TTK4115 Linear System Theory
Helicopter Lab Report

Aljumaili, Ali
772170

Espeland, Håkon
772703

August 29, 2018

Group 63

NTNU – Trondheim
Norwegian University of
Science and Technology

Contents

List of Figures

1 Part I - Mathmatical modeling

In this chapter we construct a mathematical model, defining the helicopters three axes; pitch, elevation and travel.

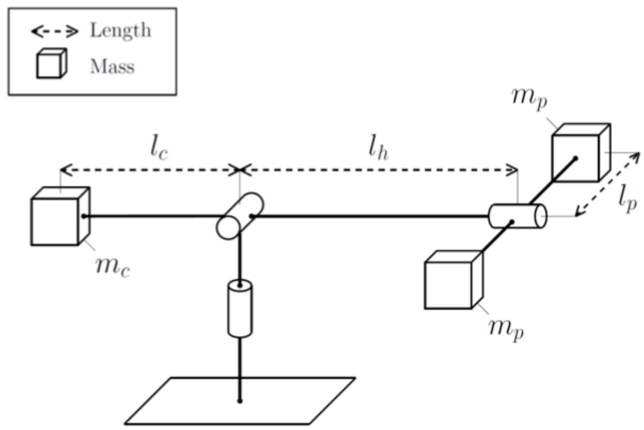

Figure 1: Model dimensions

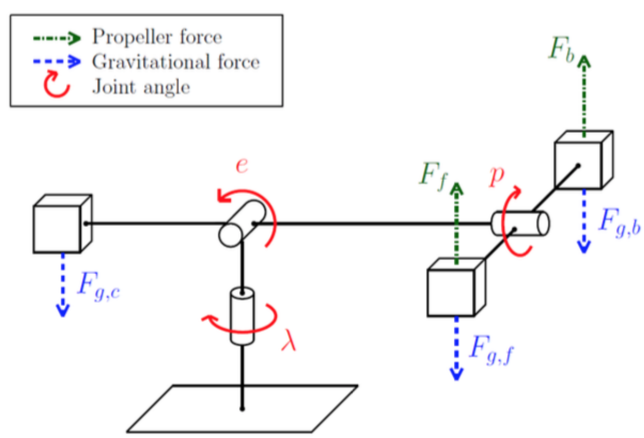

Figure 2: Model momentum and forces

A relationship between the motor force and the motor voltage is given by:

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

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

Equations (1) and (2) show that the relationship between motor force and motor voltage is proportional to a constant K_f. The equations describing the plant are then transformed into a state-space model.

1.1 Problem 1 - Derivation of the mathematical model

Newton's second law gives a relationship between the net external torque and the angular acceleration, this relationship will be used to derivate the mathmatical model for pitch, elvation and travel.

1.1.1 Derivation of pitch

Sum of torque on the pitch axis is given as: $\Sigma\tau = J_p\ddot{P}$.

Further, the magnitude of torque is given by $\tau = r \times F = |r||F|\sin(\theta)$, where θ is the angle between the force vector and the lever arm vector. We consider $\theta = 90^\circ$ to derive the following pitch equation:

$$\begin{aligned}
\Sigma\tau_\tau &= rF \\
&= (F_f - F_b)l_p \\
&= K_f l_p (V_f - V_b) && \text{(substituting 1 and 2)} \\
&= K_f l_p V_d && (V_d = V_f - V_b) \\
&= L_1 V_d = J_p\ddot{P} && \text{(Where } L_1 = K_f l_p \text{ is a constant)}
\end{aligned}$$

$$J_p\ddot{P} = L_1 V_d \tag{3}$$

1.1.2 Derivation of elevation

The forces acting on the elevation axis are $F_{g,c}$ and $F_f + F_B$ giving the vertical thrust The sum of $F_f + F_B$ determines the vertical thrust. On the elevation axis both $F_{g,c}$ and the sum of the two forces F_f and F_b giving the thrust vertical to the x axis.

$$\begin{aligned}
\Sigma\tau_e &= rF \\
&= m_c g l_c \cos(e) - 2m_p g l_h \cos(e) + (K_f V_f + K_f V_b) l_h \cos(p) \\
&= (m_c l_c - 2m_p l_h) g \cos(e) + l_h K_f V_s \cos(p) = J_e \ddot{e} && \text{(substituting 1 and 2)} \\
&= L_2 \cos(e) + L_3 \cos(p) v_s = J_e \ddot{e} && (\ L_2 = (m_c l_c - 2m_p l_h) g,\ L_3 = l_h K_f)
\end{aligned}$$

$$J_e \ddot{e} = L_2 \cos(e) + L_3 \cos(p) v_s \tag{4}$$

1.1.3 Derivation of travel axis

There are two forces acting on the travel axis, thrust force and rotor's torque has an impact on the elevation axis

$$\begin{aligned}
\Sigma\tau_\lambda &= rF \\
&= L_4 V_s \cos(e) \sin(p) = J_\lambda \ddot{\lambda} && \text{(Where } L_4 = L_3 = l_h K_f)
\end{aligned}$$

$$J_\lambda \ddot{\lambda} = L_4 V_s \cos(e) \sin(p) \tag{5}$$

1.2 Problem 2 - Linearization

In this section we approximate the equations of motions: (3), (4), (5) around our desired operating point of $p^* = e^* = \lambda^* = 0$

1.2.1 Linerizing the euqations of motions

We transform the equations of motions in (6) using a new coordinatte transofrmation (6)

$$\begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} = \begin{bmatrix} p \\ e \\ \lambda \end{bmatrix} - \begin{bmatrix} p^* \\ e^* \\ \lambda^* \end{bmatrix} \text{ and } \begin{bmatrix} \tilde{V_s} \\ \tilde{V_d} \end{bmatrix} = \begin{bmatrix} V_s \\ V_d \end{bmatrix} - \begin{bmatrix} V_s^* \\ V_d^* \end{bmatrix} \tag{6}$$

