TTK4115 – Helicopter lab

Group 69
Student 478464
Student 478404

October 16, 2018

NTNU
Department of Engineering Cybernetics

Abstract

This lab assignment first explores controlling a helicopter using monovariable controllers and a multivariable LQR controller, and then the use of a estimator for state feedback.

Contents

1 Introduction

The goal of this progject is to control and estimate a helicopter. Using different methods for controller and estimator design, the response of the helicopter is observed.

Part I gives an overview of the setup used in the project, and the modeling of the system. Part II describes how to control the system using monovariable controllers, while part III implements multivariable controllers. Finally, in part IV, the principles and consequences of using a state estimator in the system is explored.

Knowledge and experience with these techniques and principles are vital for control engineers.

2 Part 1 – Mathematical modeling

2.1 Helicopter model

To control the helicopter and analyze its behaviour it first has to be modelled mathematically. First, the physical forces, variables and constants are labelled, as shown in Figure 1. Next, the different masses and distances are labelled, as shown in Figure 2.

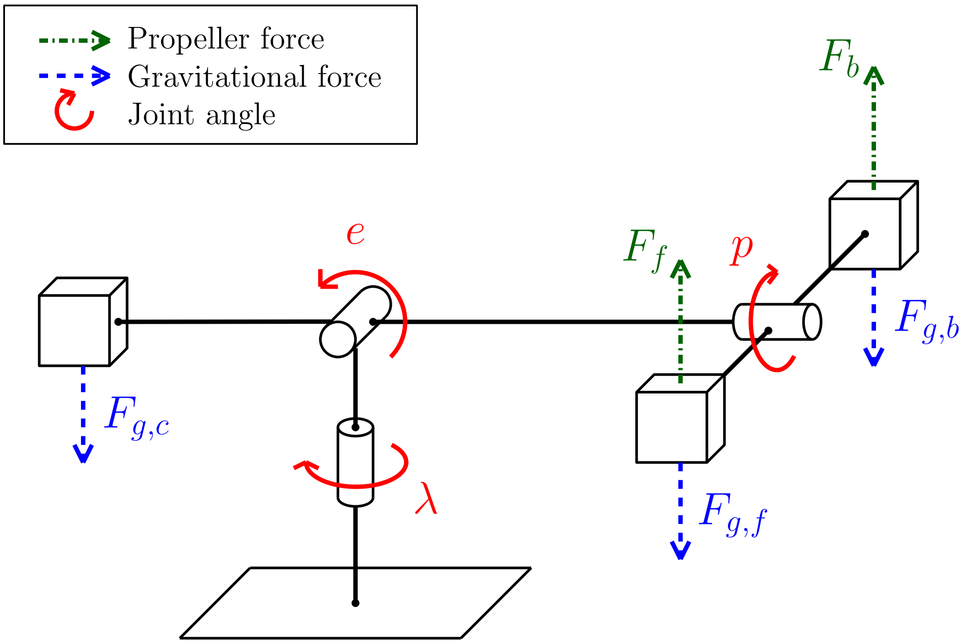

Figure 1: Helicopter model

The following is given by the assignment text: The propeller forces for the front and back propeller are given by F_f and F_b, respectively. It is assumed that there is a linear relation between the voltages F_f and F_b supplied to the motors and the forces generated by the propellers:

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

where K_f is the motor force constant.

2.2 Physical prerequisites

Applying torque to an object will cause the object to rotate. Torque is given by

$$\tau = \mathbf{F} \times \mathbf{r} \tag{2}$$

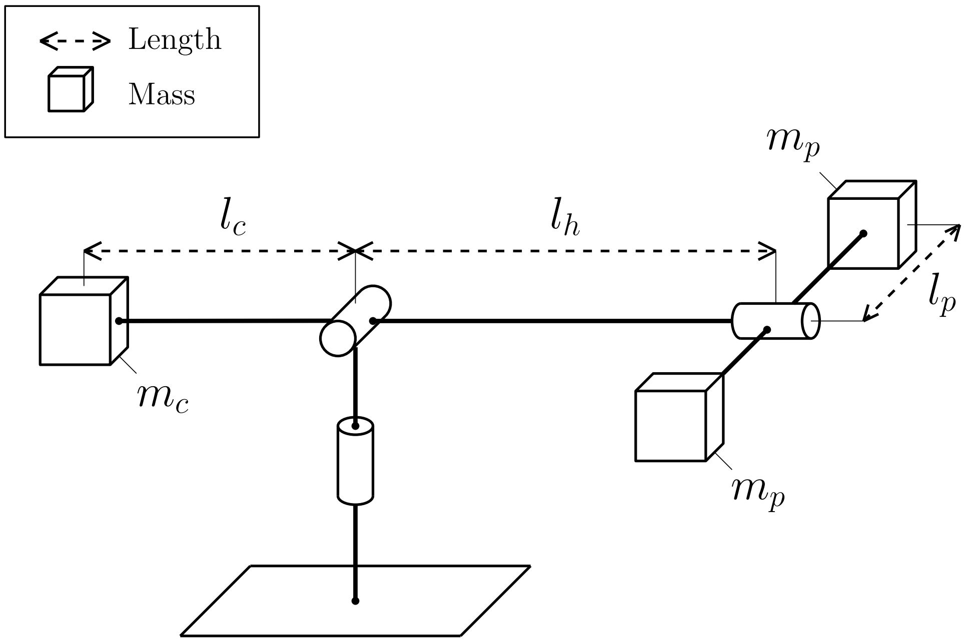

Figure 2: Masses and distances

where **F** is the force on the moment arm **r**, when **r** is at a right angle to the rotating axis.

The angular acceleration α is related to the applied torque τ by Newton's Second Law of Rotation

$$\sum \tau = J\alpha \tag{3}$$

where J is the moment of inertia about the rotating object.

2.3 Problem 1 - Modeling

For the pitch p Equation (2) and Equation (3) gives

$$\begin{aligned} J_p\ddot{p} &= l_p(F_f - F_b - F_{g,f} + F_{g,b}) \\ J_p\ddot{p} &= l_p(K_fV_f - K_fV_b - m_pg + m_pg) \\ J_p\ddot{p} &= K_fl_pV_d \end{aligned} \tag{4}$$

For the elevation e Equation (2) and Equation (3) gives

$$\begin{aligned} J_e\ddot{e} &= F_{g,c}l_c\cos(e) - 2F_{g,f}l_h\cos(e) + (F_f + F_b)l_h\cos(p) \\ J_e\ddot{e} &= g(m_cl_c - 2m_pl_h)\cos(e) + K_fl_hV_s\cos(p) \end{aligned} \tag{5}$$

For the travel λ Equation (2) and Equation (3) gives

$$J_\lambda \ddot{\lambda} = (F_f + F_b) l_h \cos(e) \cos(p)$$
$$J_\lambda \ddot{\lambda} = K_f l_h V_s \cos(e) \cos(p) \tag{6}$$

From Equation (4), Equation (5) and Equation (6) the constants can be determined:

$$L_1 = K_f l_p \tag{7a}$$
$$L_2 = g(m_c l_c - 2m_p l_h) \tag{7b}$$
$$L_3 = K_f l_h \tag{7c}$$
$$L_4 = K_f l_h \tag{7d}$$

2.4 Problem 2 - Linearization

To design a linear controller for the system, it is necessary to linearize the system around the point $(p, e, \lambda)^T = \mathbf{x}^* = (p^*, e^*, \lambda^*)^T$, where $(p^*, e^*, \lambda^*)^T$ is the equilibrium point of the system, $p^* = e^* = \lambda^* = 0$, which also implies that $(V_s, V_d)^T = \mathbf{u}^* = (V_s^*, V_d^*)$. The equilibrium point is then defined by the following:

$$p^* = e^* = \lambda^* = 0 \tag{8a}$$
$$\dot{p} = \dot{e} = \dot{\lambda} = 0 \tag{8b}$$
$$\ddot{p} = \ddot{e} = \ddot{\lambda} = 0 \tag{8c}$$

The values for V_s^* and V_f^* are then found by combining Equations (8a) and (8c) with Equations (4) and (5), which gives:

$$\frac{L_2}{J_e} + \frac{L_3}{J_e} V_s = 0 \implies V_s^* = \frac{L_2}{L_3} \tag{9a}$$
$$\frac{L_1}{J_p} V_d = 0 \implies V_d^* = 0 \tag{9b}$$

Next, the following coordinate transform is introduced:

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} = \begin{bmatrix} p \\ e \\ \lambda \end{bmatrix} - \begin{bmatrix} p^* \\ e^* \\ \lambda^* \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{u}} = \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} = \begin{bmatrix} V_s \\ V_d \end{bmatrix} - \begin{bmatrix} V_s^* \\ V_d^* \end{bmatrix} \tag{10}$$

Applying Equation (10) gives this system of equations:

$$\ddot{\tilde{p}} = \frac{L_1}{J_p} \tilde{V}_d \tag{11a}$$
$$\ddot{\tilde{e}} = \frac{L_2}{J_e} \cos(\tilde{e}) + (\frac{L_3}{J_e} \tilde{V}_s - \frac{L_2}{J_e}) \cos(\tilde{p}) \tag{11b}$$
$$\ddot{\tilde{\lambda}} = (\frac{L_4}{J_\lambda} \tilde{V}_s - \frac{L_2 L_4}{L_3 J_\lambda}) \cos(\tilde{e}) \sin(\tilde{p}) \tag{11c}$$

In order to utilize Equations (11a) to (11c), the state space variable $\mathbf{x}$ is extended to include the first order derivative of each state. Equations (11a) to (11c) can then be expressed as the following, nonlinear system:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), t) \\ \mathbf{y} &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)\end{aligned} \qquad \text{where} \quad \mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} \tag{12}$$

Using the equations described in the textbook [2, Chapter 2.4.1], the system given in Equation (12) can then be approximated to a linear system. This is done with a first order taylor expansion around the linearization point $\mathbf{x}^*$:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{13a}$$

$$\mathbf{A} := \frac{\partial \mathbf{h}}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}^*} \quad \text{and} \quad \mathbf{B} := \frac{\partial \mathbf{h}}{\partial \mathbf{u}}|_{\mathbf{u}=\mathbf{u}^*} \tag{13b}$$

Using Equation (13b), along with Equations (8a), (10), (9a) and (9b), the matrices are found to be:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ K_3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ 0 & 0 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{14}$$

which, by Equations (13a) and (14), gives the linearized equations of motion:

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

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

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

where the constants are defined as follows:

$$K_1 = \frac{L_1}{J_p} \tag{16a}$$

$$K_2 = \frac{L_3}{J_e} \tag{16b}$$

$$K_3 = \frac{-L_2 L_4}{L_3 J_\lambda} \tag{16c}$$

2.5 Problem 3 - Direct feed forward

Attempting to control the helicopter without a controller is as expected difficult. The y-axis of the joystick controlls V_s, which mainly affects the elevation, while the x-axis controlling V_d mainly affects the pitch, and therefore also the travel. The unstability of the helicopter makes the linearity of the behavior around the working point difficult to measure.

The most clear discrepancy from the model is that even with $V_d = 0$, the pitch tends to be slightly negative, causing the helicopter to rotate clockwise. It is uncertain if this was caused by differences in the weight m_p or the motor constant K_f.

The following physical factors are not captured by the model:

- Mass of the bar holding the counterweight and rotors. This affects the moments of intertia J_p, J_e and J_λ, and it will apply torque to the system due to gravity.
- Mechanical friction and drag. This will apply torque to the system, especially at high speeds.
- Saturation in actuators.

The linearization given by eq. (14) is a simplification, and only valid within a small range of the linearization point. This is another source of discrepancy between model and physical system, and it increases as the helicopter deviates from the linearization point.

2.6 Problem 4 - Offsets

Resting the helicopter at the table and scoping the measured output from the encoder gives the pitch offset, while the elevation offset is found by holding the helicopter horizontal. The offsets found are give below in degrees:

$$E_{off} = 29.2$$
$$P_{off} = 5.5$$

Attempting to estimate $V_{s,eq}$ is tough given the difficulty of controlling the helicopter without a controller. This is made significantly easier by manually balancing the helicopter while finding the elevation equilibrium. The measured value was

$$V_{s,eq} = V_s^* = 6.5V \tag{17}$$

Inserting Equation (7c) into Equation (9a) yields

$$K_f = \frac{L_2}{V_s^* l_h} \tag{18}$$

3 Part II – Monovariable control

3.1 Problem 1 - PD pitch controller

The PD contoller to implement on the system from section 2 is given by

$$\tilde{V_d} = K_{pp}(\tilde{p_c} - \tilde{p}) - K_{pd}\dot{\tilde{p}} \tag{19}$$

with $K_{pp},\ K_{pd} > 0$. Substituting into eq. (14) gives:

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

Taking the laplace tranform with $\tilde{p}(0) = \dot{\tilde{p}}(0) = 0$ gives

$$\begin{aligned}
s^2\tilde{P}(s) &= K_1 K_{pp}(\tilde{P_c}(s) - P(s)) - sK_{pd}P(s) \\
(s^2 + K_{pd}s + K_1 K_{pp})P(s) &= K_1 K_{pp}\tilde{P_c}(s) \\
\frac{\tilde{P}(s)}{\tilde{P_c}(s)} &= \frac{K_1 K_{pp}}{s^2 + K_1 K_{pd}s + K_1 K_{pp}}
\end{aligned} \tag{20}$$

Making the system critically damped will give a smooth and fast response. For a second-order linear system given by

$$h(s) = \frac{K\omega_0^2}{s^2 + 2\zeta\omega_o s + {\omega_o}^2} \tag{21}$$

this is achieved by placing both poles on the same point on the negative real axis. This is equivalent with setting the damping ratio $\zeta_p = 1$ [1, p. 143]. Comparing Equation (20) with Equation (21) reveals that $K = 1$ and $\omega_0 = \sqrt{K_1 K_{pp}} = \frac{K_1 K_{pd}}{2}$. This yields the following relation between K_{pd} and K_{pp} for achieving critical damping:

$$K_{pd} = 2\sqrt{\frac{K_{pp}}{K_1}} \tag{22}$$

It is now possible to tune the aggressiveness of the controller while (in theory) maintain critical damping by tuning the gain K_{pp}. The model discrepancies discussed in section 2.5 does however cause unstability when the gain is too high.

By experimenting, it was discovered that $K_{pp} = 1$ is not enough to maintain a pitch of 0 in order to balance the helicopter. Increasing K_{pp} to 100 introduced oscillations in the system, resulting in instability. Setting $K_{pp} = 10$ seems to give the best results.

The controller is then tested with different responses, both generated by a signal builder and by moving the joystick. The response corresponding to a step pulse in the pitch angle can be seen in figs. 3 and 4. A constant error is to be expected, using only a controller with no integral effect.

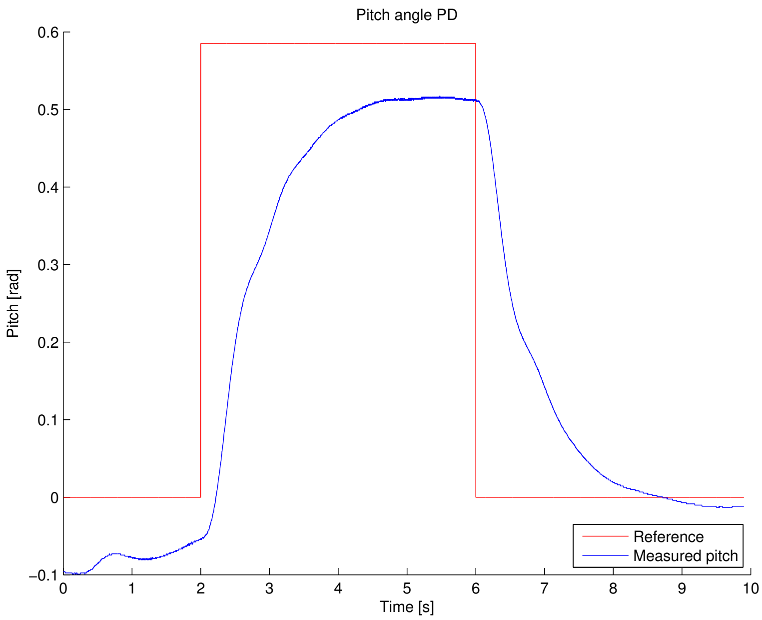

Figure 3: Pitch angle response to a step pulse with PD controller.

