LABREPORT LINEAR SYSTEM THEORY

"Helicopter lab assignment"

Delivered to

Norwegian University of Science and Technology

Written by
Magnus Bolstad
Mads-Emil Kvammen
Per Arne Kjelsvik

Faculty of Information Technology, Mathematics and Electrical Engineering
NTNU
Høyskoleringen 1, Trondheim - 7491

October 26, 2016

TTK4115

Contents

1 Part I - Mathematical modeling

1.1 Problem 1

To analyse and control the helicopter, the physics has to be modelled mathematically. The two motors, each with a rotor, are modelled as two point masses, m_f and m_b. The points are connected to an axis a distance l_h from the elevation axis. A counterweight with mass m_c is placed at the same axis, a distance l_c from the elevation axis. Sums and differences are indicated with subscript s and d respectively. Linear relationship between supplied voltage to the motors, and applied forces from the rotors are also assumed.

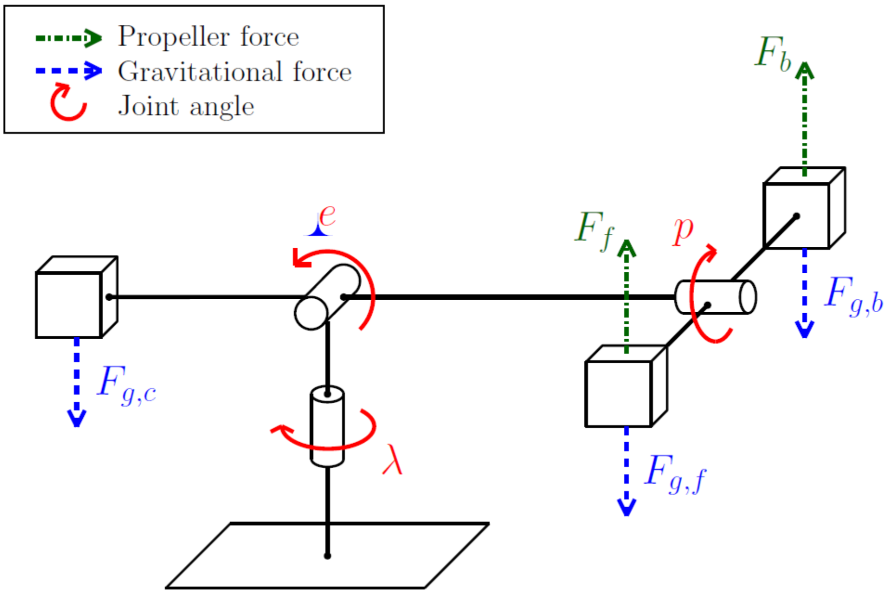


Figure 1: The helicopter with defined angles [2]

$$F_f = K_f V_f \tag{1a}$$
$$F_b = K_f V_b \tag{1b}$$

The helicopter arm can rotate in three different directions. The movement around the vertical axis is referred to as the travel of the helicopter, measured by the angle λ. Up and down is referred to as the elevation, measured by the angle e. The tilting with respect to the arm, p is referred to the pitch of the helicopter. Applying Newtons second law for rotational motion in each direction gives us a set of three equations.

$$J_p\ddot{p} = L_1 V_d \tag{2a}$$
$$J_e\ddot{e} = L_2 cos(e) + L_3 V_s cos(p) \tag{2b}$$
$$J_\lambda\ddot{\lambda} = L_4 V_s cos(e) sin(p) \tag{2c}$$

Where L_1, L_2, L_3 and L_4 are constants to be determined.

With idealized conditions, where friction is neglected, any difference in force from the two motors will result in an increased pitch angle rather than elevation or travel.

$$\begin{aligned} J_p\ddot{p} &= (F_f - F_b)l_p \\ &= (K_f V_f - K_f V_b)l_p \\ &= K_f \cdot V_d \cdot l_p \end{aligned}$$

To make the helicopter lift, the voltages supplied to the front and the back motor have to be equal. The direction of the force supplied by the motors is dependent on the joint angles, while the gravitational forces always point in a vertical direction. Decomposing the forces in vertical directions results in the following forces from the rotors:

$$F_f cos(p) + F_b cos(p) = F_s cos(p)$$

Balance of torque gives the following equation for the elevation:

$$\begin{aligned} J_e\ddot{e} &= l_h F_s cos(p) + m_c g l_c cos(e) - (m_f + m_b) l_h g cos(e) \\ &= l_h K_f V_s cos(p) + cos(e) g[m_c l_c - 2m_p l_h] \end{aligned}$$

When the rotors apply the same force, the forces can be decomposed in horizontal direction to find the equation for the travel angle. It is important to note how the torque varies with the elevation angle of the helicopter. The negative sign is due to how positive travel angle and pitch angle is defined in fig. 7 [2, p. 12], and how these affect each other.

$$\begin{aligned} J_\lambda\ddot{\lambda} &= -F_s sin(p) l_h cos(e) \\ &= -K_f V_s sin(p) l_h cos(e) \end{aligned}$$

Summarized the system can be modelled by the following equations:

Equations of motions

$$J_p\ddot{p} = K_f \cdot V_d \cdot l_p \tag{3a}$$
$$J_e\ddot{e} = l_h K_f V_s cos(p) + cos(e) g[m_c l_c - 2m_p l_h] \tag{3b}$$
$$J_\lambda\ddot{\lambda} = -K_f V_s sin(p) l_h cos(e) \tag{3c}$$

By comparing 2a-2c with 3a-3c, the values of the constants L_1, L_2, L_3 and L_4 are determined:

$$L_1 = K_f \cdot l_p \tag{4a}$$
$$L_2 = g \cdot (m_c l_c - 2m_p l_h) \tag{4b}$$
$$L_3 = K_f \cdot l_h \tag{4c}$$
$$L_4 = -K_f \cdot l_h \tag{4d}$$

1.2 Problem 2

In the second problem we want to linearize the equations of motions around the point (p, e, $\lambda)^T = (p^*, e^*, \lambda^*)^T$, with $p^* = e^* = \lambda^* = 0$. In order to do so we need to find V_s^* and V_d^* such that $(p^*, e^*, \lambda^*)^T$ is a equilibrium point of the system.

Finding V_s^* and V_d^*

Using equation 2a, 2b and the assumption that $\dot{p} = \dot{e} = \dot{\lambda} = 0$
$\Rightarrow \ddot{p} = \ddot{e} = \ddot{\lambda} = 0$. The speed is then 0.

$$J_p \cdot \dot{p}^* = J_p \cdot 0 = L_1 \cdot V_d^*$$
$$\Rightarrow V_d^* = 0$$

$$J_e \cdot \dot{e}^* = 0 = L_2 \underbrace{cos(e^*)}_{=1} + L_3 V_s^* \underbrace{cos(p^*)}_{=1}$$
$$\Rightarrow V_s^* = -\frac{L_2}{L_3} = -\frac{g(m_c l_c - 2m_p l_h)}{K_f l_h}$$

In order to linearize the system around the working point, equation (4) from the project description [2, p. 14] and the calculated values for V_s^* and V_s^* are combined.

$$\begin{bmatrix} p \\ e \\ \lambda \end{bmatrix} = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} + \underbrace{\begin{bmatrix} p^* \\ e^* \\ \lambda^* \end{bmatrix}}_{=0} \tag{5}$$

$$\begin{bmatrix} V_s \\ V_d \end{bmatrix} = \begin{bmatrix} \tilde{V_s} \\ \tilde{V_d} \end{bmatrix} + \begin{bmatrix} -\frac{L_2}{L_3} \\ 0 \end{bmatrix} \tag{6}$$

Furthermore equation 5 and 6 is inserted into the equations in 2. This gives:

$$\begin{aligned} J_p(\ddot{\tilde{p}} + 0) &= L_1(\tilde{V_d} + 0) \\ J_e(\ddot{\tilde{e}} + 0) &= L_2 cos(\tilde{e} + 0) + L_3(\tilde{V_s} - \frac{L_2}{L_3})cos(\tilde{p} + 0) \\ J_\lambda(\ddot{\tilde{\lambda}} + 0) &= L_4(\tilde{V_s} - \frac{L_2}{L_3})cos(\tilde{e} + 0)sin(\tilde{p} + 0) \end{aligned}$$

$$\Downarrow$$

$$\begin{aligned} \ddot{\tilde{p}} &= \frac{L_1 \tilde{V_d}}{J_p} \\ \ddot{\tilde{e}} &= \frac{L_2 cos(\tilde{e}) + L_3(\tilde{V_s} - \frac{L_2}{L_3})cos(\tilde{p})}{J_e} \\ \ddot{\tilde{\lambda}} &= \frac{L_4(\tilde{V_s} - \frac{L_2}{L_3})cos(\tilde{e})sin(\tilde{p})}{J_\lambda} \end{aligned}$$

