TTK4115 Helicopter Lab Report

Group 16
Odin Aleksander Severinsen (student number: 76 82 47)
Sigurd Totland (student number: 47 84 37)
Viktor Korsnes (student number:47 84 00)

October 21, 2018

Department of Engineering Cybernetics

Abstract

In this this project a physical helicopter model is controlled using different methods to obtain stability and easy controllability with a joystick. The control methods are all based on a linearized mathematical model of the system. Among them are feed forward control, monovariable feedback control, and feedback control using a linear quadratic regulator (LQR). The latter was done both with direct state feedback and with state estimation using a linear observer.

Contents

1 Introduction

In this project, a Quanser helicopter model was analysed and controlled using fundamental considerations from linear system theory. Important concepts from control theory and estimation theory were applied to achieve robust and responsive control of the helicopter.

Among the controllers studied were both a classical feedback controller with proportional and derivative effect, as well as a linear quadratic regulator. The latter was implemented with both reference feed forward as well as integral effect, but not at the same time.

In addition to implementing and tuning the controllers, the system was analysed for its observability and controllability when placed under different constraints. For instance, in part IV, the observability was analysed for different number of output variables. Linear observers were also used to test the controllers with estimated instead of measured state.

2 Problem Description

The problems revolves around a physical model of a helicopter arm which is fasten to the ground and has three degrees of freedom. Elevation (e), pitch (p) and travel (λ). The helicopter was controlled through a joystick and interfaced to the helicopter with MATLAB and Quanser's QUARC Real-Time Control Software. The following tasks are described in the report:

- Mathematical modelling of the system.
- Control of the system with mono- and multivariable controllers
- Synthesis of a linear quadratic regulator (LQR) for the system
- State estimation for states that are not directly measured

The helicopter can be modeled as three point masses: two masses represent the motors of the propellers and the third is the counterweight on the opposite side. Refer to fig. 1 and fig. 2 for depiction of setup. The cylindres in fig. 1 are the joints where the axis of the cylindres are equal to the axis of rotation of the joint. The following definitions are made: p is the pitch angle of the helicopter head, e is its elevation angle and λ is the travel angle of the helicopter. Fig. 1 shows all angles at zero and with positive orientation as indicated. For the sake of this report, it is assumed that there is a linear relationship between the forces the propellers generate and the voltage that is applied to them:

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

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

The proportionality constant, K_f, is called the *motor constant*.

Additionally, it is assumed that the weights of the motors and the counterweight are given as:

$$F_{g,f} = F_{g,b} = m_p g \tag{2a}$$
$$F_{g,c} = m_c g \tag{2b}$$

The propellers are placed symmetrically in relation to the pitch axis and the forces always attack perpendicular to the plane made by the pitch and elevation axis. The weight of the point masses are called $F_{g,f}, F_{g,b}$ and $F_{g,c}$ and attack parallel to gravity.

For the sake of this report, it is assumed that the weight of the two propeller motors are equal and given by m_p. The counterweight mass is indicated by m_c. The distance along the elevation axis between the travel axis and pitch axis is indicated by l_h. The length from the travel axis to the counterweight is indicated by l_c.

Further simplifications to the helicopter model have been made, such as neglect of the (quite prominent) ground effect.[1]

The report is divided into four parts. The first part will make mathematical descriptions of the described system. Part II will describe the implementation of monovariable control through the use of PD and P controllers. Part III will describe the implementation of an LQR controller and lastly, Part IV will describe the implementation of a state estimation model into the LQR controller.

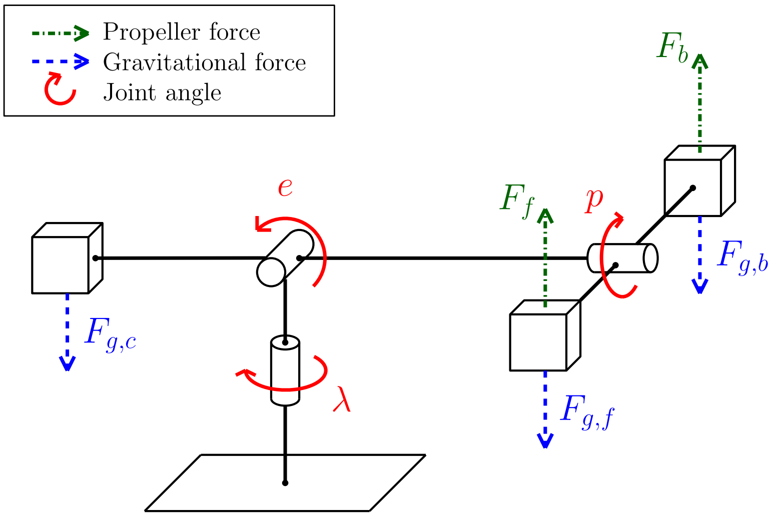

Figure 1: *Depiction of model of system. Force directions and positive angles are defined as indicated.*

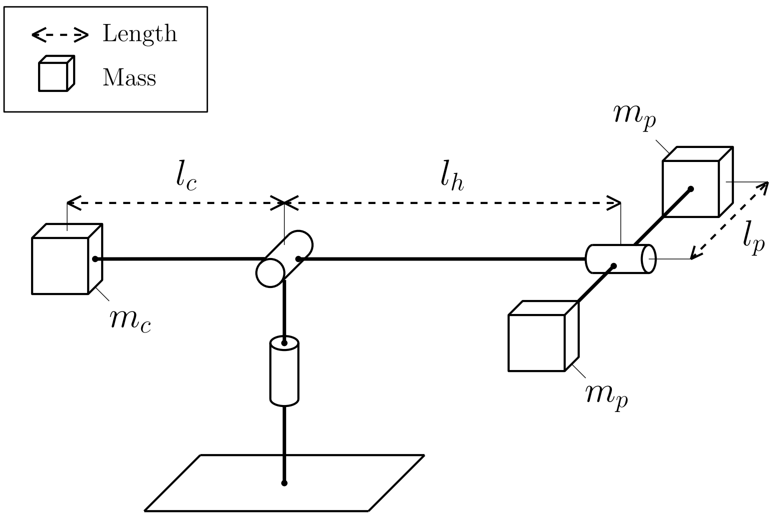

Figure 2: *Depiction of all lengths and masses of the system.*

3 Problems

3.1 Part I - Mathematical modelling

3.1.1 Problem 1

In the following derivations, Newton's 2$^{\text{nd}}$ law of motion for rotation is used:

$$J\dot{\omega} = \sum \tau = \sum (r \cdot F) \tag{3}$$

Additionally, the definitions

$$V_d = V_f - V_b \tag{4a}$$
$$V_s = V_f + V_b \tag{4b}$$

are introduced to simplify the derived equations. Lastly, it is assumed that the moments of inertia about the pitch, elevation and travel axis are, respectively

$$J_p = 2m_p l_p^2 \tag{5a}$$
$$J_e = m_c l_c^2 + 2m_p l_h^2 \tag{5b}$$
$$J_\lambda = m_c l_c^2 + 2m_p(l_h^2 + l_p^2) \tag{5c}$$

and where all constants used are according to fig. 2.

The equations of motion about the pitch axis can be computed by using eq. (3), fig. 1 and eq. (1). The forces that generate a moment about the pitch axis are the forces generated by the motors, $F_f = K_f V_f$ and $F_b = -K_f V_b$, and their weight decomposed, $F_{g,f} = -m_p g \cos(p)$ and $F_{g,b} = m_p g \cos(p)$ according to eq. (2a), where the forces are signed according to fig. 1. Given the discussed considerations, the equation of motion for rotation about the pitch axis can be derived as follows:

$$\begin{aligned}
J_p \ddot{p} &= l_p F_f - l_p F_b \\
J_p \ddot{p} &= l_p K_f V_f - l_p K_f V_b \\
J_p \ddot{p} &= l_p K_f (V_f - V_b) \\
J_p \ddot{p} &= L_1 V_d
\end{aligned} \tag{6}$$

where the definition

$$L_1 = l_p K_f \tag{7}$$

is introduced.

For the elevation, the same procedure as for pitch is used by considering how the motor forces' components about the elevation axis are a function of

p and that the weight of the motors and the counterweight have arms that are functions of e. This yields:

$$
\begin{aligned}
J_e \ddot{e} &= g l_c m_c \cos e - 2 g l_h m_p \cos e + K_f l_h V_f \cos p + K_f l_h V_b \cos p \\
J_e \ddot{e} &= L_2 \cos e + L_3 V_s \cos p \qquad (8)
\end{aligned}
$$

where the definitions

$$L_2 = g(l_c m_c - 2 l_h m_p) \qquad (9)$$

and

$$L_3 = K_f l_h \qquad (10)$$

are made.

Finally, to calculate travel, one needs to consider only the motor forces as they are the only considered forces able to generate moment about the travel axis. This component is a function of p. Additionally, the arm is a function of e. In total, this yields:

$$
\begin{aligned}
J_\lambda \ddot{\lambda} &= -(K_f l_h V_f + K_f l_h V_b) \cos e \sin p \\
J_\lambda \ddot{\lambda} &= L_4 V_s \cos e \sin p \qquad (11)
\end{aligned}
$$

Here, the definition

$$L_4 = -K_f l_h \qquad (12)$$

is made.

