Norwegian University of Science and Technology

TTK4115 Linear System Theory

Helicopter lab report

By:

Ugur Alpay Cenar(studnr)

By Shiv Jeet Rai (studnr)

Stian Grttum Danielsen (studnr)

October 18, 2014

Contents

1 Part 1 - Mathematical modeling

A mathematical model was derived to create the basis for the rest of the assignment. The motor forces were given by

$$F_f = K_f V_f \tag{1}$$

$$F_b = K_f V_b \tag{2}$$

1.1 Problem 1

To compute the equations of motion for the pitch angle p, elevation angle e, and travel angle λ, Newton's second law for rotation was applied. Newton's second law for rotation was given by:

$$\tau = \mathbf{I} \cdot \boldsymbol{\alpha} \tag{3}$$

where τ was the externally applied torque, $\mathbf{I}$ was the moment of inertia, and $\boldsymbol{\alpha}$ was the angular acceleration. The sum of the motor voltages V_s and the difference of the motor voltages V_d were given by

$$V_s = V_f + V_b \tag{4}$$

$$V_d = V_f - V_b \tag{5}$$

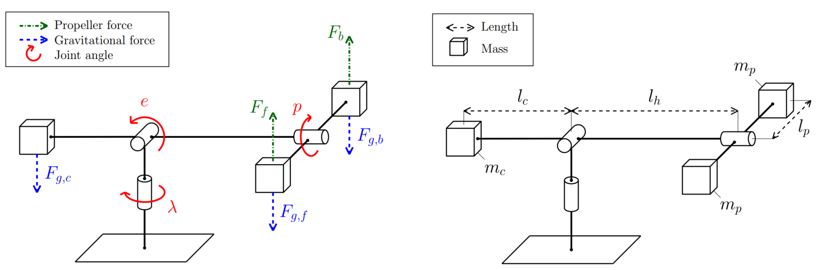

Figure 1: Helicopter model

Using equation (3) on Figure 1 with respect to the pitch gave:

$$\begin{aligned} J_p \ddot{p} &= (F_f - F_b) l_p \\ J_p \ddot{p} &= (K_f V_f - K_f V_b) l_p \\ J_p \ddot{p} &= K_f V_d l_p \end{aligned} \tag{6}$$

Similarly the equation for the travel e could be expressed as:

$$J_e\ddot{e} = F_{g,c}l_c\cos(e) + (F_f + F_b)l_h\cos(p) - (F_{g,f} + F_{g,b})l_h\cos(e)$$

Since $F_{g,f} = F_{g,b}$ we got:

$$J_e\ddot{e} = (F_{g,c}l_c - 2F_{g,f}l_h)\cos(e) + K_fV_sl_h\cos(p)$$

$$J_e\ddot{e} = (l_cm_c - 2m_pl_h)g\cos(e) + K_fV_sl_h\cos(p) \tag{7}$$

For the travel, λ, only the rotors generated torque. The magnitude of the torque generating force was given by the angle p, and the arm was given by e. By equation (3) and Figure 1 the travel e could be expressed as:

$$J_\lambda\ddot{\lambda} = -K_fV_s\sin(p)l_h\cos(e) \tag{8}$$

The three equations of motion, (6),(7) and (8) could be stated respectively on the form

$$J_p\ddot{p} = L_1V_d \tag{9}$$

$$J_e\ddot{e} = L_2\cos(e) + L_3V_S\cos(p) \tag{10}$$

$$J_\lambda\ddot{\lambda} = L_4V_s\cos(e)\sin(p) \tag{11}$$

where the constants L_1, L_2, L_3 and L_4 were given by

$$L_1 = K_fl_p \tag{12}$$

$$L_2 = F_{g,c}l_c - 2F_{g,h}l_h \tag{13}$$

$$L_3 = K_fl_h \tag{14}$$

$$L_4 = -K_fl_h \tag{15}$$

1.2 Problem 2

The equations were to be linearized around the point:

$$(p, e, \lambda)^T = (p^*, e^*, \lambda^*)^T \tag{16}$$

$$(V_s, V_d)^T = (V_s^*, V_d^*)^T \tag{17}$$

Where $p^* = e^* = \lambda^* = 0$ and $\dot{p} = \dot{e} = \dot{\lambda} = 0$ which implied $\ddot{p} = \ddot{e} = \ddot{\lambda} = 0$ Using this for equations (9),(10) and (11) we got.

$$0 = L_1V_d^*$$

$$0 = L_2cos(e^*) + L_3V_s^*cos(p^*)$$

$$0 = L_4V_s^*\cos(e^*)\sin(p^*)$$

which lead to

$$V_d^* = 0 \tag{18}$$

$$V_s^* = -\frac{L_2}{L_3} \tag{19}$$

Next we used the coordinate transformation

$$\begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} = \begin{bmatrix} p \\ e \\ \lambda \end{bmatrix} - \begin{bmatrix} p^* \\ e^* \\ \lambda^* \end{bmatrix} \qquad and \qquad \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} = \begin{bmatrix} V_s \\ V_d \end{bmatrix} - \begin{bmatrix} V_s^* \\ V_d^* \end{bmatrix} \tag{20}$$

transferring the equations to the new coordinate system using (20) yielded

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

The formula for linearizing of a multivariable function $f(x, y)$ at a point $p(a, b)$ was

$$f(x, y) \approx f(a, b) + \frac{\partial f(x, y)}{\partial x}\bigg|_{a,b}(x - a) + \frac{\partial f(x, y)}{\partial y}\bigg|_{a,b}(y - b) \tag{21}$$