Using these values the Jacobians can be calculated:

$$A = \begin{bmatrix} \frac{\partial \dot{\tilde{p}}}{\partial \tilde{p}} & \frac{\partial \dot{\tilde{p}}}{\partial \dot{\tilde{p}}} & \frac{\partial \dot{\tilde{p}}}{\partial \tilde{e}} & \frac{\partial \dot{\tilde{p}}}{\partial \dot{\tilde{e}}} & \frac{\partial \dot{\tilde{p}}}{\partial \tilde{\lambda}} & \frac{\partial \dot{\tilde{p}}}{\partial \dot{\tilde{\lambda}}} \\ \frac{\partial \ddot{\tilde{p}}}{\partial \tilde{p}} & \frac{\partial \ddot{\tilde{p}}}{\partial \dot{\tilde{p}}} & \frac{\partial \ddot{\tilde{p}}}{\partial \tilde{e}} & \frac{\partial \ddot{\tilde{p}}}{\partial \dot{\tilde{e}}} & \frac{\partial \ddot{\tilde{p}}}{\partial \tilde{\lambda}} & \frac{\partial \ddot{\tilde{p}}}{\partial \dot{\tilde{\lambda}}} \\ \frac{\partial \dot{\tilde{e}}}{\partial \tilde{p}} & \frac{\partial \dot{\tilde{e}}}{\partial \dot{\tilde{p}}} & \frac{\partial \dot{\tilde{e}}}{\partial \tilde{e}} & \frac{\partial \dot{\tilde{e}}}{\partial \dot{\tilde{e}}} & \frac{\partial \dot{\tilde{e}}}{\partial \tilde{\lambda}} & \frac{\partial \dot{\tilde{e}}}{\partial \dot{\tilde{\lambda}}} \\ \frac{\partial \ddot{\tilde{e}}}{\partial \tilde{p}} & \frac{\partial \ddot{\tilde{e}}}{\partial \dot{\tilde{p}}} & \frac{\partial \ddot{\tilde{e}}}{\partial \tilde{e}} & \frac{\partial \ddot{\tilde{e}}}{\partial \dot{\tilde{e}}} & \frac{\partial \ddot{\tilde{e}}}{\partial \tilde{\lambda}} & \frac{\partial \ddot{\tilde{e}}}{\partial \dot{\tilde{\lambda}}} \\ \frac{\partial \dot{\tilde{\lambda}}}{\partial \tilde{p}} & \frac{\partial \dot{\tilde{\lambda}}}{\partial \dot{\tilde{p}}} & \frac{\partial \dot{\tilde{\lambda}}}{\partial \tilde{e}} & \frac{\partial \dot{\tilde{\lambda}}}{\partial \dot{\tilde{e}}} & \frac{\partial \dot{\tilde{\lambda}}}{\partial \tilde{\lambda}} & \frac{\partial \dot{\tilde{\lambda}}}{\partial \dot{\tilde{\lambda}}} \\ \frac{\partial \ddot{\tilde{\lambda}}}{\partial \tilde{p}} & \frac{\partial \ddot{\tilde{\lambda}}}{\partial \dot{\tilde{p}}} & \frac{\partial \ddot{\tilde{\lambda}}}{\partial \tilde{e}} & \frac{\partial \ddot{\tilde{\lambda}}}{\partial \dot{\tilde{e}}} & \frac{\partial \ddot{\tilde{\lambda}}}{\partial \tilde{\lambda}} & \frac{\partial \ddot{\tilde{\lambda}}}{\partial \dot{\tilde{\lambda}}} \end{bmatrix} \tag{7}$$

Inserting values into matrix 7 gives:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{-L_4 \cdot L_2}{J_\lambda \cdot L_3} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{8}$$

Similarly the B matrix can be calculated.

$$B = \begin{bmatrix} \frac{\partial \dot{\tilde{p}}}{\partial \tilde{V}_d} & \frac{\partial \dot{\tilde{p}}}{\partial \tilde{V}_s} \\ \frac{\partial \ddot{\tilde{p}}}{\partial \tilde{V}_d} & \frac{\partial \ddot{\tilde{p}}}{\partial \tilde{V}_s} \\ \frac{\partial \dot{\tilde{e}}}{\partial \tilde{V}_d} & \frac{\partial \dot{\tilde{e}}}{\partial \tilde{V}_s} \\ \frac{\partial \ddot{\tilde{e}}}{\partial \tilde{V}_d} & \frac{\partial \ddot{\tilde{e}}}{\partial \tilde{V}_s} \\ \frac{\partial \dot{\tilde{\lambda}}}{\partial \tilde{V}_d} & \frac{\partial \dot{\tilde{\lambda}}}{\partial \tilde{V}_s} \\ \frac{\partial \ddot{\tilde{\lambda}}}{\partial \tilde{V}_d} & \frac{\partial \ddot{\tilde{\lambda}}}{\partial \tilde{V}_s} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \frac{L_1}{J_p} & 0 \\ 0 & 0 \\ 0 & \frac{L_3}{J_e} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{9}$$

In total the linearized system is given as:

$$\begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \dot{\tilde{e}} \\ \ddot{\tilde{e}} \\ \dot{\tilde{\lambda}} \\ \ddot{\tilde{\lambda}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{-L_4 \cdot L_2}{J_\lambda \cdot L_3} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{L_1}{J_p} & 0 \\ 0 & 0 \\ 0 & \frac{L_3}{J_e} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_d \\ \tilde{V}_s \end{bmatrix} \tag{10}$$

To sum up the linearization, the equations can be written on a more general form:

$$\ddot{\tilde{p}} = K_1 \tilde{V}_d \tag{11a}$$

$$\ddot{\tilde{e}} = K_2 \tilde{V}_s \tag{11b}$$

$$\ddot{\tilde{\lambda}} = K_3 \tilde{p} \tag{11c}$$

Finding the coefficients K_1 K_2 K_3

Combining equation 10 and equation 11 gives the coefficients:

$$K_1 = \frac{L_1}{J_p} \tag{12a}$$

$$K_2 = \frac{L_3}{J_e} \tag{12b}$$

$$K_3 = \frac{L_2}{J_\lambda} \tag{12c}$$

Inserting the calculated coefficients L_1, L_2 and L_3 as well as the given coefficients J_p, J_e and J_λ [2, equation 5, p. 14] gives:

$$K_1 = \frac{K_f}{2m_p l_p} \tag{13a}$$

$$K_2 = \frac{K_f l_h}{m_c l_c^2 + 2m_p l_h^2} \tag{13b}$$

$$K_3 = \frac{g(m_c l_c - 2m_p l_h)}{m_c l_c^2 + 2m_p(l_h^2 + l_p^2)} \tag{13c}$$

1.3 Problem 3

By investigating the helicopter, it is seen that the back motor is heavier than the front motor, $m_b > m_f$. This will cause the gravitational force to act different on the two masses, which is not taken into account in the modelling in 3a - 3c

According to 11c the acceleration of the travel angle is proportional to the pitch angle. This is only true when the rotors actually apply forces, otherwise the model is not valid.

It is assumed that the force applied by the rotors are proportional to voltage supplied to the motors, $F_{rotor} \propto V$. The accuracy or inaccuracy of this statement was difficult to measure. Friction in the joints are also neglected in the models and may be taken into account to explain discrepancies from the physical behavior.

1.4 Problem 4

To adjust the working point of the linearized model to the point where the pitch is zero when the head is horizontal and with the elevation equal to zero when the arm between the elevation axis and the helicopter head is horizontal, constants where added to the output of the encoder to give the appropriate angles. Furthermore, the value for V_s is determined by flying the helicopter up to the newly defined 0-value for the elevation angle. By approximating this value on a Simulink-scope it is found to be as displayed below.

