Helicopter lab

Group 24
Tor Istvan Stadler Kjetså 478419
Martin Lund Haug 766881
Erlend Evenstad Dahlen 767703

October 21, 2018

Department of Engineering Cybernetics

Abstract

The goal of this lab assignment was to control a helicopter using linear system theory. We did this by modelling the system, estimating unknown states and applying controllers in a real-time environment.

Contents

1 Introduction

The work we were assigned was to control a dual-rotor helicopter with a fixed arm. We started out modelling the system before we continued adding mono- and multi-variable control. After this we use LQR to make a P regulator and a PI regulator for the system. In the end we added state estimation to the our control system to get knowledge of all our states which we previously had computed using numerical differentiation. Control systems are among us in many different states, and we are highly reliant on their stability. Not all states can or will be measured in real life due to different reasons and state estimation is a powerful tool for this particular problem. The document is split into 4 parts, in which each part corresponds to a group of similar problems. At the end of the report we have included Appendix A, B and C which we will refer to throughout the different parts.

2 Part 1 - Mathematical modelling

2.1 Problem 1

The helicopter is modeled as three point masses, illustrated with cubes, and three joints, illustrated with cylinders. At one end of the movable rod is the counterweight, and at the other end is the helicopter "head", where two propellers are placed symmetrically in relation to the pitch axis. The helicopter is modeled in fig. 1 with forces and angles drawn in.

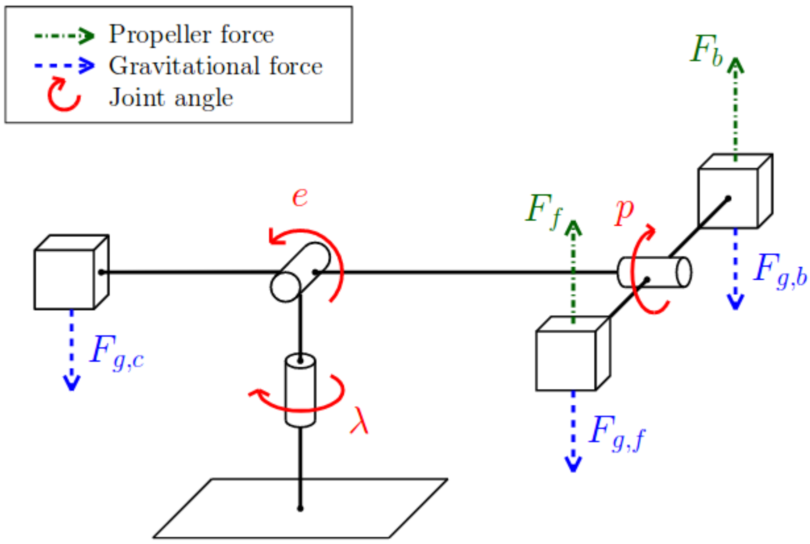

Figure 1: Helicopter model

fig. 2 illustrates the model, showing masses and distances. The values of the parameters are given in table 1

Table 1: Parameters and values.

Symbol	Parameter	Value	Unit
l_c	Distance from elevation axis to counterweight	0.46	m
l_h	Distance from elevation axis to helicopter head	0.66	m
l_p	Distance from pitch axis to motor	0.175	m
m_p	Motor mass	0.72	kg
m_c	Counterweight mass	1.92	kg

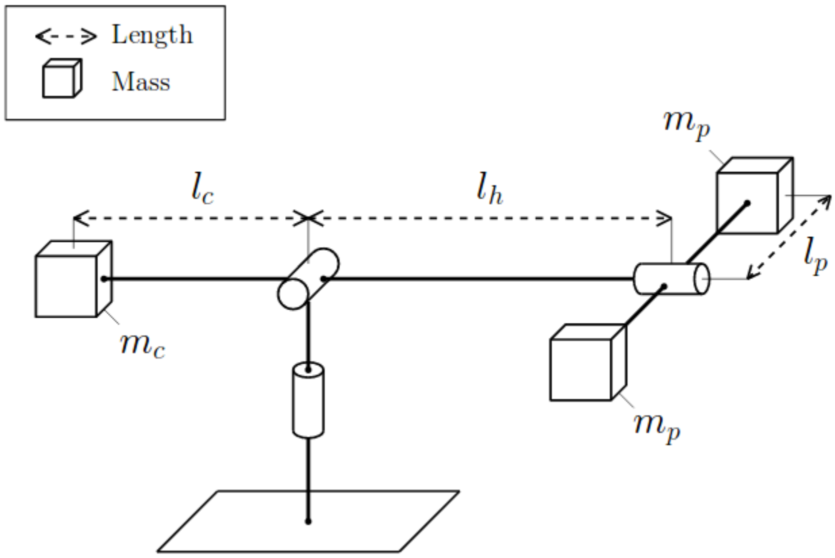

Figure 2: Masses and distances

It is assumed a linear relation between the forces generated by the propellers; F_f and F_b and the voltage supplied to the motors; V_f and V_b:

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

$$F_b = K_f V_b \tag{1b}$$

The equations of motion for the pitch angle p, the elevation angle e and the travel angle λ are computed using Newton's second law for rotation:

$$\Sigma\tau = I\alpha \tag{2}$$

Using J_p, J_e and J_λ noting the moments of inertia about the pitch, elevation and travel axes respectively, along with eq. (2) and figures fig. 1 and fig. 2 and eq. (1) , it is possible to derive the equations of motion around the axes.
Remembering $\tau = \boldsymbol{r} \times \boldsymbol{F}$;
setting up the torque balance around the pitch axis gives

$$J_p\ddot{p} = l_p(F_f + F_{g,b} - F_b - F_{g,f})$$
$$J_p\ddot{p} = l_p(F_f - F_b)$$
$$J_p\ddot{p} = l_p K_f(V_f - V_b)$$

Torque balance around the elevation axis gives

$$J_c\ddot{e} = (l_cF_{g,c} - (F_{g,f} + F_{g,b})l_h)cos(e) + l_h(F_f + F_b)cos(p)$$
$$J_c\ddot{e} = (l_cF_{g,c} - (F_{g,f} + F_{g,b})l_h)cos(e) + l_hK_f(V_f + V_b)cos(p)$$

Lastly, torque balance around the travel axis gives

$$J_\lambda\ddot{\lambda} = -l_h(F_f + F_b)cos(e)sin(p)$$
$$J_\lambda\ddot{\lambda} = -l_hK_f(V_f + V_b)cos(e)sin(p)$$

Using the torque balance and defining

$$L_1 = l_pK_f \tag{6a}$$
$$L_2 = l_cF_{g,c} - l_h(F_{g,f} + F_{g,b}) \tag{6b}$$
$$L_3 = l_hK_f \tag{6c}$$
$$L_4 = -l_hK_f \tag{6d}$$

and

$$V_d = V_f - V_b \tag{7a}$$
$$V_s = V_f + V_b \tag{7b}$$

results in eq. (8)

$$J_p\ddot{p} = L_1V_d \tag{8a}$$
$$J_e\ddot{e} = L_2cos(e) + L_3V_scos(p) \tag{8b}$$
$$J_\lambda\ddot{\lambda} = L_4V_scos(e)sin(p) \tag{8c}$$

2.2 Problem 2

In order to design a linear controller for the system the equations of motion are linearized around the point $(p, e, \lambda)^T = (p^*, e^*, \lambda^*)^T$ with $p^* = e^* = \lambda^* = 0$. Firstly, the values of the voltages V_s^* and V_d^* needs to be determined such that for all time $\dot{p} = \dot{e} = \dot{\lambda} = 0$ for $(p, e, \lambda)^T = (p^*, e^*, \lambda^*)^T$ and $(V_s, V_d)^T = (V_s^*, V_d^*)^T$.

eq. (8a) with both J_p and L_1 being nonzero gives $\underline{\underline{V_d^* = 0}}$.
Furthermore eq. (8b) gives

$$L_2cos(e^*) + L_3V_s^*cos(p^*) = 0$$
$$\underline{\underline{V_s^* = -\frac{L_2}{L_3}}}$$

Next the following coordinate transformation is introduced

$$\begin{bmatrix}\tilde{p}\\ \tilde{e}\\ \tilde{\lambda}\end{bmatrix} = \begin{bmatrix}p\\ e\\ \lambda\end{bmatrix} - \begin{bmatrix}p^*\\ e^*\\ \lambda^*\end{bmatrix} \; and \begin{bmatrix}\tilde{V}_s\\ \tilde{V}_d\end{bmatrix} = \begin{bmatrix}V_s\\ V_d\end{bmatrix} - \begin{bmatrix}V_s^*\\ V_d^*\end{bmatrix} \tag{10}$$

The system is to be linearized on the form:

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

Rewriting eq. (8) helps determining the Jacobians for

$$\mathbf{A} = \begin{bmatrix}
\frac{\partial\dot{\tilde{p}}}{\partial\tilde{p}} & \frac{\partial\dot{\tilde{p}}}{\partial\tilde{e}} & \frac{\partial\dot{\tilde{p}}}{\partial\tilde{\lambda}} & \frac{\partial\dot{\tilde{p}}}{\partial\dot{\tilde{p}}} & \frac{\partial\dot{\tilde{p}}}{\partial\dot{\tilde{e}}} & \frac{\partial\dot{\tilde{p}}}{\partial\dot{\tilde{\lambda}}}\\
\frac{\partial\dot{\tilde{e}}}{\partial\tilde{p}} & \frac{\partial\dot{\tilde{e}}}{\partial\tilde{e}} & \frac{\partial\dot{\tilde{e}}}{\partial\tilde{\lambda}} & \frac{\partial\dot{\tilde{e}}}{\partial\dot{\tilde{p}}} & \frac{\partial\dot{\tilde{e}}}{\partial\dot{\tilde{e}}} & \frac{\partial\dot{\tilde{e}}}{\partial\dot{\tilde{\lambda}}}\\
\frac{\partial\dot{\tilde{\lambda}}}{\partial\tilde{p}} & \frac{\partial\dot{\tilde{\lambda}}}{\partial\tilde{e}} & \frac{\partial\dot{\tilde{\lambda}}}{\partial\tilde{\lambda}} & \frac{\partial\dot{\tilde{\lambda}}}{\partial\dot{\tilde{p}}} & \frac{\partial\dot{\tilde{\lambda}}}{\partial\dot{\tilde{e}}} & \frac{\partial\dot{\tilde{\lambda}}}{\partial\dot{\tilde{\lambda}}}\\
\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{p}}}{\partial\dot{\tilde{p}}} & \frac{\partial\ddot{\tilde{p}}}{\partial\dot{\tilde{e}}} & \frac{\partial\ddot{\tilde{p}}}{\partial\dot{\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{e}}}{\partial\dot{\tilde{p}}} & \frac{\partial\ddot{\tilde{e}}}{\partial\dot{\tilde{e}}} & \frac{\partial\ddot{\tilde{e}}}{\partial\dot{\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}} & \frac{\partial\ddot{\tilde{\lambda}}}{\partial\dot{\tilde{p}}} & \frac{\partial\ddot{\tilde{\lambda}}}{\partial\dot{\tilde{e}}} & \frac{\partial\ddot{\tilde{\lambda}}}{\partial\dot{\tilde{\lambda}}}
\end{bmatrix}_{x=x_0} \tag{11}$$

