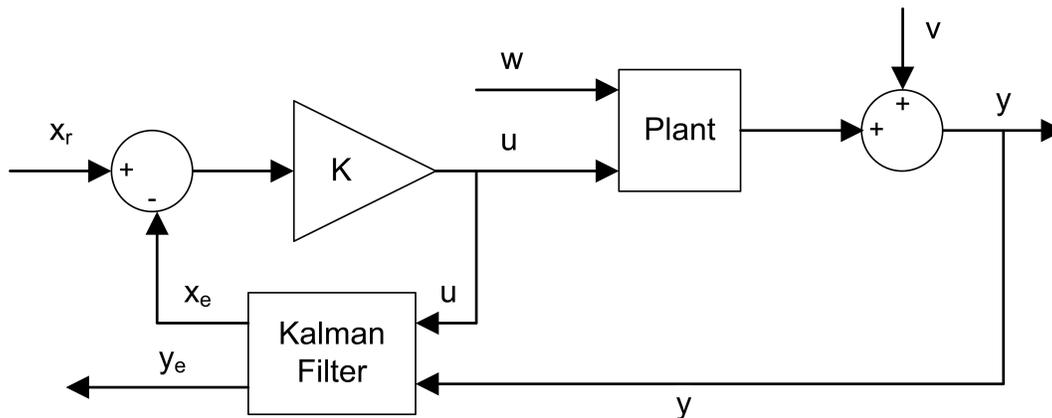


Figure 5.1: LQG controller

The Kalman filter inputs are the control signal, u , and the system output, $y + v$, where v is measurement noise. The Kalman filter generates the estimate of output vector, \hat{y} or y_e , and the estimate of state, \hat{x} or x_e . The Kalman filter estimate error can be assessed by comparing the measured and estimated outputs: y and \hat{y} . Further, with the Quanser Aero Experiment , we can compare the estimated pitch angle, $\hat{\theta}$, with the actual reading from the encoder, θ .

To design LQG controller, we will perform the following steps:

1. Construct the optimal control gain K using LQR optimization assuming all the states are being measured in the system, i.e. $\dot{x} = Ax + Bu$.
2. Construct a Kalman filter to estimate the system state, denoted as \hat{x} or x_e .
3. The state-feedback used to stabilize the system is $u = K(x_d - \hat{x})$.

5.1.1 Finding Pitch Angle from Acceleration

The angular position of the pitch can be estimated using the acceleration measurements from the three-axis accelerometer on the IMU of the Quanser Aero Experiment . The accelerations of the front rotor follow the right-hand rule and are illustrated in Figure 5.2. The accelerations acting in the x, y, and z-axes are denoted A_x , A_y , and A_z , respectively. Note the acceleration about the y-axis is pointing out of the page.

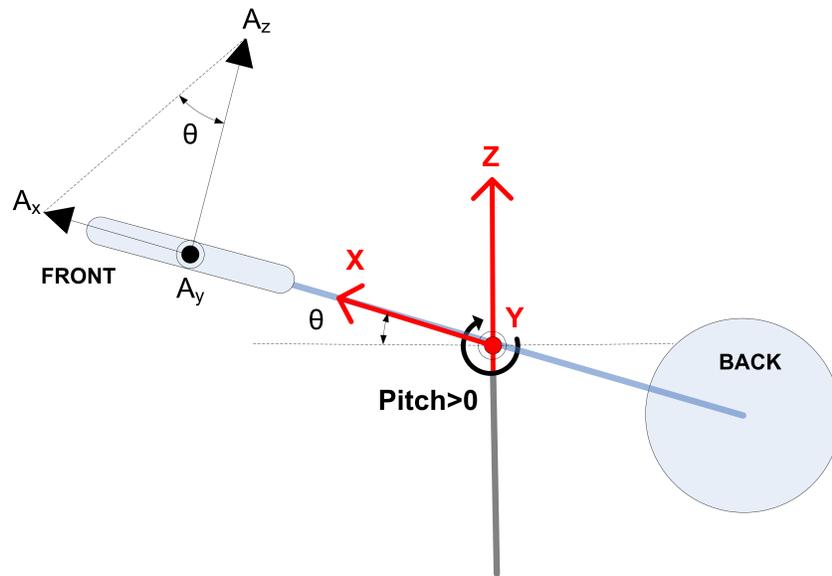


Figure 5.2: Finding pitch angle from accelerations

From the IMU acceleration measurements shown in this diagram, the angle of the pitch axis can be found with the equation