From this value the motor force constant is determined. This way of finding V_s leaves a lot of uncertainty, since the helicopter is quite uncontrollable without any controllers, and oscillates a lot around $\tilde{e} = 0$.

$$\begin{aligned}
e_{offset} &= 29\,\mathrm{deg} \\
p_{offset} &= 0.527\,\mathrm{deg} \\
V_s &= 6.5\,\mathrm{V}
\end{aligned}$$

Determining the motor force constant K_f

$$\begin{aligned}
V_s &= -\frac{L_2}{L_3} = -\frac{g(m_c l_c - 2m_p l_h)}{K_f l_h} \\
\Rightarrow K_f &= -\frac{g(m_c l_c - 2m_p l_h)}{V_s l_h} \\
&= \frac{-9.81(1.92 \cdot 0.46 - 2 \cdot 0.72 \cdot 0.66)}{6.5 \cdot 0.66} \\
&= 0.1537\,\mathrm{N\,V^{-1}}
\end{aligned}$$

This gives the values:

$$\begin{aligned}
K_1 &= 0.6098 \\
K_2 &= 0.0981 \\
K_3 &= -0.6117
\end{aligned}$$

2 Part II - Monovariable control

2.1 Problem 1

In this problem a PD controller is implemented using the equation:

$$\tilde{V}_d = K_{pp}(\tilde{p}_c - \tilde{p}) - K_{pd}\dot{\tilde{p}} \tag{14}$$

By replacing the expression in equation 11a with 14 we get:

$$\ddot{\tilde{p}} = K_1(K_{pp}(\tilde{p}_c - \tilde{p}) - K_{pd}\dot{\tilde{p}})$$
$$\ddot{\tilde{p}} + K_1K_{pd}\dot{\tilde{p}} + K_1K_{pp}\tilde{p} = K_1K_{pp}\tilde{p}_c$$

Using the laplace transform assuming $\tilde{p}(0) = 0$

$$s^2\tilde{p}(s) + sK_1K_{pd}\tilde{p}(s) + K_1K_{pp}\tilde{p}(s) = K_1K_{pp}\tilde{p}_c(s)$$
$$\tilde{p}(s)(s^2 + sK_1K_{pd} + K_1K_{pp}) = K_1K_{pp}\tilde{p}_c(s)$$

Solving this gives us the transfer function:

$$\frac{\tilde{p}(s)}{\tilde{p}_c(s)} = \frac{K_1K_{pp}}{s^2 + K_1K_{pd}s + K_1K_{pp}} \tag{15}$$

The linearized pitch dynamics can be regarded as a second-order linearized system, given by the transfer function:

$$h(s) = \frac{K\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{16}$$

To find appropriate constants Kpp and Kpd, the transfer function obtained above, 15, is compared with the general expression for a second order transfer function, 16. This gives that:

$$\omega_0 = \sqrt{K_1K_{pp}} \tag{17}$$
$$2\zeta\omega_0 = K_{pd}K_1 \tag{18}$$

ζ and ω_0 is chosen to get desired behavior of the pitch in the system, and K_{pd} and K_{pp} is then given by equation 17 and 18. The controller should control the pitch rapidly, but without excessive oscillations. The undamped resonance frequency ω_0 [rad/s] decides how fast the pitch regulation will be. By investigating the physical model of the helicopter, it is decided that $\omega_0 = \pi$ could be a nice initial guess. This is due to the fact that higher resonance frequency may damage the helicopter, or make it unstable. A large value of ζ will give a damped oscillation, while a low value of ζ results in no damping at all. For this reason, we choose $\zeta = 1$, which gives critical damping. The system will then return to equilibrium without overshooting. When K_{pp} and K_{pd} is increased, ω_0 increases too. This moves the poles further into the left half plane, and the helicopter will respond to pitch changes faster. Also, a smaller ζ increases the imaginary part of the poles, which results in less damping.

By investigating the Bode plot of the open-loop system, the stability of the closed loop system can be determined. A stable system requires that the the frequency in which the amplitude is equal to 0 dB, is lower than the frequency where the phase angle is equal to -180 degrees, $\omega_0 < \omega_{180}$. As seen from the bode plot of the open-loop system, the gain margin is infinite. This means that K_{pp}, and thus also ω_0, can be chosen as big as we want, and the system will still remain stable in theory. When a higher resonance frequency was tested, it was observed that the motors turned on and off continuously, which may damage the motors.

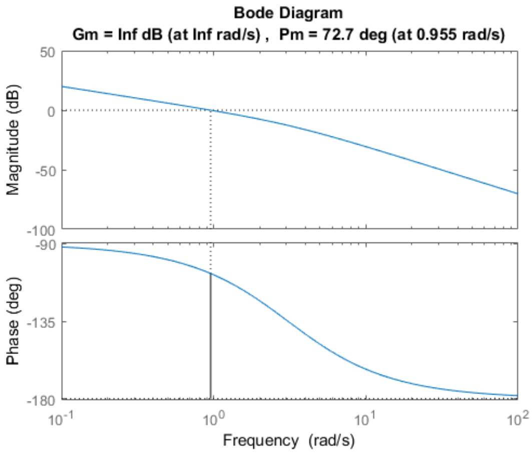


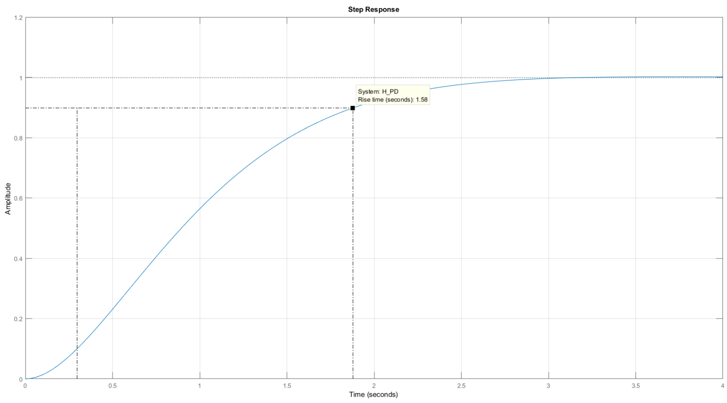
Figure 2: Frequency response of pitch controller

By investigating the step response of the transfer function 15, ζ and ω_0 was tuned to give a fast and accurate response; that is to reach the unity value as fast as possible, without overshooting. If ω_0 is set too low or too high, a static error may occur. The goal is for the step response to go from zero to one as fast as possible without overshooting. By trial and error we find that ω_0 should be equal to 0.558π, and ζ equal to 0.875.

```
%% |-- Task 5.2.1 - PD controller --|
Omega_0 = 0.558*pi;
Zeta = 0.875;
K_pp = (Omega_0)^2/K_1;
K_pd = 2*Zeta*sqrt(K_pp/(K_1));

%Step response of closed loop
s=tf('s');
H_PD = (K_1*K_pd)/(s^2 + K_1*K_pd*s + K_1*K_pp);
step(H_PD)
grid on
```

(a) Matlab code of step response



(b) Step response

Figure 3: Step response of closed loop system

Calculating K_{pp} and K_{pd}

$$K_{pp} = \frac{\omega_0^2}{K_1} \tag{19a}$$

$$K_{pd} = \frac{2\zeta\omega_0}{K_1} \tag{19b}$$

Using these equations along with the obtained values for ω_0 and ζ, K_{pp} and K_{pd} is found to be:

$$K_{pp} = 5.0395$$
$$K_{pd} = 5.0309$$

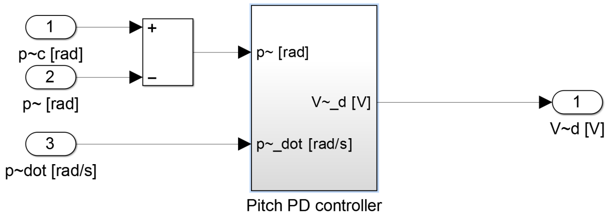


Figure 4: Pitch PD-controller

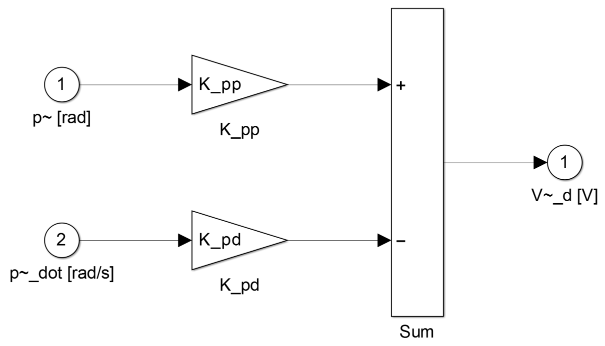


Figure 5: Implementation of PD-controller in Simulink

2.2 Problem 2

In this problem a P-controller is implemented using the equation:

$$\tilde{p}_c = K_{rp}(\dot{\tilde{\lambda}}_c - \dot{\tilde{\lambda}}) \text{ with } K_{rp} < 0 \tag{20}$$