Similarly, B is given by

$$\mathbf{B} = \begin{bmatrix}
\frac{\partial\dot{\tilde{p}}}{\partial\tilde{V}_s} & \frac{\partial\dot{\tilde{p}}}{\partial\tilde{V}_d}\\
\frac{\partial\dot{\tilde{e}}}{\partial\tilde{V}_s} & \frac{\partial\dot{\tilde{e}}}{\partial\tilde{V}_d}\\
\frac{\partial\dot{\tilde{\lambda}}}{\partial\tilde{V}_s} & \frac{\partial\dot{\tilde{\lambda}}}{\partial\tilde{V}_d}\\
\frac{\partial\ddot{\tilde{p}}}{\partial\tilde{V}_s} & \frac{\partial\ddot{\tilde{p}}}{\partial\tilde{V}_d}\\
\frac{\partial\ddot{\tilde{e}}}{\partial\tilde{V}_s} & \frac{\partial\ddot{\tilde{e}}}{\partial\tilde{V}_d}\\
\frac{\partial\ddot{\tilde{\lambda}}}{\partial\tilde{V}_s} & \frac{\partial\ddot{\tilde{\lambda}}}{\partial\tilde{V}_d}
\end{bmatrix}_{u=u_0} \tag{12}$$

It is assumed that the moments of inertia are constant and given by

$$J_p = 2m_p l_p^2 \tag{13a}$$

$$J_e = m_c l_c^2 + 2m_p l_h^2 \tag{13b}$$

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

Inserting eq. (8) and the equilibrium values to compute A and B

$$\mathbf{A} = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
\frac{L_4}{J_\lambda} & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \tag{14}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{L_1}{J_p} \\ \frac{L_3}{J_e} & 0 \\ 0 & 0 \end{bmatrix} \tag{15}$$

Thus, the linearized system on the form $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$ is

$$\begin{bmatrix} \dot{\tilde{p}} \\ \dot{\tilde{e}} \\ \dot{\tilde{\lambda}} \\ \ddot{\tilde{p}} \\ \ddot{\tilde{e}} \\ \ddot{\tilde{\lambda}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{L_4}{J_\lambda} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \\ \dot{\tilde{\lambda}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{L_1}{J_p} \\ \frac{L_3}{J_e} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{16}$$

This shows that the linearized equations of motion can be written as

$$\ddot{p} = K_1 \tilde{V}_d \tag{17a}$$

$$\ddot{e} = K_2 \tilde{V}_s \tag{17b}$$

$$\ddot{\lambda} = K_3 \tilde{p} \tag{17c}$$

with

$$K_1 = \frac{L_1}{J_p} = \frac{l_p K_f}{2 m_p l_p^2} \tag{18a}$$

$$K_2 = \frac{L_3}{J_e} = \frac{l_h K_f}{m_c l_c^2 + 2 m_p l_h^2} \tag{18b}$$

$$K_3 = \frac{L_4}{J_\lambda} = -\frac{l_h K_f}{m_c l_c^2 + 2 m_p (l_h^2 + l_p^2)} \tag{18c}$$

from eq. (6) and eq. (13).

2.3 Problem 3

The first attempt to control the helicopter was done using feed forward. The joystick outputs of the x-axis and the y-axis are connected directly to the inputs of voltage difference V_d and voltage sum V_s, respectively. Adjusting the gains on the joystick position outputs makes the helicopter easier to control by hand. We used a gain of 1 for the x-axis position and a gain of -3 for the y-axis position. Configuartion shown in fig. 21.

Figure 3 and fig. 4 shows the physical behovaiour of the helicopter to compare with the theoretical models in eq. (8) and eq. (17).

As seen the helicopter is not easy to control. It is very hard to hold a stable elevation, and controlling the travel is also proven quite difficult. In eq. (1) we assume a linear relationship between the force generated form the motors and the voltage supplied to them. In reality this relationship is probably nonlinear(cubic). Our linearized model eq. (17) shows that $\ddot{p}$ depends solely on V_s, $\ddot{e}$ solely on V_s and $\ddot{\lambda}$ solely on p. This is also a reason for the discrepancies, as the helicopter often finds itself in a state which is not its equilibrium, and the states will affect each other in a nonlinear way as shown in eq. (8). In addition to this, the helicopter isn't perfectly weighted, and the initialization process may start on an uneven surface. This causes the the states to drift even though they are given no extra input signal. This became especially evident concerning p as we saw the helicopter drifting even though $V_d = 0$.

2.4 Problem 4

The decoder values are set to zero every time Simulink is connected to the helicopter. As a result of this we need to add offset values to define elevation angle $e = 0$ when the helicopter head is horizontal. With help from a leveler we obtain an elevation angle offset of $e_{off} = 29.5^\circ = 0.5149rad$. In addition we wish to make sure that the pitch equilibrium is correct as well, in the case of the helicopter being initialized on an uneven surface. We get $p_{off} = 0^\circ = 0rad$.
We need to determine the motor force constant K_f before we can implement a controller based on the linearized equations of motion in eq. (17). By obtaining the value of V_s^* needed to keep the elevation angle $e = 0$ we can compute K_f. We obtain V_s^* by reading the voltage value when the helicopter is positioned in its equilibrium.
We obtain $V_s^* = 6.88$. Combining eq. (6), eq. (7) and the parameter values in table table 1 we can compute

$$K_f = -\frac{l_c F_{g,c} - l_h(F_{g,f} + F_{g,b})}{V_s^* l_h} \approx 0.145\text{N/V} \tag{19}$$

3 Part 2 - Monovariable control

In this part of the assignment we designed a controller for the pitch angle $\tilde{p}$ and the travel rate $\dot{\lambda}$.

3.1 Problem 1

A PD controller is used to control the pitch angle p. The controller is given as

$$\tilde{V}_d = K_{pp}(\tilde{p}_c - \tilde{p}) - K_{pd}\dot{\tilde{p}} \tag{20}$$

A model for this controller is implemented in Simulink as showed in figure 24, which is to be found in Appendix B. Substituting this equation into (17a) we get the following differential equation

$$\ddot{\tilde{p}} = K_1(K_{pp}(\tilde{p}_c - \tilde{p}) - K_{pd}\dot{\tilde{p}}) \tag{21}$$

By applying the Laplace transform we can now solve for the transfer function

$$\begin{gathered} s^2\tilde{P}(s) - s\tilde{P}(0) - \tilde{P}'(0) = K_1K_{pp}(\tilde{P}_c(s) - \tilde{P}(s)) - K_1K_{pd}(s\tilde{P}(s) - \tilde{P}(0)) \\ \implies \tilde{P}(s)(s^2 + sK_1K_{pd} + K_1K_{pp}) = \tilde{P}_c(s)K_1K_{pp} \end{gathered}$$

with $\tilde{P}'(0) = \tilde{P}(0) = 0$ from the linearization, this results in the following transfer function

$$\frac{\tilde{P(s)}}{\tilde{P}_c(s)} = \frac{K_1K_{pp}}{s^2 + sK_1K_{pd} + K_1K_{pp}} \tag{22}$$

Which can be written as

$$H(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{23}$$

with

$$\omega_0 = \sqrt{K_1K_{pp}} \tag{24}$$

$$2\zeta\omega_0 = K_1K_{pd} \tag{25}$$

By solving the characteristic equation (26), an expression for the eigenvalues of the transfer function can be obtained.

$$s^2 + 2\zeta\omega_0 s + \omega_0^2 = 0 \tag{26}$$

$$\frac{-2\zeta\omega_0 \pm \sqrt{(2\zeta\omega_0)^2 - 4\omega_0^2}}{2} \tag{27}$$

Setting the expression inside the square root equal to zero gives a double pole at the real axis. This results in $\zeta = 1$ which in theory corresponds to a critically damped response.
With regards to (25) and (24) we have

$$K_{pp} = \frac{\omega_0^2}{K_1} \tag{28}$$

$$K_{pd} = \frac{2\zeta\omega_0}{K_1} \tag{29}$$

By finding values for the damping constant ζ and the undamped frequency ω_0 we can obtain a response which is quick and without too much oscillations.
After testing for different values, $\zeta = 0.2$ was chosen, introducing complex conjugated poles. As seen in figure 5 with K_{pd1}, our theoretical value of ζ gave quite a slow and overdamped response compared to K_{pd4}. This underlines the fact that theoretical and physical results not always coincides. After tuning the natural frequency ω_0 with focus on a quick response, we got the following values for K_{pp} and K_{pd}

$$K_{pp} = 4 \tag{30}$$

$$K_{pd} = 3.9751 \tag{31}$$

It is desirable to have the poles of the system placed far to the left in the frequency plane, making the response as quick as possible. Observing that we in theory could increase K_{pp} infinitely much to achieve this, however, this would not be realizable as it would demand an infinite input signal. Since the actuator of the helicopter has a limited performance, large values of K_{pp} could cause saturation and generate a nonlinear response. Including an imaginary part to the poles introduce a slight overshoot in the response while two poles with different real parts give a slow but non-oscillating response. In conclusion, the response of the regulator is dependent on the poles of the transfer function and different system purposes and characteristics affects the pole placement.

3.2 Problem 2

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

$$\tilde{p_c} = K_{rp}(\dot{\tilde{\lambda}}_c - \dot{\tilde{\lambda}}) \tag{32}$$

A Simulink model of the travel controller is to be found in figure 25 in appendix B. Assuming pitch angle is controlled perfectly, that is $\tilde{p} = \tilde{p_c}$, and remembering $\ddot{\tilde{\lambda}} = K_3\tilde{p}$ from the linearized model, the following equation is obtained

$$\ddot{\tilde{\lambda}} = K_3 K_{rp}\dot{\tilde{\lambda}}c - K_3 K_{rp}\dot{\tilde{\lambda}} \tag{33}$$

Applying Laplace

$$\begin{aligned} S^2\tilde{\lambda}(s) - S\tilde{\lambda}(0) - \tilde{\lambda}'(0) &= K_3K_{rp}(S\tilde{\lambda}_c(s) - \tilde{\lambda}_c(0) - S\tilde{\lambda}(s) - \tilde{\lambda}(0)) \\ \implies \tilde{\lambda}(s)(S^2 + SK_3K_{rp}) - S\tilde{\lambda}(0) - \tilde{\lambda}'(0) &= \tilde{\lambda}_c(s)(K_3K_{rp}S) - \tilde{\lambda}_c(0) - \tilde{\lambda}(0) \end{aligned}$$

We have $\tilde{\lambda}'(0) = \tilde{\lambda}(0) = \tilde{\lambda}_c(0) = 0$ because of the linearizartion of the system. This give the following transfer function for the travel rate

$$\frac{\tilde{\lambda}(s)}{\tilde{\lambda}_c(s)} = \frac{K_{3K_{rp}}}{s + K_3K_{rp}} \tag{34}$$

which can be written as

$$H(s) = \frac{p}{s+p} \tag{35}$$

with $p = K_3K_{rp}$. From previous calculations, K_3 was found to be -0.0889 , therefore a negative valued K_{rp} would result in a pole at the left hand side frequency plane. We chose -1.2 as our value for K_{rp}, this gave us an accurate response which can be seen in figure 6.

4 Part 3 - Multivariable control

In this part of the assignment we controlled the pitch angle $\tilde{p}$ and the elevation rate $\dot{\tilde{e}}$ with a multivariable controller, where the reference for the pitch angle and the elevation rate were provided by the output of the joystick.