Using this formula with the appropriate number of variables for each state equation to linearize around $(\tilde{p}, \tilde{e}, \tilde{\lambda})^T = (0, 0, 0)^T$ and $(\tilde{V}_s, \tilde{V}_d)^T = (0, 0)^T$ using the moments of inertia given by

$$J_p = 2m_p l_p^2 \tag{22}$$

$$J_e = m_c l_c^2 + 2m_p l_h^2 \tag{23}$$

$$J_\lambda = m_c l_c^2 + 2m_p l_h^2 \tag{24}$$

yielded

$$\ddot{\tilde{p}} = \frac{L_1 \tilde{V}_d}{J_p} \tag{25}$$

$$\ddot{\tilde{e}} = \frac{L_3 \tilde{V}_s}{J_e} \tag{26}$$

$$\ddot{\tilde{\lambda}} = \frac{L_2 \tilde{p}}{J_\lambda} \tag{27}$$

which could be written as

$$\ddot{\tilde{p}} = K_1 \tilde{V}_d \tag{28}$$

$$\ddot{\tilde{e}} = K_2 \tilde{V}_s \tag{29}$$

$$\ddot{\tilde{\lambda}} = K_3 \tilde{p} \tag{30}$$

with the constants K_1 K_2 K_3 given by

$$K_1 = \frac{L_1}{2m_p(l_p)^2} \tag{31}$$

$$K_2 = \frac{L_3}{J_e} \tag{32}$$

$$K_3 = \frac{L_2}{J_\lambda} \tag{33}$$

1.3 Problem 3

The helicopter was controlled using direct feed forward from the joystick output. The voltage difference V_d and the voltage sum V_s were connected to the x-axis and y-axis of the joystick respectively. Comparing the behaviour of the helicopter to (6), (7) and (8) it seemed to match pretty well. Although hard to hold the helicopter steady, it seemed that $\ddot{e}$ and $\ddot{\lambda}$ changed with different angles e and p and that $\ddot{p}$ was independent of any angles. It was though observable that $\ddot{\lambda}$ reached a limit at high speed. This was likely because of drag which is not part of the model. The linearized model for elevation and travel then seemed to be a bit inaccurate, but not too bad when observed with the naked eye. This was to be expected because the system was non-linear. Both the linearized model and theoretical model of pitch was linear, so they both matched the helicopter behaviour well.

1.4 Problem 4

For the encoder outputs to be equal the joint angles p, e and λ, constants had to be added. The constant were measured by holding the helicopter in the position where all the angles were supposed to be equal to zero and reading the encoder outputs. The travel angle λ did not need any constant because the zero direction can be chosen for every start of the helicopter. The pitch angle p was very close to zero when starting from the table, so no constant was necessary. The constant for elevation angle e was measured to be

$$e_{encoderConstant} = -25.1 \tag{34}$$

To find V_s, V_s^* was measured by keeping the helicopter horizontal and reading the voltage output from the joystick. The measured voltage V_s^* was found to be

$$V_s^* = 3.6946 \tag{35}$$

$\dot{p} = \dot{e} = \dot{\lambda} = 0$ implied that

$$0 = L_1 V_d^* \implies V_d^* = 0$$

and

$$0 = L_2 \cos(e) + L_3 V_s^* \cos(p) \implies V_s^* = \frac{-L_2 \cos(e)}{L_3 cos(p)}$$

and for travel we had

$$0 = L_4 V_s \cos(e) sin(p) \implies p = 0.$$

We were measuring V_s^* when $e = 0$ so we got

$$V_s^* = \frac{-L_2}{L_3} \tag{36}$$

from the voltage transformation in equation(20) we got

$$\tilde{V}_s = V_f + V_d + \frac{L_2}{L_3}$$

$$V_s^* = \frac{-(F_{g,c} l_c - F_{g,b} l_h - F_{g,f} l_h)}{K_f l_h}$$

Solving for K_f yielded

$$K_f = 0.0832 \tag{37}$$

2 Part 2 - Monovariable control

2.1 Problem 1

In this assignment we were asked to find values for reasonable values for K_{pp},K_{pd} by transfering the following equation:

$$\tilde{V}_d = K_{pp}(\tilde{p}_c - \tilde{p}) - K_{pd}\dot{\tilde{p}} \tag{38}$$

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

From the equation (30) we had that $\ddot{\tilde{p}} = K_1 \tilde{V}_d => \tilde{V}_d = \frac{\ddot{\tilde{p}}}{K_1}$

By putting this into equation (38) we got:

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

$$S^2\tilde{p}(S) = K_1 K_{pp}(\tilde{p}_c - \tilde{p}) - K_1 K_{pd}\dot{\tilde{p}}$$
$$(S^2 + SK_{pd}K_1 + K_1K_{pp})\tilde{p}(S) = K_1K_{pp}\tilde{p}_c(S)$$

This equalled to:
$$\frac{\tilde{p}(s)}{\tilde{p}_c(s)} = \frac{K_1K_{pp}}{S^2 + SK_1K_{pd} + K_1K_{pp}} \tag{39}$$

2.1.1 Finding of K_{pp} and K_{pd}

We found K_{pp} and K_{pd} by using the equation:
$$\frac{K\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{40}$$

Where we are using that $\zeta = 1$,which gives a critically damped system with repeated poles.
This gives the equation:
$$\frac{K\omega_0^2}{(S+\omega_0)^2} \tag{41}$$
Where ω_0 is the eigenvalue of the controller.

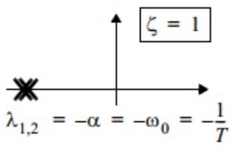

Figure 2: We have a controller with a repeated pole

By changing this you can determine the respons of the system. Large ω_0 gives fast respons, but too large value will give an underdamped system. Too small value will give over-damped system and the challenge was to find something between to have critically damped system
By comparing equation 39 and 40 you get that

$$\omega_0^2 = K_1K_{pp} => K_{pp} = \frac{\omega_0^2}{K_1}$$

and

$$2\zeta\omega_0 = K_1K_{pd} => K_{pd} = \frac{2\zeta\omega_0}{K_1} = \frac{2\omega_0}{K_1}$$

