TTK4115 Linear System Theory - Helicopter lab

Group 20:
Anette Uttisrud – 759484
Lill Maria Gjerde Johannessen – 759551
Stina Holen Friisø – 759567

October 2016

Contents

1 Part 1 - Mathematical modeling

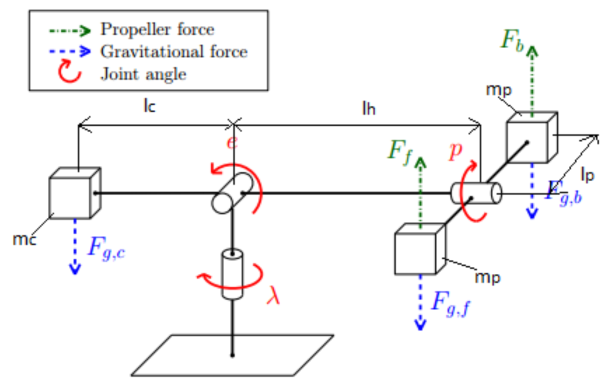


Figure 1: The figure is borrowed from the assignment text, but with added names for the masses and lengths. The figure shows the helicopter in equilibrium.

As shown in the figure, the helicopter has three masses; m_c which is the counterweight, and two m_p which are the two motors connected to the propellers. The helicopter can rotate about three different axis of rotation; the pitch axis, p, which the helicopter head rotates around, the elevation axis, e, and the travel axis λ. Furthermore, the distance between the elevation axis to the counterweight is given as l_c, the distance from the elevation axis to the pitch axis is given as l_h, and the distance from the pitch axis one of the motors is equal for both motors and is given by l_p.

In this lab the only forces taken into account are the gravitational forces $F_{g,c}$, $F_{g,f}$ and $F_{g,b}$, and the propeller forces F_f and F_b, where

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

is assumed. Here V_f and V_b are the voltages supplied to the motors, and K_f is the motor force constant.

1.1 Problem 1

We are to find the equations of motion for the pitch angle p, the elevation angle e, and the travel angle λ. First, we acquire the equation of motion on the desired form by using Newton's second law of rotation.

$$\sum \tau = I\ddot{\theta} \tag{2}$$

This means that the net torque is proportional to the angular acceleration ($\ddot{\theta}$) multiplied by the moment of inertia (I).

For the first equation, the moment of inertia about the pitch axis will be:

$$I_p = F_f \cdot arm - F_b \cdot arm = (V_f - V_b) \cdot K_f \cdot l_p$$

Here, we ignore $F_{g,f}$ and $F_{g,b}$ since the masses are equal to each other and therefore $F_{g,f} - F_{g,b} = 0$ in this direction.

For second equation, the moment of inertia about the elevation axis will be:

$$I_e = -F_{g,c} \cdot arm + F_{g,b} \cdot arm + F_{g,f} \cdot arm + F_f \cdot arm + F_b \cdot arm$$

Finally, for the third equation the moment of inertia about the travel axis will be:

$$I_\lambda = -(F_f \cdot arm + F_b \cdot arm)$$

This gives the following:

$$J_p \cdot \ddot{p} = (V_f - V_b) \cdot K_f \cdot l_p = V_d \cdot K_f \cdot l_p \tag{3a}$$

$$J_e \cdot \ddot{e} = -g \cdot [m_c \cdot l_c - 2m_b \cdot l_h] \cdot cos(e) + K_p \cdot l_h \cdot V_s \cdot cos(p) \tag{3b}$$

$$J_\lambda \cdot \ddot{\lambda} = -V_s \cdot K_f \cdot l_h \cdot cos(e) \cdot sin(p) \tag{3c}$$

Where $V_d = V_f - V_b$, $V_s = V_f + V_b$ and $m_f + m_b = 2m_b$.

Thus resulting in the following expressions for L_1, L_2, L_3 and L_4 :

$L_1 = K_f \cdot l_p$

$L_2 = -g \cdot [m_c \cdot l_c + 2 \cdot m_p \cdot l_h] \cdot cos(e)$

$L_3 = K_f \cdot l_h$

$L_4 = -L_3 = -K_f \cdot l_h$

1.2 Problem 2

We want to design a linear controller for the system, and in order to do this we need to linearize the system mathematically.

Firstly we are going to determine the voltages V_s^* and V_d^* such that $[p^*\ e^*]^T$ is an equilibrium point of the system. This beeing an equilibrium point is equivalent with $\dot{p} = \dot{e} = \dot{\lambda} = 0$ for $[p\ e]^T = [p^*\ e^*]^T$ and $[V_s\ V_d] = [V_s^*\ V_d^*]$

This gives us the following:
$\dot{p} = \dot{e} = \dot{\lambda} = 0 \implies \ddot{p} = \ddot{e} = \ddot{\lambda} = 0$

Using the information given above on equation (3a) - (3c) we get:

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

Which give the following for V_d^* and V_s^*:

$$
\begin{aligned}
V_d^* =& 0 && (4) \\
V_s^* =& -\frac{L_2}{L_3} && (5)
\end{aligned}
$$

The following is given by the exercise:

$$
\begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} = \begin{bmatrix} p \\ e \\ \lambda \end{bmatrix} - \begin{bmatrix} p^* \\ e^* \\ \lambda^* \end{bmatrix}, \begin{bmatrix} \tilde{V_s} \\ \tilde{V_d} \end{bmatrix} = \begin{bmatrix} V_s \\ V_d \end{bmatrix} - \begin{bmatrix} V_s^* \\ V_d^* \end{bmatrix} \text{ a } p^* = e^* = \lambda^* = 0
$$

By applying the coordinate transformation, further analysis is simplified, as the equilibrium point now is situated in the origin.

Therefore, firstly we apply the coordinate transforms:

$$
\begin{aligned}
p &= \tilde{p} + p^* \\
e &= \tilde{e} + e^* \\
\lambda &= \tilde{\lambda} + \lambda^*
\end{aligned}
$$

By substituting the new expressions for p, e and λ in the expression from (3a) - (3c), we get:

$$
\begin{aligned}
\ddot{\tilde{p}} + \ddot{p^*} =& \frac{(\tilde{V}_d + V_d^*)}{J_p} \cdot K_f \cdot l_p \\
\ddot{\tilde{e}} + \ddot{e^*} =& \frac{g}{J_e} \cdot [-m_c \cdot l_c + (m_f + m_p) \cdot l_h] \cdot \cos(\tilde{e} + e^*) - \frac{L_3}{J_e}(\tilde{V}_s + V_s^*) \cdot \cos(\tilde{p} + p^*) \\
\ddot{\tilde{\lambda}} + \ddot{\lambda^*} =& \frac{(\tilde{V}_s + V_s^*)}{J_\lambda} \cdot L_1 \cdot \cos(\tilde{e} + e^*) \cdot \sin(\tilde{p} + p^*)
\end{aligned}
$$

We now put the acquired equations in matrix form:

$$
\begin{bmatrix} \dot{p} \\ \ddot{p} \\ \dot{e} \\ \ddot{e} \\ \dot{\lambda} \\ \ddot{\lambda} \end{bmatrix} = \begin{bmatrix} \dot{p} \\ \frac{(\tilde{V}_d + V_d^*) K_f l_p}{J_p} \\ \dot{e} \\ \frac{g \cos(\tilde{e})(m_c l_c + 2 m_p l_n)}{J_e} - \frac{L_3 \cos(\tilde{p})(\tilde{V}_s + V_s *)}{J_e} \\ \dot{\lambda} \\ \frac{(\tilde{V}_s + V_s *) L_4 \cos(\tilde{e}) \sin(\tilde{p})}{J_\lambda} \end{bmatrix}
$$

Then we apply the state-space representation on the form $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$. Here, we have used the initial condition $V_d^* = 0$.

$$
\begin{bmatrix} \dot{p} \\ \ddot{p} \\ \dot{e} \\ \ddot{e} \\ \dot{\lambda} \\ \ddot{\lambda} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{L_3 V_s^* \sin(\tilde{p})}{J_e} & 0 & \frac{\sin(\tilde{p} L_2)}{J_e} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{(\tilde{V}_s + V_s^*) L_4 \cos(\tilde{e}) \cos(\tilde{p})}{J_\lambda} & 0 & -\frac{(\tilde{V}_s + V_s^*) L_4 \sin(\tilde{p}) \sin(\tilde{e})}{J_\lambda} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{L_1}{J_p} & 0 \\ 0 & 0 \\ 0 & \frac{L_3}{J_e} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}
$$

$$
\begin{bmatrix} \tilde{V}_d \\ \tilde{V}_s \end{bmatrix}
$$

Let $\mathbf{x} = (\tilde{p}, \tilde{e}, \tilde{\lambda})^T$ and $\mathbf{u} = (\tilde{V}_s, \tilde{V}_d)^T$ so that $\mathbf{h}(\mathbf{x}, \mathbf{u}) = (\ddot{\tilde{p}}, \dot{\tilde{p}}, \ddot{\tilde{e}}, \dot{\tilde{e}}, \ddot{\tilde{\lambda}}, \dot{\tilde{\lambda}})^T$. We then linearize about the point $\tilde{\lambda} = \tilde{e} = \tilde{p} = \tilde{V}_s = \tilde{V}_d = 0$ using:

$$
\ddot{\mathbf{x}} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\mathbf{x_0}, \mathbf{u_0})\mathbf{x} + \frac{\partial \mathbf{h}}{\partial \mathbf{u}}(\mathbf{x_0}, \mathbf{u_0})\mathbf{u}
$$

$$
\begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \dot{\tilde{e}} \\ \ddot{\tilde{e}} \\ \dot{\tilde{\lambda}} \\ \ddot{\tilde{\lambda}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{V_s^* K_f l_h}{J_\lambda} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{L_1}{J_p} & 0 \\ 0 & 0 \\ 0 & \frac{L_3}{J_e} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_d \\ \tilde{V}_s \end{bmatrix}
$$