$$\theta = \sin^{-1} \frac{A_x}{\sqrt{A_x^2 + A_y^2 + A_z^2}}. \quad (5.1)$$

This takes into account all three acceleration dimensions as would be done in a free-flying, even though the acceleration about the y-axis (i.e. the roll) does not really come into effect.

5.1.2 State-Space Representation of Aero 1 DOF attitude

Using the 2 DOF equations of motion given in Section 2.1.1, the equation for the system when in the Aero 1 DOF attitude configuration is

$$J_p \ddot{\theta} + D_p \dot{\theta} + K_{sp} \theta = K_{pp} V_p. \quad (5.2)$$

For the state-space model, we define the state

$$x^T = [\theta(t), \dot{\theta}(t)],$$

the output vector

$$y^T = [\theta_a(t), \dot{\theta}(t)],$$

and the control variable

$$u = [V_p(t)]$$

where θ_a denotes the *pitch angle estimated using the accelerometer* (see Section 5.1.1). Using these definitions and the equation of motion in Equation 5.2, the state-space matrices are

$$A = \begin{bmatrix} 0 & 1 \\ -K_{sp}/J_p & -D_p/J_p \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ K_{pp}/J_p \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{and } D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

For the Kalman filter design on the Aero 1 DOF attitude, we will assume that there is a White Gaussian noise acting on the control input, w , and in the output sensor, v . The state-space model used to generate the estimator will therefore take into account the two inputs, (u, w) , and will be augmented to the system $(A, [BB], C, [DD])$.

5.2 In-Lab: LQG Design

The Kalman estimator and controller will be designed using the **Matlab**[®] software by going through the LQG design procedure given in Section 5.1. The LQR and Kalman filter design example is given in the Matlab script *pitch_kalman.m*.

Designing the LQG control:

1. Open the Matlab script *pitch_kalman.m*.
2. The script loads the model parameters in *quanser_aero_parameters.m* and sets the state-space model given in Section 5.1.

```
% State dimensions
nx = 2; % length of state x
ny = 2; % length of output y
nu = 1; % length of input u
nw = 1; % length of process noise w
%
A = [0 1;
     -Ksp/Jp -Dp/Jp];
%
B = [0; Kpp/Jp]; % nx x nu
%
C = [1 0;
     0 1]; % ny x nx
%
D = [0; 0]; % ny x nu
%
```

3. The control gain K is generated using LQR, similarly as done in Section 4.2 for the 2 DOF configuration, with the following Q and R weighting matrices:

```
Q = diag([250 10]);
R = 0.005;
K = lqr(A,B,Q,R)
```

4. The necessary condition to design a Kalman filter is for the system to be detectable. To check this, the *rank test* is used.

```
rank = rank(observ(A,C)) % is it detectable?
```

5. For the Kalman filter design, we augment the state-space model to include the input noise w , giving us two inputs to the model: $[u, w]$. The *sys_w* model is also defined as a discrete system using the sampling time of the controller used.

```
% define state-space and include input noise 'w'
sys_w=ss(A, [B B],C, [D D],0.002,'inputname',{'u' 'w'},'outputname','y');
```

6. The Kalman observer is designed using the Matlab *kalman*. The *kest* variable contains the state-space model of the estimator, which is used in the Simulink model. These are saved under the matrices ak , bk , ck , and dk .

```
% Define weighting matrices
Qn = 50*eye(nw,nw); % nw x nw
Rn = diag([0.1 0.005]); % ny x ny
% Design Kalman observer
[kest,L,P,M] = kalman(sys_w,Qn,Rn);
[ak,bk,ck,dk,Ts]=ssdata(kest);
```


- White Gaussian noise is added in the Simulink model to the both the rate and acceleration measurements. This is all done to emulate a more realistic situation.