Assuming that $\tilde{p} = \tilde{p}_c$ in order to find the transferfunction given as:

$$\frac{\dot{\tilde{\lambda}}(s)}{\dot{\tilde{\lambda}}_c(s)} = \frac{\rho}{s + \rho} \tag{21}$$

By replacing the expression in equation 11c with 20 we get:

Assuming pitch is faster than travel rate, we can say:

$$\ddot{\tilde{\lambda}} = K_3(K_{rp}(\dot{\tilde{\lambda}}_c - \dot{\tilde{\lambda}}))$$
$$\ddot{\tilde{\lambda}} + K_3K_{rp}\dot{\tilde{\lambda}} = K_3K_{rp}\dot{\tilde{\lambda}}_c$$

Using the laplace transform assuming $\dot{\tilde{p}}(0) = 0$

$$s\dot{\tilde{\lambda}}(s) + K_3K_{rp}\dot{\tilde{\lambda}}(s) = K_3K_{rp}\dot{\tilde{\lambda}}_c(s)$$
$$\dot{\tilde{\lambda}}(s)(s + K_3K_{rp}) = K_3K_{rp}\dot{\tilde{\lambda}}_c(s)$$

Solving this gives us the transfer function:

$$\frac{\dot{\tilde{\lambda}}(s)}{\dot{\tilde{\lambda}}_c(s)} = \frac{K_3K_{rp}}{s + K_3K_{rp}} \tag{22}$$

With $\rho = K_3K_{rp}$ from equation 21

We have an internal feedback in a nested feedback loop. This means that the outer feedback loop is dependent on the inner loop. Therefore it makes sense that the inner loop should at least be twice as fast as the outer loop, so that the "sampling" of the inner loop (which the outer performs) always has updated values. If not, we would have a significant delay in the feedback loop which would make the system sluggish and unresponsive.

Choosing $K_{rp} = -\frac{\omega_0}{2}$ since the inner loop should be at least twice as fast as the outer. Since the inner loop is given by $\omega_0 = 0.558\pi$, K_{rp} is chosen as half.

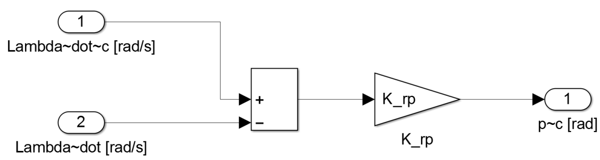


Figure 6: Implementation of travel rate-controller in Simulink

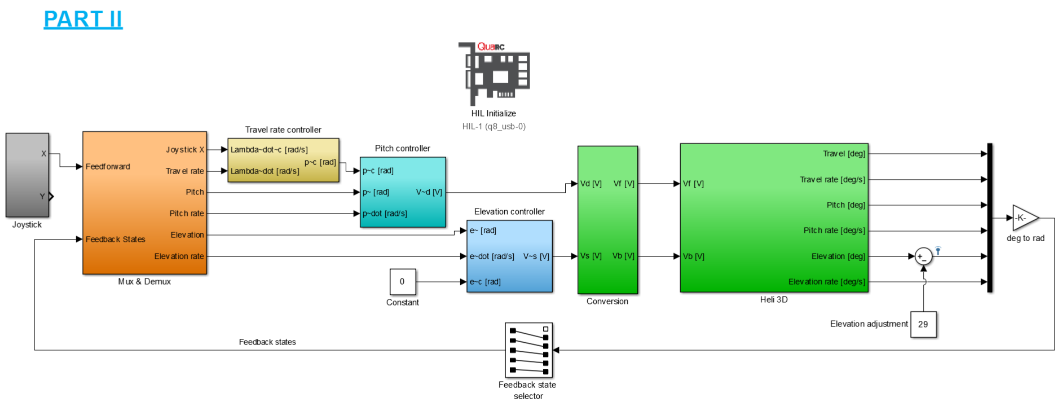


Figure 7: Simulink model after part II

3 Part III - Multivariable control

3.1 Problem 1

We put the system on the form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

where we have:

$$\mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \end{bmatrix} \quad \text{and} \quad \mathbf{u} = \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{23}$$

which gives us the following:

$$\begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \ddot{\tilde{e}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & k_1 \\ k_2 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{24}$$

3.2 Problem 2

We now aim to track the reference $\mathbf{r} = [\tilde{p}_c, \dot{\tilde{e}}_c]^T$ for the pitch angle $\tilde{p}$ and elevation rate $\dot{\tilde{e}}$, which we feed in with the joystick output.

It's useful to examine the controllabillity of our system first:

$$\mathbf{C} = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A^2B} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & K_1 & 0 & 0 \\ 0 & K_1 & 0 & 0 & 0 & 0 \\ K_2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{25}$$

As we can see, the controllability matrix has full rank! After the definition of the controllability matrix [1, p. 185] this means that the system is controllable! Now we can make a controller on the form:

$$\mathbf{u} = \mathbf{Pr} - \mathbf{Kx} \tag{26}$$

Where the matrix $\mathbf{K}$ corresponds to the linear quadratic regulator (LQR) for which the control input $\mathbf{u} = \textbf{-}\mathbf{Kx}$ optimizes the cost function.

$$J = \int_0^\infty \left(\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)\right)\, \mathrm{d}t \tag{27}$$

The $\mathbf{K}$ matrix is obtained by using the `MATLAB` command `lqr(A,B,Q,R)`. When choosing $\mathbf{P}$, it's desirable to obtain a solution such that $\mathbf{y}(t) \rightarrow \mathbf{r}$ as $t \rightarrow \infty$ so that the correct feedforward gain is chosen. We have:

$$
\begin{aligned}
\mathbf{u} &= \mathbf{Pr} - \mathbf{Kx} \\
\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{B}(\mathbf{Pr} - \mathbf{Kx}) = 0 \\
(\mathbf{A} - \mathbf{BK})\mathbf{x}_\infty &= -\mathbf{BPr} \\
\mathbf{y}_\infty &= [\mathbf{C}(\mathbf{BK} - \mathbf{A})^{-\mathbf{1}}\mathbf{B}]\mathbf{Pr} \\
&\Downarrow \\
\mathbf{P} &= [\mathbf{C}(\mathbf{BK} - \mathbf{A})^{-\mathbf{1}}\mathbf{B}]^{-\mathbf{1}}
\end{aligned} \tag{28}
$$

For simplicity, we set $\mathbf{Q}$ and $\mathbf{R}$ diagonal. From what we gathered, Bryson's Rule [3] seemed to be an ineffective way to tune the controller. So we decided to set the diagonal elements of $\mathbf{Q}$ and $\mathbf{R}$ equal to 1 as an initial guess. From there, we tuned pitch and pitch rate first, before tuning elevation. These are the $\mathbf{Q}$ and $\mathbf{R}$ we ended up with:

$$
\mathbf{Q} = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 30 & 0 \\ 0 & 0 & 100 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 0.1 & 0 \\ 0 & 1 \end{bmatrix} \tag{29}
$$

This gives us the following $\mathbf{K}$ and $\mathbf{P}$ matrices:

$$
\mathbf{K} = \begin{bmatrix} 0 & 0 & 31.6228 \\ 10.00 & 7.9245 & 0 \end{bmatrix}, \mathbf{P} = \begin{bmatrix} 0 & 31.6228 \\ 10.00 & 0 \end{bmatrix} \tag{30}
$$

It should be noted that the controllers are tuned aggressively, with high values of $\mathbf{Q}$-elements, and low values of the $\mathbf{R}$-elements. This makes the cost for mistakes in the states higher, and also makes input power more available. This aggressive tuning makes the motors vibrate slightly, and the controller adjusts to small changes more often. We found it more important to have a faster and more accurate behaviour, rather than smooth motor operation for this project. If we had been hired to make an LQR-controller for a real-world contractor, we would probably value operating time higher than we have now.

With this tuning, the system behaved desirable. The pitch is fast and accurate without excessive oscillations, and the elevation adjusts pretty well too. However, when we let go of the joystick further away from the linearized area, the helicopter would slowly go back to the equilibrium point. We were able to make the controller resist the adjustment better with a higher cost of mistakes in $\dot{\tilde{\mathbf{e}}}$, but we were not able to resist the change completely. We believe this is because our controller is optimized for our linearized model which is linear around the equilibrium point. It is not possible to maintain zero error for a closed-loop system without integral effect if there are disturbances in the system. Also, we noticed it went to equilibrium faster if we set $\dot{\tilde{\mathbf{e}}} = \mathbf{0}$ above the equilibrium point, which might be due to gravitational disturbance.