The reason for the negative sign is because a positive pitch gives a negative travel rate, in accordance to fig. 1. Summarised

$$\ddot{p} = \frac{L_1 V_d}{J_p} \qquad (13a)$$

$$\ddot{e} = \frac{L_2 \cos e + L_3 V_s \cos p}{J_e} \qquad (13b)$$

$$\ddot{\lambda} = \frac{L_4 V_s \cos e \sin p}{J_\lambda} \qquad (13c)$$

3.1.2 Problem 2

In this part we wish to linearise at the point $\mathbf{x}^* = \begin{bmatrix} p^* & e^* & \lambda^* \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ and $\mathbf{u}^* = \begin{bmatrix} V_s^* & V_d^* \end{bmatrix}^T$ where

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

is a coordinate transformation introduced to simplify further analysis. The linearisation will be with the helicopter horizontal in both pitch and elevation, as well as no travel (see fig. 1 to see how angles are defined). For the equilibrium point, V_d^* and V_s^* needs to be determined. An equilibrium point implies that $\dot{p} = \dot{e} = \dot{\lambda} = 0$, which again implies that $\ddot{p} = \ddot{e} = \ddot{\lambda} = 0$. Based on this and eq. (6), one can deduce that V_d^* will be zero, as anything else would eventually give a $\dot{p} \neq 0$ as $t \to \infty$.

To derive V_s^* it is possible to use eq. (8):

$$
\begin{aligned}
& J_e\ddot{e} = L_2 \cos e + L_3 V_s \cos p\Big|_{\mathbf{x}=\mathbf{x}^*} \\
\implies \quad & J_e \cdot 0 = L_2 \cdot 1 + L_3 V_s^* \cdot 1 \\
& V_s^* = -\frac{L_2}{L_3} \\
& V_s^* = -\frac{g(l_c m_c - 2 l_h m_p)}{K_f l_h}
\end{aligned}
$$

To summarise:

$$V_s^* = -\frac{g(l_c m_c - 2 l_h m_p)}{K_f l_h} \tag{14a}$$

$$V_d^* = 0 \tag{14b}$$

To simplify further analysis, the following coordinate transformation is introduced.

$$\begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} = \begin{bmatrix} p \\ e \\ \lambda \end{bmatrix} - \begin{bmatrix} p^* \\ e^* \\ \lambda^* \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} = \begin{bmatrix} V_s \\ V_d \end{bmatrix} - \begin{bmatrix} V_s^* \\ V_d^* \end{bmatrix} \tag{15}$$

Which gives the following motion equations.

$$\begin{bmatrix} p \\ e \\ \lambda \end{bmatrix} = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} V_s \\ V_d \end{bmatrix} = \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} + \begin{bmatrix} -\frac{L_2}{L_3} \\ 0 \end{bmatrix} \tag{16}$$

To linearise, we use the Jacobian and evaluate in our equilibrium

$$a_{i,j} = \frac{\partial f_i}{\partial x_j}\bigg|_{\tilde{p}=\tilde{e}=\tilde{\lambda}=0} \tag{17}$$

Where f_i represents the function from problem 1, eq. (13) and x_j represents the states. We define the following point $P = (\tilde{p} = 0, \tilde{e} = 0, \tilde{\lambda} = 0)$

From this the matrices become

$$
\begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \dot{\tilde{e}} \\ \ddot{\tilde{e}} \\ \dot{\tilde{\lambda}} \\ \ddot{\tilde{\lambda}} \end{bmatrix} = \begin{bmatrix}
\frac{\partial \dot{\tilde{p}}}{\partial \tilde{p}}\Big|_P & \frac{\partial \dot{\tilde{p}}}{\partial \dot{\tilde{p}}}\Big|_P & \frac{\partial \dot{\tilde{p}}}{\partial \tilde{e}}\Big|_P & \frac{\partial \dot{\tilde{p}}}{\partial \dot{\tilde{e}}}\Big|_P & \frac{\partial \dot{\tilde{p}}}{\partial \tilde{\lambda}}\Big|_P & \frac{\partial \dot{\tilde{p}}}{\partial \dot{\tilde{\lambda}}}\Big|_P \\
\frac{\partial \ddot{\tilde{p}}}{\partial \tilde{p}}\Big|_P & \frac{\partial \ddot{\tilde{p}}}{\partial \dot{\tilde{p}}}\Big|_P & \frac{\partial \ddot{\tilde{p}}}{\partial \tilde{e}}\Big|_P & \frac{\partial \ddot{\tilde{p}}}{\partial \dot{\tilde{e}}}\Big|_P & \frac{\partial \ddot{\tilde{p}}}{\partial \tilde{\lambda}}\Big|_P & \frac{\partial \ddot{\tilde{p}}}{\partial \dot{\tilde{\lambda}}}\Big|_P \\
\frac{\partial \dot{\tilde{e}}}{\partial \tilde{p}}\Big|_P & \frac{\partial \dot{\tilde{e}}}{\partial \dot{\tilde{p}}}\Big|_P & \frac{\partial \dot{\tilde{e}}}{\partial \tilde{e}}\Big|_P & \frac{\partial \dot{\tilde{e}}}{\partial \dot{\tilde{e}}}\Big|_P & \frac{\partial \dot{\tilde{e}}}{\partial \tilde{\lambda}}\Big|_P & \frac{\partial \dot{\tilde{e}}}{\partial \dot{\tilde{\lambda}}}\Big|_P \\
\frac{\partial \ddot{\tilde{e}}}{\partial \tilde{p}}\Big|_P & \frac{\partial \ddot{\tilde{e}}}{\partial \dot{\tilde{p}}}\Big|_P & \frac{\partial \ddot{\tilde{e}}}{\partial \tilde{e}}\Big|_P & \frac{\partial \ddot{\tilde{e}}}{\partial \dot{\tilde{e}}}\Big|_P & \frac{\partial \ddot{\tilde{e}}}{\partial \tilde{\lambda}}\Big|_P & \frac{\partial \ddot{\tilde{e}}}{\partial \dot{\tilde{\lambda}}}\Big|_P \\
\frac{\partial \dot{\tilde{\lambda}}}{\partial \tilde{p}}\Big|_P & \frac{\partial \dot{\tilde{\lambda}}}{\partial \dot{\tilde{p}}}\Big|_P & \frac{\partial \dot{\tilde{\lambda}}}{\partial \tilde{e}}\Big|_P & \frac{\partial \dot{\tilde{\lambda}}}{\partial \dot{\tilde{e}}}\Big|_P & \frac{\partial \dot{\tilde{\lambda}}}{\partial \tilde{\lambda}}\Big|_P & \frac{\partial \dot{\tilde{\lambda}}}{\partial \dot{\tilde{\lambda}}}\Big|_P \\
\frac{\partial \ddot{\tilde{\lambda}}}{\partial \tilde{p}}\Big|_P & \frac{\partial \ddot{\tilde{\lambda}}}{\partial \dot{\tilde{p}}}\Big|_P & \frac{\partial \ddot{\tilde{\lambda}}}{\partial \tilde{e}}\Big|_P & \frac{\partial \ddot{\tilde{\lambda}}}{\partial \dot{\tilde{e}}}\Big|_P & \frac{\partial \ddot{\tilde{\lambda}}}{\partial \tilde{\lambda}}\Big|_P & \frac{\partial \ddot{\tilde{\lambda}}}{\partial \dot{\tilde{\lambda}}}\Big|_P
\end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} + \begin{bmatrix}
\frac{\partial \dot{\tilde{p}}}{\partial \tilde{V}_s}\Big|_P & \frac{\partial \dot{\tilde{p}}}{\partial \tilde{V}_d}\Big|_P \\
\frac{\partial \ddot{\tilde{p}}}{\partial \tilde{V}_s}\Big|_P & \frac{\partial \ddot{\tilde{p}}}{\partial \tilde{V}_d}\Big|_P \\
\frac{\partial \dot{\tilde{e}}}{\partial \tilde{V}_s}\Big|_P & \frac{\partial \dot{\tilde{e}}}{\partial \tilde{V}_d}\Big|_P \\
\frac{\partial \ddot{\tilde{e}}}{\partial \tilde{V}_s}\Big|_P & \frac{\partial \ddot{\tilde{e}}}{\partial \tilde{V}_d}\Big|_P \\
\frac{\partial \dot{\tilde{\lambda}}}{\partial \tilde{V}_s}\Big|_P & \frac{\partial \dot{\tilde{\lambda}}}{\partial \tilde{V}_d}\Big|_P \\
\frac{\partial \ddot{\tilde{\lambda}}}{\partial \tilde{V}_s}\Big|_P & \frac{\partial \ddot{\tilde{\lambda}}}{\partial \tilde{V}_d}\Big|_P
\end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{18}
$$

With values

$$
\begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \dot{\tilde{e}} \\ \ddot{\tilde{e}} \\ \dot{\tilde{\lambda}} \\ \ddot{\tilde{\lambda}} \end{bmatrix} = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
-\frac{L_4 L_2}{J_\lambda L_3} & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} + \begin{bmatrix}
0 & 0 \\
\frac{L_1}{J_p} & 0 \\
0 & 0 \\
0 & \frac{L_3}{J_e} \\
0 & 0 \\
0 & 0
\end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{19}
$$

This simplifies to

$$
\dot{\tilde{p}} = \dot{\tilde{p}}
$$

$$
\ddot{\tilde{p}} = K_1 \tilde{V}_s \tag{20a}
$$

$$
\dot{\tilde{e}} = \dot{\tilde{e}}
$$

$$
\ddot{\tilde{e}} = K_2 \tilde{V}_d \tag{20b}
$$

$$
\dot{\tilde{\lambda}} = \dot{\tilde{\lambda}}
$$

$$
\ddot{\tilde{\lambda}} = K_3 \tilde{p} \tag{20c}
$$

Where

$$
K_1 = \frac{L_1}{J_p} \tag{21a}
$$

$$
K_2 = \frac{L_3}{J_e} \tag{21b}
$$

$$
K_3 = -\frac{L_4 L_2}{J_\lambda L_3} \tag{21c}
$$

3.1.3 Problem 3

Controlling the helicopter with feed forward is very difficult. This stems from an inaccurate model. The model has multiple simplifications compared to the actual system.