We set the derivative of our full state vector to zero, to find the equilibruim points of the system. Assuming $(\dot{p}, \dot{e}, \dot{\lambda})^T = (0, 0, 0)^T \Rightarrow (\ddot{p}, \ddot{e}, \ddot{\lambda})^T = (0, 0, 0)^T$

$$J_p \ddot{p} = L_1 V_d^* = 0 \qquad \Rightarrow \qquad V_d^* = 0 \tag{7}$$

$$J_e \ddot{e} = L_2 \cos(e^*) + L_3 V_s^* \cos(p^*) = 0 \qquad \Rightarrow \qquad V_s^* = -\frac{L_2}{L_3} \tag{8}$$

$$J_\lambda \ddot{\lambda} = L_4 V_s \cos(e^*) \sin(p^*) = 0 \tag{9}$$

Transforming the equations of motions (3) - (5) using equation (6) we get the following:

$$J_p\ddot{p} = L_1 V_d \quad \Rightarrow \quad J_p\ddot{\tilde{p}} = L_1\left(\tilde{V}_d - V_d^*\right) \tag{10a}$$

$$J_e\ddot{e} = L_2\cos(e) + L_3 V_s\cos(p) \quad \Rightarrow \quad J_e\ddot{\tilde{e}} = L_2\cos(\tilde{e}) + L_3\left(\tilde{V}_s + V_s^*\right)\cos(\tilde{p}) \tag{10b}$$

$$J_\lambda\ddot{\lambda} = L_4 V_s\cos(e)\sin(p) \quad \Rightarrow \quad J_\lambda\ddot{\tilde{\lambda}} = L_4\left(\tilde{V}_s + V_s^*\right)\cos(\tilde{e})\sin(\tilde{p}) \tag{10c}$$

The system of equations (10 a) - (10 c) is linearised around the point $(p^*, e^*, \lambda^*)^T = (0,0,0)^T$ and $(\tilde{V}_s, V_s{}^*)^T = (0, -\frac{L2}{L3})^T$:

$$\begin{bmatrix} \ddot{\tilde{p}} \\ \ddot{\tilde{e}} \\ \ddot{\tilde{\lambda}} \end{bmatrix} = \begin{bmatrix} \frac{\partial\ddot{\tilde{p}}}{\partial\tilde{p}} & \frac{\partial\ddot{\tilde{p}}}{\partial\tilde{e}} & \frac{\partial\ddot{\tilde{p}}}{\partial\tilde{\lambda}} \\ \frac{\partial\ddot{\tilde{e}}}{\partial\tilde{p}} & \frac{\partial\ddot{\tilde{e}}}{\partial\tilde{e}} & \frac{\partial\ddot{\tilde{e}}}{\partial\tilde{\lambda}} \\ \frac{\partial\ddot{\tilde{\lambda}}}{\partial\tilde{p}} & \frac{\partial\ddot{\tilde{\lambda}}}{\partial\tilde{e}} & \frac{\partial\ddot{\tilde{\lambda}}}{\partial\tilde{\lambda}} \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} + \begin{bmatrix} \frac{\partial\ddot{\tilde{p}}}{\partial\tilde{V}_d} & \frac{\partial\ddot{\tilde{p}}}{\partial\tilde{V}_s} \\ \frac{\partial\ddot{\tilde{e}}}{\partial\tilde{V}_d} & \frac{\partial\ddot{\tilde{e}}}{\partial\tilde{V}_s} \\ \frac{\partial\ddot{\tilde{\lambda}}}{\partial\tilde{V}_d} & \frac{\partial\ddot{\tilde{\lambda}}}{\partial\tilde{V}_s} \end{bmatrix} \begin{bmatrix} \tilde{V}_d \\ \tilde{V}_s \end{bmatrix}$$

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

$$\begin{bmatrix} \ddot{\tilde{p}} \\ \ddot{\tilde{e}} \\ \ddot{\tilde{\lambda}} \end{bmatrix} 2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ K_3 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} + \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_d \\ \tilde{V}_s \end{bmatrix}$$

Which is the same as the following equations:

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

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

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

where the constants K_1, K_2 and K_3 are given by

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

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

$$K_3 = -\frac{L_4 L_2}{J_\lambda L_3} \tag{12c}$$

The momement of inertia constants are given by:

$$J_p = 2m_p l_p^2 \tag{13a}$$
$$J_e = m_c l_c^2 + 2m_p l_h^2 \tag{13b}$$
$$J_\lambda = m_c l_c^2 + 2m_p(l_h^2 l_p^2) \tag{13c}$$

Making a script in MATLAB to calculate all of the constants is done, in order to quickly change values when changing helicopters in the lab.

Constants MATLAB-script:

```
% Contains the physical constants for the helicopter
init_heli

% Measuring V at equilibrium position:
V_s_eq = 7.23169164775339;
V_f = V_s_eq/2;
V_b = V_s_eq/2;

K_f = -(m_c*l_c-2*m_p*l_h)*g/(l_h*V_s_eq);

% Calculating initial conditions:
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));

% Calculating L1-L4:
L1 = K_f*l_p;
L2 = (m_c*l_c-2*m_p*l_h)*g;
L3 = l_h*K_f;
L4 = l_h*K_f;

% Calculating K-constants:
K1 = L1/J_p;
K2 = L3/J_e;
K3 = (-L4*L2)/(J_l*L3);
```

1.3 Problem 3 - Feed forward control

The first attempt to control the helicopter was done by using feed forward control, connecting the joystick x-axis directly to the voltage difference V_d and the y-axis directly to the voltage sum V_s.