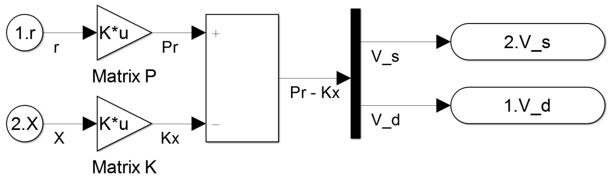


Figure 8: Implementation of LQR in Simulink

3.3 Problem 3

The controller was then modified to include an integral effect for the elevation rate and the pitch angle. This gives two additional states, given by the equations:

$$\dot{\gamma} = \tilde{p} - \tilde{p}_c \tag{31}$$

$$\dot{\zeta} = \dot{\tilde{e}} - \dot{\tilde{e}}_c \tag{32}$$

As discussed in section 3.2, the helicopter will return to the equilibrium point when $\dot{\tilde{\mathbf{e}}} = \mathbf{0}$. Ideally, it should be able to maintain its position, but because of the angles of the forces from the motors, it will not maintain its position outside its linearized area. With integral effect, the helicopter tracks the given reference $\dot{\tilde{\mathbf{e}}}$ much better. This is because the integral effect removes stationary error. At the same time, the integral effect results in two new states, where $\zeta = \tilde{e} - \tilde{e}_c$ will be minimized by the LQR controller. This is the same as trying to make $\tilde{e}$ equal to $\tilde{e}_c$.

It also becomes obvious that that control of elevation becomes better around its linearized area. Here the helicopter will maintain its position perfectly when $\dot{\tilde{e}} = 0$. We combined two step blocks in Simulink on the input of the Y-joystick, so that it generated a 0.5 second pulse. The measured data were saved to file and then plotted in MATLAB. The difference with and without integral effect is clearly shown in fig. 9.

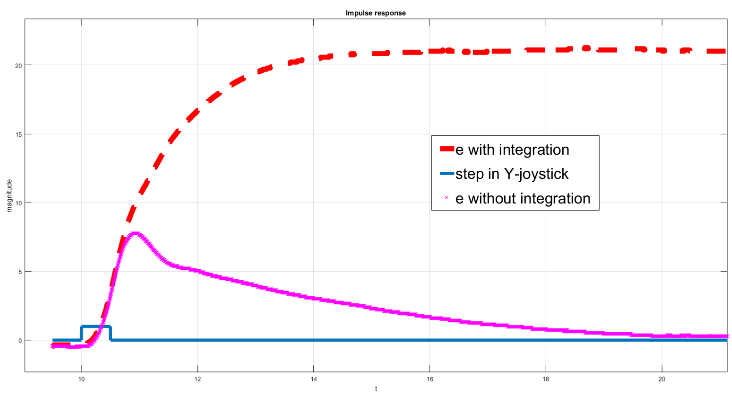


Figure 9: Impulse response of elevation with and without integral effect

After tuning in the same manner as without integral effect, with the previous matrices as an initial guess, the following $\bar{\mathbf{Q}}$, $\bar{\mathbf{R}}$ and $\bar{\mathbf{K}}$ matrices were chosen:

$$\bar{\mathbf{Q}} = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 & 50 \end{bmatrix}, \bar{\mathbf{R}} = \begin{bmatrix} 0.1 & 0 \\ 0 & 1 \end{bmatrix} \tag{33}$$

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 38.154 & 0 & 22.361 \\ 13.294 & 8.579 & 0 & 4.472 & 0 \end{bmatrix} \tag{34}$$

To find the new $\bar{\mathbf{P}}$ matrix, the state-vectors has to be manipulated. This is done by:

$$\mathbf{u} = -\bar{\mathbf{K}}\mathbf{x} + \bar{\mathbf{P}}\mathbf{r} + input\ of\ integral \tag{35}$$

We define the states like this:

$$\mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \end{bmatrix} \tag{36}$$

$$\mathbf{x_a} = \begin{bmatrix} \gamma \\ \zeta \end{bmatrix} \tag{37}$$

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{38}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{39}$$

$$\dot{\mathbf{x}}_\mathbf{a} = \begin{bmatrix} \tilde{p} - \tilde{p_c} \\ \dot{\tilde{e}} - \dot{\tilde{e}}_c \end{bmatrix} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{e}} \end{bmatrix} - \begin{bmatrix} \tilde{p_c} \\ \dot{\tilde{e}}_c \end{bmatrix} \tag{40}$$

Where

$$\mathbf{C}\mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{e}} \end{bmatrix} \tag{41}$$

and

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_\mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_\mathbf{a} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \mathbf{u} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \mathbf{r} \tag{42}$$

Where we define new matrices:

$$\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{C} & 0 \end{bmatrix} \tag{43}$$

and

$$\bar{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \tag{44}$$

Using the new matrices, a new $\bar{\mathbf{K}}$ can be defined.

$$\bar{\mathbf{K}} = lqr(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{Q}}, \bar{\mathbf{R}}) \tag{45}$$

This gives us:

$$u = -\bar{\mathbf{K}} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_\mathbf{a} \end{bmatrix} + \bar{\mathbf{P}}\mathbf{r} \tag{46}$$

The matrix $\bar{\mathbf{K}}$ can be split up in accordance with the new $\mathbf{x}$ vector. This results in two matrices: $\mathbf{K_1}$ and $\mathbf{K_2}$

$$\bar{\mathbf{K}} = \begin{bmatrix} \mathbf{K_1} & \mathbf{K_2} \end{bmatrix} \tag{47}$$

Implementing this in the input from (46) gives:

$$u = -\begin{bmatrix} \mathbf{K_1} & \mathbf{K_2} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_\mathbf{a} \end{bmatrix} + \bar{\mathbf{P}}\mathbf{r} \tag{48}$$

We now have a new representation for the state-space model:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_\mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_\mathbf{a} \end{bmatrix} - \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \begin{bmatrix} \mathbf{K_1} & \mathbf{K_2} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_\mathbf{a} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \bar{\mathbf{P}}\mathbf{r} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \mathbf{r} \tag{49}$$

After simplifying, we get:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_\mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{A} - \mathbf{BK_1} & \mathbf{BK_2} \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x_a} \end{bmatrix} + \begin{bmatrix} \mathbf{B\bar{P}} \\ -1 \end{bmatrix} \mathbf{r} \tag{50}$$

The derivates of the states will be equal to zero in equilibrium, which gives:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} (\mathbf{A} - \mathbf{BK_1})\mathbf{x} - \mathbf{BK_2}\mathbf{x_a} + \mathbf{B\bar{P}r} \\ \mathbf{Cx} - \mathbf{r} \end{bmatrix} \tag{51}$$

As we now can see from column two in (51), any value of $\mathbf{\bar{P}}$ will work, as $\mathbf{Cx} - \mathbf{r}$ will become:

$$\mathbf{Cx} = \mathbf{r} \tag{52}$$

when $t \to \infty$. All values of $\mathbf{\bar{P}}$ will therefore make the output equal to the reference values. However, there exists an optimal $\mathbf{\bar{P}}$ so that the output will equal to the reference values faster. We can derive:

$$0 = (\mathbf{A} - \mathbf{BK_1})\mathbf{x} - \mathbf{BK_2}\mathbf{x_a} + \mathbf{B\bar{P}r} \tag{53}$$

from column one in (51). The integral inputs, $\mathbf{x_a}$, will be equal to zero when the states reach the values of the reference. This gives us:

$$0 = (\mathbf{A} - \mathbf{BK_1})\mathbf{x} + \mathbf{B\bar{P}r} \tag{54}$$

$$\mathbf{x} = -(\mathbf{A} - \mathbf{BK_1})^{-1}\mathbf{B\bar{P}r} \tag{55}$$

If we add $\mathbf{C}$ to both sides, we will get:

$$\mathbf{r} = \mathbf{y} = \mathbf{Cx} = -\mathbf{C}(\mathbf{A} - \mathbf{BK_1})^{-1}\mathbf{B\bar{P}r} \tag{56}$$

when $t \to \infty$. For equation (56) to be true, we see that it is only possible if:

$$\mathbf{\bar{P}} = \left[\mathbf{C}(\mathbf{BK_1} - \mathbf{A})^{-1}\mathbf{B}\right]^{-1} \tag{57}$$

$$\mathbf{\bar{P}} = \begin{bmatrix} 0 & 38.154 \\ 13.294 & 0 \end{bmatrix} \tag{58}$$