3.1.4 Problem 4

In order to implement the controllers required in the later parts, the encoder outputs from the helicopter need to be converted from degrees to radians. This is done by adding gains of value $\frac{\pi}{180^\circ}$ to the outputs. These gains, labeled D2R can be seen in most of the simulink diagrams in Appendix A, for example in fig. 22.

Furthermore, since the elevation is defined to be zero when the helicopter is horizontal, an offset voltage V_s^* is needed. This voltage was measured by slowly increasing the voltage V_s with the joystick, all the while restraining the other two axes to zero by holding the helicopter loosely at its head. By doing so in a controlled manner, the voltage offset was found to be

$$V_s^* \approx 6.5\text{V}. \tag{22}$$

From the voltage offset, the motor force constant K_f can be calculated. By rearranging eq. (14a) and inserting values for the constants, K_f can be calculated to be

$$K_f = -\frac{g(l_c m_c - 2l_h m_p)}{V_s^* l_h} \approx 0.1537\frac{\text{Nm}}{\sqrt{\text{W}}}. \tag{23}$$

The constant values can be found in table 1 in Appendix B.

3.2 Part II - Monovariable Control

Part II takes a look at PD and P controllers. The PD is for pitch, and P for travel rate. The following formulas describe the controllers.

$$\tilde{V}_d = K_{pp}(\tilde{p}_c - \tilde{p}) - K_{pd}\dot{\tilde{p}} \tag{24}$$

$$\tilde{p}_c = K_{lp}(\dot{\tilde{\lambda}}_c - \dot{\tilde{\lambda}}) \tag{25}$$

In Simulink, eq. (24) and eq. (25) are implemented as **??** and fig. 23, respectively.

3.2.1 Problem 1

System Dynamics

It is possible to derive the dynamics of eq. (24) by replacing $\tilde{V}_d$ with the linearised equation from section 3.1, eq. (20a).

$$\begin{aligned}
\tilde{V}_d &= K_{pp}(\tilde{p}_c - \tilde{p}) - K_{pd}\dot{\tilde{p}} \\
\frac{\ddot{\tilde{p}}}{K_1} &= K_{pp}(\tilde{p}_c - \tilde{p}) - K_{pd}\dot{\tilde{p}} \\
\ddot{\tilde{p}} &= K_1\left(K_{pp}(\tilde{p}_c - \tilde{p}) - K_{pd}\dot{\tilde{p}}\right)
\end{aligned}$$

By assuming initial conditions to be zero, one arrives at the following Laplace transform:

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

From this, the following damping ratio and natural frequency follows

$$\zeta = \frac{K_1 K_{pd}}{2\sqrt{K_1 K_{pp}}} \tag{26}$$

$$\omega_0 = \sqrt{K_1 K_{pp}} \tag{27}$$

For optimal behaviour, $\zeta = 1$ is chosen as this corresponds to critical damping of the system, where the system is neither under- nor overdamped.
With $\zeta = 1$, eq. (26) and eq. (27) yields

$$K_{pd} = 2\sqrt{\frac{K_{pp}}{K_1}} \tag{28}$$

$$K_{pp} = \frac{\omega_0^2}{K_1}. \tag{29}$$

Controller Tuning
Finding an optimal ω_0 is difficult, but as can be seen from eq. (28), K_{pd} is purely dependent on K_{pp} and the constant K_1. Thus, the pitch controller was tuned simply by testing different values of K_{pp}.

To systematically test different values of K_{pp}, a simple step block in the Simulink model was used that took the pitch reference from 45° to 0°[1]. Having the reference converge to 0° is chosen as the system model is linearised around that value. The response was tested for values of K_{pp} ranging from 4 to 20. The findings, shown in fig. 3, show that a K_{pp} of about 10 gives a rapid response without overshoot. This ultimately gives

- $K_{pp} = 10$
- $K_{pd} = 8.11$

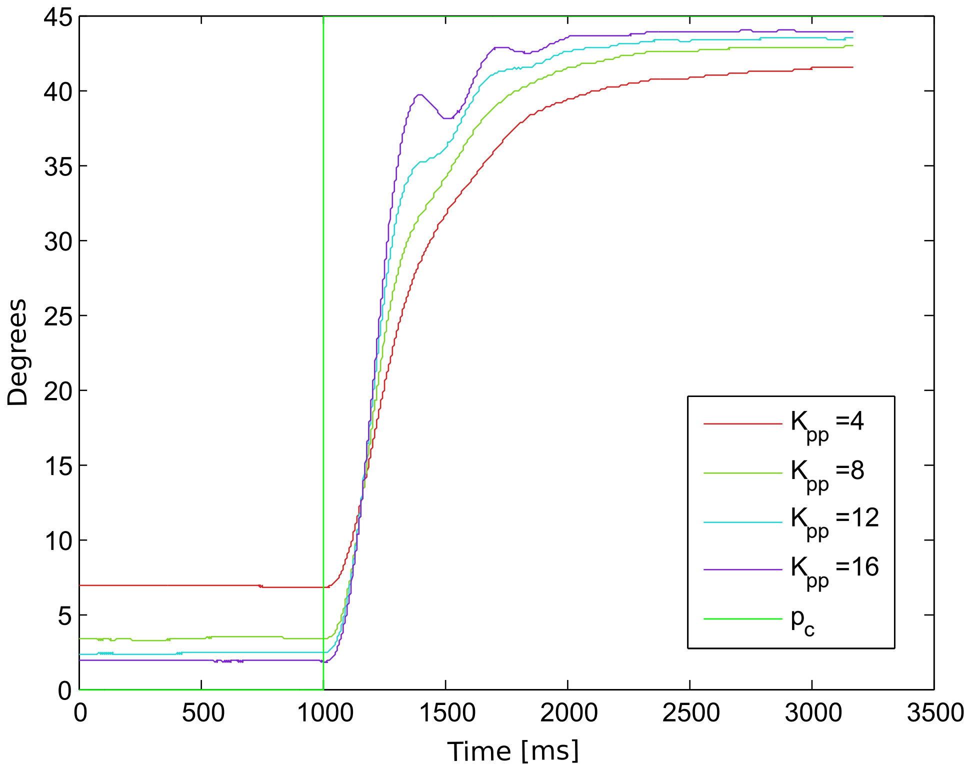

Figure 3: *Pitch response with PD control*

[1]fig. 3 shows the response going from 0° to 45°. The plot is intentionally presented this way for easier visualisation.

3.2.2 Problem 2

Transfer Function

The transfer function for the travel controller in eq. (25) can be written as

$$\frac{\dot{\tilde{\lambda}}(s)}{\dot{\tilde{\lambda}}_c(s)} = \frac{\rho}{s+\rho}, \tag{30}$$

where ρ is a constant. Assuming that the pitch angle is controlled perfectly, this can be shown by rewriting eq. (20c) so that

$$\tilde{p} = \frac{\ddot{\tilde{\lambda}}}{K_3}. \tag{31}$$

Inserting this into eq. (25) then yields

$$\ddot{\tilde{\lambda}} = K_3 K_{lp}(\dot{\tilde{\lambda}}_c - \dot{\tilde{\lambda}}),$$

which in the laplace domain is

$$s\dot{\tilde{\lambda}} = K_3 K_{lp}(\dot{\tilde{\lambda}}_c - \dot{\tilde{\lambda}}).$$

Finally, by rearranging the terms, the transfer function becomes

$$h(s) = \frac{\dot{\tilde{\lambda}}}{\dot{\tilde{\lambda}}_c} = \frac{K_3 K_{lp}}{s + K_3 K_{lp}}, \tag{32}$$

which when letting $\rho = K_3 K_{lp}$ is the same as eq. (30).