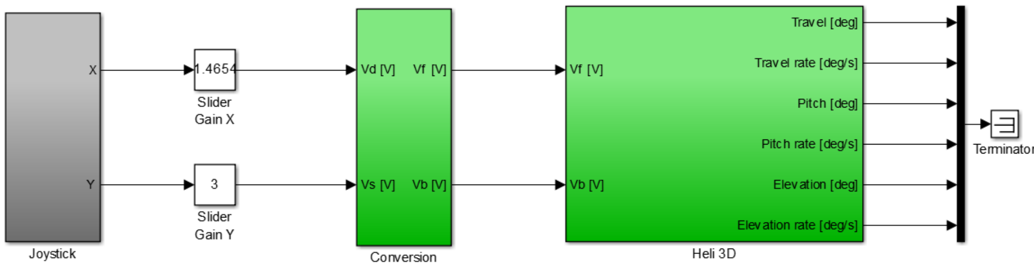

Figure 3: Feed forward control with simulink

1.3.1 Discussion: Physical behaviour vs. theoretical model

In this part we will discuss some of the observations we obtained using feed forward control.

When we first attempted to control the helicopter for the first time, we began with a slider gain between the y-axis on the joystick and the voltage sum V_s, to find an appropriate gain to satisfy our demands on sensitivity.

Connecting the x-axis to V_d would allow the joystick to interfere with the voltage difference that implies a difference in force from the two propellers. This would cause pitch angle $\neq 0$ and the helicopter starts travelling around the travel-axis. To make observations possible, we needed a gain < 1 in order to get a descent pitch sensitivity.

The theoretical model is described by the following equations from the assignment text: (3-5) and the linearised version around our operation point (

When actually controlling the helicopter, there are some discrepancies to look into. For instance, the theoretical model does not account for external disturbances both in environment and electronics. Feeding the system with a static input does not imply a static behaviour. The non-static behaviour makes the system more sensitive to disturbances, making it rather hard to control. Also if you approach a more dynamic kind of control, the intensity of the behaviour outperforms the human mind when it comes to time-responses on corrective inputs.

Comparing the behaviour of the system with the linearised equations, they are intended to the equilibrium point. Deviation from this point will create large differences between the theoretical linearised model and the physical behaviour.

At last, there are no feedback as long as the system is connected this way, meaning that the only feedback is what you see, and thats rather difficult to rely on.

1.4 Problem 4 - Estimation of K_f

Before we can implement a controller that is based on the linearized equiations of motion, we need to determine the motor force constant K_f. By adding a "to-file" measurement on both V_s and elevation angle $[e]$ in the simulink model, we were able to obtain V_s^*, the voltage sum at the equilibrium point.

Measurement resulted in $V_s^* = 7.23V$.

$$L_2 = (m_c l_c - 2m_p l_h)g \tag{14}$$

$$L_3 = l_h K_f \tag{15}$$

$$V_s{}^* = -\frac{L_2}{L_3} \tag{16}$$

By using (14), (15) and (16), we calculated the motor force constant K_f to be:

$$K_f = -\frac{(m_c l_c - 2m_p l_h)g}{l_h V_s{}^*}$$

$$K_f = -\frac{(1,92 \cdot 0,46 - 2 \cdot 0,72 \cdot 0,66) \cdot 9,81}{0,66 \cdot 7,23} \left[\frac{kg \cdot m \cdot m/s^2}{m \cdot V}\right] \tag{17}$$

$$K_f = 0,1382[N/V]$$

2 Part II – Mono-variable control

2.1 Problem 1 - PD controller

A PD controller is added to control the pitch angle p. This controller is given as:

$$\widetilde{V}_d = K_{pp}(\widetilde{p}_c - \widetilde{p}) - K_{pd}\dot{\widetilde{p}} \tag{18}$$

with constants K_{pp}, $K_{pd} > 0$, where $\widetilde{p}_c$ is the desired reference for the pitch angle $\widetilde{p}$.

from (6a):

$$\ddot{\widetilde{p}} = K_1 \widetilde{V}_d \tag{19}$$

Substituting $\widetilde{V_d}$

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

Applying laplace transformation to obtain the transferfunction H(s):

$$s^2 \tilde{P} = K_1 K_{pp}(\tilde{P}_c - \tilde{P}) - K_1 K_{pd} \tilde{P} s$$

$$\frac{\tilde{P}(s)}{\tilde{P}_c(s)} = \frac{K_1 K_{pp}}{s^2 + sK_1 K_{pd} + K_1 K_{pp}} \tag{20}$$

Criterias for the controller:

- No oscillations (Critically damped $\zeta = 1$).
- Responsive pitch regulator.

$$H(s) = \frac{w_n^2}{s^2 + 2\zeta w_n s + w_n^2}$$

The general equation H(s) provides a relationship between w_n and ζ, which we can use to obtain the relationship between K_{pp} and K_{pd}.

Looking at Equation:(20), we extracts values to calculate K_{pp} and K_{pd}:

$$K_1 K_{pd} = 2\zeta\omega_n$$

$$K_1 K_{pp} = {\omega_n}^2 \rightarrow \omega_n = \sqrt{K_1 K_{pp}}$$

Substituting ω_n:

$$K_1 K_{pd} = 2\zeta\sqrt{K_1 K_{pp}}$$

Computing the equation provides:

$$\left(\frac{K_1 K_{pd}}{2\zeta}\right)^2 = K_1 K_{pp}$$

$$K_{pp} = \frac{K_1 {K_{pd}}^2}{4\zeta^2}$$

Choosing ζ to be equal to 1, avoiding oscillations, we obtain:

$$K_{pp} = \frac{K_1 {K_{pd}}^2}{4} \tag{21}$$

From (21) the relationsship between K_{pp} and K_{pd} is expressed. As of the transfer function $H(s)$, changing these values result in change in the placement of the poles. Solving the equation for the polynomial denominator in the transfer function we see that negative gains places the poles in the right-half-plane, which will result in an unstable system. To get a stable system we need positive gains, placing the poles in the left-half-plane.

$$s = \frac{-K_1 K_{pd} \pm \sqrt{{K_1}^2 {K_{pd}}^2 - 4K_1 K_{pp}}}{2} \tag{22}$$

In theory increasing the gains would decrease the response time, but the hardware does have a limit on maximum input and output values. Closing in on these limit-values might cause oscillations due to limitations in system performance.