We implemented equation (57) in MATLAB like this to compute our **P** Matrix:

```
%% |-- Task 5.3.3 - Integral effect --|
A_PI = [0 1 0 0 0; 0 0 0 0 0; 0 0 0 0 0; 1 0 0 0 0; 0 0 1 0 0];
B_PI = [0 0; 0 K_1; K_2 0; 0 0; 0 0];
C_PI = [1 0 0 0 0; 0 0 1 0 0];
D_PI = 0;
SYS_LQR_I = ss(A_PI, B_PI, C_PI, D_PI, ...
    'StateName',{'p'; 'p_dot' ;'e_dot' ; 'gamma' ; 'zeta'}, ...
    'InputName', {'V_s';'V_d'}, 'OutputName', {'p';'e_dot'});

Q_PI = diag([100 30 100 20 50]);
R_PI = diag([0.1 1]);
K_PI = lqr(A_PI,B_PI,Q_PI,R_PI);

K_P_PI = K_PI(1:2,1:3);
P_PI = inv(C*inv(B*K_P_PI-A)*B);
%
```

Figure 10: MATLAB code for integral effect

This gave us these values of $\bar{\mathbf{P}}$ after tuning:

$$\bar{\mathbf{P}} = \begin{bmatrix} 0 & 38.1541 \\ 13.2942 & 0 \end{bmatrix} \tag{59}$$

And these values of $\bar{\mathbf{K}}$:

$$\bar{\mathbf{K}} = \begin{bmatrix} 0 & 0 & 38.1541 & 0 & 22.3607 \\ 13.2942 & 8.5792 & 0 & 4.4721 & 0 \end{bmatrix} \tag{60}$$

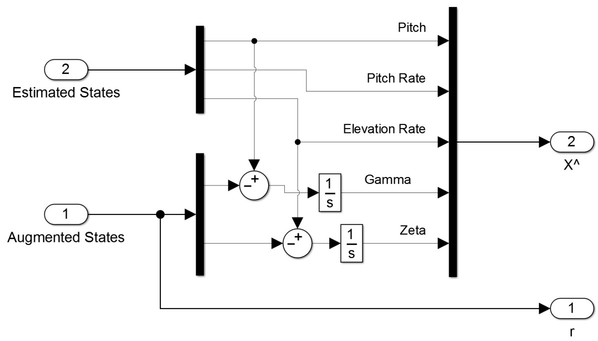


Figure 11: Simulink implementation of integral effect controller

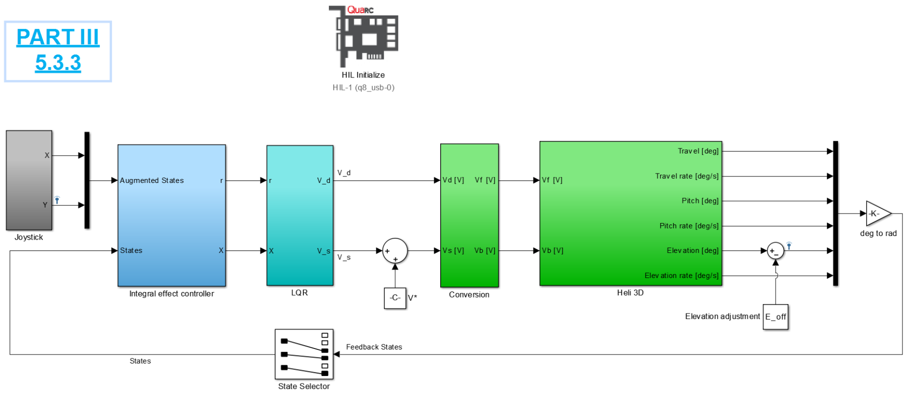


Figure 12: Simulink model after part III

4 Part IV - State estimation

4.1 Problem 1

Deriving a state-space formulation of the system in 11 of the form:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x}\end{aligned}$$

Given:

$$\mathbf{x} = \begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \dot{\tilde{e}} \\ \ddot{\tilde{e}} \\ \dot{\tilde{\lambda}} \\ \ddot{\tilde{\lambda}} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix}$$

Combining these we get the following state-space model:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \dot{\tilde{e}} \\ \ddot{\tilde{e}} \\ \dot{\tilde{\lambda}} \\ \ddot{\tilde{\lambda}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ K_3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ 0 & 0 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{61}$$

$$\mathbf{y} = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} \tag{62}$$

4.2 Problem 2

The observability matrix [1, p. 197] is computed using:

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{63}$$

In this case, only the first two computations are needed to obtain full rank. This gives:

$$\mathcal{O} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{64}$$

The observability matrix has full rank, and the system is indeed observable. The system is described by the following equation:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{L}(\mathbf{y} - \hat{\mathbf{y}}), \quad \hat{\mathbf{y}} = \mathbf{C}\hat{\mathbf{x}} \tag{65}$$

This gives us the error:

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}} \tag{66}$$

Given that our measured $\mathbf{y}$ has noise $\mathbf{n}$ ($\mathbf{y_m} = \mathbf{C}\mathbf{x} + \mathbf{n}$), we can derive the following equation:

$$\dot{\mathbf{e}} = (\mathbf{A} - \mathbf{L}\mathbf{C})\mathbf{e} - \mathbf{L}\mathbf{n} \tag{67}$$

Since the system is observable, it is possible to place the poles of the estimator arbitrarily by choosing an appropriate gain matrix, $\mathbf{L}$. The poles of the estimator is the same as the poles of $\mathbf{A} - \mathbf{LC}$. As a rule of thumb, the error dynamics should be faster than the plant itself. The system dynamics is given by the poles of $\mathbf{A} - \mathbf{BK}$, and hence the poles of $\mathbf{A} - \mathbf{LC}$ should be further into the left half plane, than the poles of $\mathbf{A} - \mathbf{BK}$. To find the gain matrix, $\mathbf{L}$, that will result in the desired estimator poles, the Matlab function `place` is used. The estimator poles are distributed at a circular arc in the left half plane [1, p. 302]. The radius of the circle is set to a multiple times the distance to the most negative pole of the system itself, $\mathbf{A} - \mathbf{BK}$. The `place` function then finds the gain matrix, $\mathbf{L}$, that gives the desired eigenvalues.

```matlab
%% |-- Oppgave 5.4.2 - Observer --|
Q_L = diag([30 30 100 20 60]);
R_L = diag([1 1]);
K_L = lqr(A_PI,B_PI,Q_L,R_L);

K_P_L = K_L(1:2,1:3);
P_L = inv(C*inv(B*K_P_L-A)*B);

system_poles = eig(A_PI-B_PI*K_L);

r0 = max(abs(system_poles));

fr = 15;
phi = pi/8;
r = r0*fr;

spread = -phi:(phi/(2.5)):phi;

p=-r*exp(1i*spread);

figure(2)
plot(real(system_poles),imag(system_poles),'sb',real(p), ...
    imag(p),'rx');grid on; axis equal

L = transpose(place(transpose(A_L),transpose(C_L),p));
%
```

Figure 13: Matlab code for placing poles of **L**

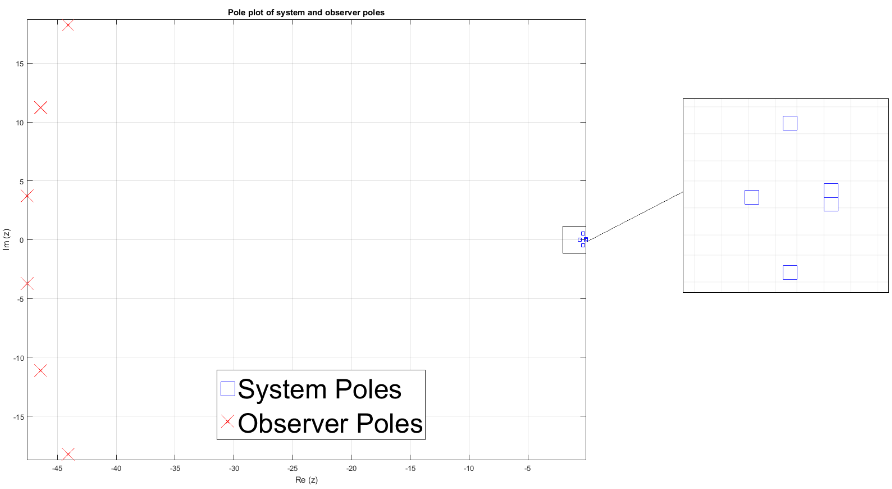


Figure 14: Poles of observer and system

At the same time, the higher values **L** has, the more will noise in measurements be amplified, as seen in equation (67). On the other hand, if **L** has low values, it means that the model and the dynamics of the state estimator contribute more. Disturbances and inaccuracies will then make a greater impact on the results. Lower values of **L** will also result in slower estimators.