4.1 Problem 1

We put the system given by the equations in (6a)-(6b) on the form:

$$\dot{x} = Ax + Bu \tag{36}$$

where **A** and **B** are matrices,

$$x = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} \quad and \;\; u = \begin{bmatrix} \tilde{V_s} \\ \tilde{V_d} \end{bmatrix} \tag{37}$$

which results in the following system:

$$\begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \ddot{\tilde{e}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & k_1 \\ k_2 & 0 \end{bmatrix} * \begin{bmatrix} \tilde{V_s} \\ \tilde{V_d} \end{bmatrix} \tag{38}$$

4.2 Problem 2

We aim to track the reference r = $[\tilde{p_c} \; \dot{\tilde{e_c}}]^T$ for the pitch angle $\tilde{p}$ and the elevation rate $\dot{\tilde{e}}$, which will be given by the joystick output. First, we examined the controllability of the system with the MATLAB command C = ctrb(A,B):

$$\mathbf{C} = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & K_1 & 0 & 0 \\ 0 & K_1 & 0 & 0 & 0 & 0 \\ K_2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{39}$$

By the definition of the controllability matrix we know that a matrix with full rank is controllable. The matrix **C** has full rank and is therefore, by definition, controllable. We can now make a controller on the form

$$u = Pr - Kx \tag{40}$$

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

$$J = \int_0^\infty (x^T(t)\mathbf{Q}x(t) + u^T(t)\mathbf{R}u(t))dt \tag{41}$$

We let the weighting matrices **Q** and **R** be diagonal and chose the values to get a fast response for the helicopter. We obtained the corresponding matrix **K** using the

MATLAB command K=lqr(A,B,Q,R). When we chose **P** we wanted the theoretical values to follow $\lim_{t\to\infty}\tilde{p}(t) = \tilde{p_c}$ and $\lim_{t\to\infty}\dot{\tilde{e}}(t) = \dot{\tilde{e}}_c$ for fixed values of $\tilde{p_c}$ and $\dot{\tilde{e_c}}$. By inserting eq. (40) into eq. (36) and following the wanted theoretical values we get

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{B}(\mathbf{Pr} - \mathbf{Kx}) = 0$$
$$(\mathbf{A} - \mathbf{BK})\mathbf{x}_\infty = -\mathbf{BPR}$$
$$\mathbf{y}_\infty = -[\mathbf{C}(\mathbf{BK} - \mathbf{A})^{-\mathbf{1}}\mathbf{B}]\mathbf{Pr}$$
$$\Downarrow$$

$$\mathbf{P} = -[\mathbf{C}(\mathbf{BK} - \mathbf{A})^{-\mathbf{1}}\mathbf{B}]^{-\mathbf{1}} \tag{43}$$

To begin we set **Q** and **R** as the identity matrix. This did not give the desired response and we tuned the values to optimize the response. The tuning resulted in these values for **Q** and **R**:

$$\mathbf{Q} = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 50 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 0.1 & 0 \\ 0 & 1 \end{bmatrix} \tag{44}$$

From **Q** and **R** we get the following **K** and **P** matrices:

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 23,36 \\ 4,47 & 15,19 & 0 \end{bmatrix}, \mathbf{P} = \begin{bmatrix} 0 & 22,36 \\ 4,47 & 0 \end{bmatrix} \tag{45}$$

The main idea in LQR control design is to minimize the cost function given in eq. (41) while manipulating the values in the matrices **Q** and **R**. The **Q** matrix penalizes the states and large values in **Q** stabilizes with least possible changes, while small values in **Q** have less concern about changes. In addition to penalizing the state variables with **Q**, the **R** matrix is utilized to penalize the control signals. The values in **R** imply how much energy is used to stabilize the system. Further, large values correspond to stabilization with less energy while small values correspond to stabilization with more energy.

To begin with, we set both **Q** and **R** as identity matrices. In our tuning we first focused on the response with regards to elevation, and gradually ended up with rather aggressive values in both **Q** and **R**. Secondly, we tuned the values with focus on the response with regards to the pitch. In the same manner as with the elevation the values were chosen appropriately, however more frugally. No matter how well we tuned the values, we ended up with some error to the reference signal. The linearized model, which is optimized around the equilibrium point, could be one reason for this error. Additionally, a drawback of a P-regulator is that it cannot maintain zero error in a closed-loop system. To overcome this, a PI-regulator should be introduced.

4.3 Problem 3

To address the problem discussed in section 3.2 the controller was modified to include an integral effect for the elevation rate and the pitch angle. This results in two additional states given by

$$\dot{\gamma} = \tilde{p} - \tilde{p}_c \tag{46}$$

$$\dot{\zeta} = \dot{\tilde{e}} - \dot{\tilde{e}}_c \tag{47}$$

After tuning in the same manner as previously described, we obtained the following $\bar{\mathbf{Q}}$, $\bar{\mathbf{R}}$ and $\bar{\mathbf{K}}$ matrices:

$$\bar{\mathbf{Q}} = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 100 \end{bmatrix}, \bar{\mathbf{R}} = \begin{bmatrix} 0.1 & 0 \\ 0 & 3 \end{bmatrix} \tag{48}$$

$$\bar{\mathbf{K}} = \begin{bmatrix} 0 & 0 & 41.01 & 0 & 31.62 \\ 8.97 & 21.13 & 0 & 1.83 & 0 \end{bmatrix} \tag{49}$$

We know have to manipulate the the state-vectors to calculate $\bar{\mathbf{P}}$. We have that

$$\mathbf{u} = \bar{\mathbf{K}}\mathbf{x} + \bar{\mathbf{P}}\mathbf{r} + effect\ of\ integral \tag{50}$$

use the state from eq. (36), and define the states

$$\mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \end{bmatrix}, \ \mathbf{x_a} = \begin{bmatrix} \gamma \\ \zeta \end{bmatrix} \tag{51}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \dot{\mathbf{x_a}} = \begin{bmatrix} \tilde{p} - \tilde{p_c} \\ \dot{\tilde{e}} - \dot{\tilde{e_c}} \end{bmatrix} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{e}} \end{bmatrix} - \begin{bmatrix} \tilde{p_c} \\ \dot{\tilde{e_c}} \end{bmatrix} \tag{52}$$

$$\mathbf{C}\mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{e}} \end{bmatrix} \tag{53}$$

This results in the state-space representation

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x_a}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x_a} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \mathbf{r} \tag{54}$$

From eq. (54) the matrices $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ are defined

$$\bar{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & 0 \\ \mathbf{C} & 0 \end{bmatrix}, \bar{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \tag{55}$$

With the new matrices, $\bar{\mathbf{K}}$ can now be defined using the MATLAB command $\bar{\mathbf{K}}$ = lqr($\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{Q}}, \bar{\mathbf{R}}$). By defining $\bar{\mathbf{K}}$ as $\left[\bar{K_1}\ \bar{K_2}\right]$ eq. (50) becomes

$$u = \begin{bmatrix}\mathbf{K_1} & \mathbf{K_1}\end{bmatrix} \begin{bmatrix}\mathbf{x} \\ \mathbf{x_a}\end{bmatrix} + \bar{\mathbf{P}}r \tag{56}$$

By inserting eq. (56) into eq. (54) the representation of the state-space model is obtained:

$$\begin{bmatrix}\dot{\mathbf{x}} \\ \dot{\mathbf{x_a}}\end{bmatrix} = \begin{bmatrix}\mathbf{A} & 0 \\ \mathbf{C} & 0\end{bmatrix} \begin{bmatrix}\mathbf{x} \\ \mathbf{x_a}\end{bmatrix} - \begin{bmatrix}\mathbf{B} \\ 0\end{bmatrix} \begin{bmatrix}\mathbf{K_1} & \mathbf{K_1}\end{bmatrix} \begin{bmatrix}\mathbf{x} \\ \mathbf{x_a}\end{bmatrix} + \begin{bmatrix}\mathbf{B} \\ 0\end{bmatrix} \bar{\mathbf{P}}\mathbf{r} + \begin{bmatrix}0 \\ -1\end{bmatrix} \mathbf{r} \tag{57}$$

The equation is then simplified to

$$\begin{bmatrix}\dot{\mathbf{x}} \\ \dot{\mathbf{x_a}}\end{bmatrix} = \begin{bmatrix}\mathbf{A} - \mathbf{BK_1} & \mathbf{BK_2} \\ \mathbf{C} & 0\end{bmatrix} \begin{bmatrix}\mathbf{x} \\ \mathbf{x_a}\end{bmatrix} - \begin{bmatrix}\mathbf{B}\bar{\mathbf{P}} \\ -1\end{bmatrix} r \tag{58}$$

Any value of will work since $\mathbf{Cx} = \mathbf{r}$ with $\mathbf{t}_\infty$. The integral inputs of $\mathbf{x_a}$ will be equal to zero when the states reach the values of the reference. This gives

$$\mathbf{x} = -(\mathbf{A} - \mathbf{BK_1})^{-\mathbf{1}}\mathbf{B}\bar{\mathbf{P}}\mathbf{r} \tag{59}$$

By adding $\mathbf{C}$ to both sides we get

$$\mathbf{r} = \mathbf{y} = \mathbf{Cx} = -\mathbf{C}(\mathbf{A} - \mathbf{BK_1})^{-\mathbf{1}}\mathbf{B}\bar{\mathbf{P}}\mathbf{r} \tag{60}$$

with $\mathbf{t}_\infty$. Finally, from eq. (60) we get:

$$\bar{\mathbf{P}} = [\mathbf{C}(\mathbf{BK} - \mathbf{A})^{-\mathbf{1}}\mathbf{B}]^{-\mathbf{1}} \tag{61}$$

$$\bar{\mathbf{P}} = \begin{bmatrix}0 & 41.01 \\ 8.97 & 0\end{bmatrix} \tag{62}$$

In fig. 7 and fig. 8 the behaviour of the helicopter is compared with and without the integral effect. The integrator is low at the start but increases action in relation both error and time and fig. 8 illustrates how the integrator part eliminates the constant error in the P-regulator.

5 Part 4 - State estimation

The system has sensors to measure the pitch, elevation and travel angles. In the previous parts the angular velocities corresponding to these have been computed using numerical differentiation. In this part an observer will be developed to estimate these nonmeasured states instead.

5.1 Problem 1

To derive a state space formulation on the form

$$\dot{x} = \mathbf{A}x + \mathbf{B}u \tag{63a}$$

$$y = \mathbf{C}x \tag{63b}$$

given the state vector x, the input vector u and the output vector y

$$x = \begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \dot{\tilde{e}} \\ \ddot{\tilde{e}} \\ \dot{\tilde{\lambda}} \\ \ddot{\tilde{\lambda}} \end{bmatrix}, u = \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \, and \, y = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} \tag{64}$$

Equation (17) or rewriting eq. (16) gives

$$\dot{x} = \begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \dot{\tilde{e}} \\ \ddot{\tilde{e}} \\ \dot{\tilde{\lambda}} \\ \ddot{\tilde{\lambda}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ K_3 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ 0 & 0 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{65}$$

$$y = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} \tag{66}$$

5.2 Problem 2

The observability matrix is defined as

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \tag{67}$$

With eq. (65) and calculating the two first computations we obtain

$$\mathcal{O} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{68}$$

The observability matrix has full rank so the system is observable. Thus it is possible to create a linear observer of the form

$$\dot{\hat{x}} = \mathbf{A}\hat{x} + \mathbf{B}u + \mathbf{L}(y - \hat{y}) \tag{69}$$

with

$$\hat{y} = \mathbf{C}\hat{x} \tag{70}$$

Defining the error e

$$e = x - \hat{x} \tag{71}$$

leads to

$$\dot{e} = \dot{x} - \dot{\hat{x}} \tag{72}$$

By inserting (69) and (63a) into the equation (72) we get the following expression

$$\dot{e}(t) = (\mathbf{A} - \mathbf{LC})e(t) \tag{73}$$

If $(\mathbf{A}, \mathbf{C})$ is observable, then all the eigenvalues of $(\mathbf{A} - \mathbf{LC})$ can be assigned by choosing an $\mathbf{L}$. By this fact and observing (73), one can choose how fast the estimated state $\hat{x}$ shall converge to the actual state x, simply by choosing an appropriate $\mathbf{L}$. The poles for $\mathbf{L}$ were placed as illustrated in figure 10 and the poles for the system is illustrated in figure 9. Both figures are found in Appendix A. How the poles are placed and $\mathbf{L}$ is computed is shown in fig. 34 and fig. 35 in Appendix C.