Controller Tuning

To find a proper value for K_{lp} one can look at the poles for both the pitch and travel controller. Because travel is dependant on pitch, pitch control should be faster than travel. From the perspective of pole placement, this means that the magnitude of the pitch poles should be greater than the travel poles. After some trial and error, $K_{lp} = -1.15$ was found to produce a rapid response without any considerable overshoot, as can be seen in fig. 4. From the pole plot in fig. 5 we can see that the pitch controller is much faster.

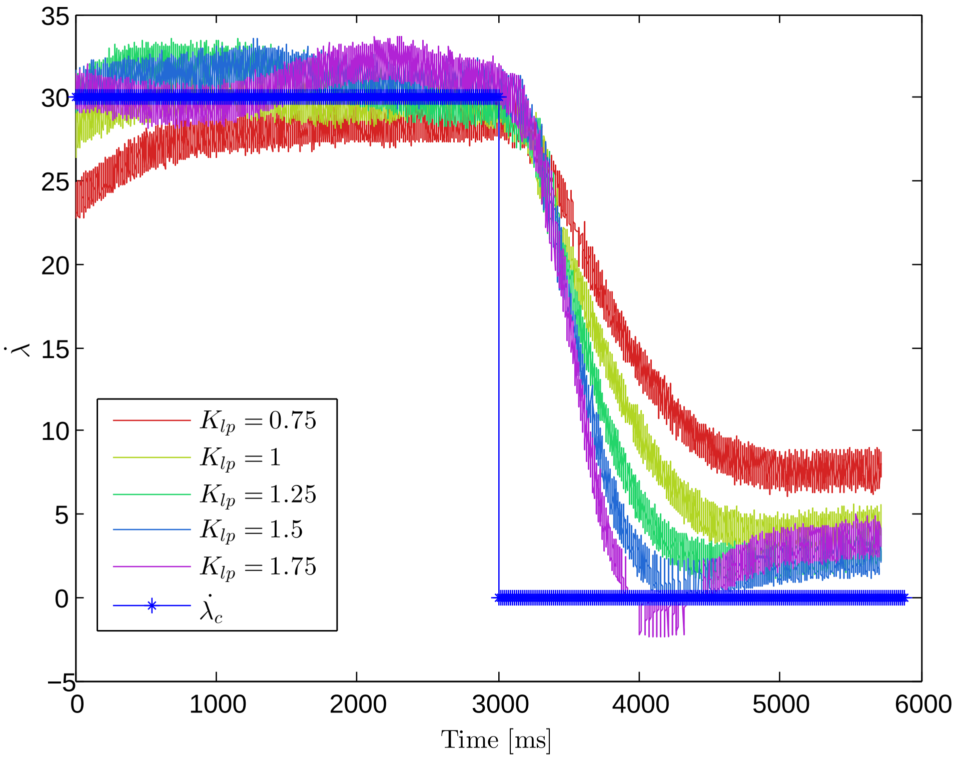

Figure 4: *Travel response with P control*

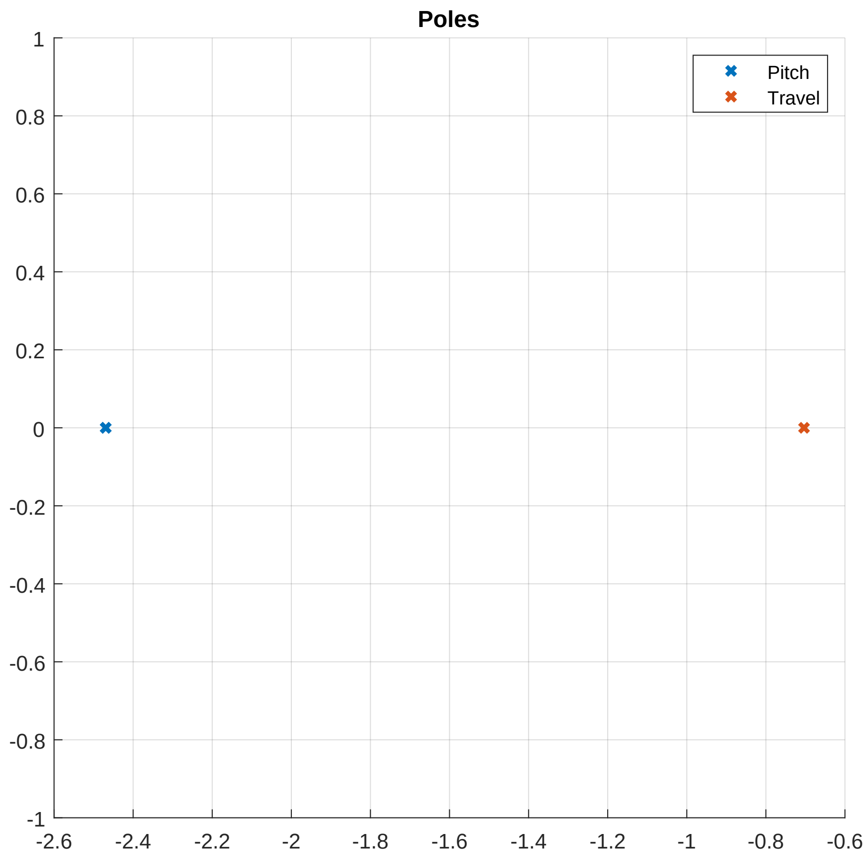

Figure 5: *Travel and pitch controller poles*

3.3 Part III - Multivariable control

3.3.1 Problem 1

Considering the linearised system in eq. (20) with a state and input vector of

$$\mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \end{bmatrix} \quad \text{and} \quad \mathbf{u} = \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{33}$$

and a state-space formulation of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \tag{34}$$

the matrices $\mathbf{A}$ and $\mathbf{B}$ become

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{35a}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ K_1 & 0 \\ 0 & K_2 \end{bmatrix}. \tag{35b}$$

3.3.2 Problem 2

Controllability Analysis
In this problem, the purpose was to track a multivariable reference

$$\mathbf{r} = \begin{bmatrix} \tilde{p_c} \\ \dot{\tilde{e}}_c \end{bmatrix}. \tag{36}$$

Firstly, the controllability of the system is examined. Due to eq. (33) having three states, the controllability matrix is given by

$$\mathcal{C} = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A^2B} \end{bmatrix} = \begin{bmatrix} 0 & 0 & K_1 & 0 & 0 & 0 \\ K_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_2 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{37}$$

With the values $K_1 = 0.9046$ and $K_2 = 0.1555$ inserted, this becomes

$$\mathcal{C} = \begin{bmatrix} 0 & 0 & 0.9046 & 0 & 0 & 0 \\ 0.9046 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1555 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{38}$$

Since $\mathcal{C}$ has full row rank, the system is controllable.
Controller Synthesis
The controller to be implemented for this part is a controller of the form

$$\mathbf{u} = \mathbf{Pr} - \mathbf{Kx} \tag{39}$$

with both state feedback and reference feed forward. The controller will implement a linear quadratic regulator (LQR) which means it minimises the cost function

$$J = \int_0^\infty (\mathbf{x}^{\mathrm{T}}(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^{\mathrm{T}}\mathbf{R}\mathbf{u}(t))\mathrm{d}t. \tag{40}$$

The optimal K is then given by

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{S} \tag{41}$$

and $\mathbf{S}$ is found by solving the continuous algebraic Ricatti equation (CARE)

$$\mathbf{A}^{\mathrm{T}}\mathbf{S} + \mathbf{SA} + \mathbf{C}^{\mathrm{T}}\mathbf{QC} - \mathbf{SBS}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{S} = \mathbf{0}. \tag{42}$$

The LQR is especially handy for controller tuning, as one can have precise control over which state variables and which reference variables should have the most "cost" assigned to them. The term cost in this sense, refers to how "expensive" deviation from the set point for one particular state parameter should be, i.e. how quickly the controller should counteract such errors. These cost variables are given in the diagonal matrices $\mathbf{R}$ and $\mathbf{Q}$, where each entry along the diagonal of $\mathbf{Q}$ correspond to the cost of each state in $\mathbf{x}$, and each entry along the diagonal of $\mathbf{R}$ correspond to the cost of using the outputs in $\mathbf{u}$.

The matlab function `lqr(A,B,Q,R)` can be used to calculate the needed feedback gain matrix K. It minimises the cost function in eq. (40) and solves the Ricatti equation internally.