After testing with different values for the gains, we found the appropriate gains to be: $K_{pd} = 10$ $K_{pp} = 21.5$

Pitch MATLAB-script

```
%Pitch Controller Gains
K_pd = 10;
K_pp = K1*K_pd^2/4;
```

2.2 Problem 2 - Travel rate

Estimating the travel rate $\dot{\tilde{\lambda}}$, using a simple P controller:

$$\tilde{p}_c = K_{rp}(\dot{\tilde{\lambda}}_c - \dot{\tilde{\lambda}}) \tag{23}$$

Inserting (11c) into (23), while assuming $\widetilde{p}_c = p$ we obtain the following:

$$\ddot{\tilde{\lambda}} = K_3 K_{rp}(\dot{\tilde{\lambda}}_c - \dot{\tilde{\lambda}}) \tag{24}$$

Applying Laplace transform:

$$s\dot{\tilde{\lambda}}(s) = K_3 K_{rp}(\dot{\tilde{\lambda}}_c(s) - \dot{\tilde{\lambda}}(s)) \tag{25}$$

$$\dot{\tilde{\lambda}}(s) = \frac{K_3 K_{rp} \dot{\tilde{\lambda}}_c(s)}{s + K_3 K_{rp}} \tag{26}$$

$$\frac{\dot{\tilde{\lambda}}(s)}{\dot{\tilde{\lambda}}_c(s)} = \frac{p}{s + p} \tag{27}$$

where $p = K_3 K_{rp}$. Desired maximum travel rate is set to be $\frac{\pi}{4}$ rad/s, keeping $\left|\dot{\tilde{p}}_c\right| < \frac{\pi}{4}$.

3 Part III - Multi-variable control

The objective of this chapter is to **control** the helicopter using a multiple-input/multiple-output (MIMO) control system, to obtain a desirable behaviour of several output states by simultaneously manipulating multiple input channels.

3.1 Problem 1 - Deriving a multi-varible controller

We put the linerized equations **(11a - 11c**) into the (28) state-space representation.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{28}$$

We get the following state space system

$$\begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \ddot{\tilde{e}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{29}$$

3.2 Problem 2- LQR

Controlling the system using the LQR method, allows more flexibility when tuning the system, and is the preferred method for multi-input/multi-output (MIMO) systems. The LQR control method places the poles of the system at the optimal place

The poles of the system are placed at the optimal place, by selecting on the weighing of matrices Q and R. In order to to do this the system must be controllable.

Controllability is a requirement that the output of the system can be steered to any desired position by manipulating the input. The state-space model in (36) is fully controllable if and only if the matrix C= $\begin{bmatrix} B & AB & A^2B \end{bmatrix}$ has full row rank. We test this in our matlab script and it is fully controllable,

LQR MATLAB-script

```
LQR - Without Integral effect
A_LQR = [0 1 0; 0 0 0; 0 0 0];
B_LQR = [0 0; 0 K1; K2 0];
C_LQR = [1 0 0; 0 0 1];
D_LQR = 0;

Rank_LQR = rank(ctrb(A_LQR,B_LQR)); %check if A Matrix is full rank

R_LQR = [1 0;0 1];
Q_LQR = 200*diag([2, %pitch
                  2/3, %pitch rate
                    3]); % elevation rate

 [K_LQR,S_LQR,E_LQR]=lqr(A_LQR,B_LQR,Q_LQR,R_LQR); % Calculate the gain, and the eigenvalues
P_LQR = inv(C_LQR*inv(B_LQR*K_LQR-A_LQR)*B_LQR);
```

A controller of the following form is desired

$$\mathbf{u} = \mathbf{Pr} - \mathbf{Kx} \tag{30}$$

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

$$J = \int_0^\infty \left(\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)\right) dt \tag{31}$$

Bryson's rule was used as a starting point for finding reasonable matrices $\mathbf{Q}$ and $\mathbf{R}$. It states that they should be chosen to be diagonal with

$$Q_{ii} = \frac{1}{\text{maximum acceptable value of } z_i^2}, \quad i \in 1, 2, \ldots, l \tag{32a}$$

$$R_{jj} = \frac{1}{\text{maximum acceptable value of } u_j^2}, \quad i \in 1, 2, \ldots, k \tag{32b}$$

The chosen values is showed in the MATLAB script.

The choice of weighting matrices Q and R is a tradeoff between control performance (Q Large) and low input energy (R large), increasing both Q and R by the same factor leaves the optimal solution invariant. We tune the Q and R to the following:

$$\mathbf{Q} = \begin{bmatrix} 400 & 0 & 0 \\ 0 & 133.33 & 0 \\ 0 & 0 & 600 \end{bmatrix} \quad \text{and} \quad \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{33}$$

The corresponding matrix **K** was then obtained by using the MATLAB command `K=lqr(A,B,Q,R)`:

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 24.49 \\ 20 & 14.3636 & 0 \end{bmatrix} \tag{34}$$

Figure 4 and 5 show how the final system was setup in Simulink.