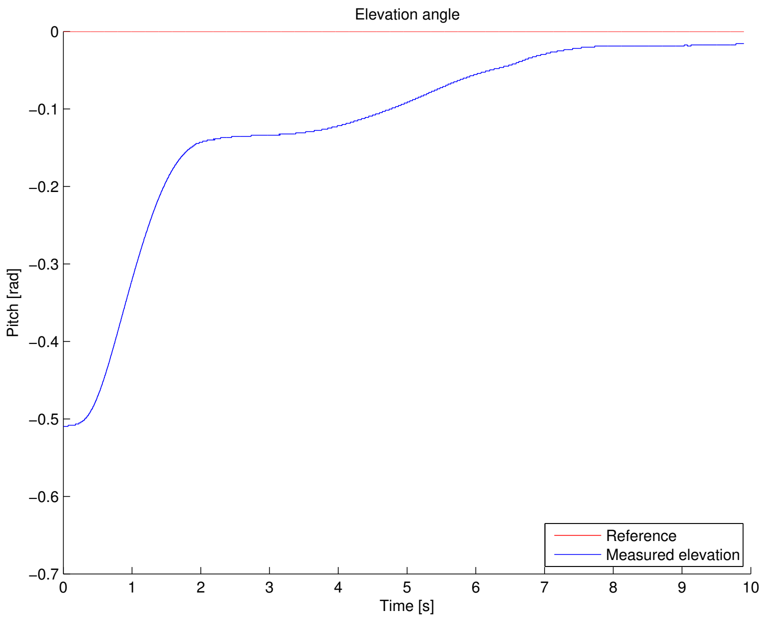

Figure 4: Corresponding elevation angle when applying a step pulse in pitch angle.

3.2 Problem 2 - P travel rate controller

The P controller given to control the travel rate $\dot{\lambda}$ is given by

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

with $K_{rp} < 0$. Assuming that the pitch angle is controlled perfectly gives

$$\tilde{p} = \tilde{p}_c \tag{24}$$

Combining Equations (15c), (23) and (24) gives the following

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

Taking the laplace transform gives

$$\begin{aligned} s\dot{\tilde{\lambda}}(s) &= K_3 K_{rp}(\dot{\tilde{\lambda}}_c(s) - \dot{\tilde{\lambda}}(s)) \\ \implies h(s) = \frac{\dot{\tilde{\lambda}}}{\dot{\tilde{\lambda}}_c}(s) &= \frac{\rho}{s+\rho} \end{aligned}$$

where $\rho = K_3 K_{rp}$.

Next, the controller gain K_{rp} is chosen to give the helicopter the desired response. First and foremost, $K_{rp} < 0$ in order for the system to stay stable, as Equation (16c) gives $K_3 < 0$, and choosing $K_{rp} > 0$ would result in a pole in the right half plane. By experimenting, choosing $K_{rp} = -2$ gives a good response, even though there are still oscillations. To further improve performance, the PD controller from problem 1 is adjusted by increasing the controller gain such that $K_{pp} = 13$. This gives quicker control towards the desired pitch p, which led to lower amounts of drift and oscillations.

The travel rate $\dot{\lambda}$ in response to a step signal can be seen in fig. 5, with the corresponding responses for the travel λ and elevation e in figs. 6 and 7.

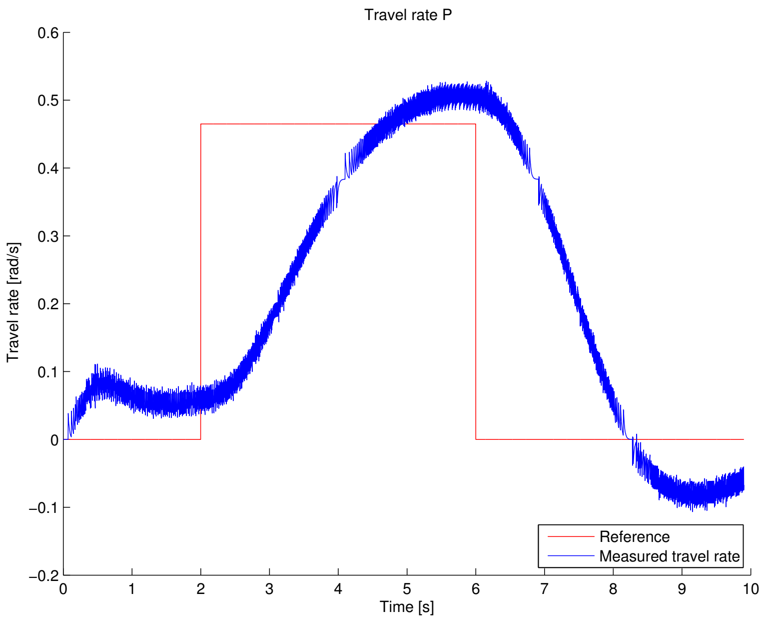

Figure 5: Travel rate response to a step pulse with P controller.

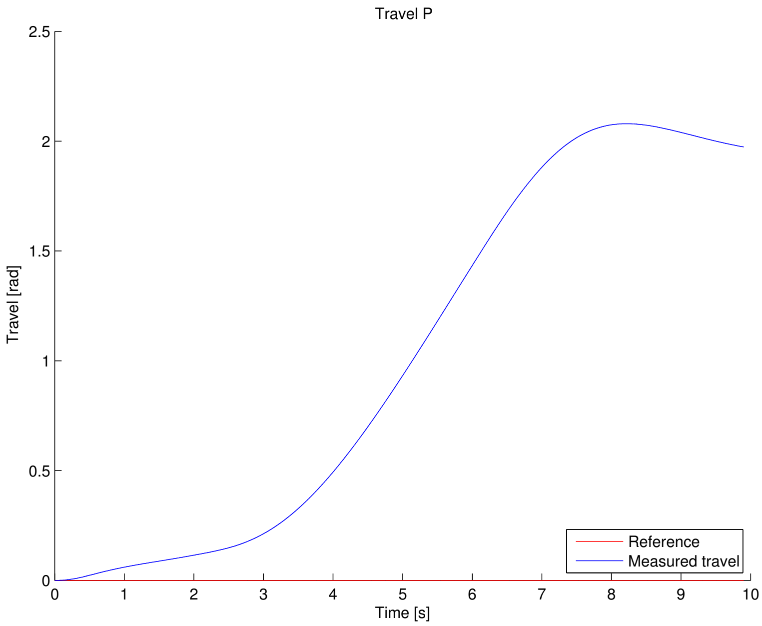

Figure 6: Corresponding travel when applying a step pulse in the travel rate.

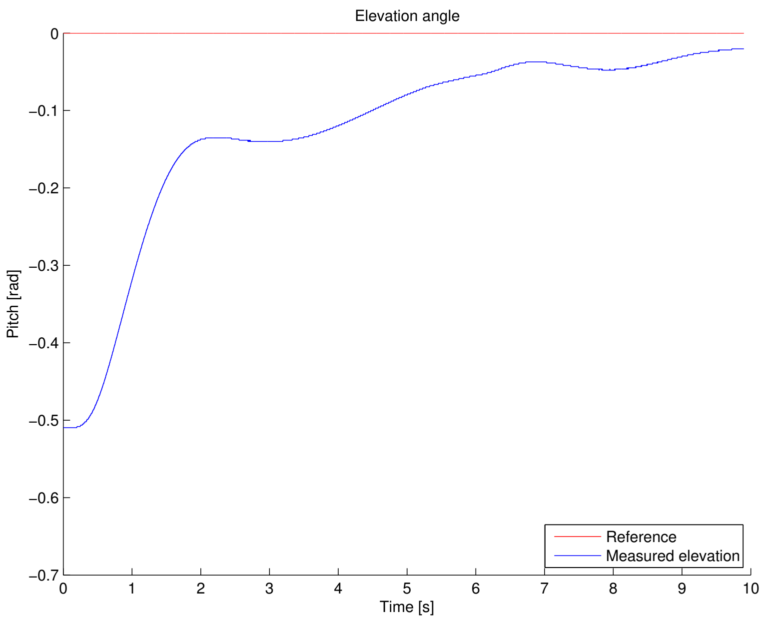

Figure 7: Corresponding elevation when applying a step pulse in the travel rate.

4 Part III – Multivariable control

4.1 Problem 1 - Pitch and elevation state space model

The new state, input and reference of the system is given by

$$\mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \end{bmatrix} \quad , \quad \mathbf{u} = \begin{bmatrix} \tilde{V_s} \\ \tilde{V_d} \end{bmatrix} \quad \text{and} \quad \mathbf{r} = \begin{bmatrix} \tilde{p_c} \\ \dot{\tilde{e_c}} \end{bmatrix} \tag{26}$$

Using eq. (15a) and eq. (15b) gives the following system matrices

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad , \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{27}$$

4.2 Problem 2 - Reference feed forward and State feedback

The reference is now chosen to be $\mathbf{r} = [\tilde{p}_c, \dot{\tilde{e}}_c]^T$, where $\tilde{p}_c$ and $\dot{\tilde{e}}_c$ are given by the joystick output on the x-axis and y-axis, respectfully. The output is subsequently chosen to be $\mathbf{y} = [\tilde{p}, \dot{\tilde{e}}]^T$

Firstly, the controllability of the system is examined by calculating the controllability matrix $\mathcal{C}$ (named `Co` in the Matlab script):

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & K_1 & 0 & 0 \\ 0 & K_1 & 0 & 0 & 0 & 0 \\ K_2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{28}$$

It is clear that $rank(\mathcal{C}) = 3$, which gives $\mathcal{C}$ full rank, and thus, according to [2, Theorem 6.1] the system is controllable.