We acquire the following equations from the matrices:

$$
\begin{aligned}
\dot{\tilde{p}} &= \dot{\tilde{p}} && \text{(7a)} \\
\ddot{\tilde{p}} &= \frac{L_1 \tilde{V}_d}{J_p} = K_1 \tilde{V}_d && \text{(7b)} \\
\dot{\tilde{e}} &= \dot{\tilde{e}} && \text{(7c)} \\
\ddot{\tilde{e}} &= \frac{L_3 \tilde{V}_s}{J_e} = K_2 \tilde{V}_s && \text{(7d)} \\
\dot{\tilde{\lambda}} &= \dot{\tilde{\lambda}} && \text{(7e)} \\
\ddot{\tilde{\lambda}} &= \frac{V_s^* L_4 \tilde{p}}{J_\lambda} = \frac{-L_3 L_2 \tilde{p}}{-L_3 J_\lambda} = \frac{L_2 \tilde{p}}{J_\lambda} = K_3 \tilde{p} && \text{(7f)}
\end{aligned}
$$

Where $V_s* = -\frac{L_2}{L_3}$ and

$K_1 = \frac{L_1}{J_p} = \frac{K_f l_p}{2 m_p l_p^2}$,

$K_2 = \frac{L_3}{J_e} = \frac{K_f l_h}{m_c l_c^2 + 2 m_p l_h^2}$,

$K_3 = \frac{L_2}{J_\lambda} = \frac{-g[m_c l_c + 2 m_p l_h] cos(e)}{m_c l_c^2 + 2 m_p (l_h^2 + l_p^2)}$

1.3 Problem 3

The first attempt to control the helicopter will be done using feed forward. The signal from the x-axis of the joystick is connected directly to the voltage difference V_d, and the signal from the y-axis of the joystick is connected directly to the voltage sum V_s. We added a small gain to the output of y-axis of the joystick, because the motors do not need a large voltage difference to increase the pitch angle. We found that 0.8 is a suitable gain. A guesstimate of the input needed to control the helicopter is $V_s^* = 12V$. This seems to be a bit large value of V_s^*. The helicopter is hard to control using only feed forward, and we did not manage to control it in a good way even with different values.

In addition, the theoretical model differs from the actual behaviour of the helicopter due to multiple reasons. First of all, we calculate the behaviour only by taking the masses of the motors and the counter mass into account, leaving out the mass of the rest of the helicopter (such as the pole between the motors and the counterweight). Secondly, we do not take into account that the axis' have limitation; the head of the helicopter can barely turn 180 degrees, and the elevation axis can only go as high as the physical layout of the helicopter will allow. Thirdly, no limitations of the helicopter are taken into account, for instance will the helicopter not be controllable outside its limits when the helicopter head is as high as it gets.

Furthermore, outside influences like worn equipment, slightly uneven weight of motors (our helicopter is by default slightly tilted in the pitch axis), loss in voltage to ground and neglected drag force are all neglected, and of course make our theoretical model differ from the actual behaviour of the helicopter.

Lastly, the linearization of the model further simplifies the model (which already is very simplified).

1.4 Problem 4

The helicopter head has so far rested on the table when the helicopter is started, now we want it to fly straight up to equilibrium when it is started.

From the graphs in Simulink/Matlab, we observe that the helicopter is in equilibrium when the elevation degree is 30°, hence we added that as a constant to get equilibrium to be 0°, and then observed that the helicopter now started in equilibrium when the system ran. We also observe in a scope that the helicopter head is horizontal by adding -4.5 to the pitch angle. Then we added a gain $K = \frac{3,14}{180}$ after the encoder to convert the values from degrees to radians.

Furthermore, we observed the value of V_s^* to be approximately 6.9V when the helicopter was in equilibrium by adding a scope after V_s in Simulink, and used this to calculate K_f:

$$\begin{aligned} V_s^* &= -\frac{L_2}{L_3} = \frac{g \cdot [-m_c \cdot l_c + 2 \cdot m_p \cdot l_h]}{K_f \cdot V_s*} \\ \implies K_f &= \frac{g \cdot [-m_c \cdot l_c + 2 \cdot m_p \cdot l_h]}{l_h \cdot V_s*} = \frac{9,81 \cdot [-1,92 \cdot 0.46 + 2 \cdot 0,72 \cdot 0,06]}{0,66 \cdot 6,9} = 0,145 \end{aligned}$$

2 Part 2 - Monovariable control

2.1 Problem 1

A PD-controller is a controller where the a proportional gain and a derivative term is added to control the system. The proportional gain simply eliminates constant error, while the derivative gain makes the system faster.

A PD-controller is added to control the pitch angle p. This controller is given by

$$\tilde{V_d} = K_{pp}(\tilde{p_c} - \tilde{p}) - K_{pd}\dot{\tilde{p}} \tag{8}$$

with $K_{pp}, K_{pd} > 0$

From (7b) we know that $\ddot{\tilde{p}} = K_1\tilde{V_d}$

We replace $\tilde{V_d}$ with the expression given, and get:

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

We apply laplace transformation to the equation and get the transfer function:

$$\begin{aligned}
s^2\tilde{p}(s) =& K_1(K_{pp}(\tilde{p_c}(s) - \tilde{p}(s) - K_{pd}\tilde{p}(s)s) \\
&\implies \tilde{p}s^2 + K_1K_{pd}\tilde{p}s + K_1K_{pp}\tilde{p} = K_1K_{pp}\tilde{p} \\
&\implies \tilde{p}(s^2 + K_1K_{pd}s + K_1K_{pp}) = K_1K_{pp}\tilde{p_s} \\
\implies \frac{\tilde{p}(s)}{\tilde{p_c}(s)} =& \frac{K_1K_{pp}}{(s^2 + k_1K_{pd}s + K_1K_{pp}})
\end{aligned}$$

Then we used the transfer function to find reasonable values for K_{pp} and K_{pd} by using the formula for a damped system. In this formula we have that ζ is the damping factor and ω_0 is the resonans frequency:

$$h(s) = \frac{K}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{9}$$

Where $\omega_0 = \sqrt{K_1K_{pp}}$ and $\zeta = \frac{1}{2}\frac{K_{pd}}{K_{pp}}\sqrt{K_1K_{pp}}$

Since we want our system to be critically damped, we set $\zeta = 1$, and thus we get:

$\omega_0 = \sqrt{K_1K_{pp}}$ and $1 = \frac{1}{2}\frac{K_{pd}}{K_{pp}}\sqrt{K_1K_{pp}} \implies \omega_0 = 2\frac{K_{pp}}{K_{pd}}$

Further, for tuning of the controller gains K_{pp} and K_{pd} we decided to do this through testing different values of the resonans frequency of the system, ω_0. We plotted everything in Matlab, and tried out different values of ω_0. We discovered that our helicopter was easy to control, and behaved in a desired way when $\omega_0 = 1.5$, which gave the resulting $K_{pd} = 5.2225$ and $K_{pp} = 3.9160$. This gives the ratio between K_{pp} and K_{pd} equal to 0.75, which means that our damping coefficient is relatively high in comparison to our proportional coefficient due to the fact that $\zeta = 1$. Here, we could have tried an overdamped system to see if that would have eliminated external factors more efficiently. This would have made our system slower, but might have eliminated the slight oscillation in the very beginning more efficiently.

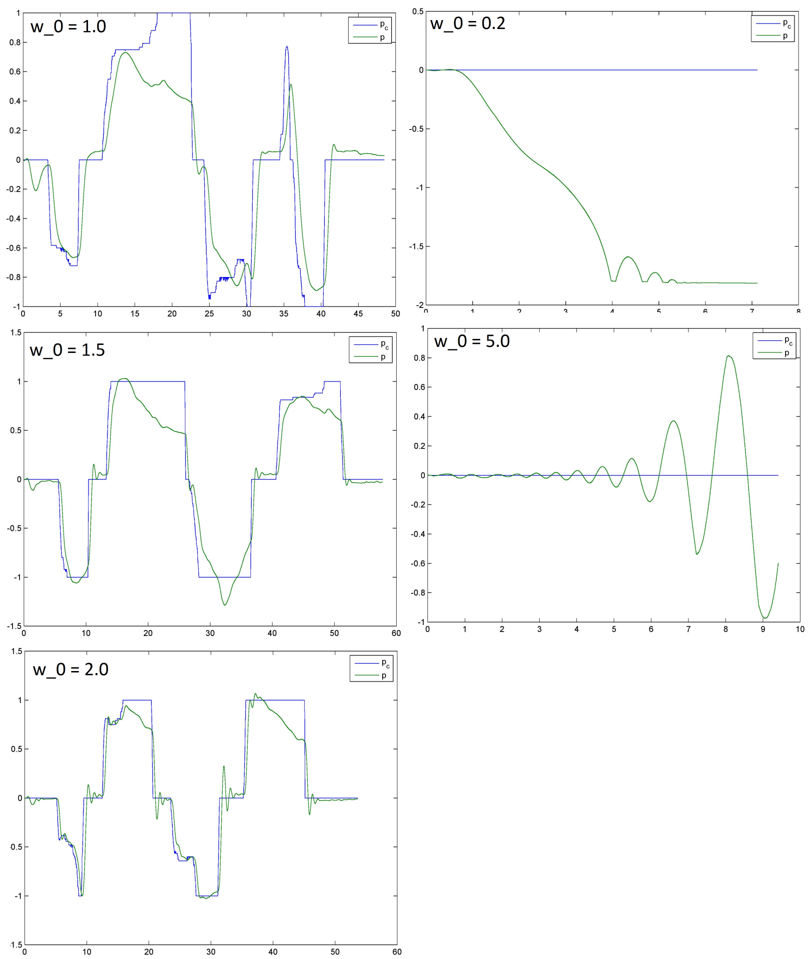


Figure 2: Plots of the pitch $\tilde{p}$, and the pitch reference $\tilde{p_c}$ for different values of ω_0.