The equations is:

$$K_{pp} = \frac{\omega_0^2}{K_1} \tag{42}$$

$$K_{pd} = \frac{2\omega_0}{K_1} \tag{43}$$

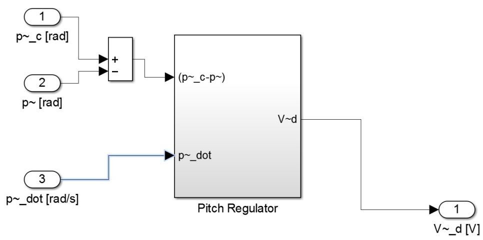

Figure 3: Simulink diagram of Pitch Controller

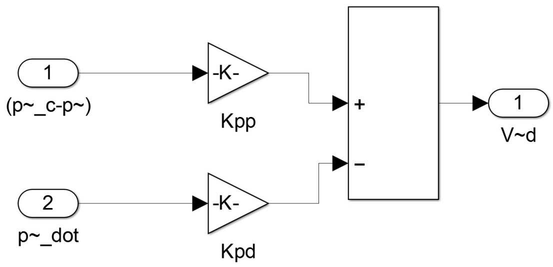

Figure 4: Simulink diagram of Pitch Controller

By experiment we found that $\omega_0 = 2$ gave us the best control with parameters:

$$K_{pd} = 25.4762\frac{V}{rad} \tag{44}$$

$$K_{pp} = 25.4762\frac{V}{rad} \tag{45}$$

2.1.2 Discussion

This gave fast and accurate control. If we had w_0 too low the pitch controller was very slow and had problem with reaching the reference, and if we had it too large the pitch became under-damped and in worst case started to oscillate.

2.2 problem 2

Travel rate is controlled by a simple P-controller

$$\tilde{p}_c = K_{rp}(\dot{\tilde{\lambda}}_c - \dot{\tilde{\lambda}}) \tag{46}$$

Assuming that the pitch angle is controlled perfectly $\tilde{p} = \tilde{p}_c$ and using the formula (ref) we get $\ddot{\tilde{\lambda}} = K_3\tilde{p} => \tilde{p} = \frac{\ddot{\tilde{\lambda}}}{K_3}$ which gives:

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

This equals to:

$$\frac{\dot{\tilde{\lambda}}(s)}{\dot{\tilde{\lambda}}_c(s)} = \frac{K_{rp}K_3}{S + K_{rp}K_3} \tag{47}$$

K_{rp} was found by using the formula:

$$h(S) = \frac{-K\lambda}{S - \lambda}$$

where $\lambda = -\frac{1}{T}$. This is the pole of the transfer function and T gives the time the travel rate will arrive to 63 percent of the reference. By changing T, the response-time of the controller could be changed.

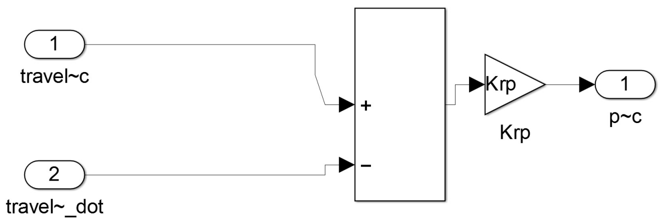

Figure 5: Simulink diagram for Travel-Controller

The controller was added to the simulink diagram and by experiment it was found that with $T = 6s$ and

$$K_{rp} = \frac{1}{TK_3} = -1.0827\frac{V}{rad} \tag{48}$$

gave fast and accurate control of the travel rate.

2.2.1 Discussion

If T was too large the travel controller had problem with reaching the reference and it could take very long time. If T was too small the travel rate became under-damped and started to oscillate when it tried to break. T=6s gave the best response.
From the figure 6 you can see the step response of the travel rate. The system reaches the reference in 6 seconds, which is faster than expected. This can be because the elevation controller have to give extra power to hold the helicopter at the reference and this makes the response-time faster. But that is not a problem and the travel controller have a fast and accurate response.

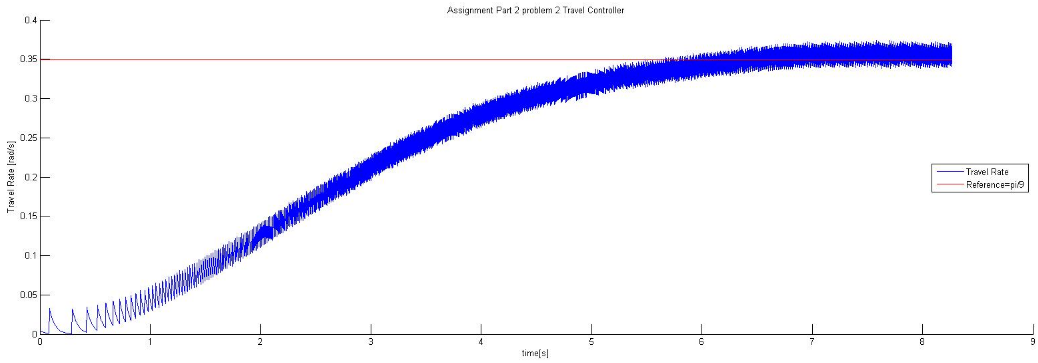

Figure 6: Step Response of the Travel Rate

3 Part III - Multivariable control

3.1 5.3.1 Problem 1

In this problem we were told to put the system of equations in a state-space formulation of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{49}$$

where:

$$\mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \end{bmatrix} \qquad and \qquad \mathbf{u} = \begin{bmatrix} \tilde{V_s} \\ \tilde{V_d} \end{bmatrix} \tag{50}$$

By using the linearized equations of motions (6a-c) and saying $\dot{\tilde{p}} = \dot{\tilde{p}}$ we got:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \ddot{\tilde{e}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \end{bmatrix} \mathbf{u}$$

From this we could conclude:

$$\underline{\underline{\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \textit{and} \qquad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \end{bmatrix}}} \tag{51}$$

3.2 5.3.2 Problem 2

Here our aim was to track the $\mathbf{r} = \begin{bmatrix} \tilde{p}_c & \dot{\tilde{e}} \end{bmatrix}^T$ by using a controller of the form:

$$\mathbf{u} = \mathbf{Pr} - \mathbf{Kx} \tag{52}$$

The matrix $\mathbf{K}$ corresponded to the linear quadratic regulator (LQR) for which the term $\mathbf{u}$ = -$\mathbf{Kx}$ optimized the cost function:

$$J = \int_0^\infty (\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t))dt \tag{53}$$

By using $\mathbf{Q}$ and $\mathbf{R}$ to be diagonal matrices, $\mathbf{K}$ to choosed by the MATLAB command $\boldsymbol{K}$=*lqr(*$\boldsymbol{A}$,$\boldsymbol{B}$,$\boldsymbol{Q}$,$\boldsymbol{R}$), we found a $\mathbf{P}$ such that (in theory):

$$lim_{t\to\infty}\tilde{p}(t) = \tilde{p}_c \qquad \textit{and} \qquad \dot{\tilde{e}}(t) = \dot{\tilde{e}}_c \tag{54}$$