Next, it is desired to implement a controller of the form

$$\mathbf{u} = \mathbf{Pr} - \mathbf{Kx} \tag{29}$$

where $\mathbf{Pr}$ represents the reference feed forward, and $\mathbf{Kx}$ represents the state feedback.

$\mathbf{K}$ is found such that $\mathbf{u} = -\mathbf{Kx}$ optimizes the cost function

$$J = \int_0^\infty (\mathbf{x}^T\mathbf{Qx} + \mathbf{u}^T\mathbf{Ru})dt$$

where $\mathbf{Q}$ and $\mathbf{R}$ are weighting matrices given as

$$\mathbf{Q} = \begin{bmatrix} q_1 & 0 & 0 \\ 0 & q_2 & 0 \\ 0 & 0 & q_3 \end{bmatrix} \quad \text{and} \quad \mathbf{R} = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix} \tag{30}$$

representing the costs for state error and the costs for input, respectively. $\mathbf{Q}$ and $\mathbf{R}$ are chosen to be diagonal matrices, and the weights are chosen by experimentation, aiming to get a fast and accurate response, while at the same time minimizing oscillations in the input $\mathbf{u}$. The optimal gain matrix $\mathbf{K}$ is calculated numerically using Matlab's built-in function `lqr(A,B,Q,R)`.

Reference feed forward is implemented to achieve $\lim_{t\to\infty} \mathbf{y} = \mathbf{r}$. $t \to \infty$ implies $\dot{\mathbf{x}} = \mathbf{0}$, which, combined with eqs. (27) and (29) gives:

$$\begin{aligned}
\mathbf{0} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\
\mathbf{0} &= (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} + \mathbf{B}\mathbf{P}\mathbf{r} \\
\mathbf{y} &= \mathbf{C}(\mathbf{B}\mathbf{K} - \mathbf{A})^{-1}\mathbf{B}\mathbf{P}\mathbf{r}
\end{aligned}$$

where it is clear that

$$\mathbf{P} = [\mathbf{C}(\mathbf{B}\mathbf{K} - \mathbf{A})^{-1}\mathbf{B}]^{-1} \tag{31}$$

gives $\mathbf{y} = \mathbf{r}$ when $\dot{\mathbf{x}} = \mathbf{0}$ and $t \to \infty$. The final value of $\mathbf{P}$ is hence calculated by Matlab using the formula given in eq. (31).

The response in the elevation rate $\dot{e}$ and corresponding elevation e, using the controller given by eq. (29), can be seen in Figures 8 and 9. Note that some constant error is to be expected, using only a P controller.

4.3 Problem 3 - LQR-controller with integral effect

Now the controller from Section 4.2 are modified to include an integral effect. This is done by introducing two new states γ and ζ given by

$$\dot{\mathbf{x}}_a = \begin{bmatrix} \dot{\gamma} \\ \dot{\zeta} \end{bmatrix} = \begin{bmatrix} \tilde{p} - \tilde{p_c} \\ \dot{\tilde{e}} - \dot{\tilde{e_c}} \end{bmatrix} \tag{32}$$

These states are the error in the system, meaning that they are the difference between the reference and the actual state. By eqs. (27) and (32) The augmented state space model is now given by

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_a \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x_a} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \mathbf{r} \tag{33a}$$

$$\dot{\bar{\mathbf{x}}} = \bar{\mathbf{A}}\bar{\mathbf{x}} + \bar{\mathbf{B}}\mathbf{u} + \mathbf{F}\mathbf{r} \tag{33b}$$

Where $\bar{\mathbf{x}}$ is the new state vector and $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ are the new system matrices. With the introduction of two new states the cost matrix $\mathbf{Q}$ must be expanded to include the costs of error in the new states

$$\bar{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_a} \end{bmatrix} \quad \text{where} \quad \mathbf{Q_a} = \begin{bmatrix} q_4 & 0 \\ 0 & q_5 \end{bmatrix} \tag{34}$$

The weights Q_4 and Q_5 will tune the controller gain matrix $\bar{\mathbf{K}}$ with respects to the error in the new states. Since the new states is the integral of the error between the state and reference over time, increasing these weights will place more emphasis on the integrated values.

Again the gain matrix $\bar{\mathbf{K}}$ can be calculated by the `lqr` command in Matlab yielding

$$\bar{\mathbf{K}} = \begin{bmatrix} \mathbf{K} & \mathbf{K_a} \end{bmatrix} \tag{35}$$

and $\mathbf{P}$ is calculated with eq. (31) using the new value of $\mathbf{K}$, the submatrix of $\bar{\mathbf{K}}$ controlling the states from eq. (26) such that

$$\bar{\mathbf{u}} = \begin{bmatrix} \mathbf{K} & \mathbf{K_a} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x_a} \end{bmatrix} + \mathbf{Pr} \tag{36}$$

To adjust for the integral effect V^* is increased to 10. This is because the helicopter will integrate the error in the elevation e.

The integral effect greatly reduces the drift in the helicopter. This is because without an integral effect, the controller will have a constant error. The integration of the error solves this problem by increasing its effect on the input $\bar{\mathbf{u}}$ until $\tilde{p} = \tilde{p_c}$ and $\dot{\tilde{e}} = \dot{\tilde{e}}_c$. At that point the error is 0, and further integration will not affect the input. With the new states and controller the helicopter can keep an elevation over the equilibrium point. By experimenting, the weight for the elevation error integral Q_4 needs to be quite high to eliminate drift, values around 20 gives the best result. For Q_5, values around 40 seems to give a good result.

The response in the elevation rate $\dot{e}$ and corresponding elevation e, using the controller given by eq. (36), can be seen in figs. 10 and 11. Note that the constant error seen in figs. 8 and 9 is almost completely gone.

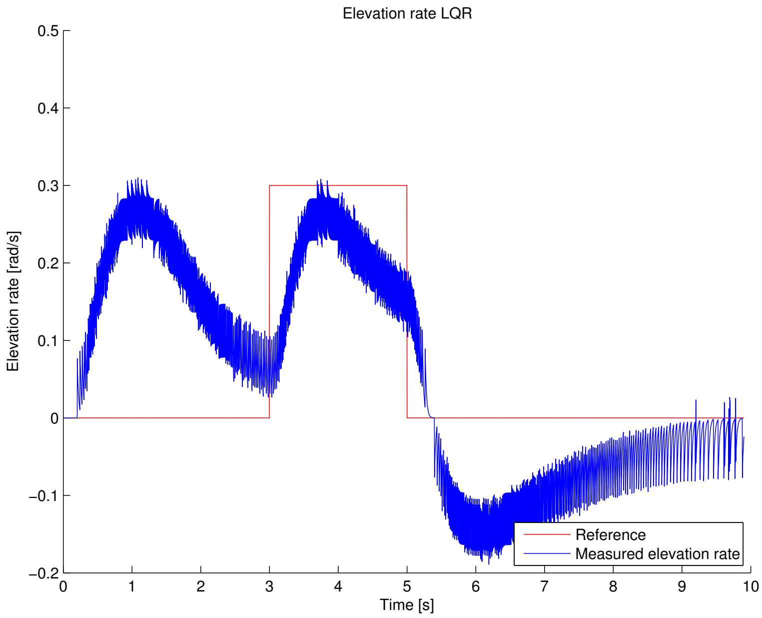

Figure 8: Elevation rate in response to a step signal with the normal LQR controller.