The matrices Q and R are tuned with the following values:

$$\mathbf{Q} = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 100 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 0.1 & 0 \\ 0 & 3 \end{bmatrix} \tag{74}$$

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 41.01 & 0 & 31.62 \\ 8.97 & 21.13 & 0 & 1.83 & 0 \end{bmatrix} \tag{75}$$

$$\mathbf{L} = \begin{bmatrix} 0.17 & 0.01 & 0 \\ 6.85 & 0.62 & 0 \\ 0.01 & 0.17 & 0 \\ 0.74 & 6.89 & 0 \\ 0 & 0 & 0.12 \\ 0 & 0 & 3.42 \end{bmatrix} * 10^3 \tag{76}$$

In the pole placement, a large distance from the imaginary axis is desirable to achieve fast response. If all the eigenvalues are placed at the same point, the response will often become slower and the actuating signal will become large. By this fact, spreading the poles along the real axis is beneficial to the system. Since we are using the estimator in state feedback, the estimator eigenvalues should be faster than the eigenvalues of the state feedback. It is important to keep in mind that the system will never respond faster than the eigenvalues of the observer, therefore large negative values for the observer is preferred. Because of the separation property we can choose the poles of the state feedback and the estimator independent of each other.

5.3 Problem 3

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

$$\boldsymbol{\mathcal{C}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{77}$$

and using the MATLAB command O = obsv(A,C) the observability matrix becomes

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -k_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -k_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{78}$$

The observability matrix has full rank so the system is observable. If one only measures $\tilde{e}$ and $\tilde{p}$ the C matrix becomes

$$\boldsymbol{\mathcal{C}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{79}$$

and using the MATLAB command O = obsv(A,C) the observability matrix becomes

$$
\mathcal{O} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \tag{80}
$$

The observability matrix has rank equal to 4,which is not full rank, thus the system is not observable.

If one only measures $\tilde{e}$ and $\tilde{\lambda}$ the system is observable. If we look at equations eq. (17) this makes sense. It is possible to differentiate $\tilde{e}$ two times to obtain $\ddot{\tilde{e}}$ and $\tilde{\lambda}$ to obtain $\ddot{\tilde{\lambda}}$. $\tilde{p}$ is then given by $\ddot{\tilde{\lambda}}$, hence it is possible to differentiate $\tilde{p}$ two times to obtain $\ddot{\tilde{p}}$. However, if one only measures $\tilde{e}$ and $\tilde{p}$ the system is not observable. Initially, it is possible to differentiate $\tilde{e}$ two times to obtain $\ddot{\tilde{e}}$ and $\tilde{p}$ to obtain $\ddot{\tilde{p}}$. Notwithstanding this, to obtain $\tilde{\lambda}$ it is now necessary to integrate $\ddot{\tilde{\lambda}}$. Opposed to differentiating, integration results in unknown constants and this makes the system not observable.

In theory, it should now be possible to control the helicopter by only measuring $\tilde{e}$ and $\tilde{\lambda}$. In practice, this turned out to be rather difficult. To obtain $\dot{\tilde{p}}$ with these measurements, $\tilde{\lambda}$ had to be differentiated three times. The problem is that the measurement noise is differentiated as well, and the differentiation amplifies the noise. Because this is done three times, the noise propagates through each iteration with increasing magnitude. To reduce the noise propagated, the values in row 1 in the **L** matrix are significantly reduced. In addition, all the poles are still placed on the real axis to achieve high damping and few oscillations. The measured and estimated states are plotted with pitch in fig. 17, pitch rate in fig. 18 and elevation rate in fig. 19.

We ended up with a lot of noise in $\tilde{p}$ and had to tune the matrices **Q** and **R** accordingly. To avoid the noise in $\tilde{p}$ the **R** matrix was chosen to respond with extremely low energy. In some sense this eliminated the effect of the noise, however the system became unstable with even small external variations in the pitch. The values for the matrices **Q**, **R**, **K** and **L** were following.

$$\mathbf{Q} = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 100 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 0.1 & 0 \\ 0 & 300000000 \end{bmatrix} \tag{81}$$

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 41.01 & 0 & 31.62 \\ 0.02 & 0.96 & 0 & 0 & 0 \end{bmatrix} \tag{82}$$

$$\mathbf{L} = \begin{bmatrix} -137.60 & -1090.30 \\ -0.9 & -6.80 \\ 0.2 & 0 \\ 6.60 & 0.5 \\ 0 & 0.2 \\ 0.9 & 0.94 \end{bmatrix} * 10^3 \tag{83}$$

6 Appendix A - Plots

6.1 Part 1

6.1.1 Problem 3

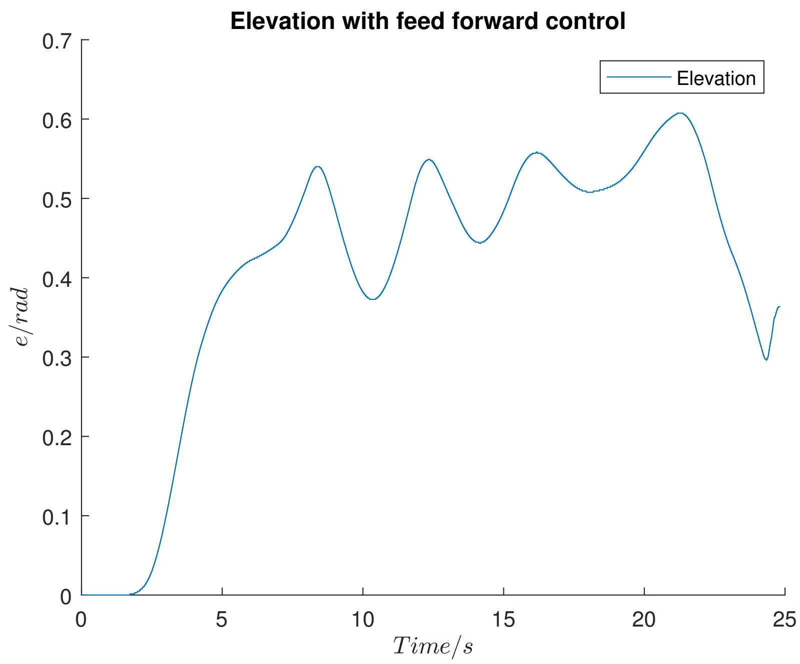

Figure 3: Elevation with feed forward

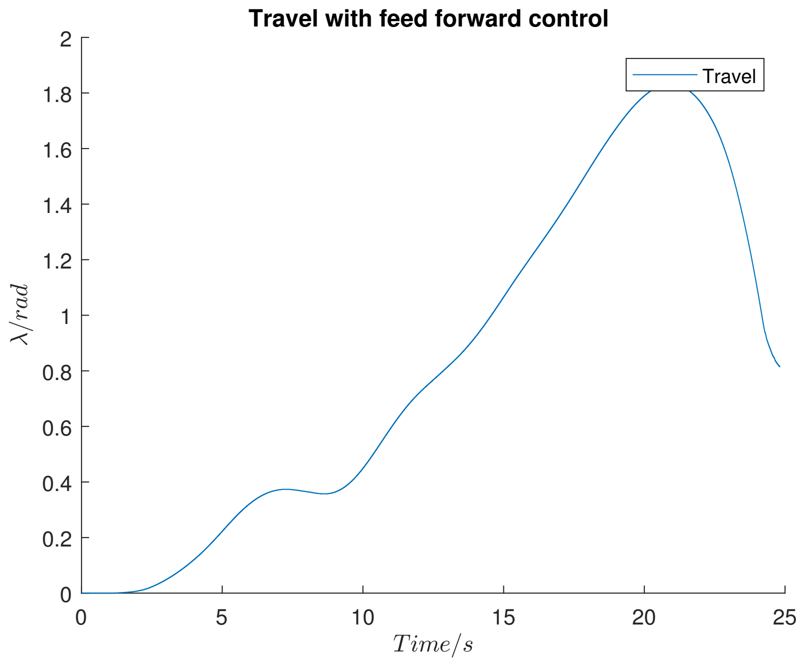

Figure 4: Travel with feed forward

6.2 Part 2

6.2.1 Problem 1

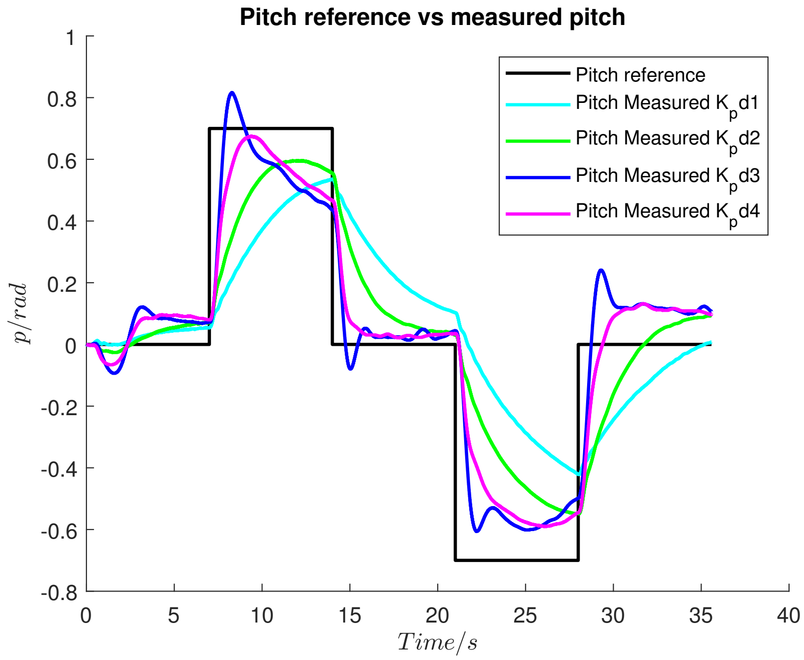

Figure 5: Pitch step response.