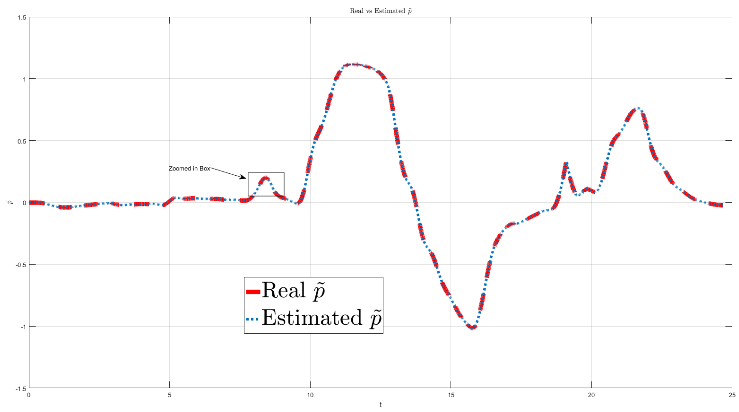
In this project, the closed-loop observer is used to low-pass-filter the measurements. We could easily increase the speed of the estimators to become fast enough to follow the actual measurements more or less perfectly, without suffering of measurement noise. We decided to use estimator poles at a circle of radius 15 times the distance of the left-most pole of the system.

As before, the matrices $\mathbf{Q}$ and $\mathbf{R}$ are tuned until desired behavior is achieved. The following matrices were chosen:

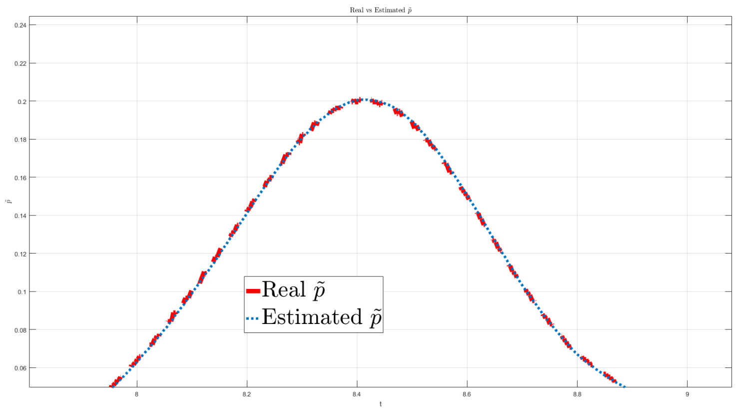
$$\mathbf{Q} = \begin{bmatrix} 30 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 & 60 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{68}$$

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 16.06 & 0 & 7.75 \\ 10.05 & 7.93 & 0 & 4.47 & 0 \end{bmatrix} \tag{69}$$

$$\mathbf{L} = \begin{bmatrix} 0.093 & 0.0051 & -0.013 \\ 2.22 & 0.24 & -0.65 \\ -0.020 & 0.093 & 0.0016 \\ -0.091 & 2.28 & 0.0723 \\ 0.0147 & -0.0020 & 0.0907 \\ 0.7043 & -0.103 & 2.11 \end{bmatrix} * 10^3 \tag{70}$$

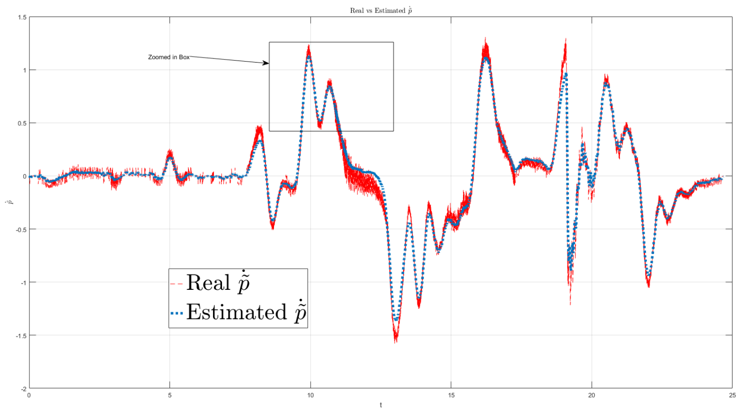


(a) 25 seconds run with observer

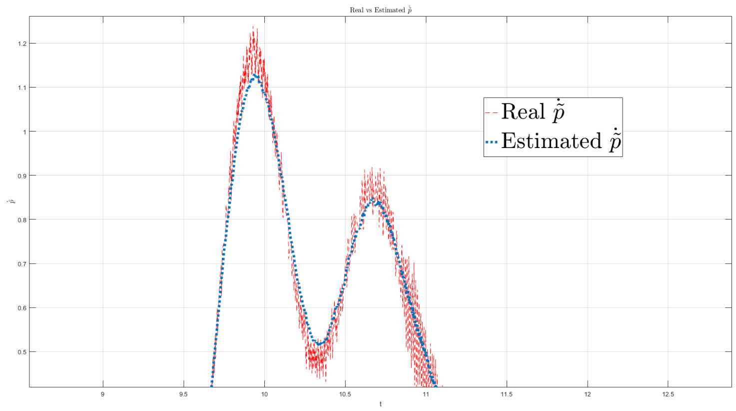


(b) Zoomed-in view

Figure 15: Pitch: Estimated and measured states

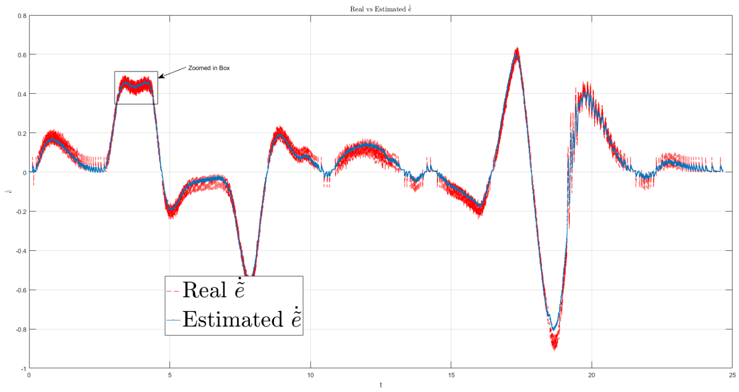


(a) 25 seconds run with observer

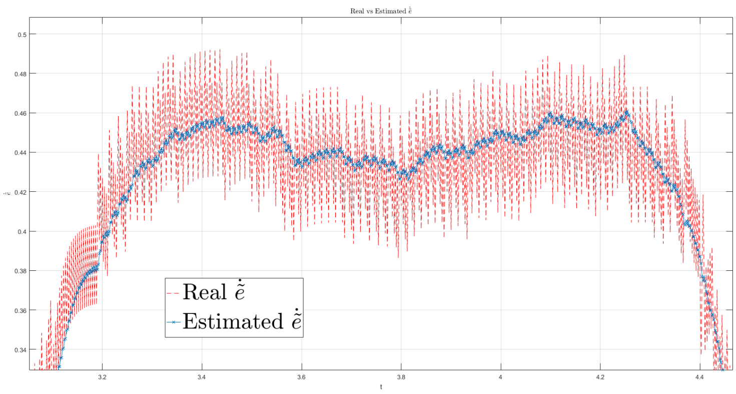


(b) Zoomed-in view

Figure 16: Pitch rate: Estimated and measured states



(a) 25 seconds run with observer



(b) Zoomed-in view

Figure 17: Elevation rate: Estimated and measured states

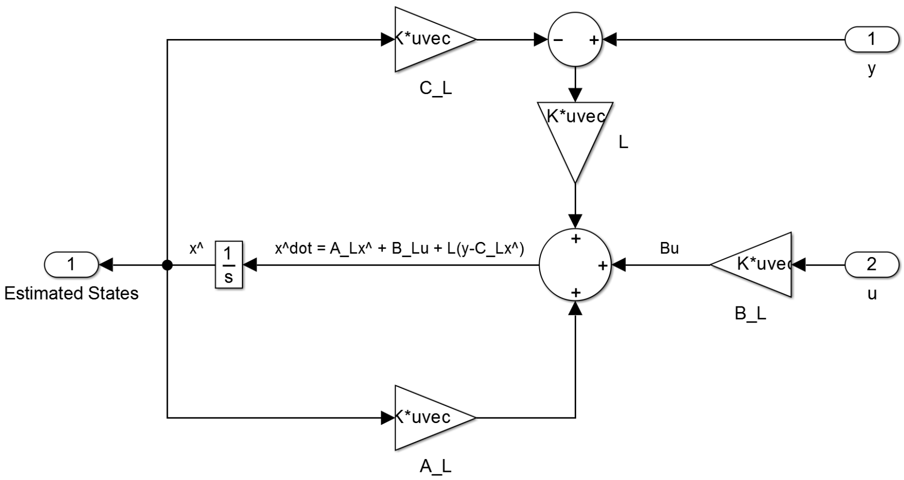


Figure 18: Simulink implementation of observer

4.3 Problem 3

If one measures $\tilde{e}$ and $\tilde{\lambda}$, the C matrix becomes:

$$\mathcal{C} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{71}$$

The observability matrix is then computed by 63:

$$\mathcal{O} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -k_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -k_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{72}$$

This observability matrix has rank equal to 6, which equals full rank. The system is then observable.