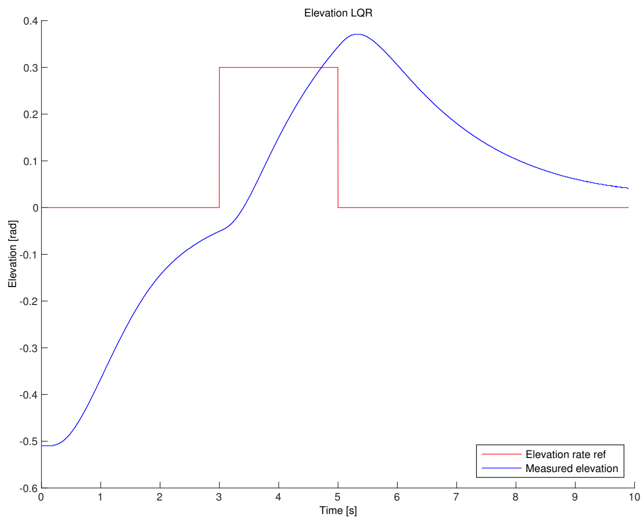

Figure 9: Elevation in response to a step signal in the elevation rate, using the normal LQR controller.

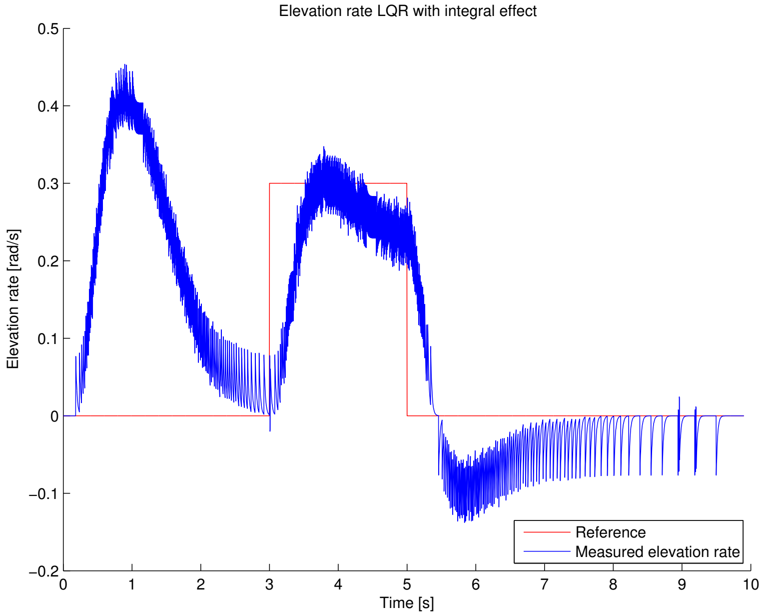

Figure 10: Elevation rate in response to a step signal with the LQR controller with an integral effect.

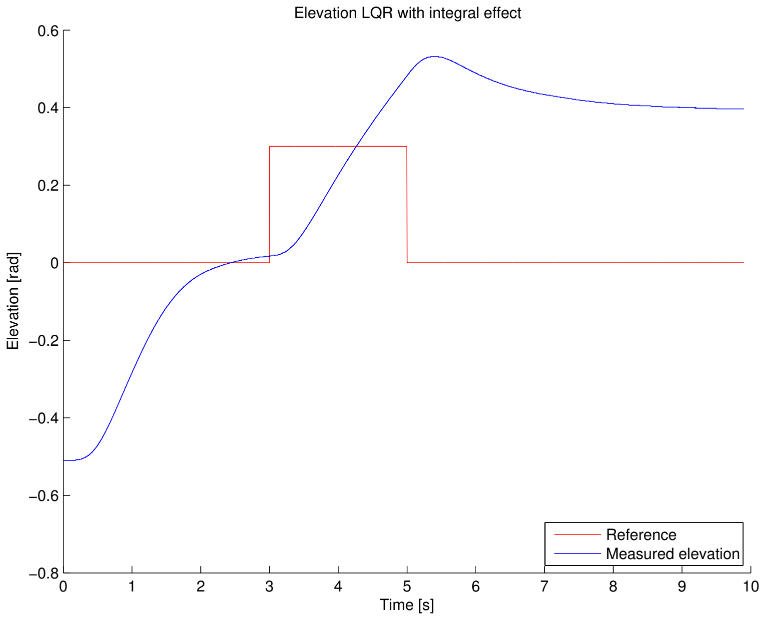

Figure 11: Elevation in response to a step signal in the elevation rate, using the LQR controller with integral effect.

5 Part IV – State estimation

5.1 Problem 1 – State space formulation

The state space model is derived based on the given system form, the given state vector $\mathbf{x}$, the given input vector $\mathbf{u}$, and the given output vector $\mathbf{y}$:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}_{se}\mathbf{x} + \mathbf{B}_{se} + \mathbf{u} \\ \mathbf{y} &= \mathbf{C}_{se}\mathbf{x}\end{aligned} \quad , \quad \mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} \quad , \quad \mathbf{u} = \begin{bmatrix} \tilde{V_s} \\ \tilde{V_d} \end{bmatrix} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} \tag{37}$$

This gives the state space model with the same system matrices $\mathbf{A}$ and $\mathbf{B}$ as in the linearized model from eqs. (13a) and (14), but with a $\mathbf{C}$ required to produce the output $\mathbf{y}$. The system is then given in its entirety as follows:

$$\mathbf{A_{se}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ K_3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad , \quad \mathbf{B_{se}} = \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ 0 & 0 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{C_{se}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{38}$$

where the constants K_1, K_2 and K_3 are given by eq. (16).

5.2 Problem 2 – Observability and Closed-loop estimation

First, the observability of the system is examined by calculating the observability matrix:

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^5 \end{bmatrix} \tag{39}$$

where the complete observability matrix $\mathcal{O}$ can be found in the appendix, in eq. (46). From the calculated matrix $\mathcal{O}$, it is clear that $rank(\mathcal{O}) = 6$, which gives $\mathcal{O}$ full rank, and thus, according to [2, Theorem 6.O1] the system is observable.

A linear observerer is then to be implemented where the system is of the form

$$\begin{aligned}\dot{\hat{\mathbf{x}}} &= \mathbf{A_{se}}\hat{\mathbf{x}} + \mathbf{B_{se}}\mathbf{u} + \mathbf{L}(\mathbf{y} - \mathbf{C_{se}}\hat{\mathbf{x}}) \\ &= (\mathbf{A}_{se} - \mathbf{L}\mathbf{C}_{se})\hat{\mathbf{x}} + \mathbf{B}_{se}\mathbf{u} + \mathbf{L}\mathbf{y} \end{aligned} \tag{40}$$

where $\mathbf{L}$ is the observer gain matrix, and $\hat{\mathbf{x}}$ is the closed-loop estimator for the state vector $\mathbf{x}$.

In order to measure the performance of the implemented observer, one has to define a measurement of the error between the actual state $\mathbf{x}$ and the estimated state $\hat{\mathbf{x}}$:

$$\mathbf{e} := \mathbf{x} - \hat{\mathbf{x}} \tag{41}$$

Differentiating $\mathbf{e}$ and then substituting eqs. (37) and (40) into eq. (41) yields

$$\begin{aligned} \dot{\mathbf{e}} &= \dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}} \\ &= \mathbf{A}_{se}\mathbf{x} + \mathbf{B}_{se}\mathbf{u} - (\mathbf{A}_{se} - \mathbf{L}\mathbf{C}_{se})\hat{\mathbf{x}} - \mathbf{B}_{se}\mathbf{u} - \mathbf{L}\mathbf{y} \\ &= (\mathbf{A}_{se} - \mathbf{L}\mathbf{C}_{se})(\mathbf{x} - \hat{\mathbf{x}}) \\ \implies \dot{\mathbf{e}} &= (\mathbf{A}_{se} - \mathbf{L}\mathbf{C})\mathbf{e} \end{aligned} \tag{42}$$

Equation (42) shows that the error $\mathbf{e}$ can be seen as its own dynamic system, where its behaviour is determined by the eigenvalues of the matrix $(\mathbf{A} - \mathbf{L}\mathbf{C})$.

The system is previously shown to be observable, which, according to [2, Theorem 8.O3], implies that the poles of $(\mathbf{A}_{se} - \mathbf{L}\mathbf{C}_{se})$ can be arbitrarily assigned by selecting a real constant matrix $\mathbf{L}$, a result one gets directly from the duality theorem given in [2, Theorem 6.5].

It is then desirable to choose the eigenvalues of $(\mathbf{A}_{se} - \mathbf{L}\mathbf{C}_{se})$ such that $\mathbf{e} \to \mathbf{0}$ as fast as possible, or, equivalently, $\hat{\mathbf{x}} \to \mathbf{x}$. This will give the controller the most recent and precise state, which will result in the correct output. In theory one could assign the eigenvalues of $(\mathbf{A}_{se} - \mathbf{L}\mathbf{C}_{se})$ to be arbitrarily large to make $\mathbf{e} \to \mathbf{0}$ quickly. However, in practice the unlinear nature of the system makes the estimator very susceptible to noise as L would not filter out the small changes in y. A good rule of thumb is therefore to choose the eigenvalues of $(\mathbf{A}_{se} - \mathbf{L}\mathbf{C}_{se})$ such that the system for $\mathbf{e}$ in eq. (42) has a response which is two to twenty times faster than the system one wishes to control [3].