We tried out several values for ω_0 before we decided to set $\omega_0 = 1.5$. For a critically damped system, we have that the eigenvalues conhencide on the negative part of the real axis, $\lambda_{1,2} = -\omega_0$. For $\omega_0 > 5$, the eigenvalues are < -5, the system will oscillate with a decreasing amplitude, and do not have a proper behavior. For smaller values closer to zero, as $\omega_0 = 0.2$, the system drifts fast to one side. We found out that a suitable value for ω_0 is somewhere around 1 and 2, with eigenvalues between -1 and -2 . For $\omega_0 = 1$, the system is a bit slow, and for $\omega_0 = 2$ we get some small oscillations with an amplitude that is gradually decreasing to zero when we have a fast change in the reference. Hence, we decided $\omega_0 = 1.5$.

As expected our helicopter was easier to control when the PD-controller was added. This was expected as the derivative in the system has a damping effect and makes the system settle on the reference value in a stable way.

After testing different values of joystick gain, we agreed that the joystick input worked appropriately without a gain.

2.2 Problem 2

The travel rate $\dot{\tilde{\lambda}}$ is to be controlled using a simple P controller:

$$\tilde{p_c} = K_{rp}(\dot{\tilde{\lambda}}_c - \dot{\tilde{\lambda}}) \tag{10}$$

with constant $K_{rp} < 0$

When the pitch angle is controlled perfectly $\tilde{p_c} = \tilde{p}$. From the linearization we have:

$$\tilde{p} = \frac{\ddot{\lambda} J_\lambda}{V_s * L_3}$$
$$\implies \frac{\ddot{\lambda} J_\lambda}{L_2} = K_{rp}(\dot{\tilde{\lambda}}_c - \dot{\tilde{\lambda}})$$

Lpalace transform the expression:

$$\frac{s\dot{\tilde{\lambda}}(s) J_\lambda}{L_2} = K_{rp}(\dot{\tilde{\lambda}}_c(s) - \dot{\tilde{\lambda}}(s))$$

Rearrange the expression and get:

$$\dot{\tilde{\lambda}}(s)(\frac{J_\lambda}{L_2}(s) + K_{rp}) = \tilde{\lambda_c}(s) K_{rp}$$

$$\implies \frac{\dot{\tilde{\lambda}}(s)}{\tilde{\lambda}_c(s)} = \frac{K_{rp}}{\frac{J_\lambda}{L_2}s + K_{rp}} = \frac{\frac{K_{rp}}{J_\lambda}}{s + \frac{K_{rp}L_2}{J_\lambda}}$$

Which gives the following transfer function:

$$\frac{\dot{\tilde{\lambda}}(s)}{\tilde{\lambda}_c(s)} = \frac{\rho}{s + \rho} \tag{11}$$

with $\rho = \frac{K_{rp}L_2}{J_\lambda}$

After testing a while, we concluded that $K_{rp} = -1$ was an appropriate value as the helicopter was fast and accurate at this value. While with a value higher than this, we observed that the helicopter responded very slow. In our case, we tried with $K_{rp} = -0.2$. On the other hand, when we try to decrease the value to -4, the helicopter moved very fast, but that resulted in the helicopter not beeing accurate.

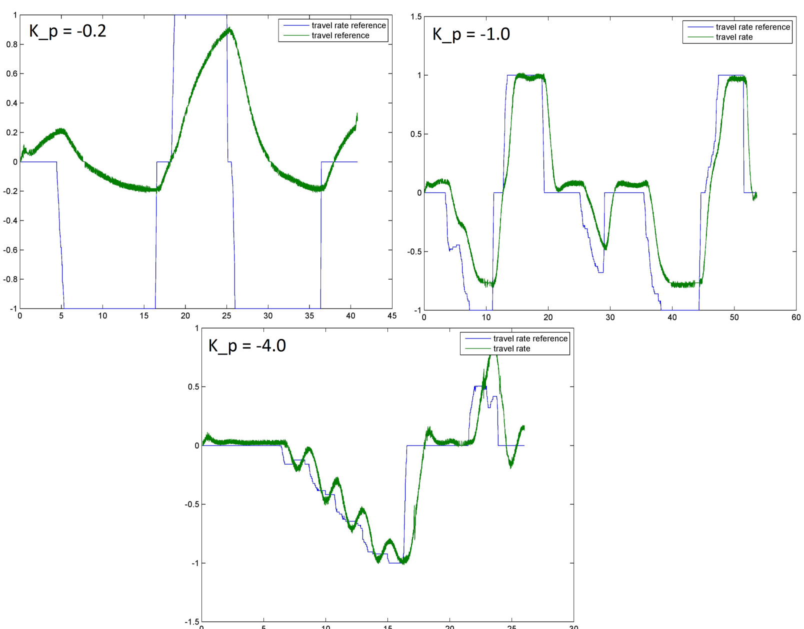


Figure 3: Plots of the travel rate $\dot{\tilde{\lambda}}$ and travel rate reference $\dot{\tilde{\lambda}}_c$ for different values of K_{rp}.

After testing different values of joystick gain, we agreed that the joystick input worked appropriately without a gain.

3 Part 3 - Multivariable control

3.1 Problem 1

We have the system

$$\ddot{\tilde{p}} = K_1 \tilde{V}_d \tag{12a}$$
$$\ddot{\tilde{e}} = K_2 \tilde{V}_s \tag{12b}$$

which we can write in a state-space formulation of the form

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

$$\mathbf{y} = \mathbf{C}\mathbf{x}$$

with the equations

$$\begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \ddot{\tilde{e}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix}$$

$$\begin{bmatrix} \tilde{p} \\ \dot{\tilde{e}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \end{bmatrix}$$

3.2 Problem 2

We have a reference $r = [\tilde{p}_c, \dot{\tilde{e}}_c]^T$, for the pitch angle $\tilde{p}$ and the elevation rate $\dot{\tilde{e}}_c$ given by the joystick output. Firstly, we examine the controllability of the system, by computing the controllability matrix M:

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

We can see that $Rank(\mathbf{M}) = 3 \implies \mathbf{M}$ has full rank, and the system is controllable.

We are looking for a controller of the form $\mathbf{u} = \mathbf{Pr} - \mathbf{Kx}$. The matrix $\mathbf{K}$ corresponds to the linear quadratic regulator (LQR) for which the control input $\mathbf{u} = -\mathbf{Kx}$ optimizes the cost function

$$J = \int_0^\infty (\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t))dt \tag{13}$$

We are only interested in weighting the states and inputs individually, and not the combination of them. Therefore, we choose $\mathbf{Q}$, which are weighting the states, and $\mathbf{R}$, which are weighting the inputs, to be diagonal. We choose to set $\mathbf{R} = I$, such that we are only weighting $\mathbf{Q}$, where q_{11} punishes the pitch error, q_{22} is damping the pitch rate and q_{33} is damping the elevation rate.

$$\mathbf{Q} = \begin{bmatrix} q_{11} & 0 & 0 \\ 0 & q_{22} & 0 \\ 0 & 0 & q_{33} \end{bmatrix} \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

We obtain the corresponding matrix $\mathbf{K}$ by using the MATLAB lqr-command, as shown in figure 4.

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 3 \\ 4 & 4 & 0 \end{bmatrix}$$

The $\mathbf{K}$ matrix weights $\dot{\tilde{e}}$ for $\tilde{V_s}$ and $\tilde{p}$ and $\dot{\tilde{p}}$ for $\tilde{V_d}$, which makes sense.

Then we choose the matrix $\mathbf{P}$ such that $\lim_{t\to\infty} \tilde{p}(t) = \tilde{p_c}$ and $\lim_{t\to\infty} \tilde{e}(t) = \tilde{e_c}$ for fixed values of $\tilde{p_c}$ and $\tilde{e_c}$, so that $\mathbf{P}$ is weighted in such a way that $\tilde{p_c}$ is only codependent on V_d and $\dot{\tilde{e_c}}$ is only codependent V_s.
From the given equations

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

$$\mathbf{u} = \mathbf{P}\mathbf{r} - \mathbf{K}\mathbf{x}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x}$$

which gives

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{P}\mathbf{r} - \mathbf{B}\mathbf{K}\mathbf{x}$$

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} + \mathbf{B}\mathbf{P}\mathbf{r}$$

we obtain $\mathbf{P}$ by setting $\dot{\mathbf{x}} = \mathbf{0}$ and $\mathbf{y_{eq}} = \mathbf{r} = \mathbf{C}\mathbf{x} =$ and get:

$$\mathbf{P} = (\mathbf{C}[\mathbf{B}\mathbf{K} - \mathbf{A}]^{-\mathbf{1}}\mathbf{B})^{-\mathbf{1}}$$

```
%Part 3 problem 2

A3_2 = [0 1 0; 0 0 0; 0 0 0];
B3_2 = [0 0; 0 k_1; k_2 0];
C3_2 = [1 0 0; 0 0 1];
Q3_2 = [20 0 0; 0 2 0; 0 0 10]; %Q11 - pitch, Q22 - damping/pitch rate, Q33 - damping/elevation rate
R = [1 0; 0 1];
K3_2 = lqr(A3_2,B3_2,Q3_2,R);
P3_2 = inv(C3_2*inv(B3_2*K3_2 -A3_2)*B3_2);
pol = eig(A3_2 - B3_2*K3_2);
```

Figure 4: MATLAB code for Problem 3.2.

As shown in figure 4, we ended up with the **Q** matrix:

$$\mathbf{Q} = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 10 \end{bmatrix}$$

which means that we have a proportional gain on the pitch $\tilde{p}$ which is ten times larger than the damping constant on the pitch rate $\dot{\tilde{p}}$, and a damping constant on the elevation rate $\dot{\tilde{e}}$ that is half of the proportional gain on the pitch $\tilde{p}$.