3.2.1 Controllability of the system

To verify that the system was controllable

3.2.2 Finding matrix P

We inserted equation(52) into equation(49) and by using the limits in (54) we knew that $\dot{\mathbf{x}} = 0$. We got after placing the term on both sides of the equal operand:

$$-(\mathbf{A} - \mathbf{BK})\mathbf{x}_\infty = \mathbf{BPr}$$

$$-(\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \end{bmatrix} \begin{bmatrix} k_1 & k_3 & k_5 \\ k_2 & k_4 & k_6 \end{bmatrix}) \begin{bmatrix} \tilde{p}_c \\ 0 \\ \dot{\tilde{e}}_c \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \end{bmatrix} \begin{bmatrix} p_1 & p_3 \\ p_2 & p_4 \end{bmatrix} \begin{bmatrix} \tilde{p}_c \\ \dot{\tilde{e}}_c \end{bmatrix}$$

This can be written as:

$$
\begin{aligned}
0 &= 0 \\
K_1 k_2 \tilde{p}_c + K_1 k_6 \dot{\tilde{e}}_c &= K_1 P_2 \tilde{p}_c + K_1 P_4 \dot{\tilde{e}}_c \\
K_2 k_1 \tilde{p}_c + K_2 k_5 \dot{\tilde{e}}_c &= K_2 P_1 \tilde{p}_c + K_2 P_3 \dot{\tilde{e}}_c
\end{aligned} \tag{55}
$$

From equation(55) we clearly see that:

$$
\underline{\underline{\mathbf{P} = \begin{bmatrix} p_1 & p_3 \\ p_2 & p_4 \end{bmatrix} = \begin{bmatrix} k_1 & k_5 \\ k_2 & k_6 \end{bmatrix}}} \tag{56}
$$

3.2.3 Estimating matrices Q and R

To track the reference $\mathbf{r} = \begin{bmatrix} \tilde{p}_c & \dot{\tilde{e}} \end{bmatrix}^T$ in the best possible way with LQR the values of the matrices **Q** and **R** had to be estimated. Since there is a relationship between the matrices, we chose to let **R** remain as the identity and only change **Q**. Our approach was to use Bryson's rule as intial values, which tells to use:

$$
Q_{ii} = \frac{1}{maximum\ acceptable\ value\ of\ x_i^2}[1] \tag{57}
$$

By using equation(57) our initial Q was:

$$
\mathbf{Q} = k_u \begin{bmatrix} \frac{1}{max\ \tilde{p}^2} & 0 & 0 \\ 0 & \frac{1}{max\ \dot{\tilde{p}}^2} & 0 \\ 0 & 0 & \frac{1}{max\ \dot{\tilde{e}}^2} \end{bmatrix} = k_u \begin{bmatrix} \frac{1}{\frac{\pi}{3}^2} & 0 & 0 \\ 0 & \frac{1}{\frac{\pi}{9}^2} & 0 \\ 0 & 0 & \frac{1}{\frac{\pi}{9}^2} \end{bmatrix} \simeq k_u \begin{bmatrix} 0.911 & 0 & 0 \\ 0 & 2.74 & 0 \\ 0 & 0 & 2.74 \end{bmatrix} \tag{58}
$$

where k_u is the "extra" cost of matrix **Q**. With "extra" cost we mean in addition to the cost values for each state we have at the diagonal in matrix **Q**. Since a higher cost gives higher control input **u** we used this to decide how large we wanted the matrix **u** to be. The Q_{ii}-values were the values we thought were the maximal radians we would our helicopter on.

For choosing the values of **Q**, while **R** remained an identity matrix, we looked mainly at diagrams to see if the system became over-, under- or critically damped when following the reference from the joystick and some self made step responses. In addition we saw at the helicopter's level of aggressiveness which we took in consideration. After trying different values we concluded with:

$$
\mathbf{Q} \simeq 50 \begin{bmatrix} 1.46 & 0 & 0 \\ 0 & 1.83 & 0 \\ 0 & 0 & 1.2 \end{bmatrix} \qquad and \qquad \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{59}
$$

Significance of the states and references in matrices Q and R

When we were tuning the controller we tried different values. We found out that changing the value of $\tilde{p}$ resulted in a higher/lower slope for the response, where a higher value gave a faster response, but too large value gave an under damped system and opposite for lower value. Similarly changing $\dot{\tilde{p}}$ gave higher/lower damping of the pitch,where higher value gave higher damping which lead to slower response. And by changing $\dot{\tilde{e}}$ higher/lower slope for the elevation rate, where higher value gave faster response, but too high value gave under damping. We used this principle when we were tuning the parameters.

3.2.4 Comments on the choices for the controller design:

The controller system was fast, but unfortunately not quite accurate with the chosen parameters for the **Q** matrix. The response of the system can be seen in Figure 7 and Figure 8. We tried to minimize the inaccuracies by amplifying the gain for the different states which helped, but also made the helicopter very aggressive. Thus we decided to rather let it be. However we think that these inaccuracies were meant to happen due to that we didn't have any integral effect in our controller. Therefore there was no good tool to minimize the stationary difference which we clearly see in Figure 7.

In conlusion we were satisfied with the choices since the controller system was fast and more than average accurate.