Running the LQG control on the Aero 1 DOF attitude :

1. Set up the Aero 1 DOF attitude configuration, as shown in Figure 1.1. Make sure:

- (a) Front rotor is horizontal and parallel to the ground.
- (b) *Yaw axis is locked* and the pitch axis is free to rotate.
- (c) Body is horizontal and parallel with the ground (i.e. pitch angle $\theta = 0$).

Note: If the Quanser Aero Experiment body does not remain horizontal and parallel with the ground at rest, then you can add large (e.g. paper clamp) to either the front or back rotor. If there is a mechanical bias and **the controller starts when the pitch is not initially zero, then the Kalman estimate will not be as accurate.**

2. Run the Matlab script *pitch_kalman.m* to generate control gain K using LQR and the Kalman filter state-space model to estimate the position and rate of the pitch angle. Some of the design was discussed in Section 5.2

3. Build and run the *q_aero_pitch_kalman* QUARC controller to implement the LQG control.

4. See the sample scope response in Figure 5.4.

- The *Pitch (deg)* scope displays the setpoint (yellow), measured (purple), and estimated (cyan) pitch angle. The measured signal shown is directly from the encoder (i.e. not the from the acceleration-based estimate).
- The *Position (deg)* scope shows the measured (yellow) position calculated from the accelerometers and the estimated position from the Kalman filter.
- The *Pitch Speed (rad/s)* scope displays the measured (yellow) rate from the gyroscope and the estimated (purple) rate from the Kalman filter.
- The *Pitch Motor (V)* scope shows the motor voltage applied to the front rotor.