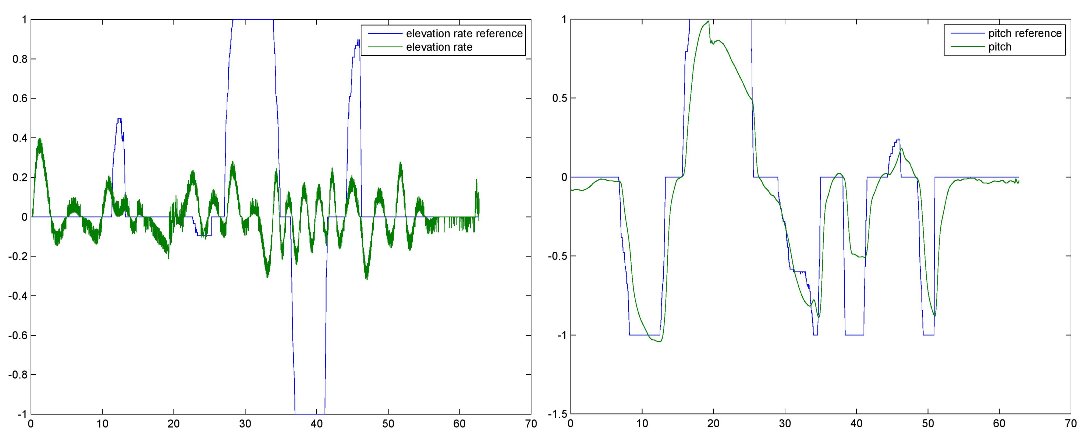


Figure 5: Plots for the elevation rate $\dot{\tilde{e}}$ and the elevation rate reference $\dot{\tilde{e}_c}$ and the pitch $\tilde{p}$ and the pitch reference $\tilde{p_c}$ for the chosen **Q** matrix in problem 3.2.

As we see in the plots in Figure 5, this gave us a quite good regulation of the helicopter. We could have scaled down the output from the joystick on the y-axis.

The helicopter does not stay up/down when we let go of the joystick, as we only have a PD-controller. Thus, the error is not integrated, and the helicopter then seeks to equilibrium as it is the given offset. We then anticipate that when adding the integral effect, the helicopter will stay up when moving it up with the joystick.

3.3 Problem 3

We modify the controller $\mathbf{u} = \mathbf{Pr} - \mathbf{Kx}$ to include an integral effect for the pitch angle and the elevation rate. This results in two additional states γ and ζ, for which the differential equations are given by

$$\dot{\gamma} = \tilde{p} - \tilde{p_c} \tag{14a}$$

$$\dot{\zeta} = \dot{\tilde{e}} - \dot{\tilde{e_c}} \tag{14b}$$

From earlier problems we have that

$$\ddot{\tilde{p}} = K_1 \tilde{V_d}$$
$$\ddot{\tilde{e}} = K_2 \tilde{V_s}$$
$$\ddot{\tilde{\lambda}} = K_3 \tilde{p}$$

We obtain the system with the additional states, given on state-space formulation of the form $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} + \mathbf{Fr}$, from the given equations:

$$\begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \ddot{\tilde{e}} \\ \dot{\gamma} \\ \dot{\zeta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \\ \gamma \\ \zeta \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V_s} \\ \tilde{V_d} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \tilde{p_c} \\ \tilde{e_c} \end{bmatrix}$$

This satisfies:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x_a}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x_a} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \mathbf{u} \text{ - } \begin{bmatrix} 0 \\ I \end{bmatrix} \mathbf{r}$$

which is reasonable, since the augmented part of the system is derived from the outputs of the system, $\mathbf{y} = \mathbf{Cx} = [\tilde{p}, \dot{\tilde{e}}]$ and the reference $r = [\tilde{p_c}, \dot{\tilde{e_c}}]^T$.

```
%Part 3 problem 3

Q3_3 = [5 0 0 0 0; 0 10 0 0 0; 0 0 20 0 0; 0 0 0 5 0; 0 0 0 0 10];
% Q11 - p, Q22 - damping/p rate, Q33 - damping/e rate, Q44 - gamma (integral of p error), Q55 - zeta (integral of e error)
A3_3 = [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];
B3_3 = [0 0; 0 k_1; k_2 0; 0 0; 0 0];
C3_3 = [1 0 0 0 0: 0 0 0 0 1];
%R = [1 0; 0 1];
K3_3 = lqr(A3_3,B3_3,Q3_3,R);
F = [0 0; 0 0; 0 0; -1 0;0 -1];
```

Figure 6: MATLAB code for problem 3.3

As shown in the MATLAB code in Figure 6, the matrix **K** are calculated in the same way as in problem 3.2.

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 17 & 0 & 3 \\ 5 & 5 & 0 & 2 & 0 \end{bmatrix}$$

We see that the **K** Matrix weights the elevation rate $\dot{\tilde{e}}$ and ζ for $\tilde{V}_s$, and $\tilde{p}$, $\dot{\tilde{p}}$ and γ for $\tilde{V}_d$.

q_{11}, q_{22} and q_{33} have the same functions as in problem 3.2. q_{44} weights the integral of the pitch error, and q_{55} weights the integral og the elevation rate error. This time we ended up with the **Q** matrix:

$$\mathbf{Q} = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 20 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 10 \end{bmatrix}$$

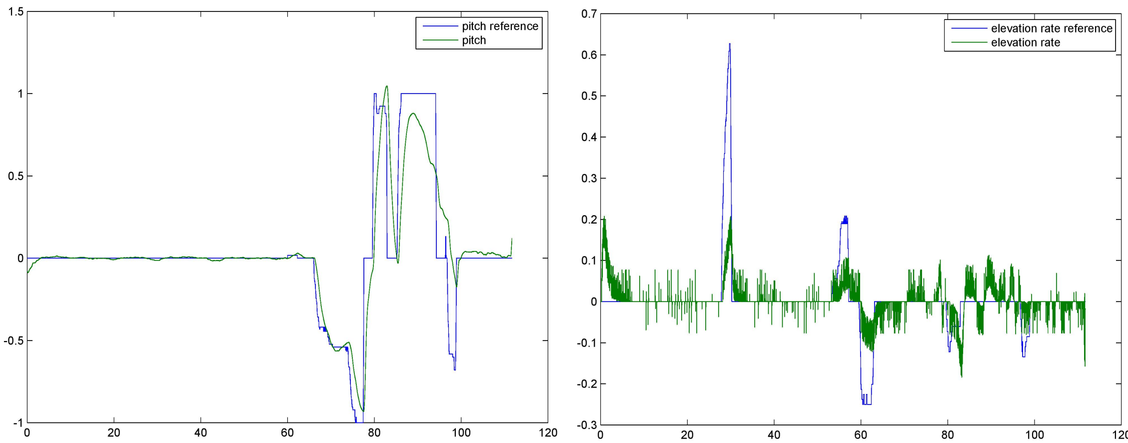


Figure 7: Plots for the pitch $\tilde{p}$ and the pitch reference $\tilde{p_c}$ and the elevation rate $\dot{\tilde{e}}$ and the elevation rate reference $\dot{\tilde{e}}_c$ for the chosen $\mathbf{Q}$ matrix in problem 3.3.

We noticed that when we increased the value of the weighting of ζ, q_{55}, the helicopter does not fly as close to the reference from start as it is supposed to. However, it is more stable, and therefore we chose to punish it (10), but not too much (we tried 50, but that was too much) to find a golden middle road.

The main difference in the system with integral effect versus without, is that the helicopter now stays when for instance driving the helicopter up and then releasing the joystick. This is due to the fact that ζ integrates the elevation rate error. In other words, we are telling the helicopter that when we are not moving the joystick, the acceleration should be zero, hence the helicopter should stay at the height it is. Furthermore, this is the reason why the helicopter now does not reach equilibrium when started, as the integral effect works against the motion. γ, on the other hand, is the integrate of the pitch error, which makes the pitch easier to control as it is a bit slower than without the integral effect.

4 Part 4 - State estimation

4.1 Problem 1

We have the system

$$\ddot{\tilde{p}} = K_1 \tilde{V}_d \tag{15a}$$

$$\ddot{\tilde{e}} = K_2 \tilde{V}_s \tag{15b}$$

$$\ddot{\tilde{\lambda}} = K_3 \tilde{p} \tag{15c}$$

which we can write in state-space formulation of the form:

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

We get the following equations:

$$\begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \dot{\tilde{e}} \\ \ddot{\tilde{e}} \\ \dot{\tilde{\lambda}} \\ \ddot{\tilde{\lambda}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ k_3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ 0 & 0 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_s \\ V_d \end{bmatrix}$$

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

4.2 Problem 2

We are now going to create a linear observer which has the following form:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{L}(\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}) \tag{16}$$

In order to make an observer, we need to know that our system is observable. We examine the observability of the system by obtaining the rank of the observability matrix $\mathbf{O}$:

$$\mathbf{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA^2} \\ \mathbf{CA^3} \\ \mathbf{CA^4} \\ \mathbf{CA^5} \end{bmatrix}$$

By using Matlab to compute the observability matrix $\mathbf{O}$, and to calculate the rank we get that $\text{rank}(\mathbf{O}) = 6$. With $\text{rank}(\mathbf{O}) = 6$ we have full rank and therefore the system is observable.

Hence, we can start creating the linear observer. We want to make the observer such that $\hat{\mathbf{x}} \to \mathbf{x}$ so that the error given by $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$ goes towards zero. This gives us the following:

$$\begin{aligned} \dot{\mathbf{e}} =& \dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}} \\ \implies \dot{\mathbf{e}} =& \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} - \dot{\hat{\mathbf{x}}} \end{aligned}$$

With the observer given by (16), we get:

$$\begin{aligned} \dot{\mathbf{e}} =& \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} - \mathbf{A}\hat{\mathbf{x}} - \mathbf{B}\mathbf{u} - \mathbf{L}(\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}) \\ \implies \dot{\mathbf{e}} =& \mathbf{A}(\mathbf{x} - \hat{\mathbf{x}}) - \mathbf{L}\mathbf{C}(\mathbf{x} - \hat{\mathbf{x}}) \\ \implies \dot{\mathbf{e}} =& \mathbf{A}\mathbf{e} - \mathbf{L}\mathbf{C}\mathbf{e} \\ \implies \dot{\mathbf{e}} =& (\mathbf{A} - \mathbf{L}\mathbf{C})\mathbf{e} \end{aligned}$$

Hence, the **L** matrix is of the dimension 6x3. To find **L** we had to compute the placement of the poles of the system. The placement of the closed-loop observer poles have an influence on the helicopter since the poles corresponds directly to the eigenvalues of the system. And since the eigenvalues of the system control the characteristics of the response of the system, that means that the placement of the poles corresponds to the response of the helicopter.

```matlab
%Problem 4.1 and 4.2

A4_2 = [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];
B4_2 = [0 0; 0 k_1; 0 0; k_2 0; 0 0; 0 0];
C4_2 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];
O1 = obsv(A4_2, C4_2);

n2 = 6; %Rank A4_2/observability matrix

%for placing poles in the form of a fan
%theta = 3.141592/40;
%alpha = 3.141592*9/10;
%r = 10;
%j = sqrt(-1);
%p = zeros(n2,1); %Lager en array med 0

%for i = 1:2:n2
   % alpha = alpha + theta;
   % pole = r*cos(alpha) + j*r*sin(alpha);
   % p(i) = pole;
   % p(i+1) = conj(pole);
%end;

for i=1:1:n2
    pole = -20 -2*i;
    p(i) = pole;
end;

%Placing poles manually
%p = [-6 -8 -10 -12 -14 -16]

L = place(A4_2', C4_2', p)';
```

Figure 8: MATLAB code for problem 4.2

We want to place the poles such that our system responds quickly, but not too fast, as this often leads to oscillations and makes the system less accurate. We also need to consider how big cut off frequency we wish to have in our system. We want a cut of frequency big enough to be able to trust our measurements, but not to big as it leads to too much noise.