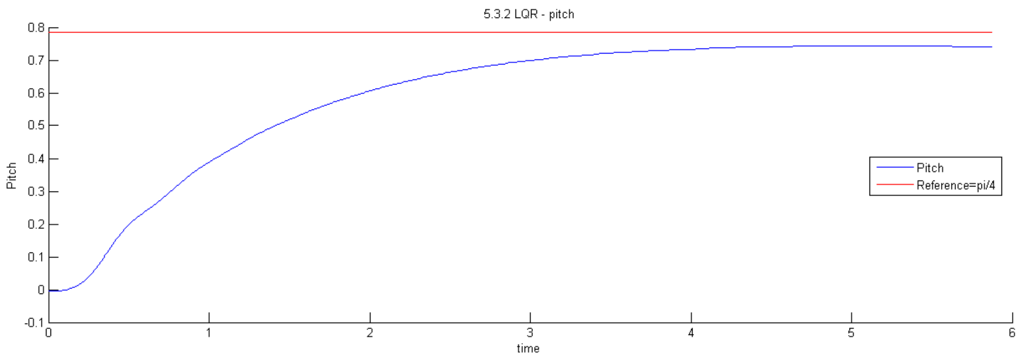

Figure 7: LQR - Pitch

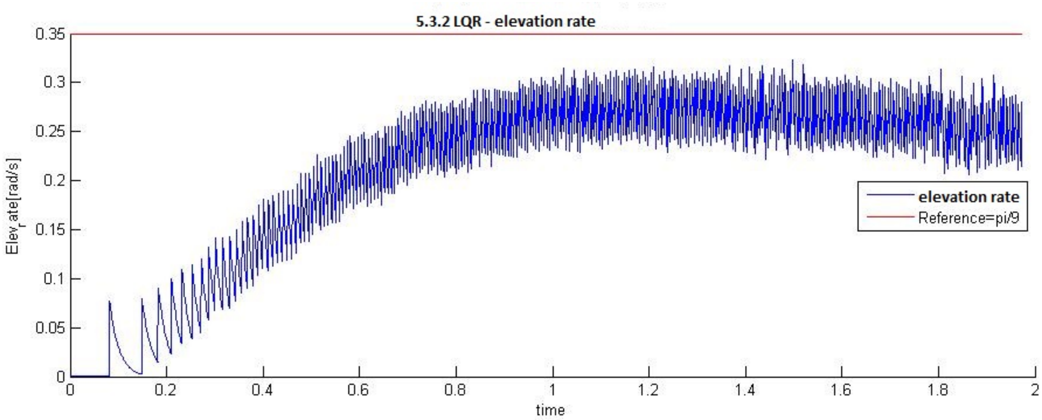

Figure 8: LQR - elevation rate

3.2.5 Matlab code

```
%LQR
A=[0,1,0;0,0,0;0,0,0];
B=[0,0;0,K1;K2,0];
Q=25*diag([0.2*14.6,1*0.6,0.5*4.8]);
R=diag([1,1]);
K=lqr(A,B,Q,R);
P=[K(1),K(5);K(2),K(6)];
```

3.3 5.3.3 Problem 3

Like problem 2 in 5.3.2 our aim was to track the $\mathbf{r} = \begin{bmatrix}\tilde{p_c} & \dot{\tilde{e}}\end{bmatrix}^T$ by using the controller from equation(52), which was:

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

The difference in this problem compared to problem 2 was that the state vector now was given by:

$$\mathbf{x} = \begin{bmatrix}\tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \\ \gamma \\ \zeta\end{bmatrix} \tag{60}$$

where:

$$\begin{aligned}\dot{\gamma} &= \tilde{p} - \tilde{p}_c \\ \dot{\zeta} &= \dot{\tilde{e}} - \dot{\tilde{e}}_c\end{aligned} \tag{61}$$

3.3.1 Finding matrix $\mathbf{P}$

To find $\mathbf{P}$ we first added $\mathbf{Cr}$ to equation(1) due to equation(61) so that we got:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{C}\mathbf{r} \tag{62}$$

By using same assumption as earlier, such that the limits from equation(54) gave $\dot{\mathbf{x}} = 0$, we got:

$$-(\mathbf{A} - \mathbf{BK})\mathbf{x}_\infty = (\mathbf{BP} + \mathbf{C})\mathbf{r}$$

$$-(\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} k_1 & k_3 & k_5 & k_7 & k_9 \\ k_2 & k_4 & k_6 & k_8 & k_{10} \end{bmatrix}) \begin{bmatrix} \tilde{p_c} \\ 0 \\ \dot{\tilde{e_c}} \\ 0 \\ 0 \end{bmatrix}$$
$$= (\begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_1 & p_3 \\ p_2 & p_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}) \begin{bmatrix} \tilde{p_c} \\ \dot{\tilde{e_c}} \end{bmatrix}$$

which can be written as:

$$\begin{gathered} 0 = 0 \\ K_1 k_2 \tilde{p_c} + K_1 k_6 \dot{\tilde{e_c}} = K_1 P_2 \tilde{p_c} + K_1 P_4 \dot{\tilde{e_c}} \\ K_2 k_1 \tilde{p_c} + K_2 k_5 \dot{\tilde{e_c}} = K_2 P_1 \tilde{p_c} + K_2 P_3 \dot{\tilde{e_c}} \\ -\tilde{p_c} = -\tilde{p_c} \\ -\dot{\tilde{e_c}} = -\dot{\tilde{e_c}} \end{gathered} \tag{63}$$

From equation(63) we clearly see that P is the same as in problem 2 in 5.3.2, which was equation(56):

$$\underline{\underline{\mathbf{P} = \begin{bmatrix} p_1 & p_3 \\ p_2 & p_4 \end{bmatrix} = \begin{bmatrix} k_1 & k_5 \\ k_2 & k_6 \end{bmatrix}}}$$

Note: This could also be seen from equation(52), since the matrix $\mathbf{P}$ is dependent of the reference $\mathbf{r}$. Since $\mathbf{r}$ remained unchanged so will the matrix $\mathbf{P}$.

3.3.2 Estimating matrices $\mathbf{Q}$ and $\mathbf{R}$

To track the reference $\mathbf{r} = \begin{bmatrix} \tilde{p_c} & \dot{\tilde{e}} \end{bmatrix}^T$ in the best possible way we followed the same procedure as in problem 2 in 5.3.2, except from that we just had γ and ζ as 0s in the beginning.