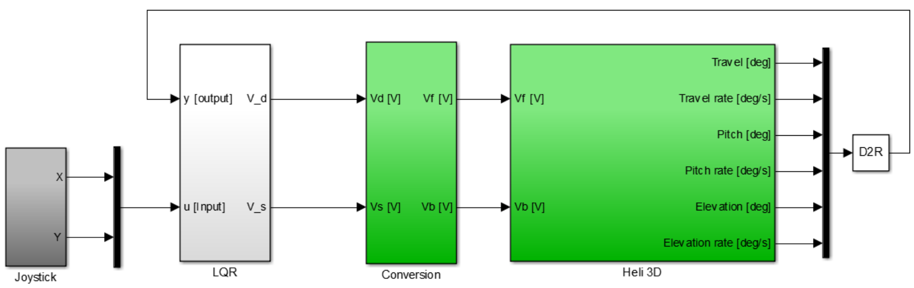

Figure 4: LQR with simulink

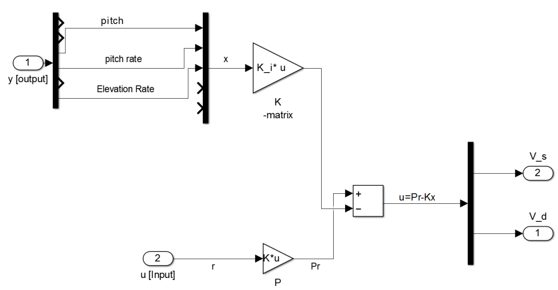

Figure 5: Inside LQR-Block

3.3 Problem 3 - Adding an integral effect

Due to model imperfections, external disturbances and similar effects the controller in its current form will not be able to reach a steady-state tracking error of zero. To account for this we can add an "integral" part. This is done by adding two new state variables:

$$\begin{aligned} \dot{\gamma} &= \tilde{p} - \tilde{p}_c \\ \dot{\zeta} &= \dot{\tilde{e}} - \dot{\tilde{e}}_c \end{aligned} \tag{35}$$

The integral effect accounts for disturbances on the system. The output will tend towards a constant reference.

The new system is given by the following equation:

$$\begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \ddot{\tilde{e}} \\ \dot{\gamma} \\ \dot{\zeta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \\ \gamma \\ \zeta \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{36}$$

The system is tested for controllability and the weighing Q matrix is first tested with the Bryson's Rule start values, further a trial and error approached was used to find a satisfactory tuning. This is shown in the following MATLAB-Script

LQR with integral MATLAB-script

```
%Including Integral effect.
A_i = [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_i = [0 0;0 K1;K2 0; 0 0; 0 0];
C_i = [1 0 0 0 0;0 0 1 0 0];
D_i = 0;
Rank_i = rank(ctrb(A_i,B_i)); %check if A Matrix is full rank

R_i = diag([1,1]);
Q_bry = [(1/(pi/4)^2) 0 0;0 (1/(pi/16)^2) 0;0 0 1/(pi/16)^2];
Q_i = 100 . [[2.2 0 0 0 0];[0 0.8 0 0 0];[0 0 2 0 0];[0 0 0 1/20 0];[0 0 0 0 5]];

[K_i,S_i,E_i] = lqr(A_i,B_i,Q_i,R_i); %Find state feedback K gain vector
P_i = pinv(B_i)*(B_i*K_i-A_i)*pinv(C_i);
%P_i = K_i*C_i'; %another method
```

The LQR-block is changed to add two

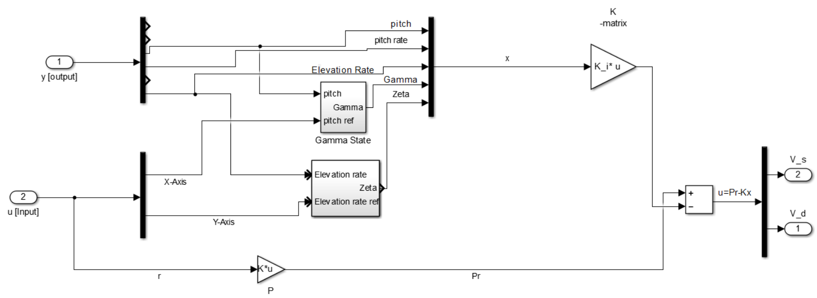

Figure 6: Inside LQR-Block with Integral

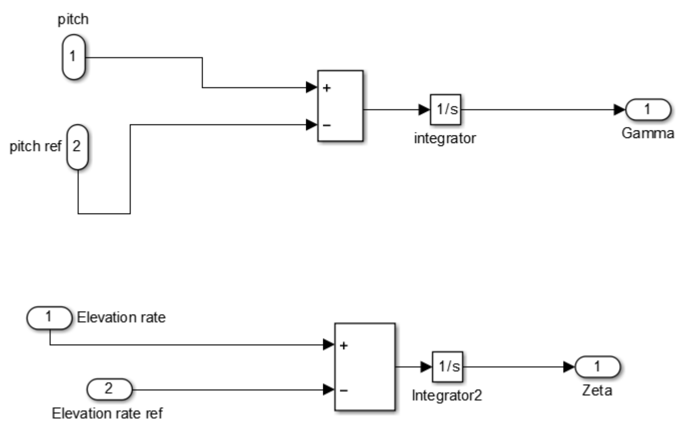

Figure 7: Inside Gamma and Zeta

4 Part IV – State estimation

In this section an observer is developed in order to estimate the non-measured states instead of using numerical differentiation.