At first, we tried placing the poles in the form of a fan. The distance between the real parts and imaginary parts of the poles influences the degree of oscillations in our system, as well as making the system faster. We observed that the transient response of the helicopter worked relatively good and fast, but the helicopter slightly oscillated which is not optimal.

Next we tried placing the poles on the real axis. Since we do not have any imaginary parts here, we will have an overdamped system whit no oscillations. Further, we need to find out how far from origin we want to place the poles. The closer the poles are to origin, the shorter cut off frequency and slower system. And vica verca. When we tried placing the poles along the real axis between -6 and -16 we observed that the helicopter worked relatively fast and stabile. We therefore concluded that having the poles here gives us a cut off frequency large enough to trust our mesurements and short enough to not have too much noise, and that the poles were placed so far from origin that the helicopters transient respons was fast enough.

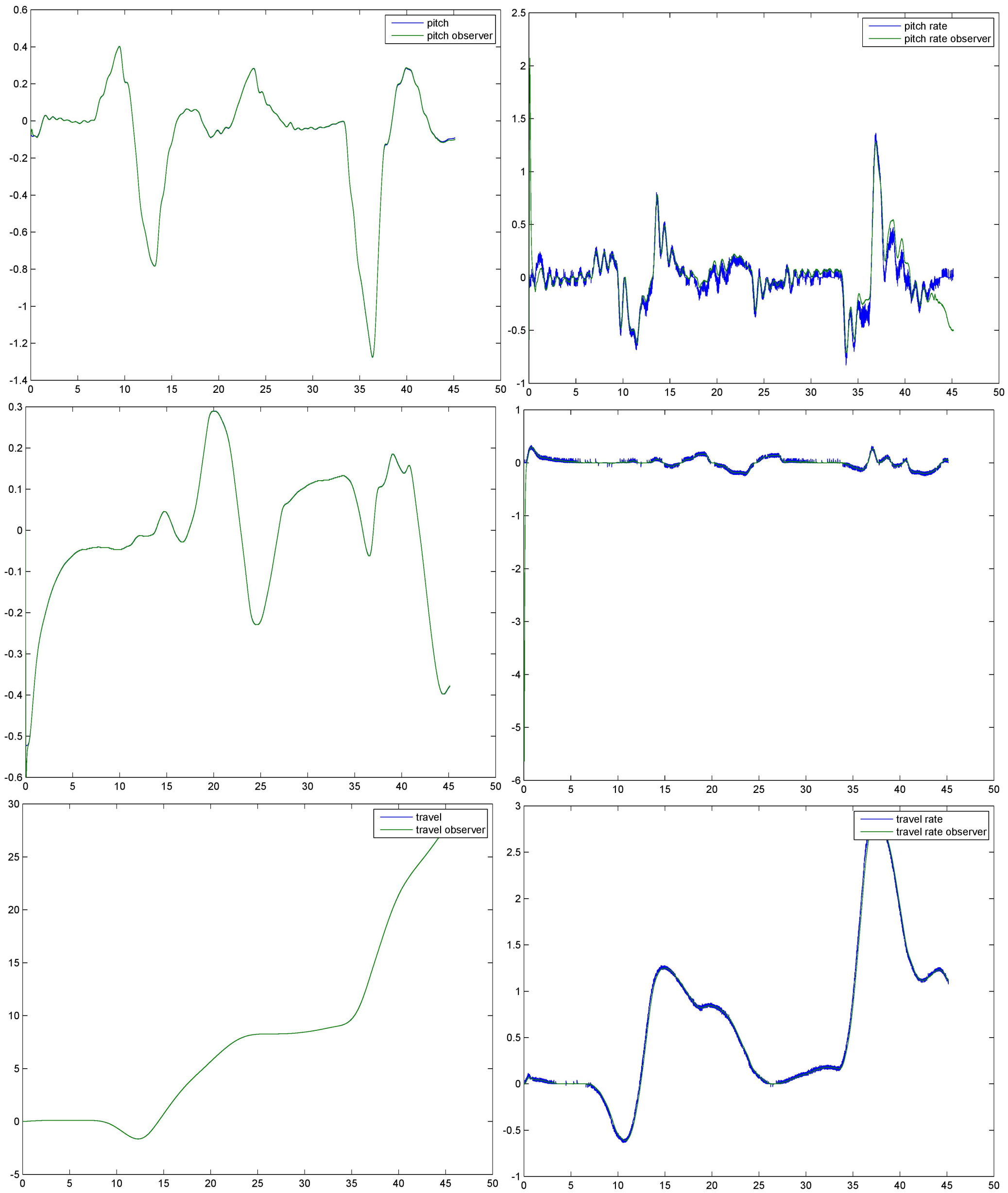


Figure 9: Plots of the states and the state estimates with the integral effect. On the top, pitch $\tilde{p}$ and pitch rate $\dot{\tilde{p}}$, in the middle elevation $\tilde{e}$ and elevation rate $\dot{\tilde{e}}$, and on the bottom travel $\tilde{\lambda}$ and travel rate $\dot{\tilde{\lambda}}$.

To find the linear observer matrix **L**, we used the MATLAB function "place". This function calculates L, based on the input, the output and the poles, and uses this to predict how the system will behave. By using the poles along the real axis, we ended up with the following **L** matrix:

$$\mathbf{L} = \begin{bmatrix} 58 & 2 & 0 \\ 848 & 54 & 0 \\ 2 & 58 & 0 \\ 60 & 828 & 0 \\ 0 & 0 & 46 \\ 1 & 0 & 528 \end{bmatrix}$$

Here we see that **L** weights $\tilde{p}$, $\tilde{e}$ and $\tilde{\lambda}$ about 55 times, and the rates $\dot{\tilde{p}}$, $\dot{\tilde{e}}$ and $\dot{\tilde{\lambda}}$ about 500-850 times the difference between the corresponding measured and observed states. We also se that the pitch rate is weighted about 60 times the difference between the measured and observed value for elevation, and the other way around. $\tilde{\lambda}$ and $\dot{\tilde{\lambda}}$ are only weighted by the difference of the measured and the observed value for $\tilde{\lambda}$.

4.3 Problem 3

First, we want to show that the system is observable when measuring $\tilde{e}$ and $\tilde{\lambda}$. To show this, we compute the observability matrix **O** in the same way as in the previous problem. Now we are only measuring $\tilde{e}$ and $\tilde{\lambda}$, and the **C** matrix is therefore:

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

By using MATLAB to calculate the obsevarbility, we get that $\text{rank}(\mathbf{O}) = 6$ which is a full rank. Hence, the system is observable when measuring only $\tilde{e}$ and $\tilde{\lambda}$.

Next, we want to show that the system is not observable when only measuring $\tilde{p}$ and $\tilde{e}$. To show this, we follow the exact approach as previously. Now, we get the following **C** matrix:

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

By using MATLAB the same way as previously, we see that $\text{rank}(\mathbf{O}) = 4$, which is not a full rank. Therefore, the system is not observable when we are only measuring $\tilde{p}$ and $\tilde{e}$.

Furthermore, we are going to create a linear observer based on the measurement vector y $= [\tilde{e}\ \tilde{\lambda}]^T$.

To do this we follow the same approch as in 5.4.2 and get the following equation:

$\dot{\mathbf{e}} = (\mathbf{A} - \mathbf{LC})\mathbf{e}$

Since the dimension of the **C** matrix has changed since 5.4.2 due to only measuring to inputs, our linear observer matrix L will have other dimensions as well. Since **C** is a 2x6 matrix, **L** becomes a 6x2 matrix

We did not manage to get our system stable. However, even though it is doable, we expected it to be very difficult as we are only measuring two states. To get the system controllable the states measured should be theoretically independent of each other, as is elevation and travel, but not pitch and elevation.

When we were trying to get the system stable we tried a variation of different poles. First, we tried placing the poles in a fan form, but this did not work. This was expected, as poles in fan form creates more oscillations than along the real axis, and since the system already is very unstable, oscillations are not wanted in the system. Hence, we tried placing the poles along the real axis. Though we did not manage to actually get the system stable, we expect that the poles that makes the system controllable are somewhere on the real axis, and coinciding as this will make the system critically damped. It may also have different real poles, which make the system overdamped, this may also be a good solution as the system then would be slower and maybe reduce the chance of oscillating out of control.

We would also expect that the poles should not be too far away from the origin, as the the system is quicker the further away from the origin it is, and we want a slow system in order to be able to control it.

5 Bibliography

- visited 23.10.2016, last modified on 22 October 2016, at 17:58:

https://en.wikipedia.org/wiki/PID_controller/PID_controller_theory- Kompendiumkomiteen for kybernetikk 2016, Reguleringsteknikk Kompendium 2016.

- Balchen, J.G. Andresen, T. Foss, B.A.(2003). Reguleringsteknikk. Institutt for teknsik kybernetikk, NTNU.