In this experiment, $\mathbf{L}$ is chosen such that it places the eigenvalues of the matrix $(\mathbf{A}_{se} - \mathbf{L}\mathbf{C}_{se})$ evenly out on the negative real axis, such that the estimator is stable. The closest eigenvalue is being scaled by a given constant to move it away from the fastest pole of the closed system. This is calculated in Matlab, using the function `place(A_se', C_se', eigenvalues)'` to calculate $\mathbf{L}$. It is however worth noting that despite placing the eigenvalues evenly out on a straight line seems to give a desirable response, [2, p. 302] suggests placing the poles evenly along a circle inside a given sector. This has not been tested in this experiment.

Using the LQR controller given by $\mathbf{u} = \mathbf{Pr} - \mathbf{Kx}$ from section 4.2 given in eq. (29), one gets the closed loop system

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{BK})\mathbf{x} + \mathbf{BPr} \tag{43}$$

where $\mathbf{A}$ and $\mathbf{B}$ are given by eq. (27).

By experimenting, a scaling factor of 15 seems to give an accurate estimator as well as a stable system. Scaling the eigenvalues with a factor larger than 20 seems to give a system susceptible to noise; a scaling factor of 200 does for instance give an unstable estimator. The whole system turns unstable if the estimator poles are too slow, or, equivalently, the control of the error $\mathbf{e}$ is too slow.

Using the LQR controller with the integral effect given by eq. (36) in section 4.3, one gets the closed loop system

$$\dot{\bar{\mathbf{x}}} = (\bar{\mathbf{A}} - \bar{\mathbf{B}}\bar{\mathbf{K}})\bar{\mathbf{x}} + \bar{\mathbf{B}}\mathbf{Pr} \tag{44}$$

where $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$ and $\bar{\mathbf{K}}$ are given by eqs. (33b) and (35).

Using the same procedure as with the closed loop system given by eq. (43), the desired response is achieved. The scaling factor is now increased to 20, as this gave a better response. This is probably because using a LQR controller with an integral effect gives a faster control of the system, which in turns allows for faster control of the error $\mathbf{e}$.

Comparison between the estimated states and the real states can be seen in section 5.2. The plot shows the desired result, an estimation which is without noise, but at the same time up to speed with the real state.

The rate states shown in figs. 13 to 15 are measured by differentiating the corresponding encoder output. Even though they are low pass filtered from the encoder, they are filled with noise. The estimator will however reject the disturbance further, functioning as a low pass filter of its own.

5.3 Problem 3 - State estimator without pitch measurement

Measuring only $\tilde{p}$ and $\tilde{e}$ changes the system such that it is not observable. Using Matlabs `obsv(A, C)` gives the observability matrix $\mathcal{O}$ with a rank of 4, which means that two states are unobservable, more specifically $\tilde{\lambda}$ and $\dot{\tilde{\lambda}}$. This is because it is not possible to measure the effect of these states on our system, given only $\tilde{p}$ and $\tilde{e}$.

Measuring $\tilde{e}$ and $\tilde{\lambda}$ gives an observability matrix with full rank, therefore all states are observable. Even though $\tilde{p}$ and $\dot{\tilde{p}}$ are not directly measured, eqs. (37) and (38) describes how they are affecting the measured state λ, which provides information about their values.

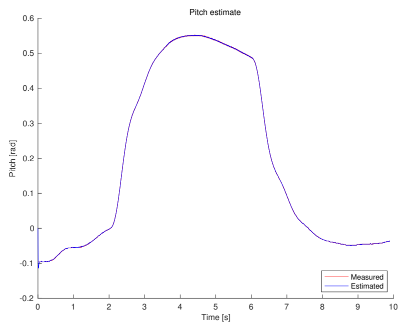

Figure 12: The estimated pitch and the measured value from the encoder (with pitch measurment)

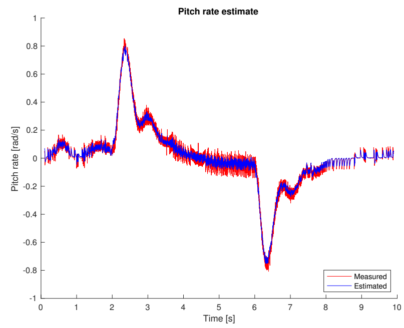

Figure 13: The estimated pitch rate and the measured value from the encoder (with pitch measurment)

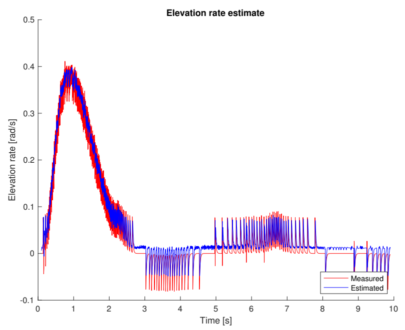

Figure 14: The estimated elevation rate and the measured value from the encoder (with pitch measurment)

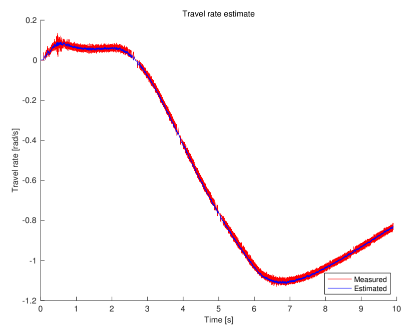

Figure 15: The estimated travel rate and the measured value from the encoder (with pitch measurment)

The system is now to be controlled without measuring the pitch $\tilde{p}$, that is

$$\mathbf{C}_{se} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} \tilde{e} \\ \tilde{\lambda} \end{bmatrix} \tag{45}$$

Tuning the estimator without measuring $\tilde{p}$ was a challenge. The best approach seemed to be to reduce the speed of the poles in $(\mathbf{A_{se}} - \mathbf{LC_{se}})$ corresponding to $\tilde{p}$ and $\dot{\tilde{p}}$ with the latter reduced the most. This is propably because $\tilde{p}$ and $\dot{\tilde{p}}$ are estimated from the higher order derivatives of the travel $\tilde{\lambda}$, which means they are very susceptible to noise. Consequently, the estimator must be less aggressive when estimating these states. Reducing the costs of $\bar{\mathbf{Q}}$ also helped stabilize the system, as a slower controller gives the estimator more time to accurately estimate the state. This is an example of how reducing the resources spent on sensors for measurements will result in a trade-off in stability and control speed.

The changes in the estimator and controller discussed above made the helicopter flyable, but noticable worse than before. The response is slower, and there are tendencies to oscillilations. Plots of the estimates is shown in section 5.3. Comparing with fig. 28 shows how the estimates for pitch and pitch rate are now more noisy and less accurate, which is the effect of eliminating the pitch sensor.

From the discussion in the paragraph above, the model explains why the estimates for the pitch $\tilde{p}$ and especially the pitch rate $\dot{\tilde{p}}$ are poor. Intuitively, the estimates for the pitch states are poor because they are not measured directly, but estimated from the measurements of the travel $\tilde{\lambda}$, elevation $\tilde{e}$, input $\bar{\mathbf{u}}$ and the model of the system given by eq. (37).