6.2.2 Problem 2

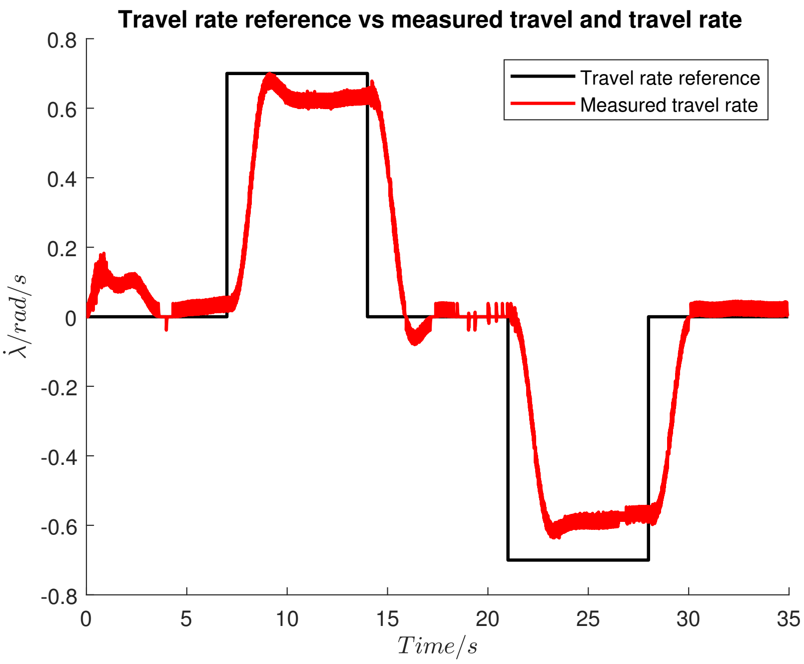

Figure 6: Travel rate step response.

6.3 Part 3

6.3.1 Problem 3

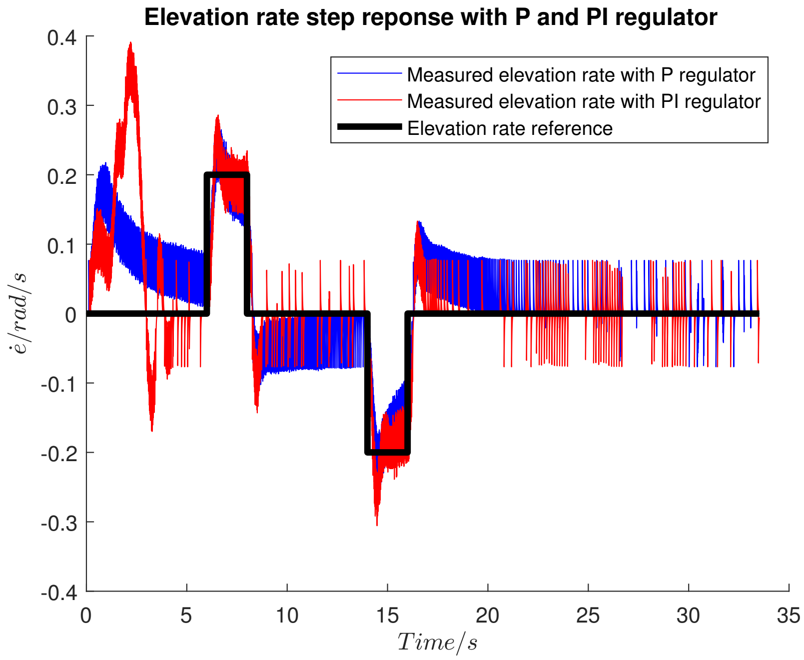

Figure 7: Elevation rate step response with P and PI regulator

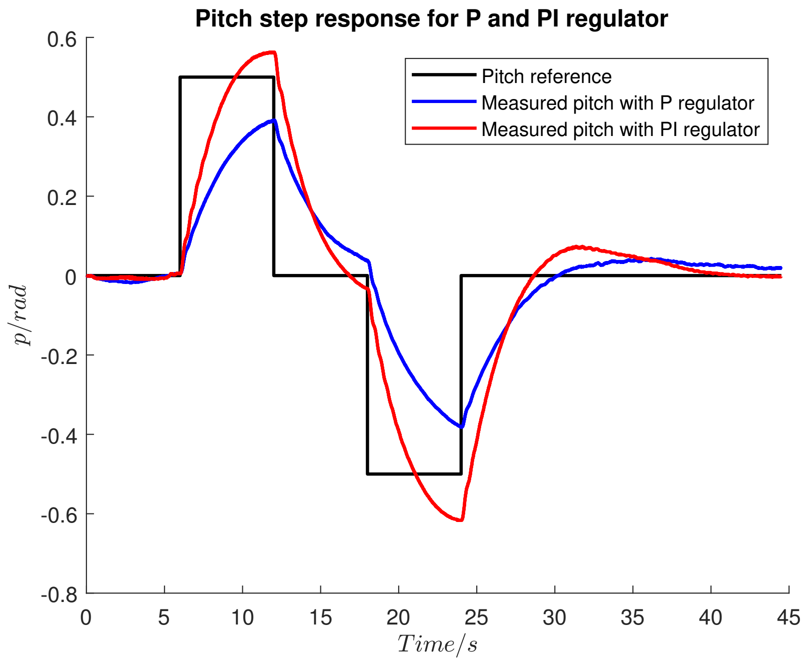

Figure 8: Pitch step response with P and PI regulator

6.4 Part 4

6.4.1 Problem 2

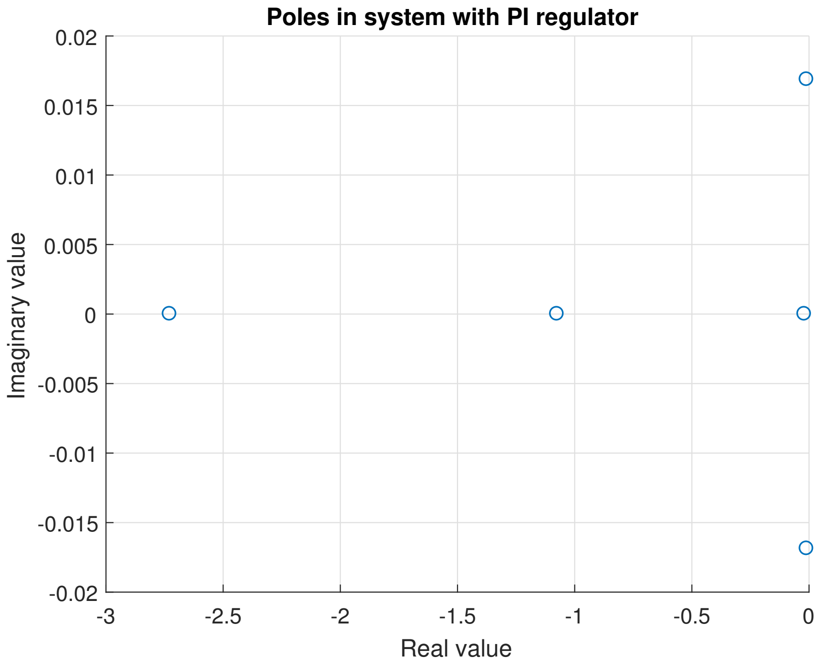

Figure 9: Poles of system.