4.1 Problem 1 - State space formulation

Derivation of state-space formulation of the system on the form:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{37}$$

where;

$$x = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix}, \quad u = \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \text{ and } \quad y = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} \tag{38}$$

By using the equations (11a) - (11c) we obtained the derived state-space formulation:

$$\dot{x} = \begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \dot{\tilde{e}} \\ \ddot{\tilde{e}} \\ \dot{\tilde{\lambda}} \\ \ddot{\tilde{\lambda}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ K_3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ 0 & 0 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{39}$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x} \tag{40}$$

4.2 Problem 2 - Observer

Considering a system of the form:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) \tag{41}$$

$$\dot{\hat{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ K_3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \hat{x} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ 0 & 0 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} u + L \left(y - \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \hat{x} \right)$$

Estimator MATLAB-script:

```
1  %Observability
2  Ob = obsv(A_s,C_s);
3  Orank = rank(Ob);
4
5  %Observer
6  es = eig(A_i-B_i*K_i);
7  r0 = max(abs(es));
8
9  %Radial multiplier & sector
10 fr = 7.5;
11 phi = pi/4;
12 r = r0*fr;
13
14 spread= -phi:(2*phi/(5)):phi;
15 poles = -r *exp(1j*spread);
16
17 %Gain matrix L
18 L_0 = place((A_s)', (C_s)', poles);
19 L_s = L_0';
```

Using the **Estimator MATLAB-script0**, we find that $\mathcal{O}$ has full rank which implies that the system is observable. By computing the eigenvalues for $A-BK$ we find the most negative pole to be $-0.3815 + 0i$, which we want to multiply with fr to obtain our eigenvalues for the observer.

As a thumb-rule, placing the new poles 5-10 times larger than the old ones, thus $fr = \frac{10+5}{2} = 7.5$. Then we multiply this with r placing 6 poles with an angle of ϕ between them. The pole-placements determines the gain matrix L_s through line [18] and [19] in the script.

Pole-placement is a method for placing the closed-loop-poles in pre-determined locations in the s-plane. Placing poles is desirable because the location of the poles determines the eigenvalues of the system, which again controls the characteristics of the response of the system. Having the poles placed in the left-half-plane is the desirable case.

Considerations when placing poles:

- Close spacing of eigenvalues results in sluggish response and a requirement of large input.
- Large σ results in fast response and requirement of large input.
- Large r results in fast response and requirement of large input.
- Large θ results in greater overshoot.

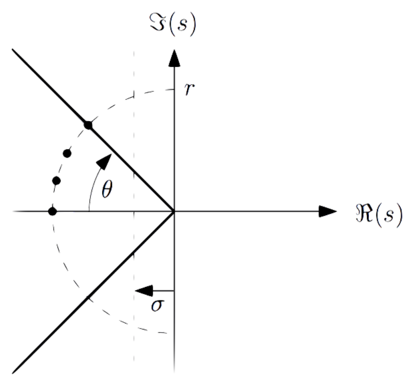

Figure 8: Poleplacement

Reflecting over the pole-placement, the hardware specifications have to be accounted for. The signals for inputs and outputs have maximum values, which causes a maximum reasonable limit for σ. Increasing σ further would maximize the output-values of the signals. By this there will be no middle-values, the controller requires larger values than what is actually possible, resulting in vibrations and oscillations.

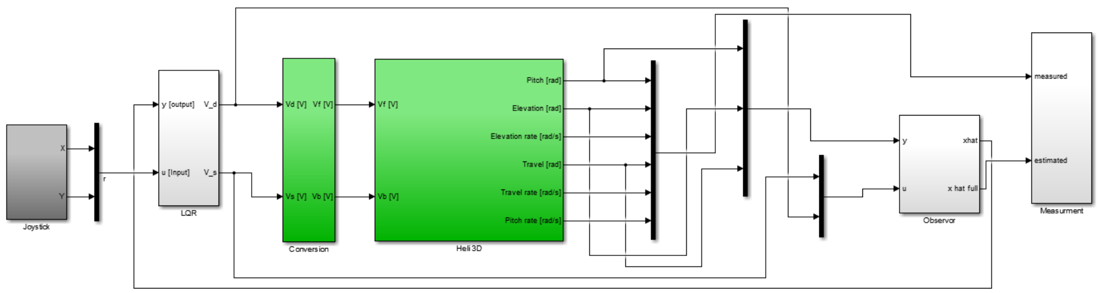

Figure 9: Observer - SimuLink Model

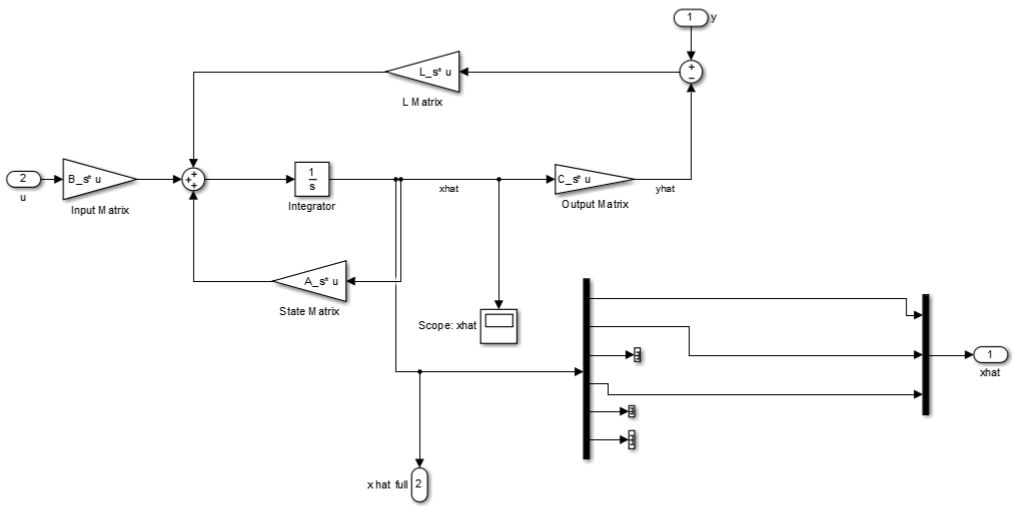

Figure 10: Observer block

4.3 Problem 3 - Measurement Observability

In this section we are going to develop a linear observer based on the measurement vector **y** and plot the measured states together with the estimated states. The measurement vector is given as:

$$y = \begin{bmatrix} \tilde{e} \\ \tilde{\lambda} \end{bmatrix} \tag{42}$$

The following MATLAB-script were used to calculate the observability of the different measured states: **Estimator MATLAB-script:**

```
%Pole-placement Problem 4.3
fr_3 = 1;
phi_3 = pi/3.5;
r_3 = r0*fr_3;
spread_3 = -phi_3:(phi_3/2.5):phi_3;
prob_3_poles = -r_3 * exp(1i * spread_3);

%Matrices
C_1 = [0 0 1 0 0 0;0 0 0 0 1 0];
C_2 = [1 0 0 0 0 0;0 0 1 0 0 0];

Ob_1 = rank(obsv(A_s,C_1)); %Observable
Ob_2 = rank(obsv(A_s,C_2)); %Not observable

L_1_s = )place((A_s)', (C_s)', poles))';
```