After trying different values we concluded with:

$$\mathbf{Q} \simeq 0.4 \begin{bmatrix} 500 & 0 & 0 & 0 & 0 \\ 0 & 200 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 500 \end{bmatrix} \qquad \textit{and} \qquad \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{64}$$

Comments on the choices for the controller design:
The controller system was fast and quite accurate with the chosen parameters for the $\mathbf{Q}$ matrix. The response of the system can be seen in Figure 9, Figure 10 and in Figure???. Due to the integral effect in form of γ the stationary difference was taken care of, and we can see that we just needed a low factor for it. A higher γ gave an under-damped system. As we can see we increased $\tilde{p}_c$ and $\dot{\tilde{p}}_c$, theoretically we shouldn't have done that, but it gave better response from the helicopter and followed the graph better. The helicopter was not very aggressive with these values

ζ took care of holding the helicopter at the last elevation position of the joystick, instead of the always falling back or rising to the intial value. However as we can see in Figure**??**e had an under-damping for the elevation. The problem with tuning the elevation was to decide for tuning when the helicopter was going upwards or downwards. The reason is that when the helicopter went upwards it had the gravitational force to help it stop at the reference so a low motor input was needed. On the downward direction however we needed a higher motor input such that the falling helicopter could come faster to the reference. In other words by making the elevation when going upwards better, we made the elevation downwards worse. We ended up between both of them, such that neither the upward or downward direction gave a very bad response.
BILDER

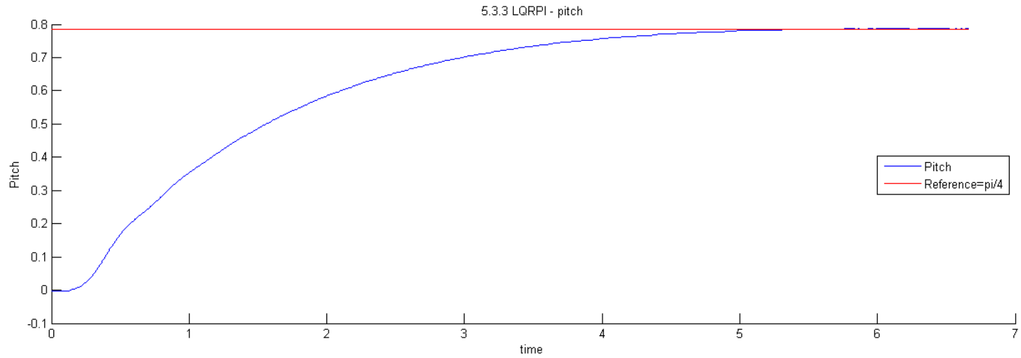

Figure 9: LQRPI - Pitch

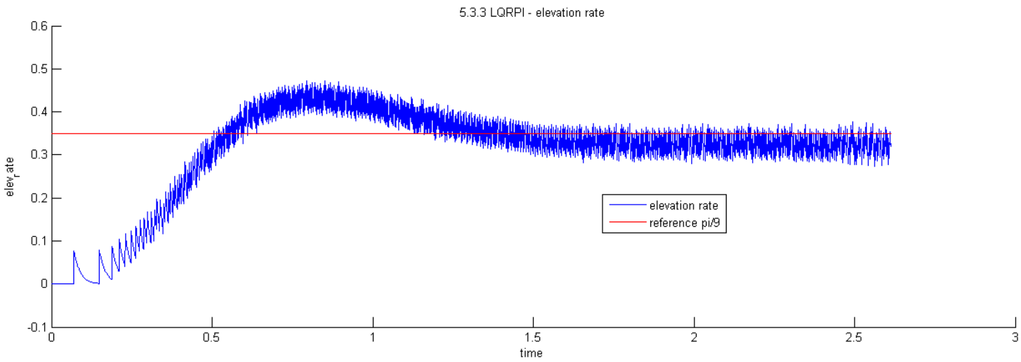

Figure 10: LQRPI - Elevation rate

3.3.3 Matlab code

```
%LQR-with PI
Q_int=0.4*diag([500,200,5,1,500]);
A_int=[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_int=[0,0;0,K1;K2,0;0,0;0,0];
C_int=[0,0;0,0;0,0;-1,0;0,-1];
K_int=lqr(A_int,B_int,Q_int,R);
P_int=[K_int(1),K_int(5);K_int(2),K_int(6)];
```

References

[1] $http://www.uz.zgora.pl/ wpaszke/materialy/kss/lqrnotes.pdf$

4 Part 4 - State Estimation

4.1 problem 1

The problem was to derive a state-space formulation of the system of the form:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{65}$$

Where

$$x = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix}, u = \begin{bmatrix} V_s \\ V_d \end{bmatrix}, y = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix}$$

By using (6a-6b) it was found that:

$$\begin{aligned} \dot{\tilde{p}} &= \dot{\tilde{p}} \\ \ddot{\tilde{p}} &= K_1 \tilde{V}_d \\ \dot{\tilde{e}} &= \dot{\tilde{e}} \\ \ddot{\tilde{e}} &= K_2 \tilde{V}_s \\ \dot{\tilde{\lambda}} &= \dot{\tilde{\lambda}} \\ \ddot{\tilde{e}} &= K_3 \tilde{p} \end{aligned}$$

This was used to find the Matrices A,B and C:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ K_3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ 0 & 0 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

4.2 Problem 2

First we checked the observality of the system. We did that by using the matlab function

$$rank(obs(A, C)) = 6$$

This function finds the observablitiy matrix and calculates the rank of it. The rank was 6 which is also the size of the A matrix and it means that the observalitity matrix is full rank. This means that the system is Observable. To create a linear observer this function was used:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) \tag{66}$$

Where L is the observer gain matrix. And the error functions is:

$$e(t) = x(t) - \hat{x}(t) \tag{67}$$

$$\dot{e}(t) = \dot{x}(t) - \dot{\hat{x}}(t) \tag{68}$$

By combining equations 66 and 65 with the error equation 68 we get

$$\begin{aligned}\dot{e}(t) &= Ax + Bu - (A\hat{x} + Bu + L(y - C\hat{x}))\\ \dot{e}(t) &= (A - LC)x(t) - (A - LC)\hat{x}(t)\end{aligned}$$