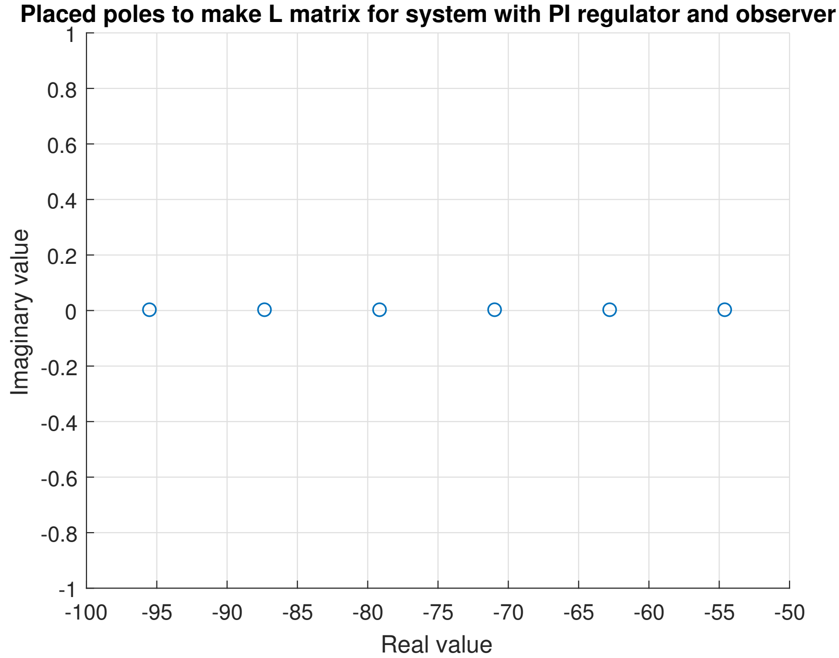

Figure 10: Poles of observator

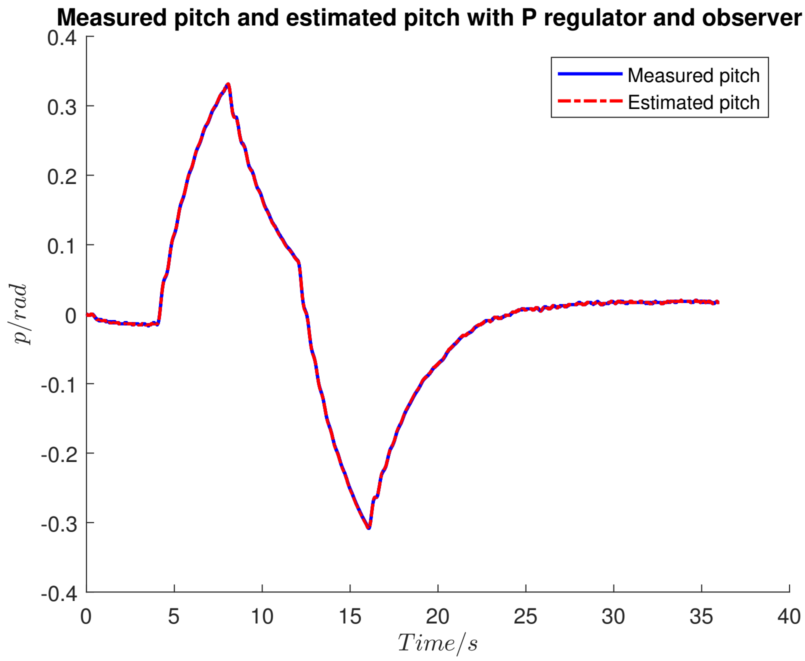

Figure 11: Measured and estimated pitch with P-regulator

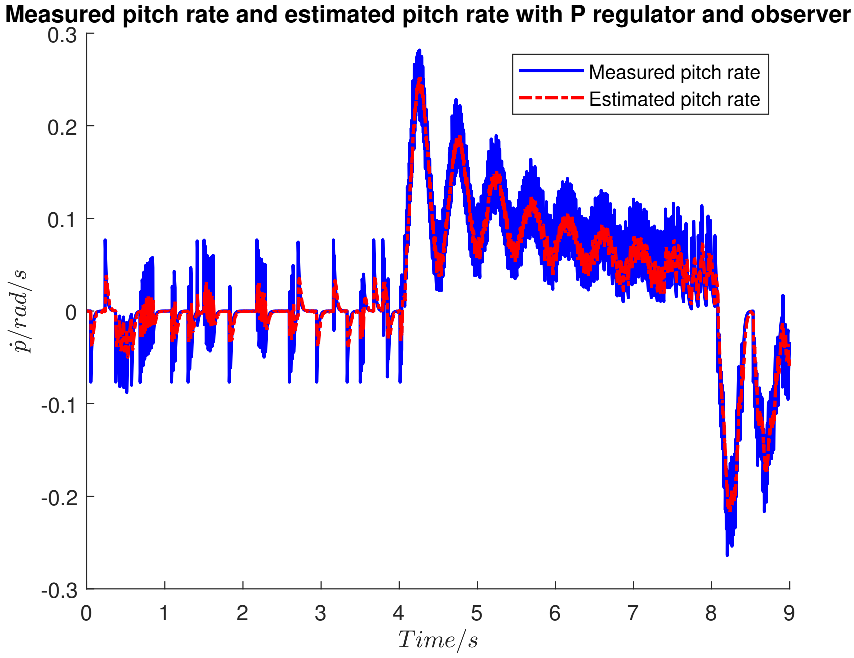

Figure 12: Measured and estimated pitch rate with P-regulator

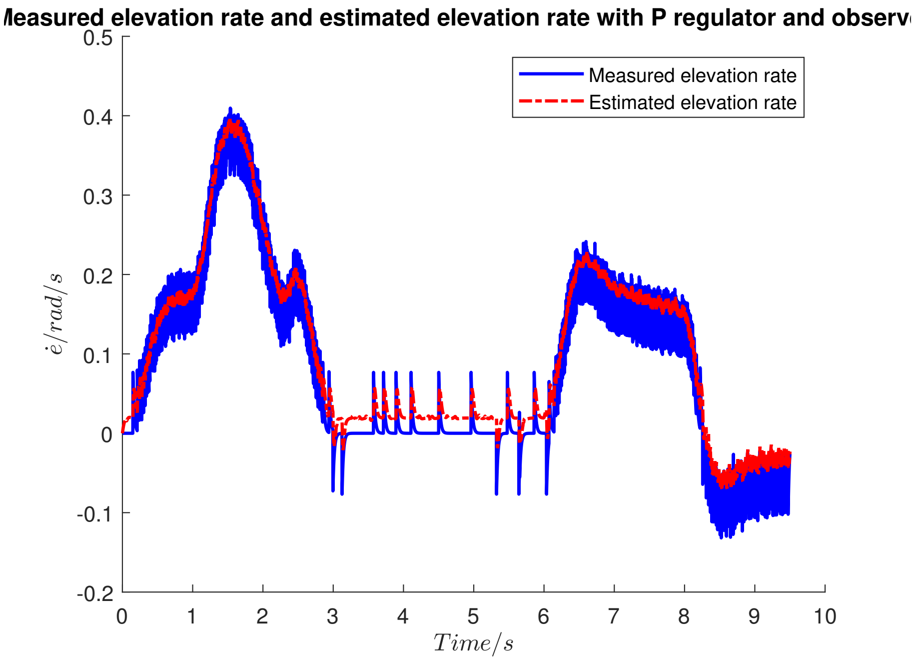

Figure 13: Measured and estimated elevation rate with P-regulator

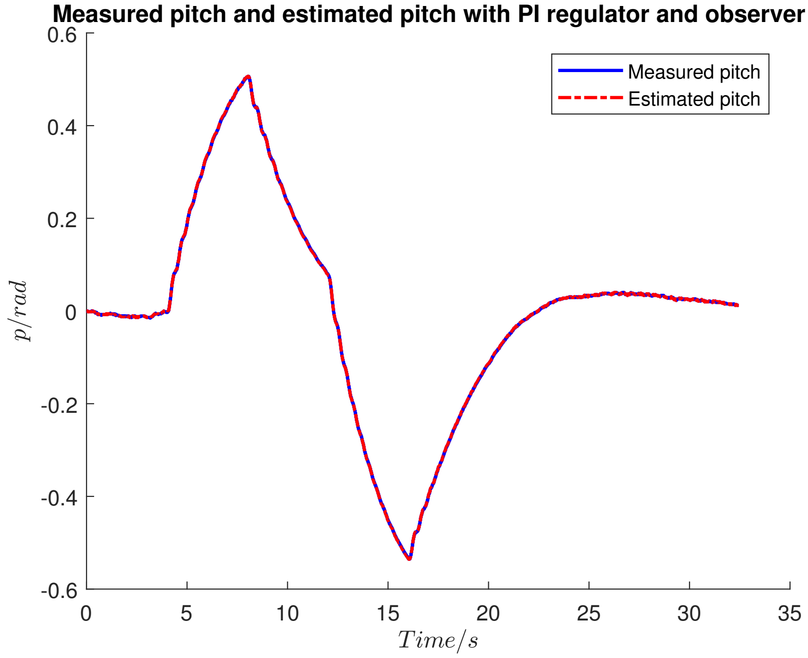

Figure 14: Measured and estimated pitch with PI-regulator

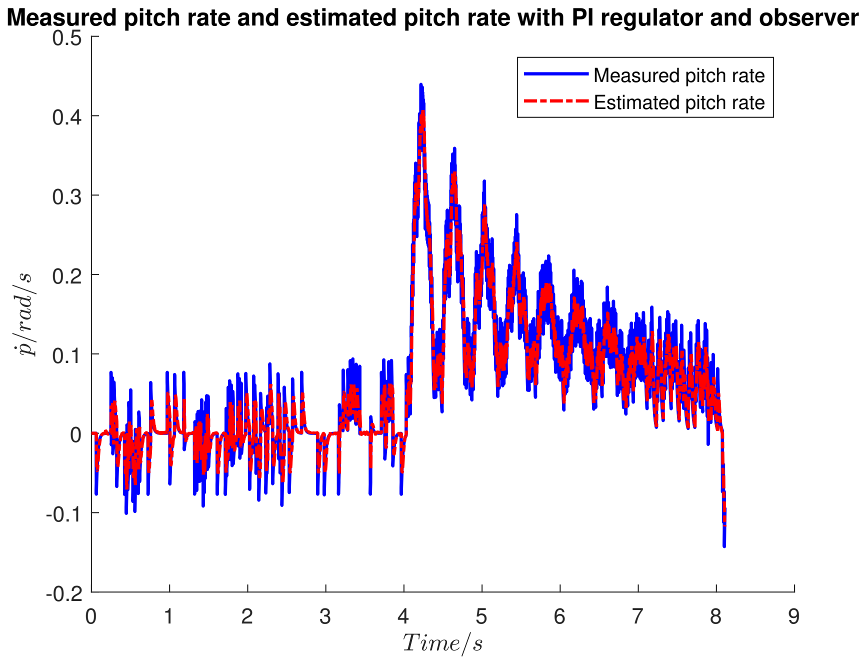

Figure 15: Measured and estimated pitch rate with PI-regulator

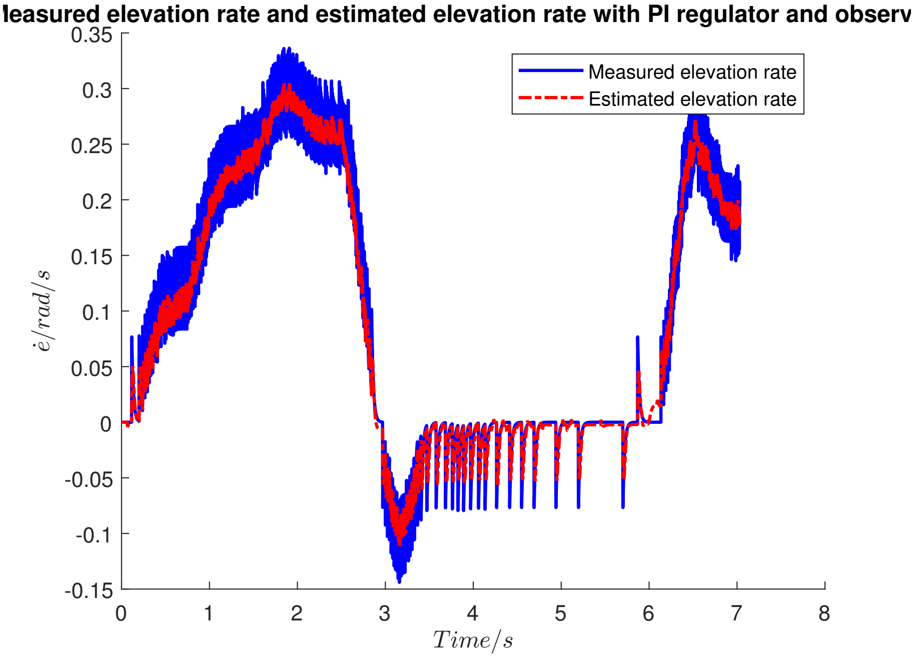