The reference feed forward matrix $\mathbf{P}$ can be found by letting $t \to \infty$, at which point

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}_\infty + \mathbf{B}\mathbf{u} = 0. \tag{43}$$

Inserting the controller from eq. (39) then yields

$$\dot{\mathbf{x}} = \mathbf{BPr} - \mathbf{BK}\mathbf{x}_\infty = \mathbf{Bpr} + (\mathbf{A} - \mathbf{BK})\mathbf{x}_\infty = 0. \tag{44}$$

Since $(\mathbf{BK} - \mathbf{A})$ is invertible,

$$\mathbf{x}_\infty = (\mathbf{BK} - \mathbf{A})^{-1}\mathbf{BPr}.$$

Now, because $\mathbf{y} = \mathbf{Cx}$,

$$\mathbf{y}_\infty = \mathbf{C}(\mathbf{BK} - \mathbf{A})^{-1}\mathbf{BPr},$$

and since the output will eventually tend towards the reference,

$$\mathbf{y}_\infty = \mathbf{r}$$

$$\mathbf{C}(\mathbf{BK} - \mathbf{A})^{-1}\mathbf{BP} = \mathbf{I},$$

where $\mathbf{I}$ is the identity matrix. So finally, by the invertability of $\mathbf{C}(\mathbf{BK} - \mathbf{A})^{-1}\mathbf{B}$,

$$\mathbf{P} = [\mathbf{C}(\mathbf{BK} - \mathbf{A})^{-1}\mathbf{B}]^{-1}. \tag{45}$$

Controller Tuning

The weighting matrices chosen are

$$\mathbf{Q} = \begin{bmatrix} 10000 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10000 \end{bmatrix} \quad \text{and} \quad \mathbf{R} = \begin{bmatrix} 75 & 0 \\ 0 & 100 \end{bmatrix}, \tag{46}$$

which corresponds to the controller matrices

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 11.547 \\ 10 & 5.7357 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{P} = \begin{bmatrix} 0 & 11.547 \\ 10 & 0 \end{bmatrix}. \tag{47}$$

Looking at the values of $\mathbf{Q}$, one can see that emphasis has been put on the pitch $\tilde{p}$ and the elevation $\tilde{e}$. The pitch was found to be one of the most important states, as a lack of pitch control often caused the helicopter to hit the ground. Furthermore, high cost on the pitch rate caused the helicopter to turn too fast, which made it difficult to control.

For the **R** matrix, too high values made the controls very slow, as the high cost penalised the inputs too much. On the other hand, having them too small made the controls helicopter jittery and the joystick very sensitive.

In the end, when using these tuning parameters, the LQR controller provided robust and responsive control of the helicopter, all though a slight drift in the elevation was impossible to circumvent.

As is shown in figure fig. 6, the elevation rate responds quite quickly to a step in reference. So does the pitch, as shown in fig. 7. A Simulink diagram showing the LQR implementation is shown in fig. 8 and a top level view is shown in fig. 24.

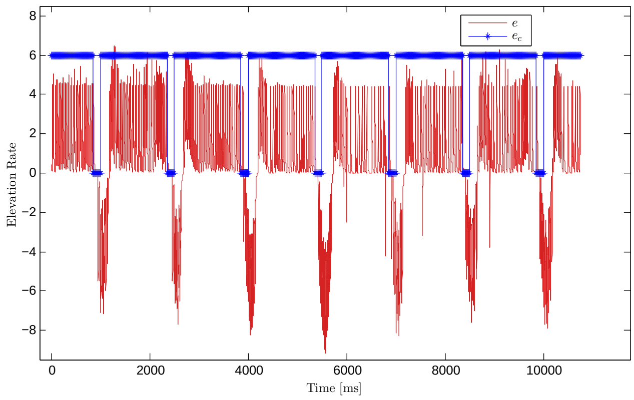

Figure 6: *Elevation rate response with LQR*

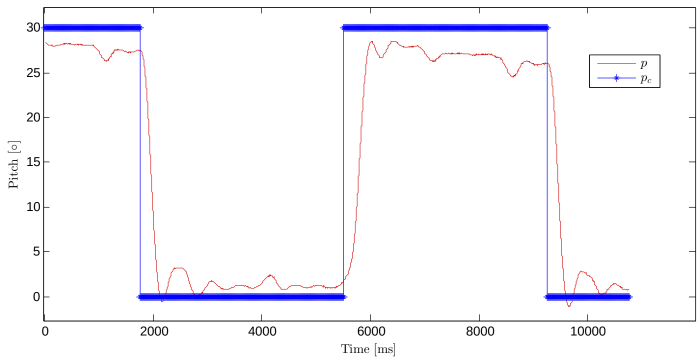

Figure 7: *Pitch response with LQR*

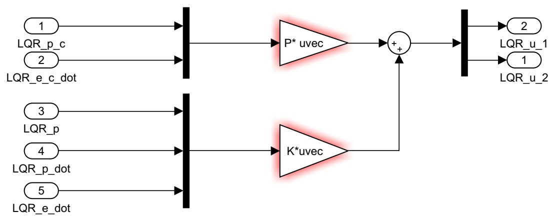

Figure 8: *LQR controller implementation in Simulink*

3.3.3 Problem 3

To achieve integral effect it is necessary to introduce two new variables.

$$\dot{\gamma} = \tilde{p} - \tilde{p}_c \tag{48}$$
$$\dot{\zeta} = \dot{\tilde{e}} - \dot{\tilde{e}}_c \tag{49}$$

By introducing an integral effect the stationary error should vindicate. This will have a colossal effect on the elevation controller, as drifting in elevation is quite noticeable. Figure 9 shows the simulink model of the new multivariable controller. Because pitch was no problem in part 2, we didn't weigh γ much,

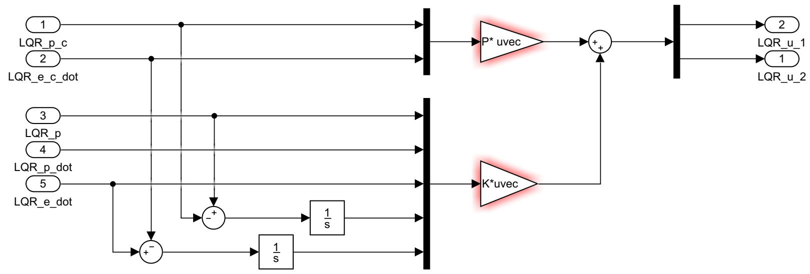

Figure 9: *Multivariable controller with integral effect.*

but since elevation was not working to well, we boosted ζ.

$$Q = \begin{bmatrix} 1000 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1500 \end{bmatrix} \tag{50}$$

$$R = \begin{bmatrix} 8 & 0 \\ 0 & 11 \end{bmatrix} \tag{51}$$

It is clear from the pitch that the linearised model is being pushed outside its comfort zone. The Elevation rate is quite disturbed by discretization

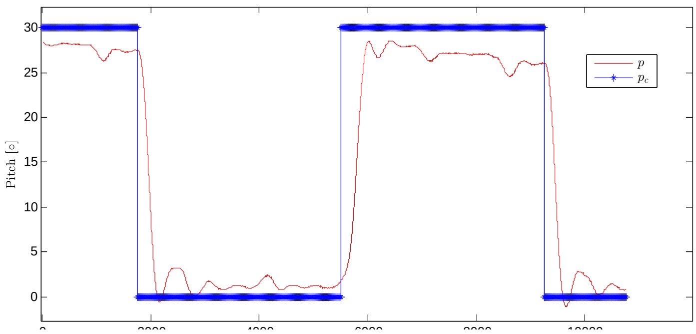

Figure 10: *Pitch.*

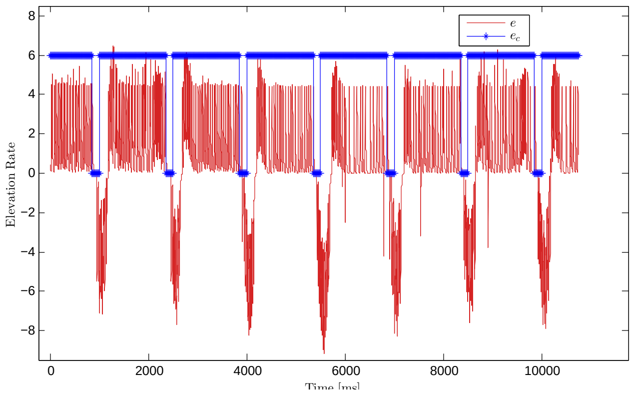

Figure 11: *Elevation rate.*

noise. The elevation rate would really benefit from some sort of clamping and low pass filter.

3.4 Part IV - State estimation

3.4.1 Problem 1

Mathematical Modelling

The system given in eq. (20) can be expressed as a state space representation with the vectors

$$\mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix}. \tag{52}$$

The system can thus be expressed on the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B} \tag{53}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x}, \tag{54}$$

with the matrices

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ K_3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ 0 & 0 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{55}$$

and

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{56}$$

3.4.2 Problem 2

Observability Analysis
From the system matrix in (55) and the output matrix in (56), the observability matrix can be found to be

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA^2} \\ \mathbf{CA^3} \\ \mathbf{CA^4} \\ \mathbf{CA^5} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.612 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.612 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{57}$$

The matrix has full column rank, which means the system is observable.