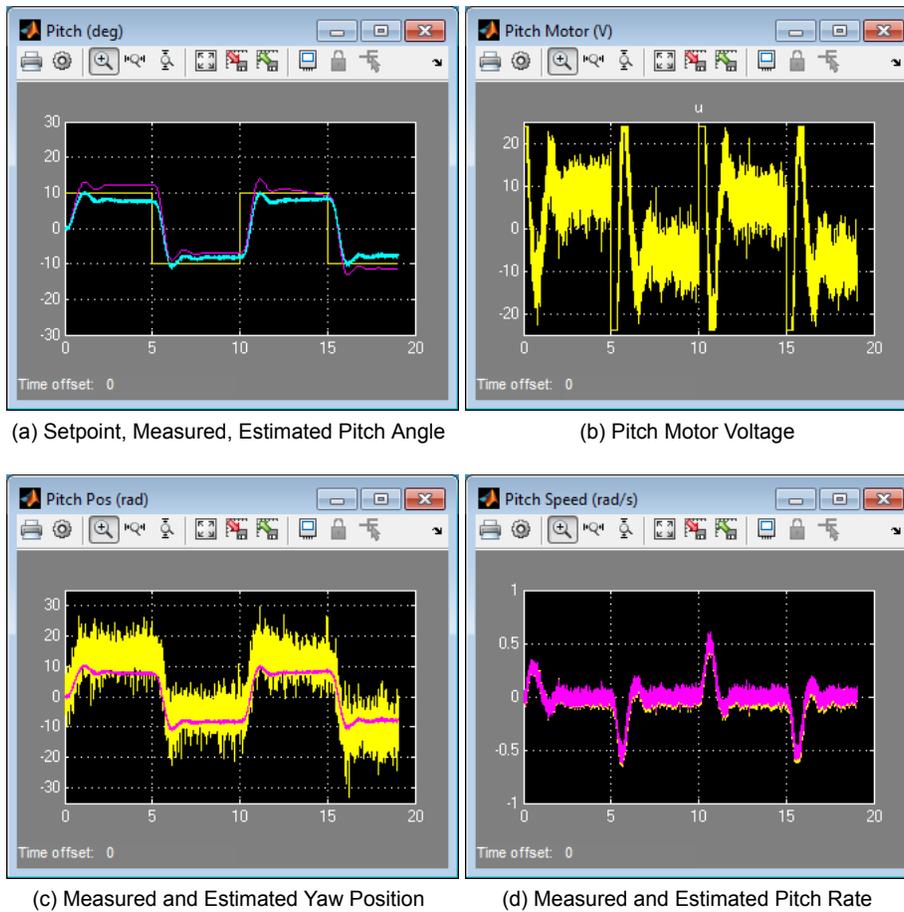


Figure 5.4: Sample response in scopes when running LQG on Aero 1 DOF attitude

- Examine the obtained closed-loop response and see if it matches the desired specifications given above. If it does not match the desired specifications, then you can tune your Kalman observer by adjusting the Q_n and R_n weighting matrices in the Matlab script, run the *pitch_kalman.m* script to generate a new Kalman observer system (ak, bk, ck, dk), and run the QUARC controller again. Similarly, you can tune the control gain by adjusting the LQR weighting matrices Q and R .

Response Analysis

The obtained response is shown in Figure 5.5. Similarly as in the scopes above, the position response in the top-left corner includes the reference command / setpoint (black dot), the direct position measurement from the encoder (solid red line), and the estimated position from the Kalman filter (dash-dot blue line). There is an offset between the measured angle (from the encoder) and the estimated angle.

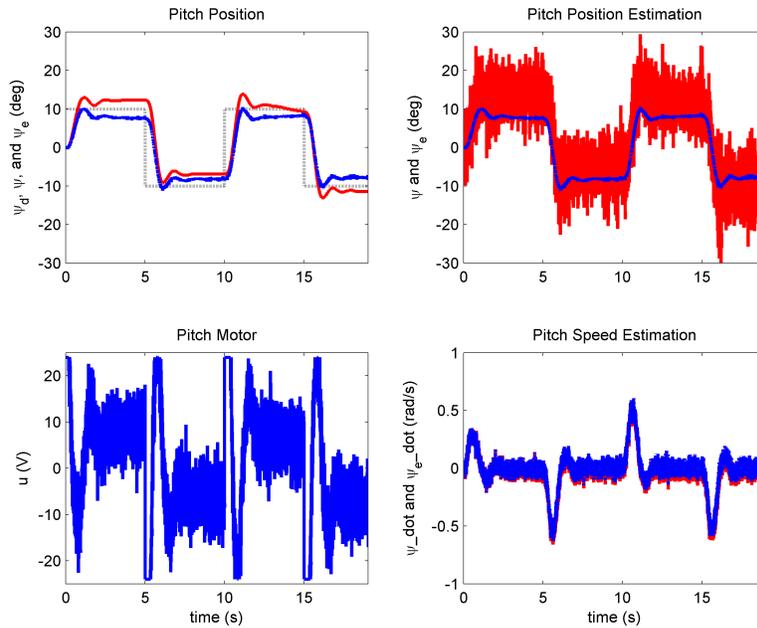


Figure 5.5: Sample LQG closed-loop response on Quanser Aero Experiment

The *Pitch Position Estimation* response shows the measured (solid red line) position calculated from the accelerometers and the estimated position from the Kalman filter (blue dash-dot). Notice the noise in the acceleration-based position signal. As discussed previously, the acceleration signals have more inherent noise due to the rotor vibrations as well as the added simulated noise. The position estimate from the Kalman filter has a lot less noise than the measured signal.

The *Pitch Speed Estimation* response shows the measured (solid red line) rate from the gyroscope and the estimated (blue dash-dot) rate from the Kalman filter.

The peak time, overshoot, and steady-state error of the *measured encoder position response* in Figure 5.5 are:

1. Steady-state error: $e_{ss} = 10 - 9.3 = 0.7 \leq 2$ deg
2. Peak time: $t_p = 11.1 - 10 = 1.1 \leq 2$ s.
3. Percent Overshoot: $\text{pitch } PO = (13.9 - 10)/20 = 19.5\% > 7.5\%$.
4. Actuator saturation: $|V_y| \leq 24V$ and $|V_p| \leq 24V$.

These measurements are taken on the second cycle. Comparing with the time-domain requirements listed in Section 4.2, the steady-state error and peak time satisfy the specifications but the percent overshoot does not (but it does in the first cycle). The motor rotors do reach their limit of $\pm 24V$, but only for a short instant. The overshoot could be compensated further by increasing the damping weighting terms in the Q matrix of the LQR control.