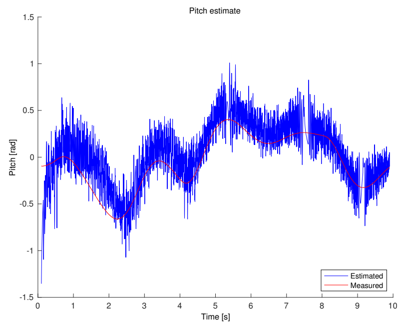

Figure 16: The estimated pitch and the measured value from the encoder (without pitch measurment)

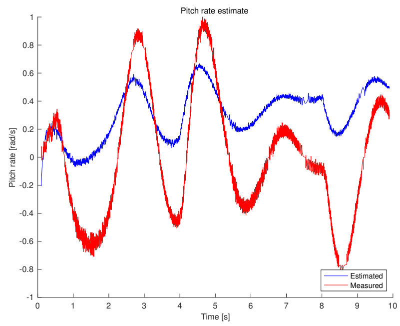

Figure 17: The estimated pitch rate and the measured value from the encoder (without pitch measurment)

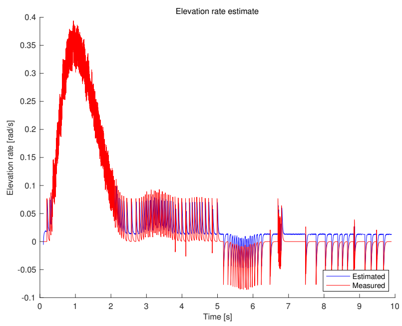

Figure 18: The estimated elevation rate and the measured value from the encoder (without pitch measurment)

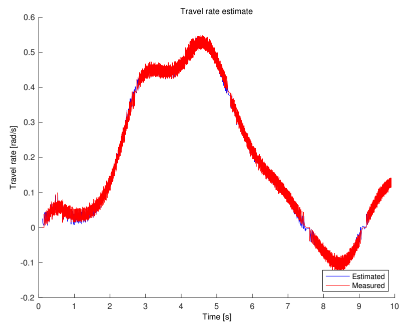

Figure 19: The estimated travel rate and the measured value from the encoder (without pitch measurment)

6 Conclusion

The assignment gave valuable insights and knowledge. The use of LQR controllers and estimators demonstrates how a more dynamical approach can be taken to control the design by giving the design engineer more options for control and sensor design. This is valuable for further development as cybernetics students.

7 Appendix

7.1 Appendix C – Matlab code

7.1.1 Part I – Mathematical modelling

```
%% Problem 1

%% TIME
t = 0:0.002:9.9;

% Joystick gains
Joystick_gain_x = 0.8;
Joystick_gain_y = -0.8;

%%%%%%%%%%% Measured constants
E_OFF = 29.2; % Elevation encoder offset
P_OFF = 5.5; % Pitch offset
V_star = 6.5; % Requiered equibalirium voltage

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

% From 5.1.4
K_f = (2*m_p*g*l_h - m_c*g*l_c)/(l_h*V_star);

% From 5.1.1
L_1 = K_f*l_p;
L_2 = m_c*g*l_c - 2*m_p*g*l_h;
L_3 = K_f*l_h;
L_4 = K_f*l_h;

% From 5.1.2
J_p = 2*m_p*l_p*l_p;
J_e = m_c*l_c*l_c + 2*m_p*l_h*l_h;
J_lambda = m_c*l_c*l_c + 2*m_p*(l_h*l_h + l_p*l_p);

K_1 = L_1/J_p;
K_2 = L_3/J_e;
```

```
K_3 = (L_2*L_4) / (L_3*J_lambda);
```

7.1.2 Part II – Monovariable control

```
%% Problem 2.1

% Joystick gains
Joystick_gain_x = 0.8;
Joystick_gain_y = -0.8;

% Controller Parameters (from problem description)
e_c = 0;
K_ei = 5;
K_ep = 15;
K_ed = 12;

K_pp = 7.5;
K_pd = 2*sqrt(K_pp/K_1);
```

```
%% Problem 2.2

K_pp = 5;
K_pd = 2*sqrt(K_pp/K_1);

K_rp = -1.5;
rho = K_3*K_rp;
```

7.1.3 Part III – Multivariable control

```
%% Problem 3.1
A = [0 1 0;
     0 0 0;
     0 0 0];

B = [0 0;
     0 K_1;
     K_2 0];

C = [1 0 0;
     0 0 1];

 %% Problem 3.2 - Controllability and LQR of the system
Co = ctrb(A, B);

q_1 = 60; % Pitch
```

```
q_2 = 10; % Pitch rate
q_3 = 80; % Elevation rate
r_1 = 1; % V_s
r_2 = 1; % V_d

Q = [q_1 0 0;
     0 q_2 0;
     0 0 q_3];

R = [r_1 0;
     0 r_2];

K = lqr(A, B, Q, R);
P = inv(C*inv(B*K - A)*B);
```

```
%% Problem 3.3 - LQR with integral effect

V_star = 10;
Joystick_gain_x = Joystick_gain_x * 0.5;

% Augmented system
A_bar = [ 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_bar = [ 0 0;
          0 K_1;
          K_2 0;
          0 0;
          0 0 ];

C_bar = [ 1 0 0 0 0;
          0 0 1 0 0 ];


% Costs for LQR with integral effect
q_1 = 30; % Pitch
q_2 = 30; % Pitch rate
q_3 = 30; % Elevation rate
q_4 = 20; % Pitch integral
q_5 = 40; % Elevation integral
r_1 = 1; % V_s
```

```
r_2 = 1; % V_d

Q_bar = [ q_1 0 0 0 0;
          0 q_2 0 0 0;
          0 0 q_3 0 0;
          0 0 0 q_4 0;
          0 0 0 0 q_5 ];

R = [r_1 0;
     0 r_2];


% LQR gain with integral effect
K_bar = lqr(A_bar, B_bar, Q_bar, R);

K = K_bar([1 2], [1 2 3]);

% Reference feed forward
P = inv(C*inv(B*K - A)*B);
```

7.1.4 Part IV – State estimation

```
%% Problem 4.2

V_star = 6.5;

A_se = [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];

B_se = [0 0;
        0 K_1;
        0 0;
        K_2 0;
        0 0;
        0 0];

C_se = [1 0 0 0 0 0;
        0 0 1 0 0 0;
        0 0 0 0 1 0];

```

```
% Observability
Ob = obsv(A_se,C_se);

%%% Closed-loop observer - Controller without integral effect
eigs_controller = eig(A-B*K);

% Use fastest pole as starting point
[~, index] = max(abs(eigs_controller));
min_eig = -abs(eigs_controller(index));
spacing = 0.15; % Spacing between the poles
eigs_gain_factor = 20; % How much each pole is scaled
eigs_estimator = 1:spacing:(1 + spacing*(size(A_se) - 1));
eigs_estimator = eigs_estimator * min_eig * eigs_gain_factor;

% Place poles as desired (Make C_se complex equivalent)
L = place(A_se', (C_se)', eigs_estimator)';
```

```
%% With integral effect

[~, index] = max(abs(eigs_controller_i));
min_eig_i = eigs_controller_i(index);
spacing_i = spacing;
eigs_gain_factor_i = eigs_gain_factor;
eigs_estimator_i = 1:spacing_i:(1 + spacing_i*(size(A_se) - 1));
eigs_estimator_i = eigs_estimator_i * min_eig_i * eigs_gain_factor_i;
L = place(A_se', C_se', eigs_estimator_i)';
```

```
%% Problem 4.3

% y without pitch
C_se_no_pitch = [0 0 1 0 0 0;
                 0 0 0 0 1 0];

% y without travel
C_se_no_travel = [1 0 0 0 0 0;
                  0 0 1 0 0 0];

% Calculate observability
Ob_no_pitch = obsv(A_se, C_se_no_pitch);
Ob_no_travel = obsv(A_se, C_se_no_travel);

% Adjust poles based on testing
% (Reduce poles for pitch and pitch rate)
eigs_estimator_i(1) = eigs_estimator_i(1) * 0.25;
```

```
eigs_estimator_i(2) = eigs_estimator_i(2) * 0.001;
% Place poles
L = place(A_se', C_se_no_pitch', eigs_estimator_i)';

q_1 = 15; % Pitch
q_2 = 15; % Pitch rate
q_3 = 30; % Elevation rate
q_4 = 2; % Pitch integral
q_5 = 20; % Elevation integral

Q_bar = [ q_1 0 0 0 0;
          0 q_2 0 0 0;
          0 0 q_3 0 0;
          0 0 0 q_4 0;
          0 0 0 0 q_5 ];

% LQR gain with integral effect
K_bar = lqr(A_bar, B_bar, Q_bar, R);
K = K_bar([1 2], [1 2 3]);

% Reference feed forward
P = inv(C*inv(B*K - A)*B);
```

7.2 Appendix B – Simulink diagrams

7.2.1 Part II – Monovariable control

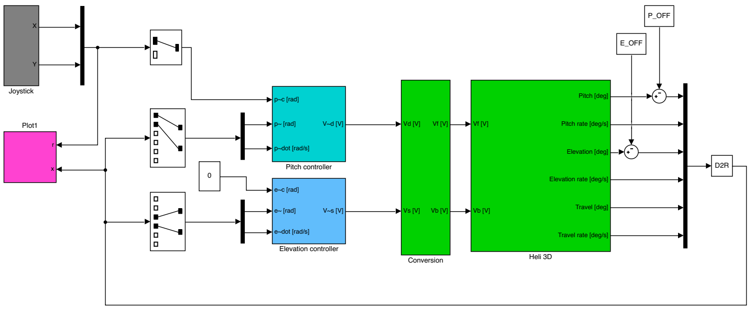

Figure 20: Simulink diagram for Part II: Problem 1.