If instead the states $\tilde{p}$ and $\tilde{e}$ are measured, the C matrix becomes:

$$\mathcal{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{73}$$

The observability matrix then becomes:

$$\mathcal{O} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{74}$$

This observability matrix has rank equal to 4, which is not full rank. Hence the system is not observable. The physical reason for why it is possible to use the first two states, but not the second pair can be seen from 11c. The pitch, $\tilde{p}$, can be found by differentiating $\tilde{\lambda}$ twice, and multiplying with the constant K_3. On the other hand, if we try to integrate the travel, $\tilde{\lambda}$, information lost during differentiating can not be obtained. That is, integration will result in unknown constants. As a result of this $\tilde{p}$ can not replace $\tilde{\lambda}$ as a measured state.

In general, the more measured states, the better. Even though it should be possible to control the helicopter by measuring only $\tilde{e}$ and $\tilde{\lambda}$, this turned out to be difficult. The estimator of the pitch behaved strange because the pitch angle was not measured. As seen from fig. 19, the estimated pitch rate was especially bad. This is a bad estimator because the travel, $\tilde{\lambda}$, is differentiated three times to get the pitch rate, $\dot{\tilde{p}}$. When this is done, the measurement noise is also differentiated, and differentiating noise is the same as amplifying it. Of this reason, the L matrix is chosen to a lower value in this part of the assignment, but still at least twice as fast as the so the bad measurements will make less impact on the results. The poles were also placed on the real axis, to achieve as high damping as possible. This is because the pitch oscillated heavily.

One problem with this implementation arises when the noisy measurements is used in the integral effect for the elevation rate and the pitch angle. Since the signal from the observer is very noisy and the signal from the joystick is not, the difference between them will be changing fast in equation 31 and 32. This results in a system where the integral effect tries to correct an error that isn't there, and it becomes more unstable. In different test runs, both high and low values of the integral effect was tested. The results showed that a higher value in the Q matrix (higher cost of error) gave a more oscillating system that was hardly stable. The final value for the two last diagonal elements of Q where then chosen to be small, in order to disregard most of the integral effect in the controller.

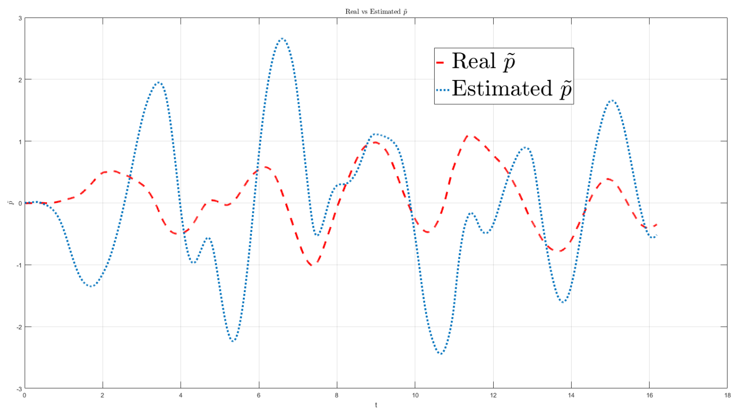


Figure 19: Pitch: Estimated and measured states

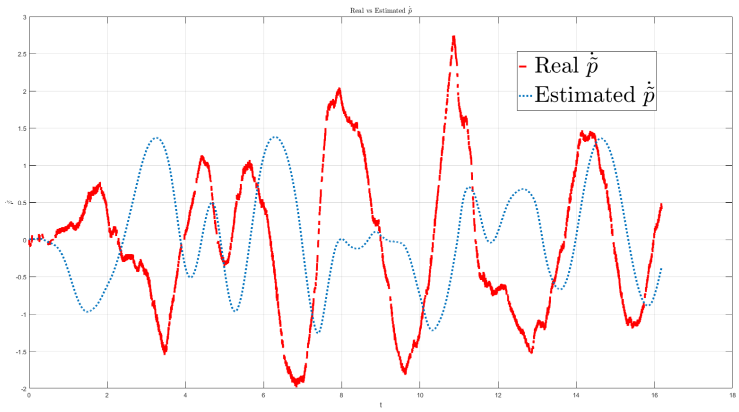


Figure 20: Pitch rate: Estimated and measured states

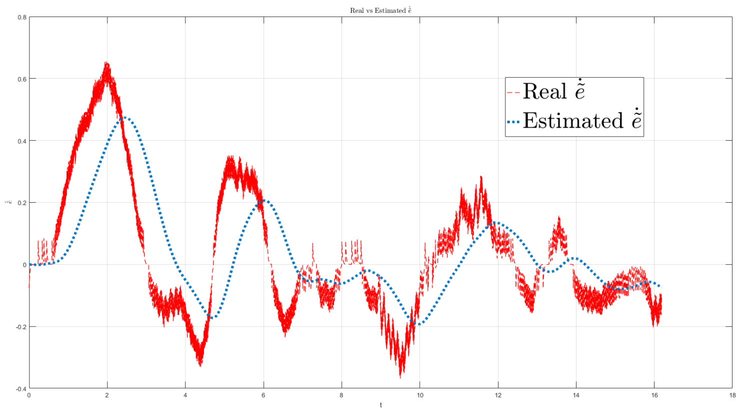


Figure 21: Elevation rate: Estimated and measured states

After tuning of the weighting matrices and the estimators, the closest we came desired behavior was with the following matrices:

$$\mathbf{Q} = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 50 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0.01 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 500 & 0 \\ 0 & 500 \end{bmatrix} \tag{75}$$

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 0.4372 & 0 & 0.0045 \\ 0.1256 & 0.6466 & 0 & 0.0045 & 0 \end{bmatrix} \tag{76}$$

$$
\mathbf{L} = \begin{bmatrix} 25.243 & -103.253 \\ 19.827 & -59.084 \\ 5.994 & -0.014 \\ 8.729 & -0.042 \\ -0.763 & 10.506 \\ -6.110 & 39.560 \end{bmatrix} \tag{77}
$$

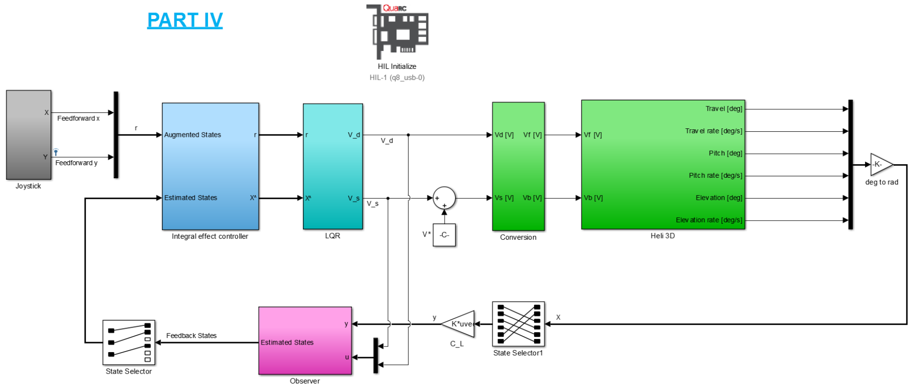


Figure 22: Simulink model after part IV

References

[1] Chi-Tsong Chen. *Linear System Theory and Design*. Oxford University Press, inernational fourth edition, 2013.

[2] Kristoffer Gryte. *TTK4115 Helicopter lab assignment*, volume 4.5. Department of Engineering Cybernetics NTNU, August 2015.

[3] Morten D. Pedersen. *Lecture 5 - State feedback*. Department of Engineering Cybernetics NTNU, 19. September 2016.