Figure 16: Measured and estimated elevation rate with PI-regulator

6.4.2 Problem 3

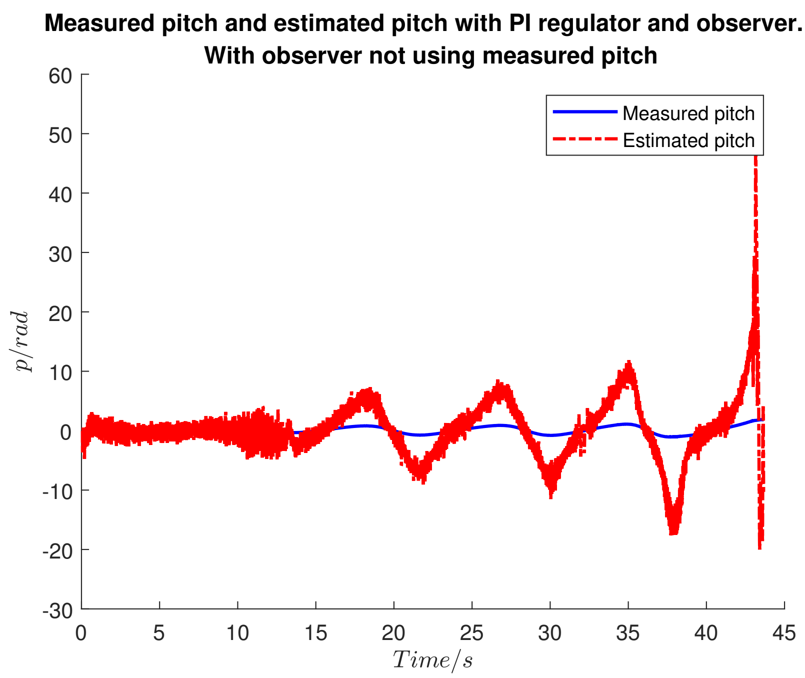

Figure 17: Measured and estimated pitch

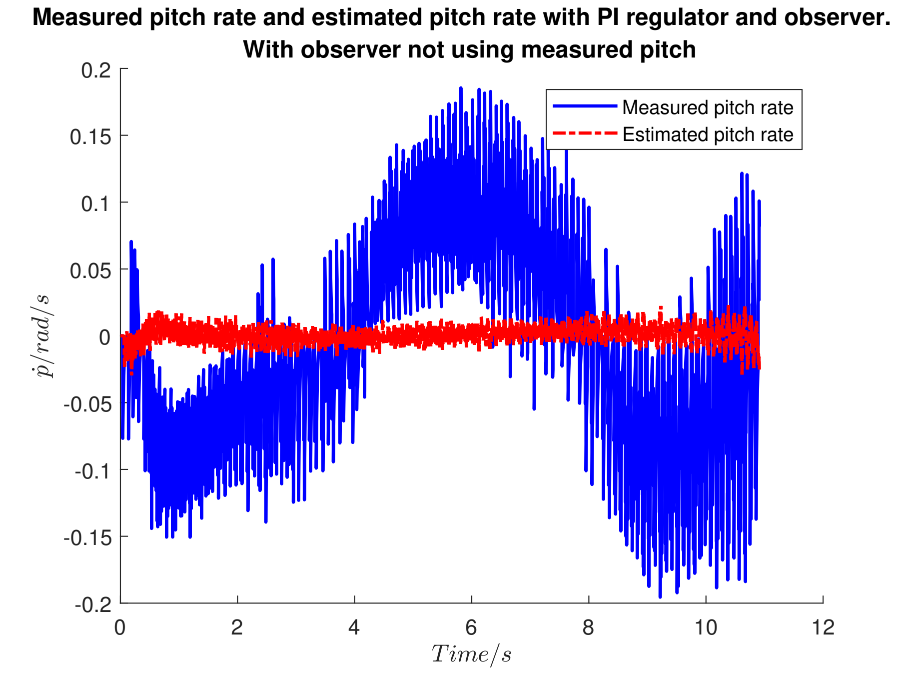

Figure 18: Measured and estimated pitch rate

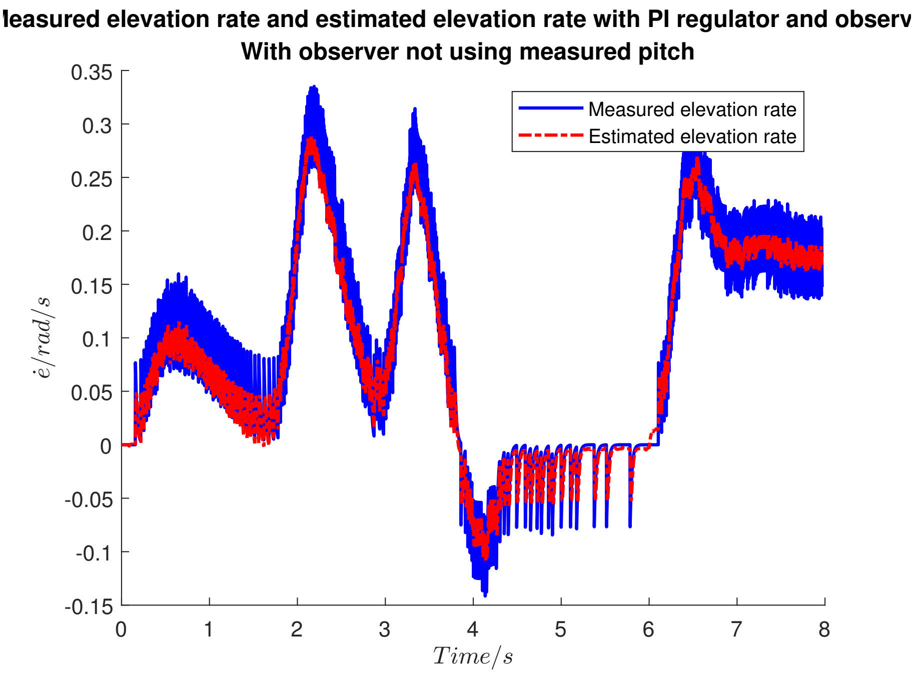

Figure 19: Measured and estimated elevation rate

7 Appendix B - Simulink models

7.1 Part 1

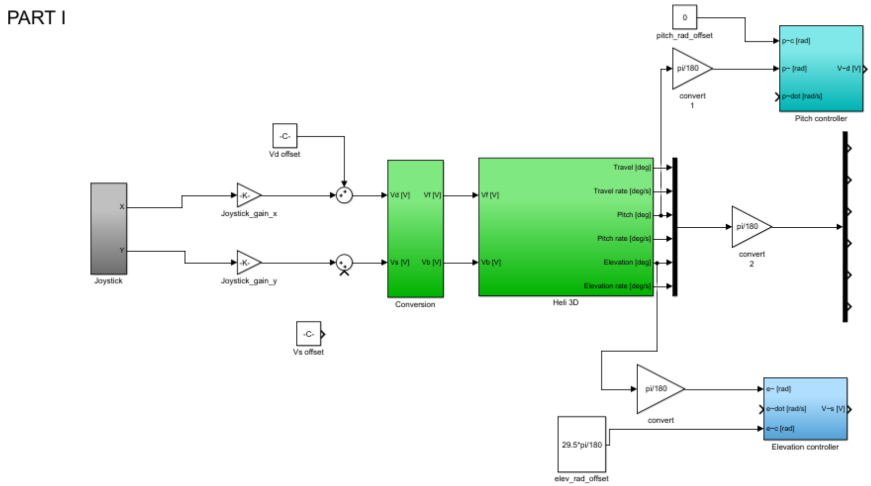

Figure 20: System overview

7.1.1 Problem 3

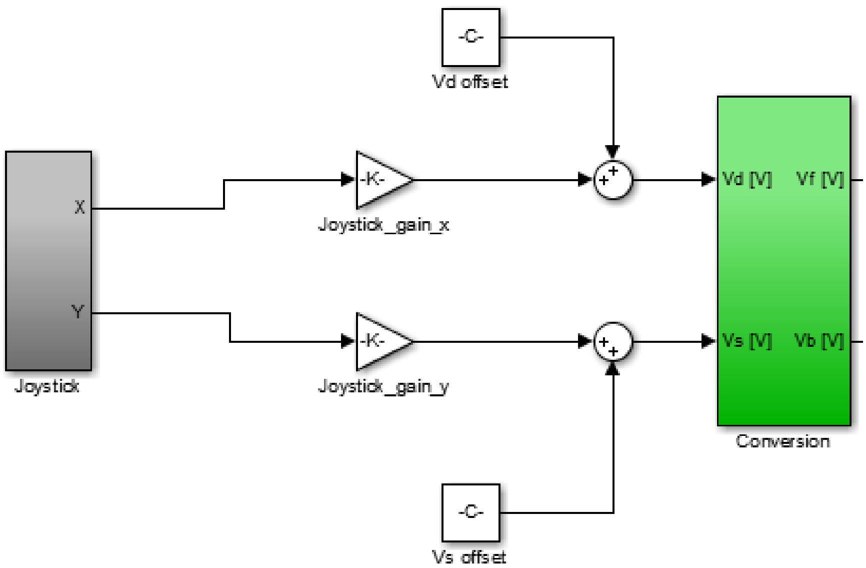

Figure 21: Configuration of connecting joystick to V_s and V_D and adding equilibrium values