Linear Observer Design
We wish to control the system using an estimated state $\hat{\mathbf{x}}$. A *linear observer* (or closed-loop estimator) is then

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{L}(\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}), \tag{58}$$

where $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are as before and $\mathbf{L}$ is the *observer gain matrix.*
Replacing the output with $\mathbf{Cx}$ and rearranging yields

$$\dot{\hat{\mathbf{x}}} = (\mathbf{A} - \mathbf{LC})(\hat{\mathbf{x}} - \mathbf{x}) + \mathbf{Ax} + \mathbf{LC}(\mathbf{x} - \hat{\mathbf{x}}),$$

and when inserting the estimation error (denoted $\mathbf{d}$ as in *deviation* here, to not confuse it with the elevation e),

$$\mathbf{d} = \hat{\mathbf{x}} - \mathbf{x}, \tag{59}$$

along with its derivative, the observer's error dynamics, become

$$\dot{\mathbf{d}} = (\mathbf{A} - \mathbf{LC})\mathbf{d}. \tag{60}$$

Good state estimation is essential to properly control the helicopter. The error dynamics of the observer thus need to be faster than the dynamics of the feedback controller. This corresponds to the real part of the poles of the closed loop observer being more *negative* than the controller poles. Conveniently, by theorem 8.O3 in Chen, the poles of the closed loop observer, i.e. the eigenvalues of $\mathbf{A} - \mathbf{LC}$, can be placed arbitrarily with appropriate choice of observer gain $\mathbf{L}$.[2]

To place the poles like so, one can use the matlab `place` command. This command will output a feedback gain matrix that will place the poles of a closed loop system at the values given in a vector $\mathbf{p}$. This applies to using a feedback controller such as $\mathbf{u} = -\mathbf{Kx}$. In that case, the closed loop dynamics of the system, $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$, will be

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{BK})\mathbf{x}.$$

Using the matlab `place` command as such:

```
K = place(A, B, p);
```

will produce a feedback gain matrix $\mathbf{K}$, that places the eigenvalues of $\mathbf{A}-\mathbf{BK}$ at the desired values.[3]

For the closed loop observer in (58), the only modifiable matrix is $\mathbf{L}$. By the properties of the transpose,

$$(\mathbf{A} - \mathbf{LC})^{\mathrm{T}} = \mathbf{A}^{\mathrm{T}} - (\mathbf{LC})^{\mathrm{T}} = \mathbf{A}^{\mathrm{T}} - \mathbf{C}^{\mathrm{T}}\mathbf{L}^{\mathrm{T}},$$

such that

$$\mathbf{L}^{\mathrm{T}} = \mathtt{Place}(\mathbf{A}^{\mathrm{T}}, \mathbf{C}^{\mathrm{T}}, \mathbf{p}).$$

And so finally, by the property that any matrix has eigenvalues identical to its transpose, the required matrix can be calculated as

$$\mathbf{L} = \mathtt{place}(\mathbf{A}^{\mathrm{T}}, \mathbf{C}^{\mathrm{T}}, \mathbf{p})^{\mathrm{T}}, \tag{61}$$

where the desired eigenvalues are given in the vector $\mathbf{p}$.
Furthermore, by substituting the estimate $\hat{\mathbf{x}}$ for $\mathbf{x}$ in the feedback controller $\mathbf{u} = -\mathbf{Kx}$ from problem 3, the estimated system becomes

$$\dot{\hat{\mathbf{x}}} = (\mathbf{A} - \mathbf{LC} - \mathbf{BK})\hat{\mathbf{x}} - \mathbf{Ly}. \tag{62}$$

Thus, the dynamics of the closed loop system, with controller and estimator are

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{d}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} - \mathbf{BK} & \mathbf{BK} \\ \mathbf{0} & \mathbf{A} - \mathbf{LC} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{d} \end{bmatrix}. \tag{63}$$

Note how, since this matrix is triangular, its eigenvalues are simply those of the diagonal matrices. The stability of the entire system can hence be determined by verifying stability in the observer and the system separately,[4] a principle known as the *separation principle.*[2]

Observer Tuning

By the separation principle, tuning of the observer can be done independently of the controller. However, as mentioned previously, the poles of the observer should be placed much further to the left in the complex plane than the poles of the closed loop feedback controller. With the `place` command, the poles are placed in a fan as illustrated in figure 12. Having the poles

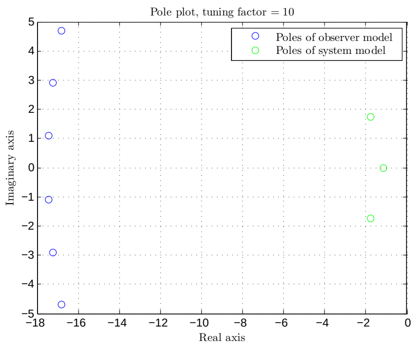

Figure 12: *Observer and system poles*

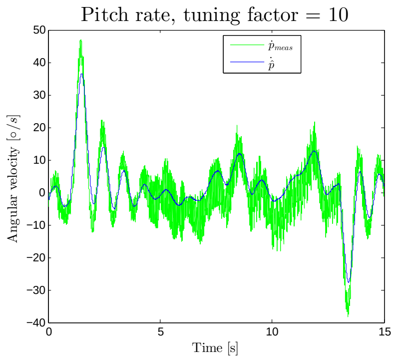

Figure 13: *Pitch rate with timid observer*

be quite negative as they are here, makes the observer fast and easily able to track the state closely, neither does it introduce any noticeable lag or instability into the system.

An example plot showing the estimated pitch rate, $\dot{\hat{p}}$ (in blue) and the measured state (in green) are shown in figure 13, which confirms these assumptions. The only considerable estimation error appears in the pitch rate, when the helicopter turns quickly. This is likely caused by the pitch rate being so far from zero, which is the value it was linearized around. Had the other two estimates, elevation rate and travel rate, displayed similar behaviour, estimation errors would probably appear there too. However, since the states are so much slower, this is not the case.

The poles in this case were placed around -17. This makes the observer fast enough for this application, but not fast enough to track the state perfectly. This slowness also has the side effect of low-pass filtering the response, which makes for clear looking plots.

Nevertheless, it can be useful to make the poles even more negative. Placing them at around -250 for instance, makes the response very noisy, but with very close tracking of the actual state, as can be shown in figure 14.

It is in fact easy to show that very negative eigenvalues of $\mathbf{L}$ scale up any unwanted measurement measurement noise heavily, just like in this case. Rewriting (58) with $\mathbf{y} = \mathbf{Cx} + \mathbf{w}$ as output, where the vector $\mathbf{w}$ represents

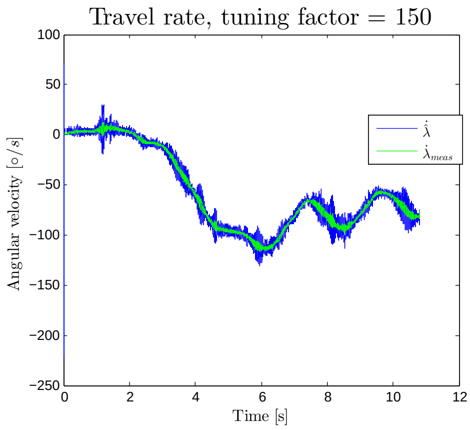

Figure 14: *Travel rate with very fast observer*

the measurement noise in the output, means (60) instead becomes

$$\dot{\mathbf{d}} = (\mathbf{A} - \mathbf{LC})\mathbf{d} - \mathbf{Lw}. \tag{64}$$

An $\mathbf{L}$ matrix with very negative eigenvalues will thus result in $|\mathbf{Lw}|$ becoming very large and introduce high amplitude noise into the estimation error dynamics.

A middle ground, with poles placed around -120, results in precise estimation of the measured value. It introduces a bit of noise, but no more than their measured counterparts do. Plots that show how well the estimated states, $\dot{\hat{p}}$, $\dot{\hat{e}}$ and $\dot{\hat{\lambda}}$ compare to the measured states, are shown in figures 15, 16 and 17, respectively. The pole placement is shown in figure 18

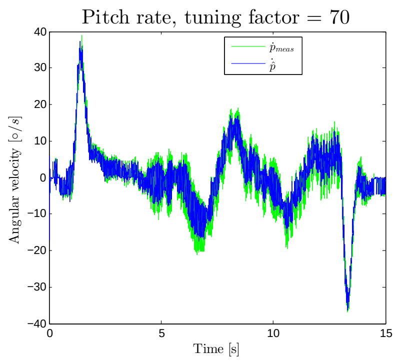

Figure 15: *Pitch rate with moderate estimator*

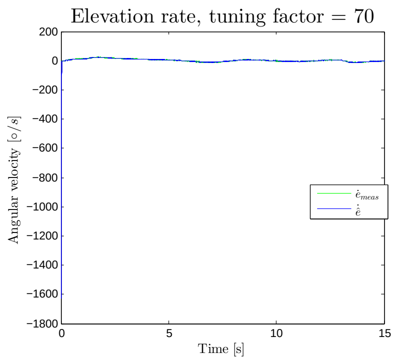

Figure 16: *Elevation rate with moderate estimator*

When the LQR controllers from Part III were tested using the estimated states, the controllers were as good as before, without any noticeable performance impact.