By using equation 67 we get

$$\dot{e}(t) = (A - LC)e(t) \tag{69}$$

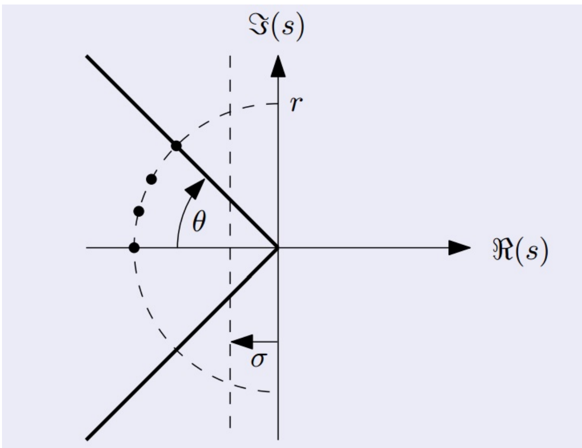

Figure 11: Finding the desired eigenvalue

This equation governs the estmation error. The eigenvalues of (A-LC) have to be assigned so the error goes to zero as time approaches infinity. We

used the figure 11 which can be found in the book "Linear system theory and design by CHI-TSONG CHEN" in page 290. This shows that greater the distance the eigenvalues have from the imaginary axis, the faster response will the system have, and greater the angle θ is, the greater overshoot will it be. But we also have to think that the eigenvalues have to be placed evenly around the circle to get fast response.
This principle was used to find the eigenvalues for the system. We wanted to have fast response and by experimenting we came up with these eigenvalues. Written in Matlab script:

$$poles = [-98 + 98i, -123 + 63i, -137 + 22i, -137 - 22i, -123 - 63i, -98 - 98i]$$

Where poles and eigenvalues is the same thing.
To find L matrix the matlab function place() was used:

$$L = place(A', C', poles).'$$

This gives the L matrix:

$$L = \begin{bmatrix} 227 & 1 & -67 \\ 15031 & 859 & -9232 \\ 3 & 248 & 28 \\ 1432 & 19121 & 5208 \\ 63 & -47 & 241 \\ 9861 & -6184 & 16465 \end{bmatrix} \tag{70}$$

This simulink diagram was made for the state estimate. The estimated

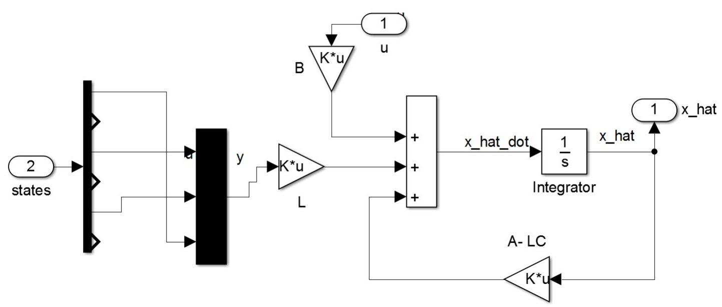

Figure 12: State Estimate simulink model

values of $\dot{\tilde{p}}$ and $\dot{\tilde{e}}$ was used as input for the LQR system

//feil med figurplasering

4.2.1 Discussion

If the eigenvalues was too small the system oscillated and was very unstable. The reason was because the estimator was too slow compared to measured values and the controller thought that the estimated values was accurate which gave oscillations. You can see the estimated and measured values of the pitch rate in figure 15.But if the eigenvalues was too large the system startet to vibrate and the controller gave to much power to the engines. The reason was because the estimator was too fast and had too high frequency and small change in system made the estimator to estimate too large values. You can see the result in figure 13. We had to find values that gave estimated values between these. And we ended up with very good estimator like you can see in figure 14

4.3 Problem 3

The same matlab script was used to find if the system was observable with the new y vector. If we only measure $\tilde{p}$ and $\tilde{e}$ we got the **C** matrix

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

Which gives us the rank:

$$rank(obs(A, C)) = 4$$

which is not full rank. This means that the system is not observable in this situation. But if we measure $\tilde{e}$ and $\tilde{\lambda}$ we get the **C** matrix:

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

Which gives the rank:

$$rank(obs(A, C)) = 6$$

This is full rank and the system is observable. We implemented this into Simulink.

4.3.1 Discussion

The same eigenvalues was used to start with. The pitch was very unstable and startet to oscillate with these eigenvalues. Even if the size of the eigenvalues was changed, the pitch was still very unstable. This shows that by only measuring $\tilde{e}$ and $\tilde{\lambda}$, estimator cannot have a good estimate for the pitch rate. This can be because of that even if the travel rate or the elevation rate is over zero we cannot be sure about what the pitch rate is. Pitch rate can be zero even if the travel rate or elevation rate is different from zero. Thats why this estimation is very poor for our system. You can see the the estimated and measured values in figure 16 where estimated value is too large and too slow compared to measured values.

Matlab code used to plot the graphs

```
%State estimated and measuered

close all;
tid_e=estimated.time;
value_e=estimated.signals.values
tid_m=measured.time;
value_m=measured.signals.values
hold on
I = (tid_e > 1.2) & (tid_e < 1.28);

plot(tid_e,value_m(:,6));
plot(tid_e,value_e(:,4),'r');
title('Comparing Pitch Rate')
ylabel('Radians/s')
xlabel('Time[s]')
legend('Measured','Estimated')
hold off
```