7.2 Part 2

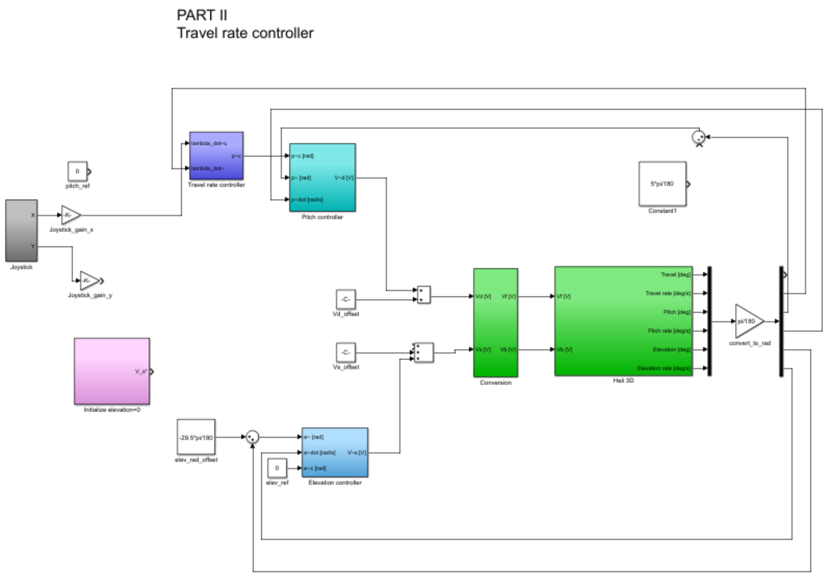

Figure 22: System overview

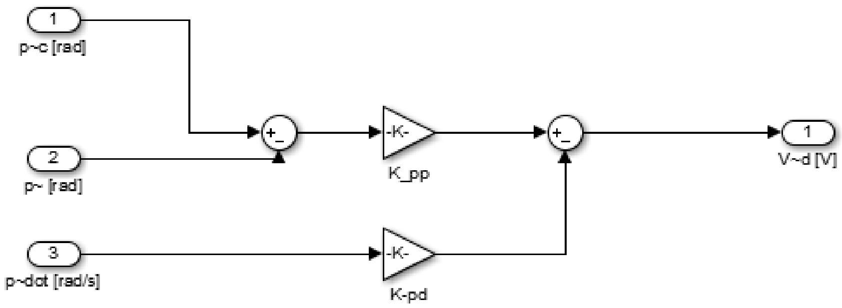

Figure 23: PD-controller for pitch angle

7.2.1 Problem 1

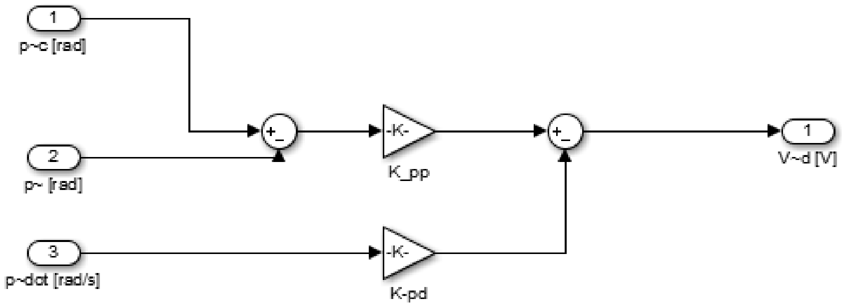

Figure 24: PD-controller for pitch angle

7.2.2 Problem 2

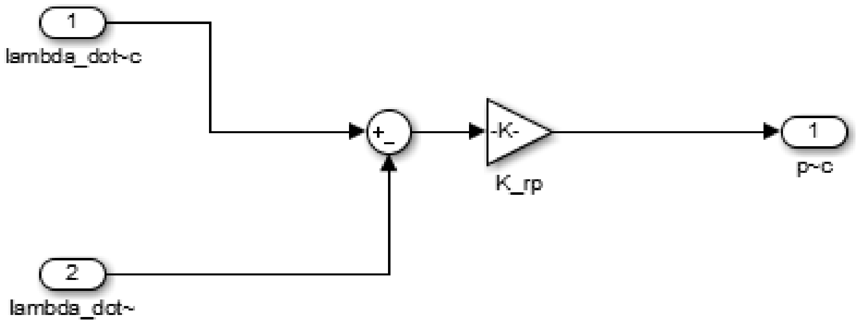

Figure 25: P-controller for travel rate

7.3 Part 3

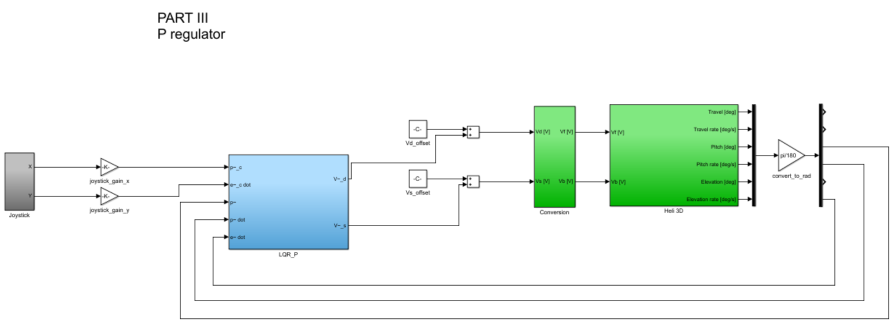

Figure 26: System overview of P-regulator

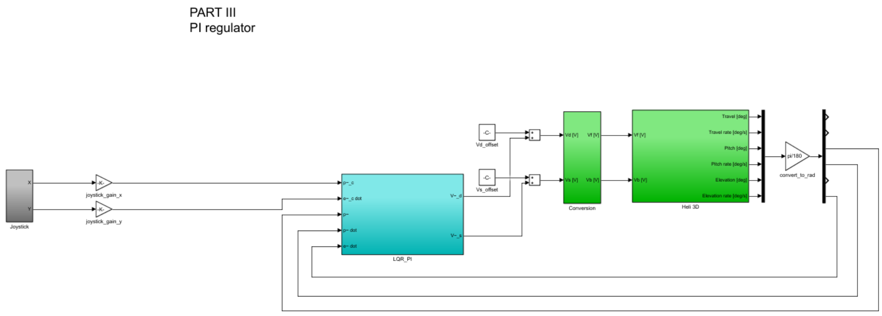

Figure 27: System overview of PI-regulator

7.3.1 Problem 2

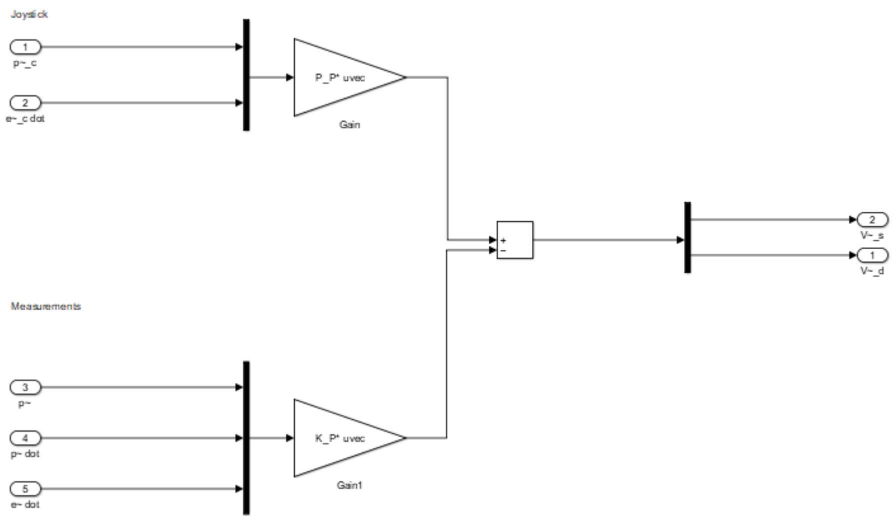

Figure 28: P-regulator

7.3.2 Problem 3

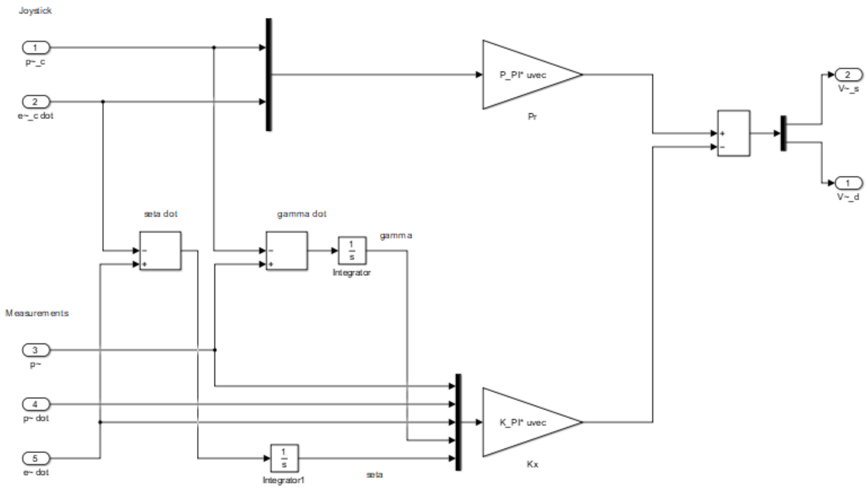

Figure 29: PI-regulator

7.4 Part 4

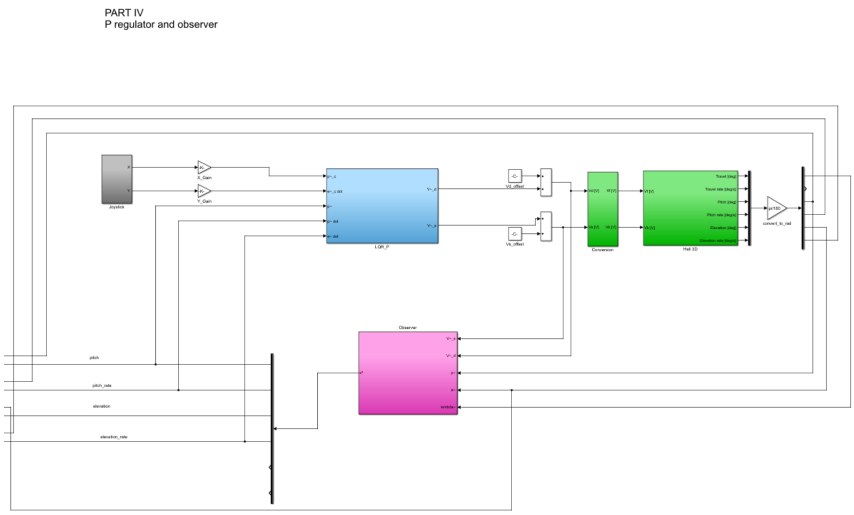

Figure 30: System overview of P-regulator and observer with three states

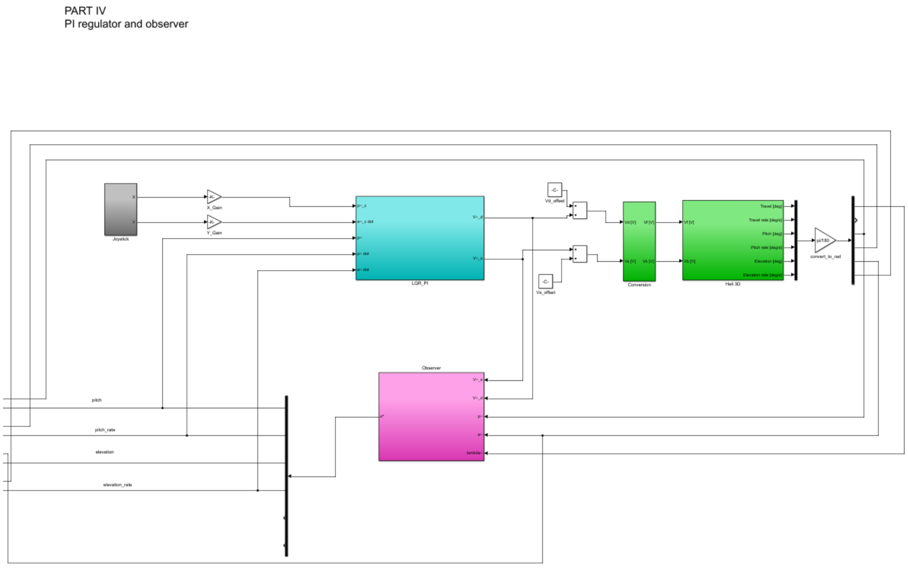

Figure 31: System overview of PI-regulator and observer with three states

7.4.1 Problem 2

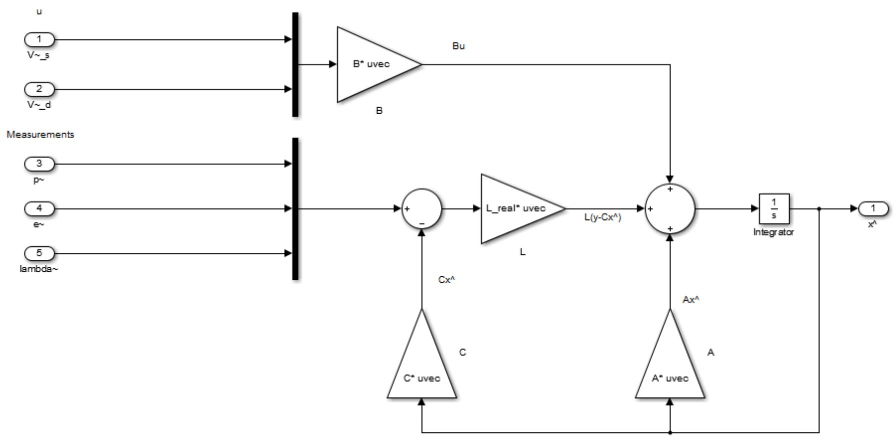

Figure 32: Observer with three states

7.4.2 Problem 3

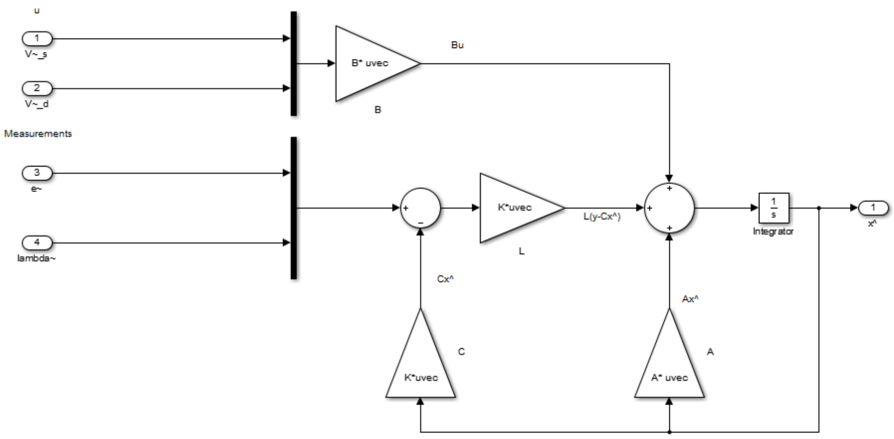

Figure 33: Observer with two states

8 Appendix C - MATLAB script

8.1 Part 4

8.1.1 Problem 2

```
[~,index_p]=max(abs(P_poles));
r_0_p=P_poles(index_p);
spacing_i_p=0.15;
k_r_p=20*r_0_p;
poles_real_p = 1:spacing_i:(1 + spacing_i*(size(A) - 1));
poles_real_p = poles_real_p * k_r;
L_real=transpose(place(A',C',poles_real_p));
```

Figure 34: Placing poles for **L** based on poles in system with P regulator and computing L

```
[~, index] = max(abs(PI_poles));
r_0 = PI_poles(index);
spacing_i = 0.15;
k_r = 20*r_0;
poles_real = 1:spacing_i:(1 + spacing_i*(size(A) - 1));
poles_real = poles_real * k_r;
L_real=transpose(place(A',C',poles_real));
```

Figure 35: Placing poles for L based on poles in system with PI regulator and computing L

8.1.2 Problem 3

```
poles_real_elev_trav=poles_real;
poles_real_elev_trav(1)=poles_real_elev_trav(1)*0.25;
poles_real_elev_trav(2)=poles_real_elev_trav(2)*0.0001;
L_elev_trav_real=transpose(place(A', C_elev_trav', poles_real_elev_trav));
```

Figure 36: Altering the poles and computing a new **L** for observator without p.

References

[1] Chen, Chi-Tsong. *Linear System Theory and Design*. Oxford University Press, Incorporated, 2014.

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