- Chen, Chi-Tsong. Linear System Theory and Design, Fourth edition. Oxford University Press. John Wiley Sons, Inc.

6 Appendix

6.1 Complete MATLAB code

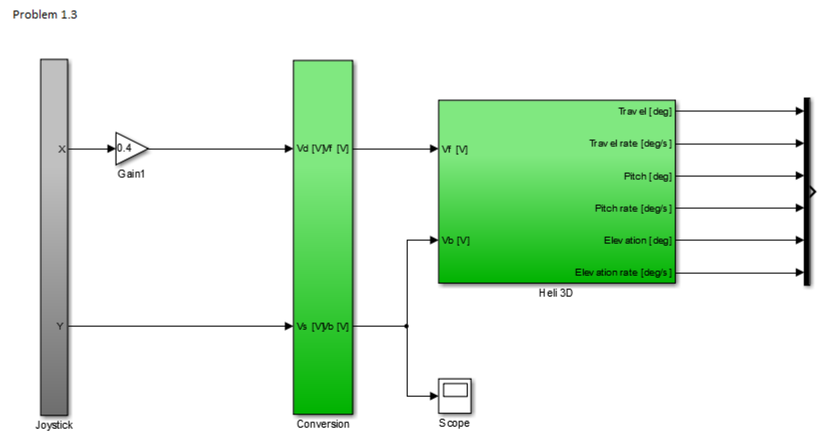
```
%%%%%%%%%%%% Calibration of the encoder and the hardware for the specific
%%%%%%%%%%%% helicopter
Joystick_gain_x = 1;
Joystick_gain_y = -1;


%%%%%%%%%%%% Physical constants
g = 9.81; % gravitational constant [m/s^2]
l_c = 0.46; % distance elevation axis to counterweight [m]
l_h = 0.66; % distance elevation axis to helicopter head [m]
l_p = 0.175; % distance pitch axis to motor [m]
m_c = 1.92; % Counterweight mass [kg]
m_p = 0.72; % Motor mass [kg]

v_s = 6.9; % v_s*

k_f = g * (-m_c * l_c + 2* m_p *l_h) / (l_h * v_s); % motor force constant

w_0 = 1.5; % omega_0, resonans frequence

j_p = 2 * m_p * l_p^2;
j_e = m_c * l_c^2 + 2 * m_p * l_h^2;
j_l = m_c * l_c^2 + 2 * m_p *(l_h^2 + l_p^2);

l_1 = k_f * l_p;
l_2 = g * (-m_c * l_c + 2 * m_p * l_h);
l_3 = k_f * l_p;
l_4 = -l_3;

k_1 = l_1/j_p;
k_2 = l_3/j_e;
k_3 = l_2/j_l;
```

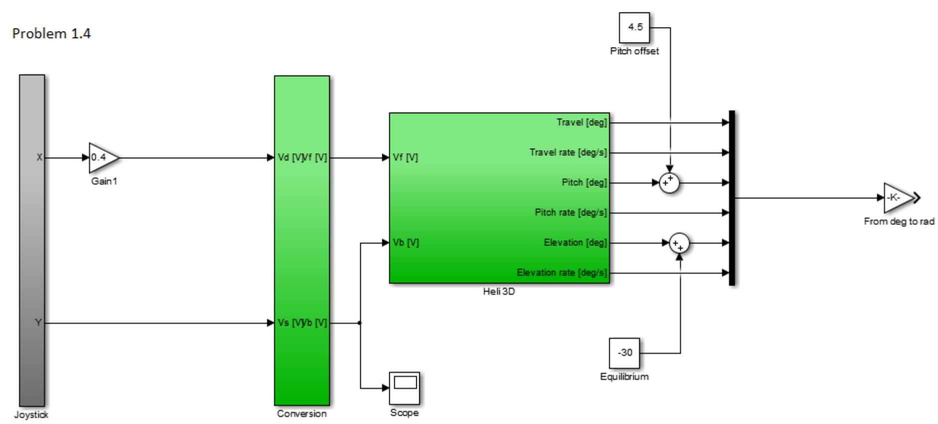
```
K_pp = w_0^2 * j_p / l_1;
K_pd = 2*w_0 * j_p/ l_1;
K_ratio = K_pp/K_pd;
K_rp = -1;

zeta = 0.5*(K_pd/K_pp)*sqrt(k_1*K_pp);

%Part 3 problem 2

A3_2 = [0 1 0; 0 0 0; 0 0 0];
B3_2 = [0 0; 0 k_1; k_2 0];
C3_2 = [1 0 0; 0 0 1];
Q3_2 = [20 0 0; 0 2 0; 0 0 10]; %Q11 - pitch, Q22 - damping/pitch rate, Q33 - damping/elevation rate
R = [1 0; 0 1];
K3_2 = lqr(A3_2,B3_2,Q3_2,R);
P3_2 = inv(C3_2*inv(B3_2*K3_2 -A3_2)*B3_2);
pol = eig(A3_2 - B3_2*K3_2);

%Part 3 problem 3

Q3_3 = [5 0 0 0 0; 0 10 0 0 0; 0 0 20 0 0; 0 0 0 5 0; 0 0 0 0 10];
% Q11 - p, Q22 - damping/p rate, Q33 - damping/e rate, Q44 - gamma (integral of p error), Q55 - zeta (integral of
A3_3 = [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];
B3_3 = [0 0; 0 k_1; k_2 0; 0 0; 0 0];
C3_3 = [1 0 0 0 0: 0 0 0 0 1];
%R = [1 0; 0 1];
K3_3 = lqr(A3_3,B3_3,Q3_3,R);
F = [0 0; 0 0; 0 0; -1 0;0 -1];
```

```
%Problem 4.1 and 4.2

A4_2 = [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];
B4_2 = [0 0; 0 k_1; 0 0; k_2 0; 0 0; 0 0];
C4_2 = [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];
O1 = obsv(A4_2, C4_2);

n2 = 6; %Rank A4_2/observability matrix

%for placing poles in the form of a fan
%theta = 3.141592/40;
%alpha = 3.141592*9/10;
%r = 10;
%j = sqrt(-1);
%p = zeros(n2,1); %Lager en array med 0

%for i = 1:2:n2
   % alpha = alpha + theta;
   % pole = r*cos(alpha) + j*r*sin(alpha);
   % p(i) = pole;
   % p(i+1) = conj(pole);
%end;

for i=1:1:n2
    pole = -20 -2*i;
    p(i) = pole;
end;

%Placing poles manually
%p = [-6 -8 -10 -12 -14 -16]

L = place(A4_2', C4_2', p)';
```

```
%Problem 4.3

A4_3 = [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];
B4_3 = [0 0; 0 k_1; 0 0; k_2 0; 0 0; 0 0];
C4_3 = [0 0 1 0 0 0; 0 0 0 0 1 0]; %Measured elevation and travel
C4_33 = [1 0 0 0 0 0; 0 0 1 0 0 0]; %Measured pitch and elevation
O2 = obsv(A4_3, C4_3); % rank(O2) = n = 6, controllable
O3 = obsv(A4_3, C4_33); %rank(O3) != n, not controllable

n3 = 6; %Rank A4_3/observability matrix

p2 = zeros(n3,1);

theta2 = 3.141592/60;
alpha2 = 3.141592*9/10;

%for i = 1:2:n3
  %  alpha2 = alpha2 + theta2;
   % pole = r*cos(alpha2) + j*r*sin(alpha2);
   % p2(i) = pole;
   % p2(i+1) = conj(pole);
%end;

p2 = [-30 -8 -10 -30 -6+0.5*i -8];

L4_3 = place(A4_3', C4_3', p2)';
```