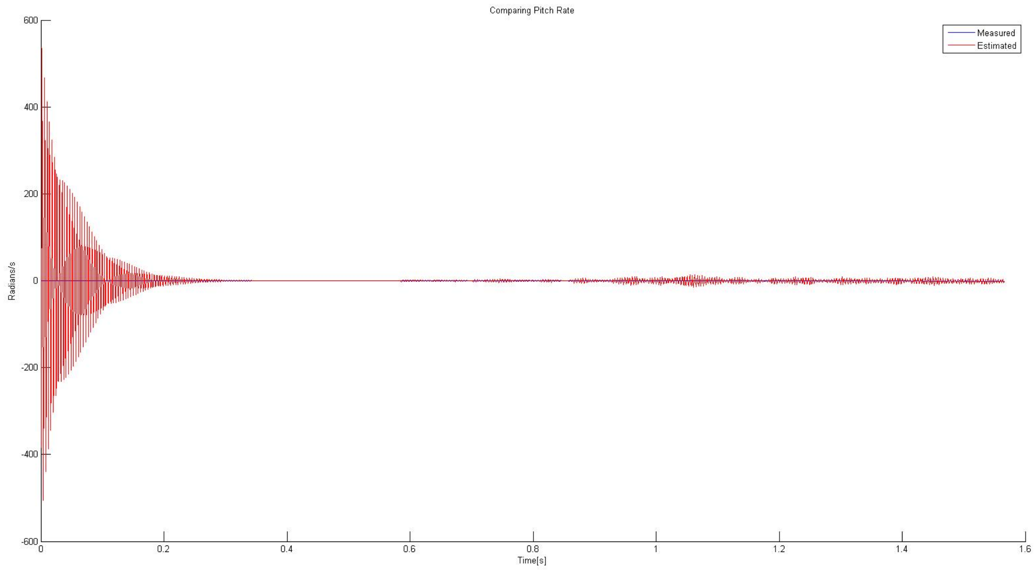

Figure 13: State Estimate with small eigenvalues

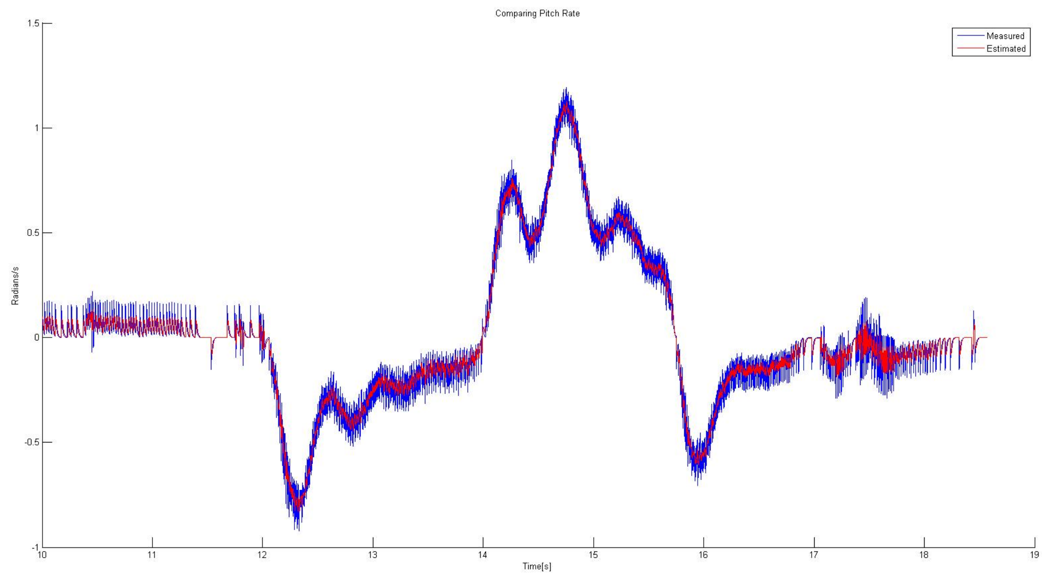

Figure 14: State estimate with choosen eigenvalues

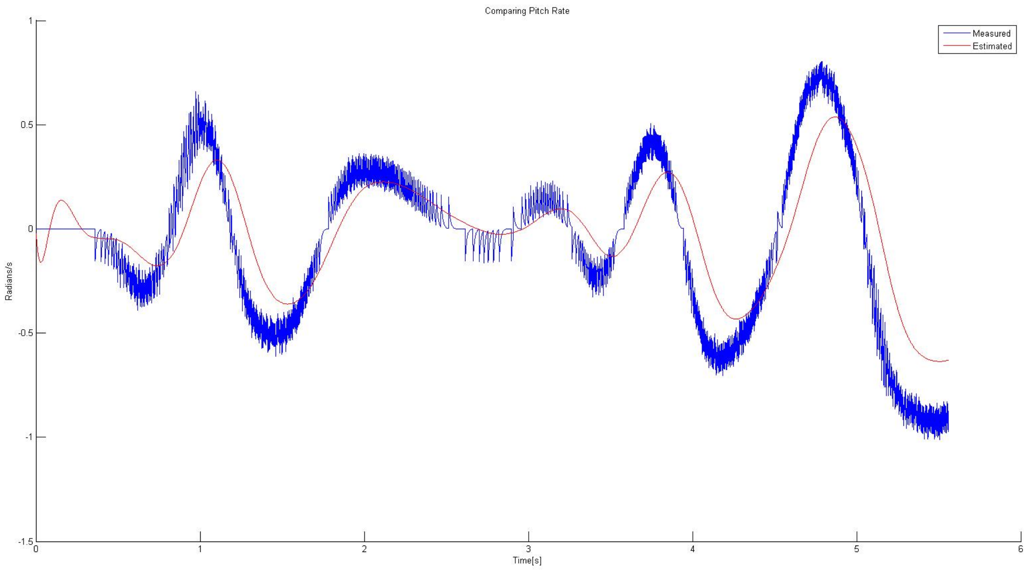

Figure 15: State estimate with large eigenvalues

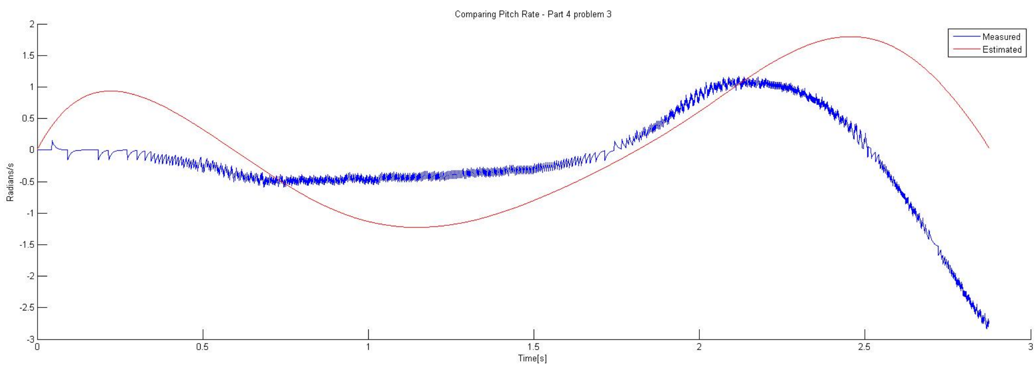

Figure 16: State estimate without measuring pitch