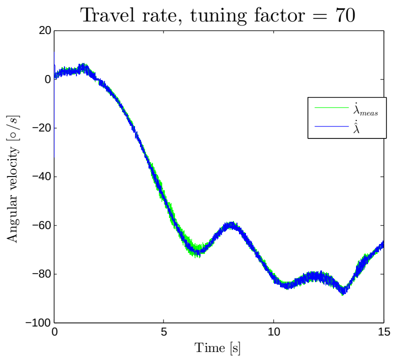

Figure 17: *Travel rate with moderate observer*

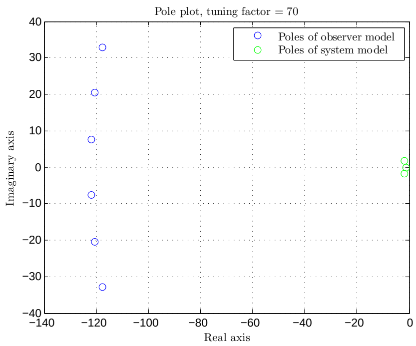

Figure 18: *Pole plot with moderate observer*

3.4.3 Problem 3

Observability Analysis
The matrices

$$\mathbf{C}_{e\lambda} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{C}_{pe} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{65}$$

correspond to the measurement vectors

$$\mathbf{y}_{e\lambda} = \begin{bmatrix} \tilde{e} \\ \tilde{\lambda} \end{bmatrix} \quad \text{and} \quad \mathbf{y}_{pe} = \begin{bmatrix} \tilde{p} \\ \tilde{e} \end{bmatrix} \tag{66}$$

respectively.
Since the observability matrix,

$$\mathcal{O}_{e\lambda} = \begin{bmatrix} \mathbf{C}_{e\lambda} \\ \mathbf{C}_{e\lambda}\mathbf{A} \\ \mathbf{C}_{e\lambda}\mathbf{A^2} \\ \mathbf{C}_{e\lambda}\mathbf{A^3} \\ \mathbf{C}_{e\lambda}\mathbf{A^4} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.612 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.612 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{67}$$

has full column rank, the system is observable when measuring only $\tilde{e}$ and $\tilde{\lambda}$. On the other hand, when measuring only $\tilde{p}$ and $\tilde{e}$, the system is *not* observable, as

$$\mathcal{O}_{pe} = \begin{bmatrix} \mathbf{C}_{pe} \\ \mathbf{C}_{pe}\mathbf{A} \\ \mathbf{C}_{pe}\mathbf{A^2} \\ \mathbf{C}_{pe}\mathbf{A^3} \\ \mathbf{C}_{pe}\mathbf{A^4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{68}$$

does *not* have full column rank.
This also makes intuitive sense, as we cannot expect to observe the travel, λ, from only knowing elevation e and the pitch p. But if we do know the travel, the pitch can be obtained by taking the derivative.
Tuning the observer based solely on the outputs e and λ proved to be much harder than when the pitch, p was also part of the measurements. The erroneous estimation is most clear in the pitch rate estimation as shown in fig. 19.[2] This is partly because the pitch has the non-linear correlation with elevation and travel given in equations (8) and (11), but the observer is based on the linearised equations given in (20). As a result, the estimation error becomes more prevalent the further away from the linearisation point the state sways. This error gets amplified further when taking the derivative, since a high frequency noise signal such as $\sin(\omega t)$, where ω is large, has the high amplitude derivative $\omega \cos(\omega t)$.

For this problem, the chosen controller was the LQR of part III with integral effect. Initially, it was clear that the original tuning of $\mathbf{Q}$ and $\mathbf{R}$ was not applicable for this modified closed-loop system, and new tuning was required. It eventually became clear that the system was so sensitive that the poles of **A**-**LC** ultimately had to be placed solely on the real axis in an attempt to reduce oscillations. After more testing and tuning, it was settled on a tuning factor of 40, which produced the responses of fig. 19. This resulted in the poles `[-23.9303 -24.0303 -24.1303 -24.2303 -24.3303 -24.4303]`. With this pole placement, the helicopter would converge towards equilibrium by itself and remain stable, though small input would cause it to diverge.

[2] The term “tuning factor“ comes from a factor introduced during tuning of the poles of $\mathbf{A} - \mathbf{LC}$. The factor indicates how much farther the poles of $\mathbf{A} - \mathbf{LC}$ are from the origin than the real part of the farthest pole of $\mathbf{A} - \mathbf{BK}$ into the left half-plane. Thus, a tuning factor of 3 means that the poles of $\mathbf{A} - \mathbf{LC}$ are three times the distance from the origin along the real axis than the farthest pole of $\mathbf{A} - \mathbf{BK}$

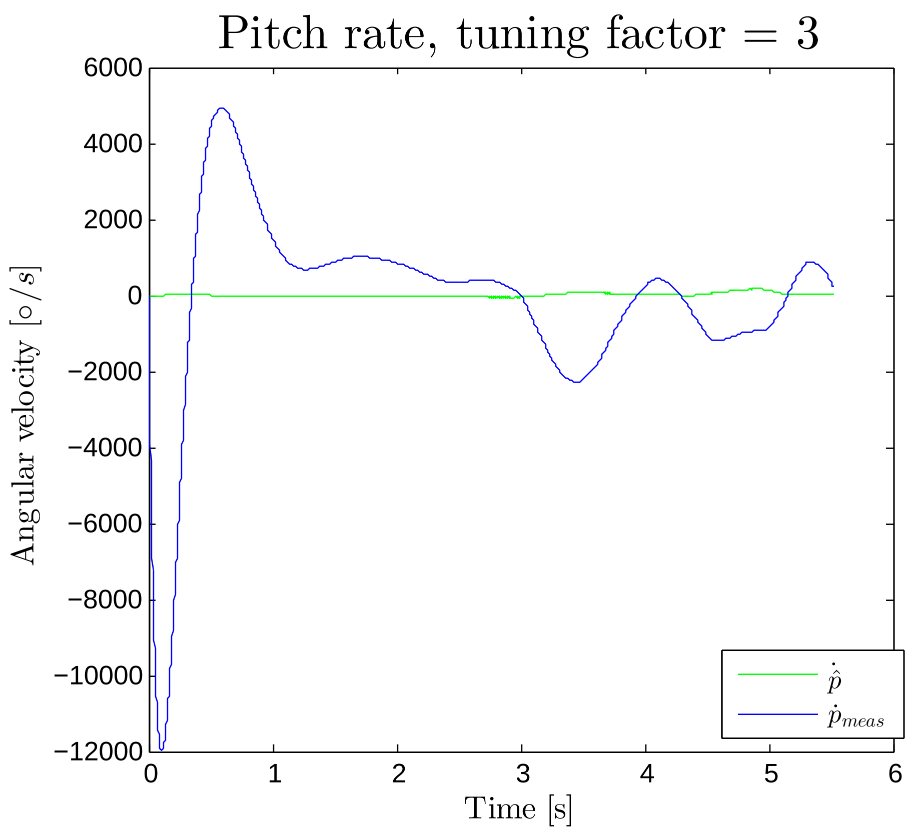

Figure 19: *Plot showing estimated pitch rate without using the measurement. The estimate deviates considerably from the measurement.*

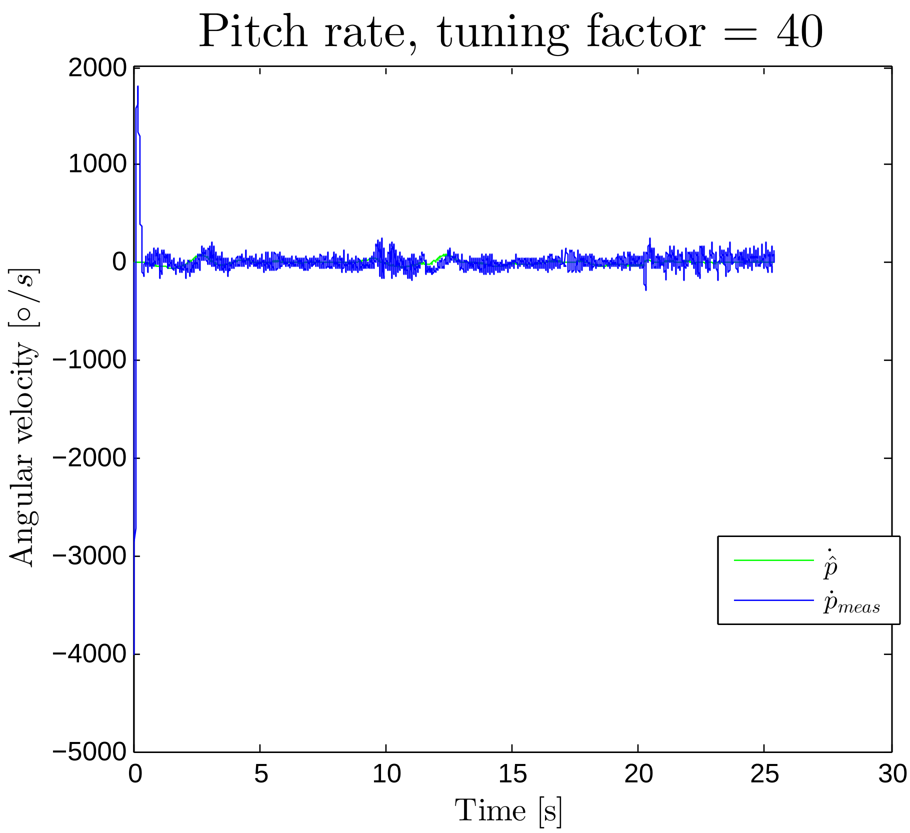

Figure 20: *Plot showing estimated pitch rate without using the measurement. The estimate approximates the measurement much better, disregarding the offset in the beginning.*

4 Conclusion

The report has analysed a system consisting of a helicopter with non-linear equations of motion. The analysis was done by first deriving a mathematical model for the system, for then to linearise it at a given equilibrium point. Then different controllers were applied to control the behaviour of the helicopter. First, a monovariable PD controller was tried to control pitch, for then to introduce an additional P controller that controlled travel.