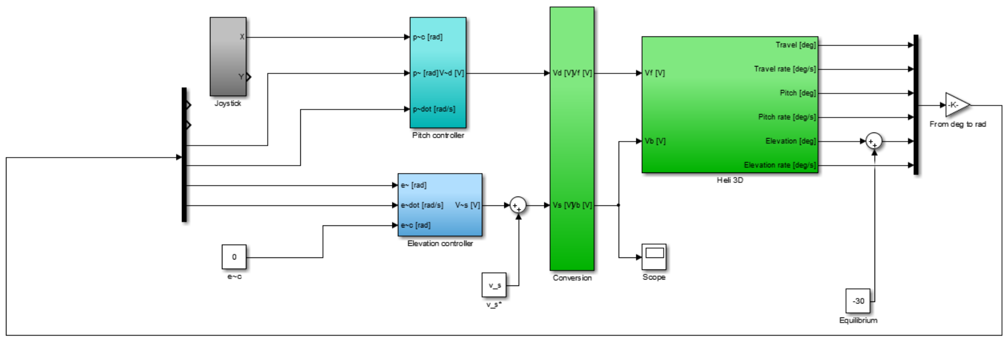
6.2 Complete Simulink diagrams





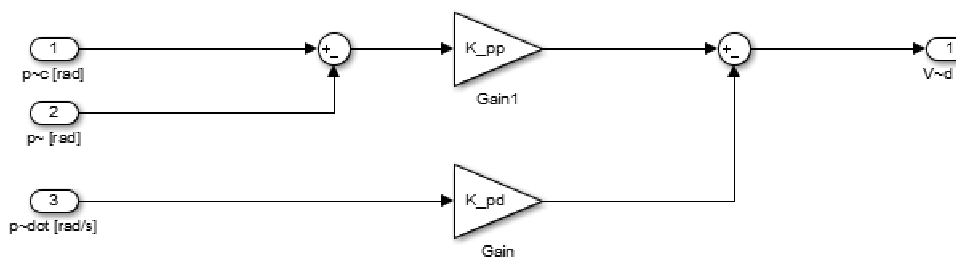
1.png

Part 2, problem 1



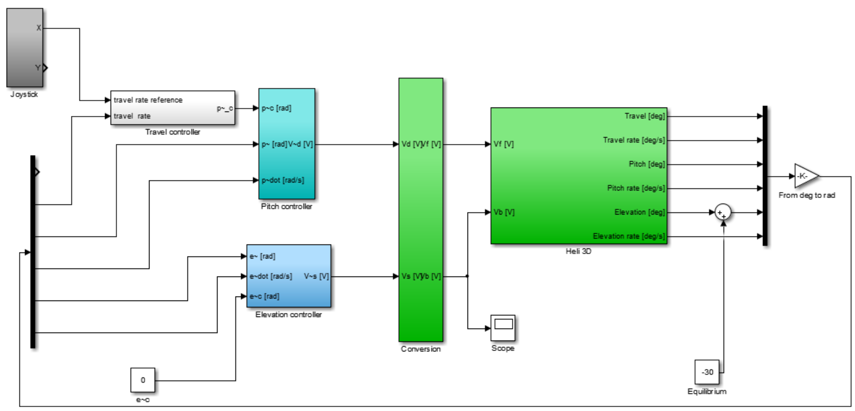
2 1.png

Pitch controller



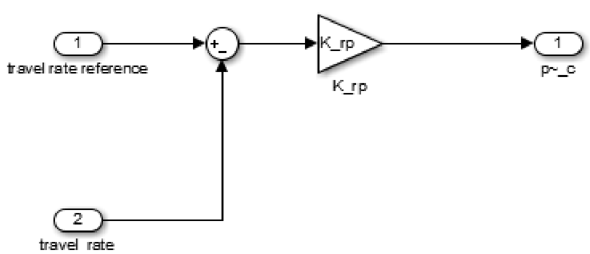
2 1 pitch controller.png

Part 2, problem 2



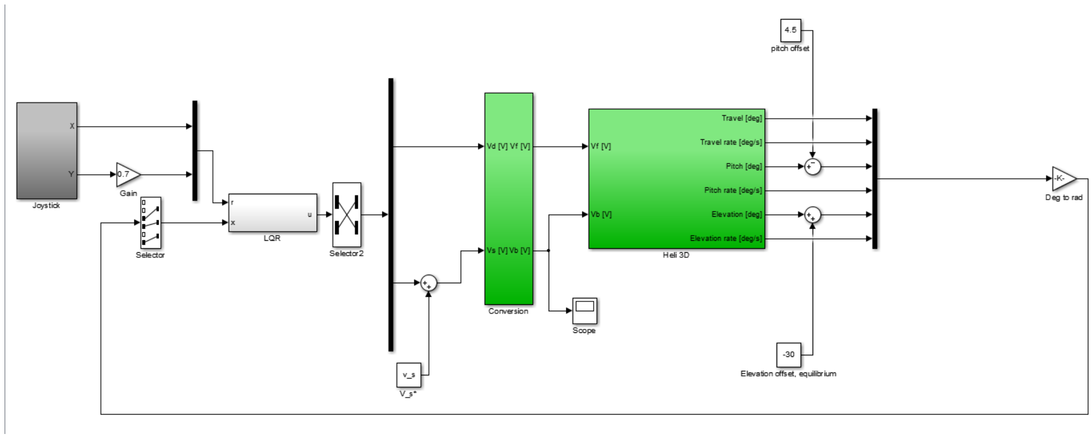
2 2.png

Travel controller



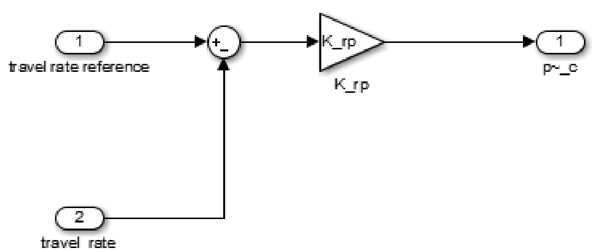
2 2 travel controller.png

Part 3, problem 2



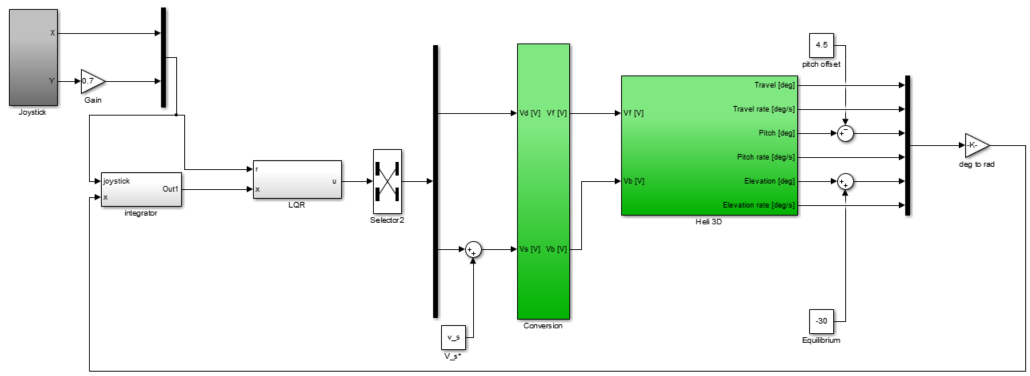
3 2.png

LQR



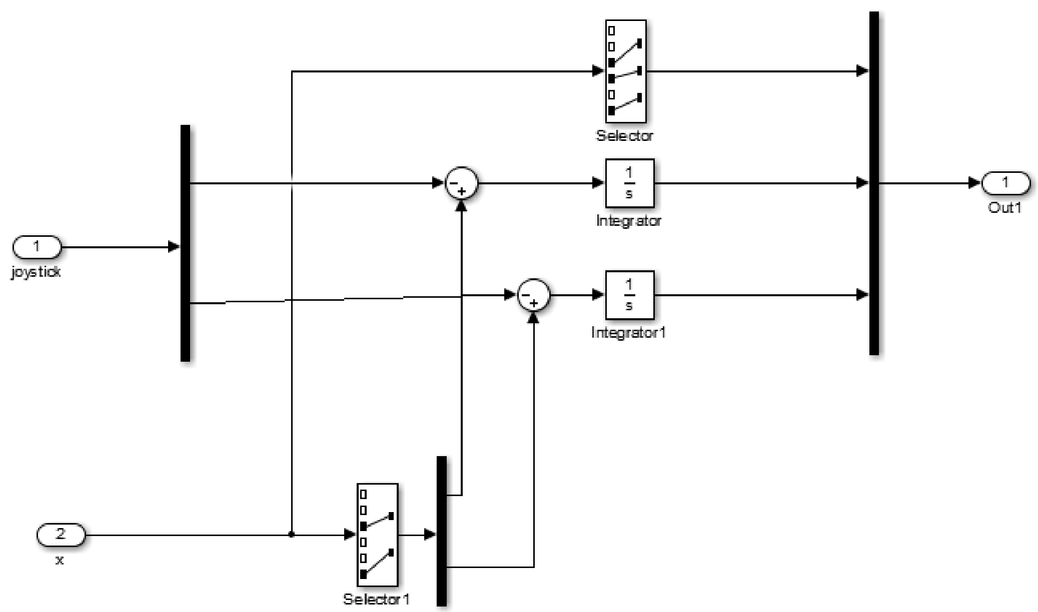
3 2 LQR.png

Part 3, problem 3



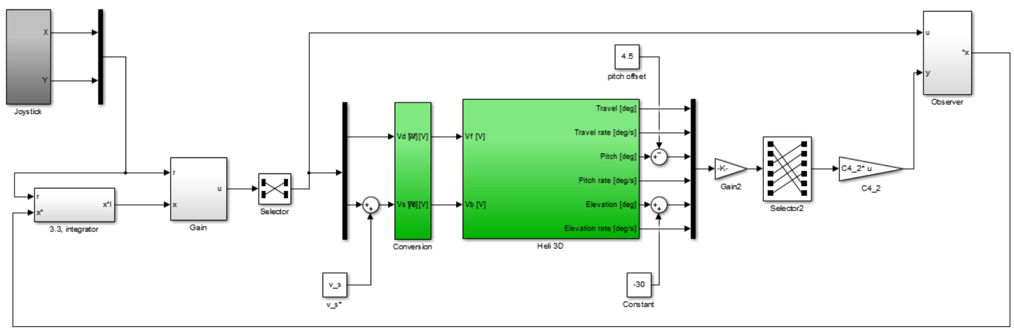
3 3.png

Integrator



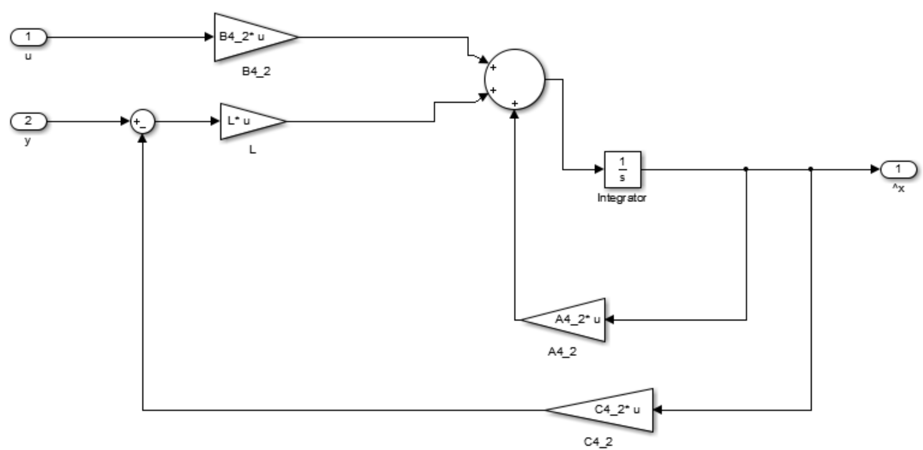
3 3 Integrator.png