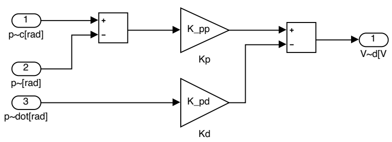

Figure 21: Part II: Problem 1. PD controller used to control the pitch.

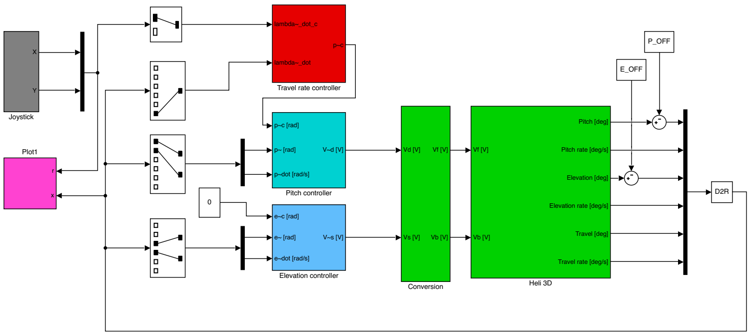

Figure 22: Simulink diagram for Part II: Problem 2.

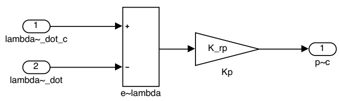

Figure 23: Part II: Problem 2. P controller used to control the travel rate.

7.2.2 Part III – Multivariable control

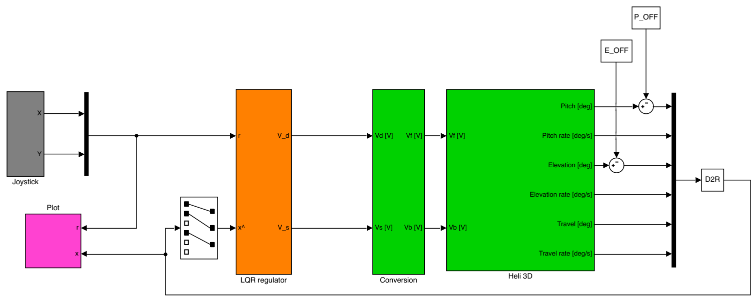

Figure 24: Simulink diagram for Part III: Problem 2.

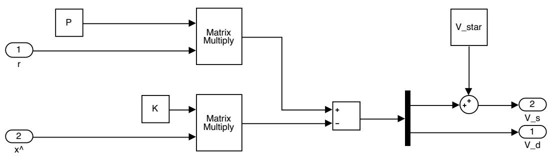

Figure 25: Part III. LQR controller used for problem 2 and 3.

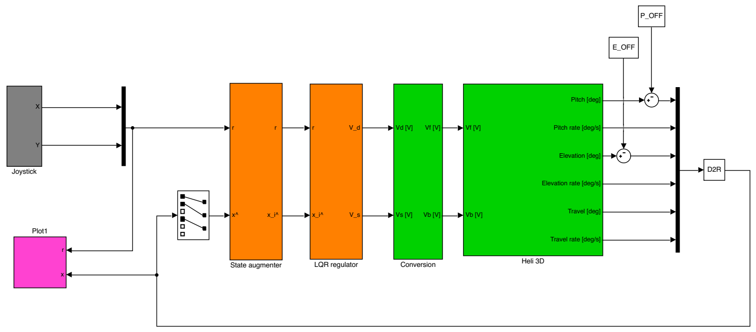

Figure 26: Simulink diagram for Part III: Problem 3.

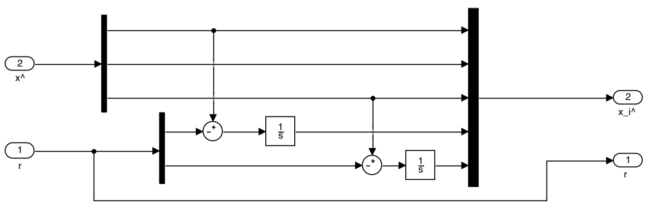

Figure 27: Part III: Problem 3. State augmenter used to add integral states.

7.2.3 Part IV – State estimation

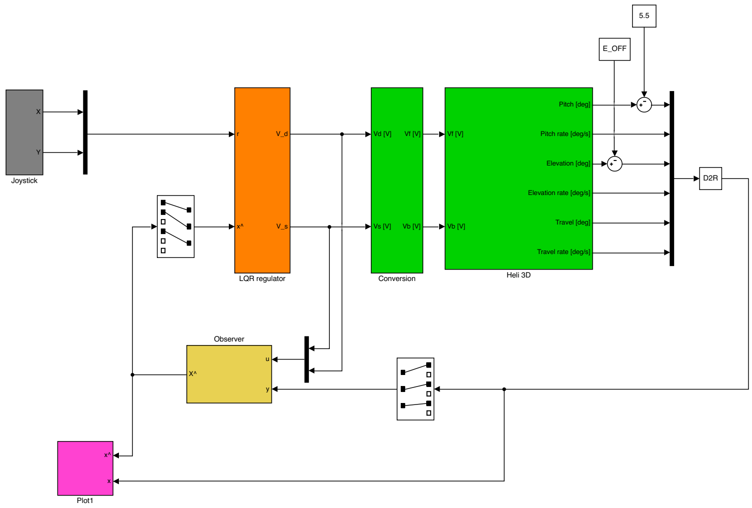

Figure 28: Simulink diagram for Part IV: Problem 2.

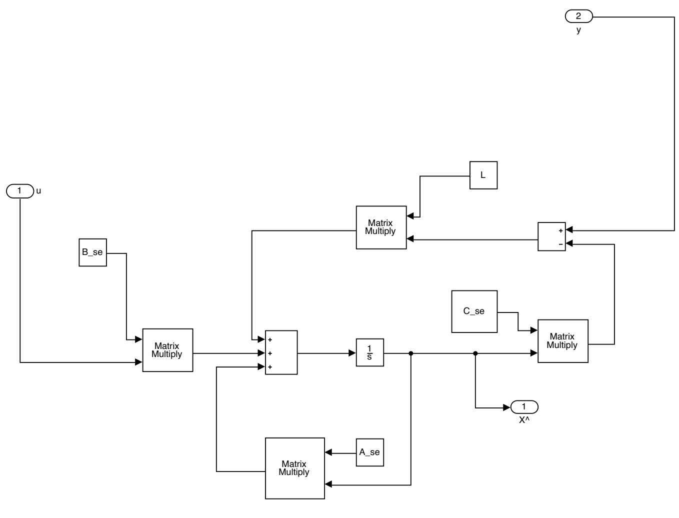

Figure 29: Part IV. Observer used to estimate states.

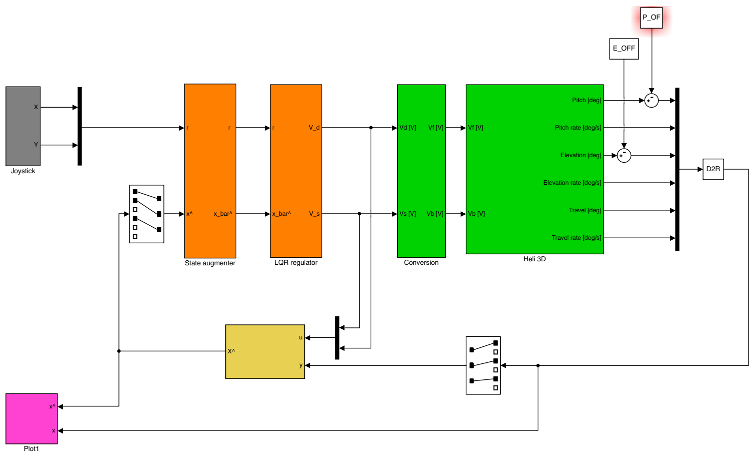

Figure 30: Simulink diagram for Part IV: Problem 2 with integral effect.

figures/simulink/P4p3.pdf

Figure 31: Simulink diagram for Part IV: Problem 3.

7.3 Appendix C – Other

7.3.1 Controllability matrix for Part IV problem 2

$$
\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ K_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{46}
$$

References

[1] Trond. Andresen, Jens. Balchen, and Bjarne. Foss. *Reguleringsteknikk*. NTNU grafiske senter, 2016.

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

[3] Morten Pedersen. *Lecture 6 - Linear System Theory*.