A Simulink Diagrams

A.1 Simulink Diagrams for Part II

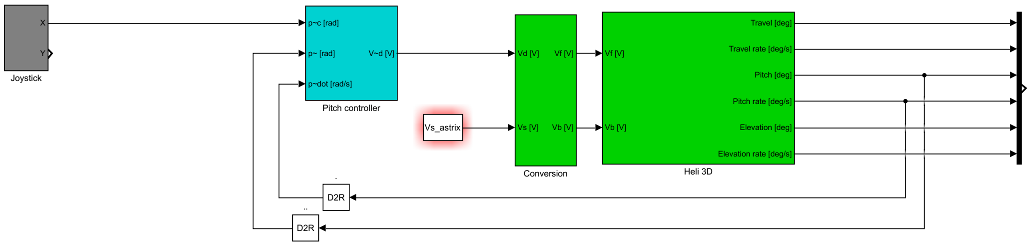

Figure 21: *Top view of model from Part 2 problem 1*

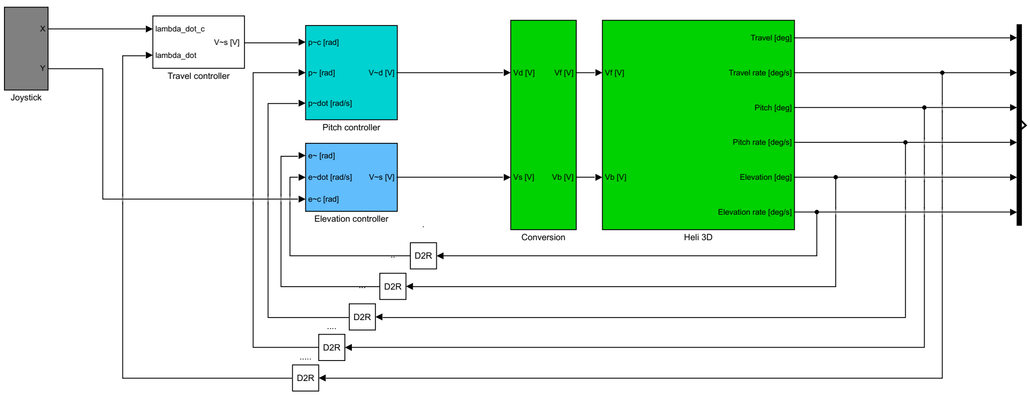

Figure 22: *Top view of model from Part 2 problem 2*

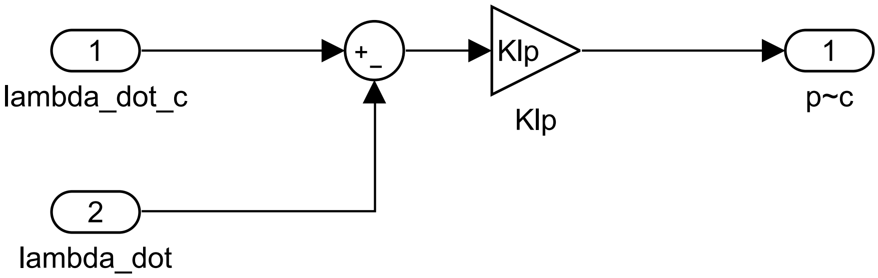

Figure 23: Inside the Travel controller

A.2 Simulink Diagrams for Part III

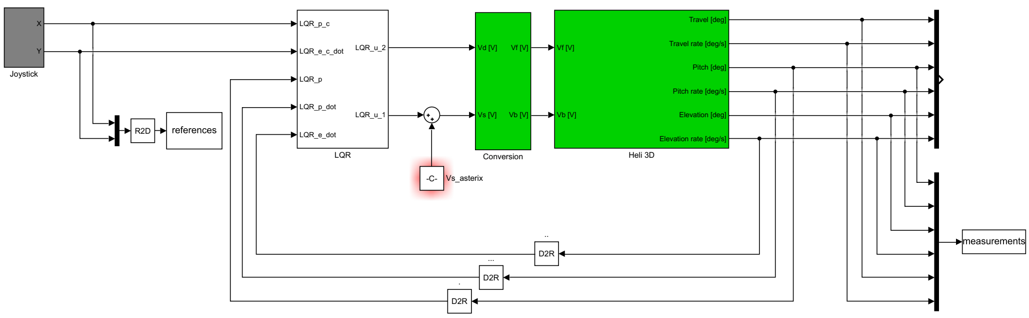

Figure 24: Top view of model from Part 3 problem 2

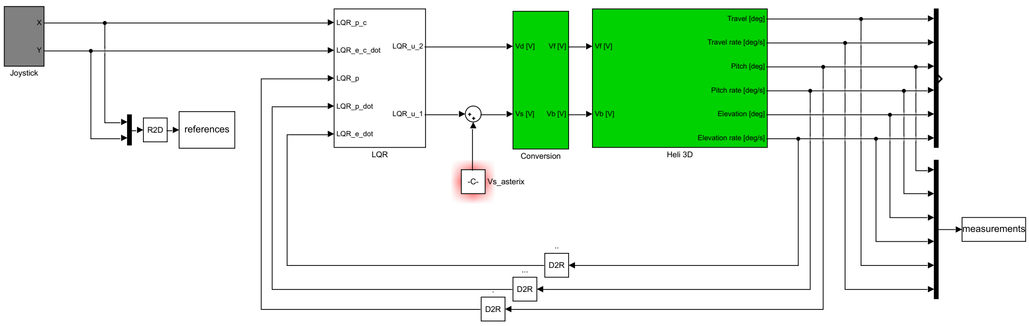

Figure 25: *Top view of model from Part 3 problem 3*

A.3 Simulink Diagrams for Part IIII

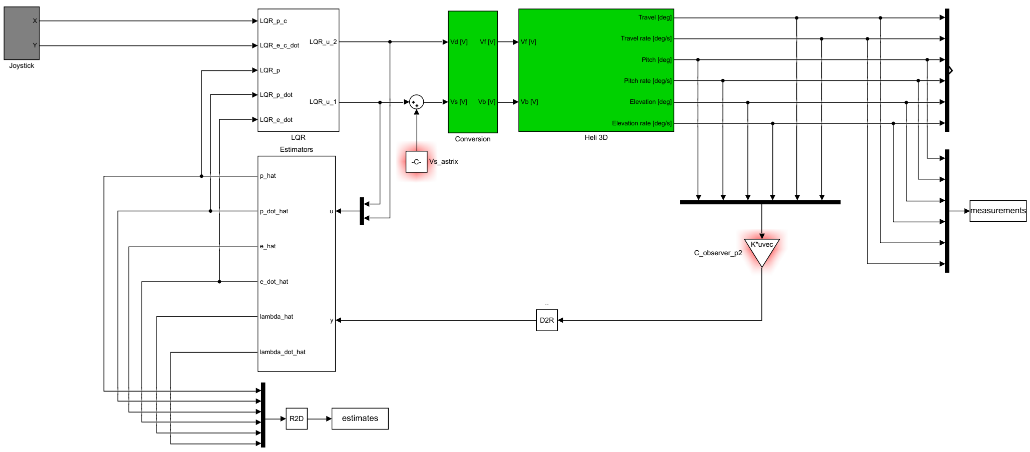

Figure 26: *Top view of model from Part 4 problem 2*

B Parameters and values

Table 1: *Parameters and values.*

Symbol	Parameter	Value	Unit
l_h	Distance from elevation axis to helicopter body	0.63	m
l_p	Distance from pitch axis to motor	0.18	m
K_f	Force constant motor	0.25	N/V
J_e	Moment of inertia for elevation	0.83	kg m^2
J_λ	Moment of inertia for travel	0.83	kg m^2
J_p	Moment of inertia for pitch	0.034	kg m^2
m_p	Mass of helicopter	1.05	kg
m_c	Weight of counterweight	1.87	kg

References

[1] I.c. Cheeseman and J.d.l. Gregory. “The Effect Of The Ground On The Performance Of A Helicopter.” In: *Performance* (1959).

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

[3] *Pole Placement*. `https://www.mathworks.com/help/control/getstart/pole-placement.html`. Accessed: 2018-10-19.

[4] *Proof that the eigenvalues of a block matrix are the combined eigenvalues of its blocks*. `https://math.stackexchange.com/questions/21454/prove-that-the-eigenvalues-of-a-block-matrix-are-the-combined-eigenvalues-of-its`. Accessed: 2018-10-19.