Part 4, problem 2, with integrator



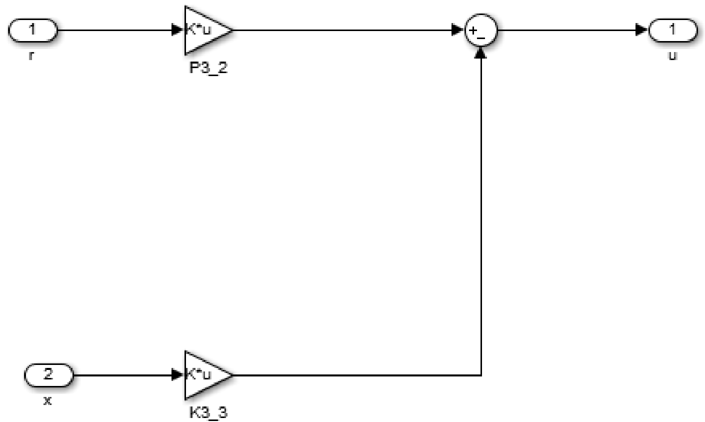
4 2.png

Observer



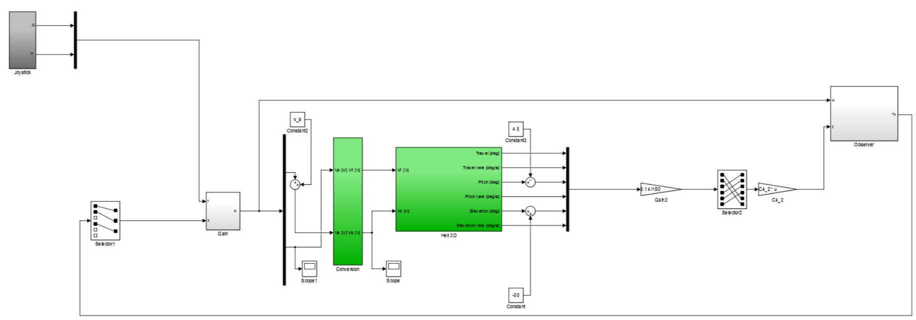
4 2 Observer.png

Gain



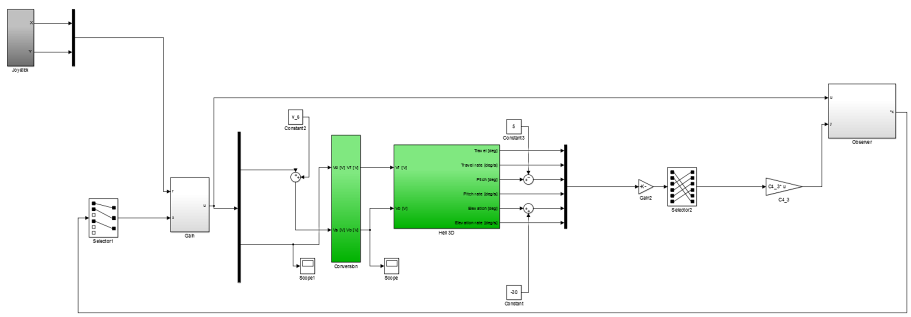
4 2 Gain.png

Part 4, problem 2, without integrator



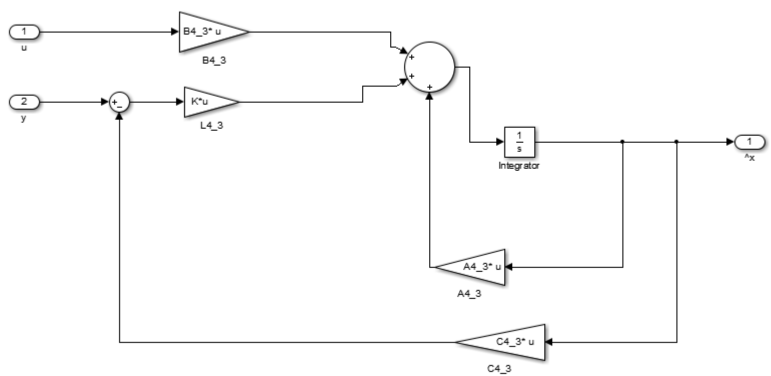
4 2 without integrator.png

Part 4, problem 3



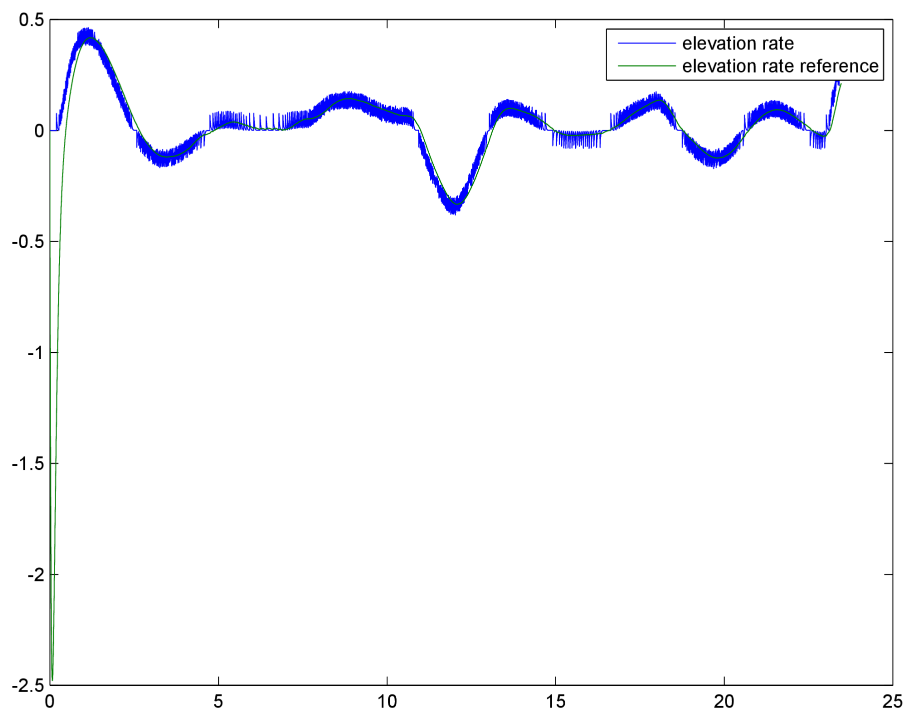
4 3.png

Observer

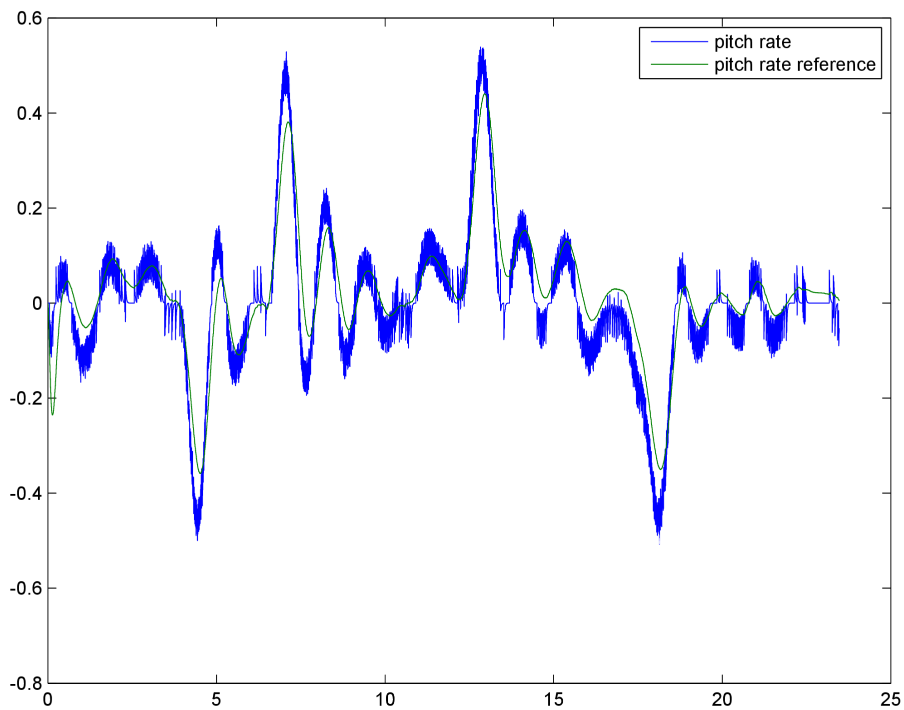


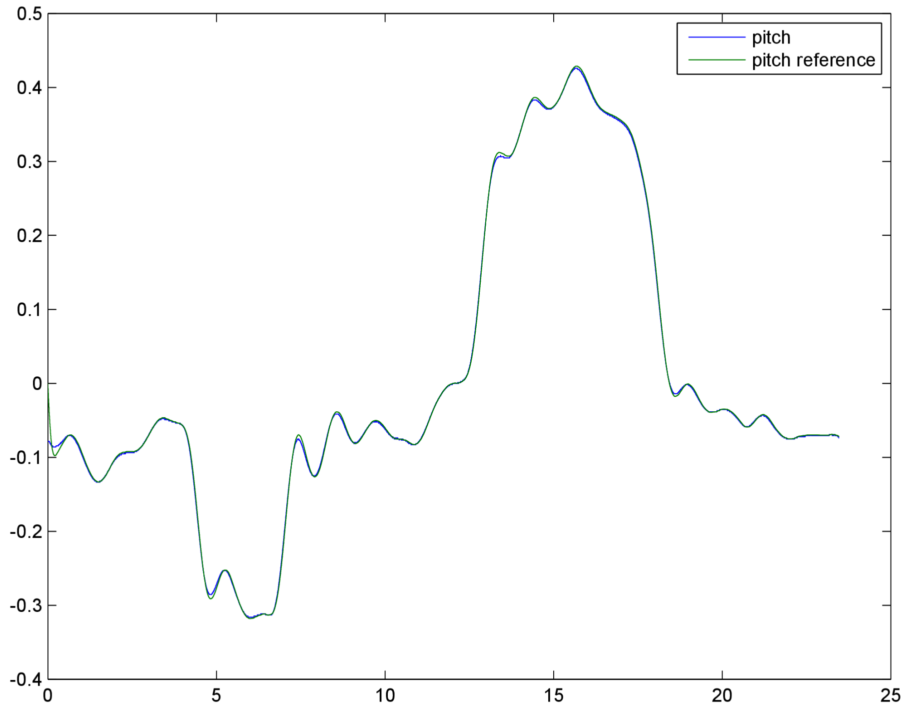
4 3 observer.png

6.3 Plots



(1).png


pitch rate
pitch rate reference
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
5
10
15
20
25



0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
0
5
10
15
20
25
pitch
pitch reference