Using the given measurement vector, new matrices for C and L had to be calculated in **MATLAB**.

Determinations:

$$y = \begin{bmatrix} \tilde{e} \\ \tilde{\lambda} \end{bmatrix} \rightarrow rank(\mathcal{O}) = 6$$

The observability matrix has rank 6 → The system is observable.

Analysing the performance of the new observer, we could clearly see that the pitch and pitch rate had poor performane, due to many factors, among that the pitch is the unmeasured state and relies only on the model.

$$y = \begin{bmatrix} \tilde{p} \\ \tilde{e} \end{bmatrix} \rightarrow rank(\mathcal{O}) = 4$$

The observability matrix has rank 4 → The system is not observable.

The state estimate for $\tilde{\lambda}$ has a second-derived $\ddot{\tilde{\lambda}}$ that is dependant of $\tilde{p}$, making it rather hard to both estimate and measure. It can still be done, if the poles are placed correctly. The model itself, is linearised, thus the state estimator is prone to linearization errors.

4.4 Graphs

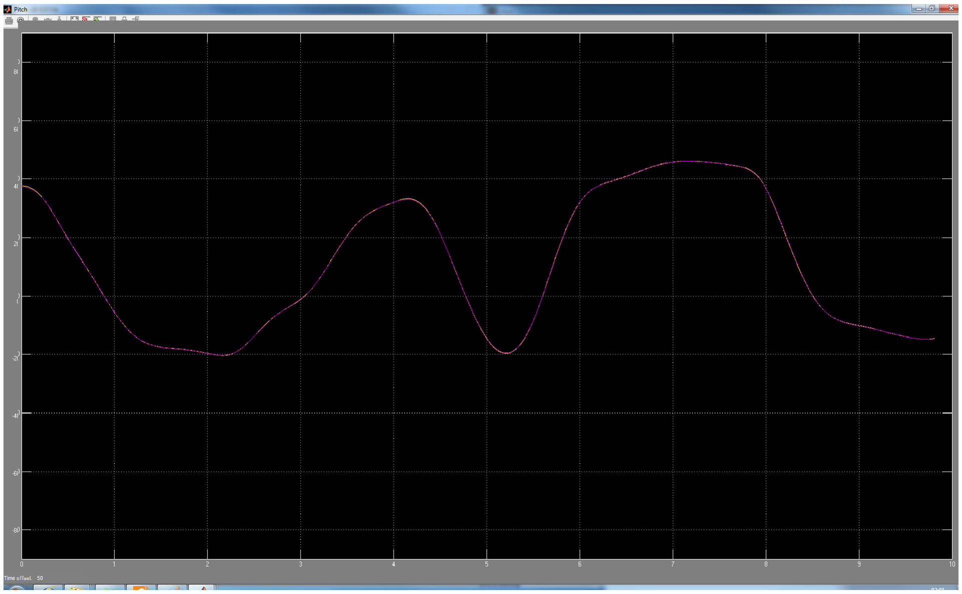

Figure 11: Pitch Estimator

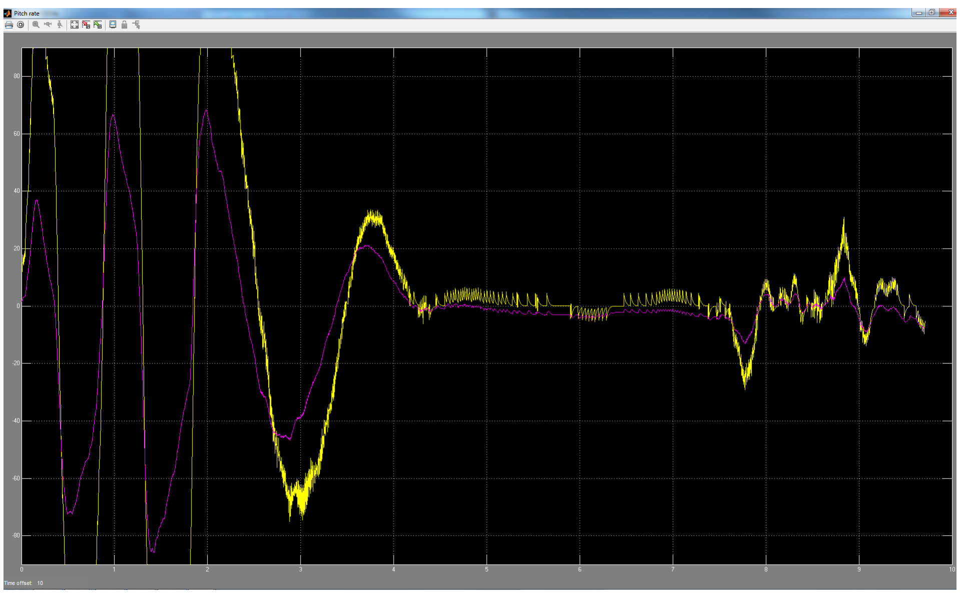

Figure 12: Pitch rate Estimator

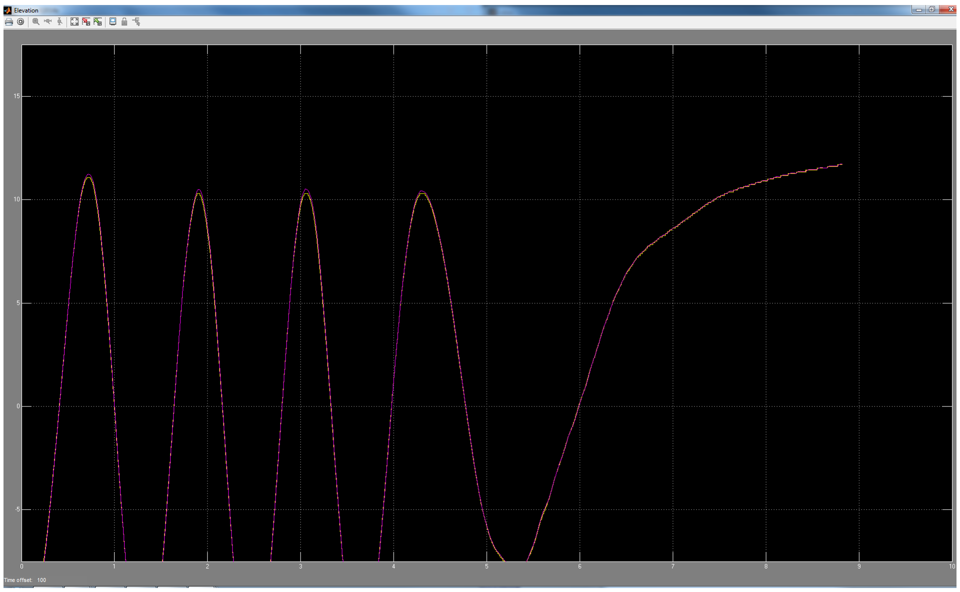

Figure 13: Elevation Estimator

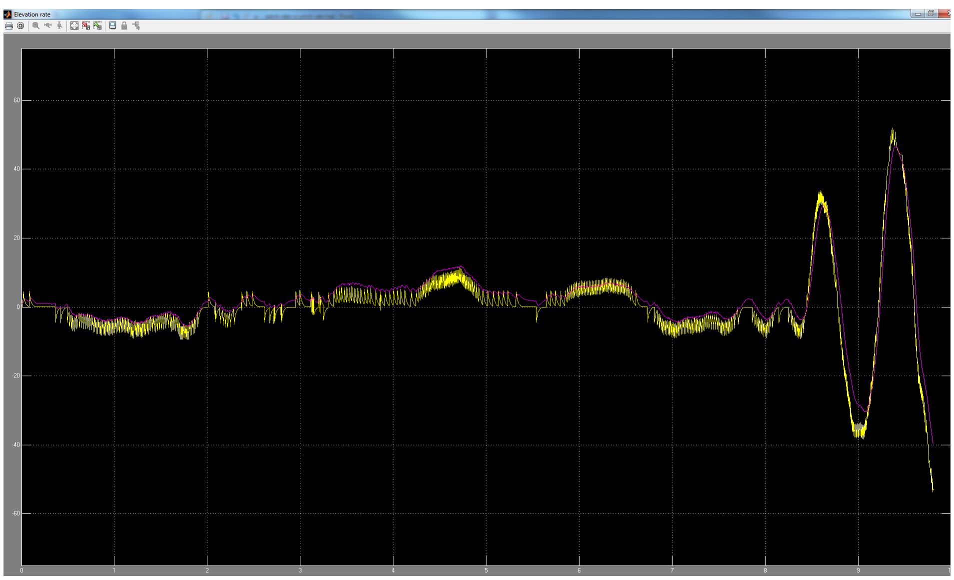

Figure 14: Elevation rate Estimator

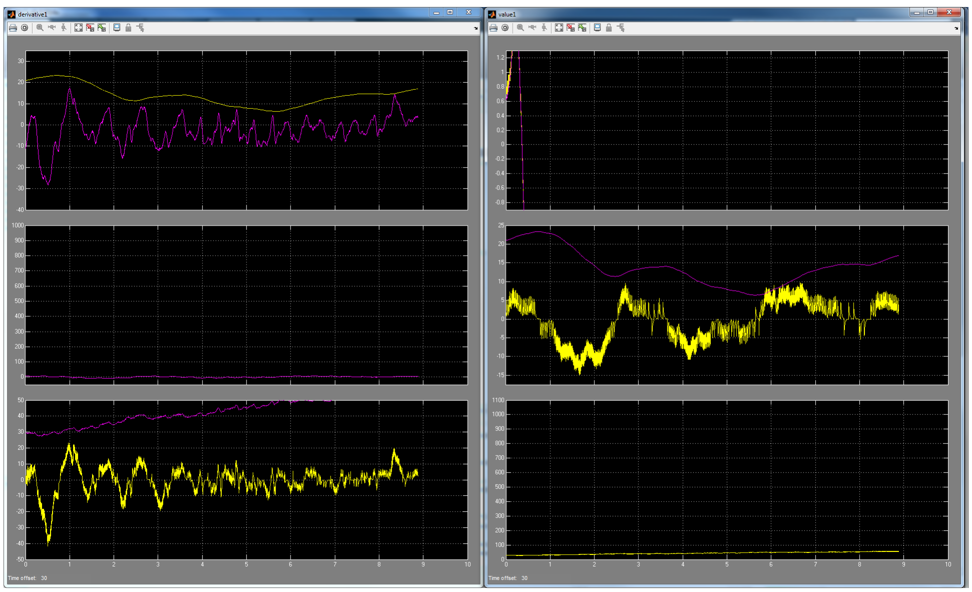

Figure 15: All states Estimator