Helicopter Lab Report

Group 8

Halvor Ødegård Teigen, 478422
Simen Krantz Knudsen, 478444
Ole-Jørgen Hannestad, 478457

October 22, 2018

Contents

1 Introduction

The main goal of this lab exercise is to control a helicopter using linear system theory. This report describes the work done to complete our assigned work; controlling a helicopter to hover at a wanted position and use a joystick to set variable references. The work consists of mathematically modelling of the helicopter dynamics before controlling the helicopter using different techniques. Subjects from control theory consists of feed-forward control, monovariable and multivariable control, P, PI and PD controllers, determining controllability and observability, LQR, and state estimation.

Throughout the report, our findings and the mathematical derivations are presented. The reader is assumed to have a basic understanding of calculus and control theory, as we skip some of the simplest steps in the derivations.

2 Part 1 - Mathematical modeling

A fundamental part of control theory is to make a mathematical model of the process that is being controlled. The following part consists of mathematical derivations to achieve this model. Figure 1 shows a simplified graphical model of the helicopter where each propeller and the counterweight is modelled as a point mass, and the arms are considered to be massless.

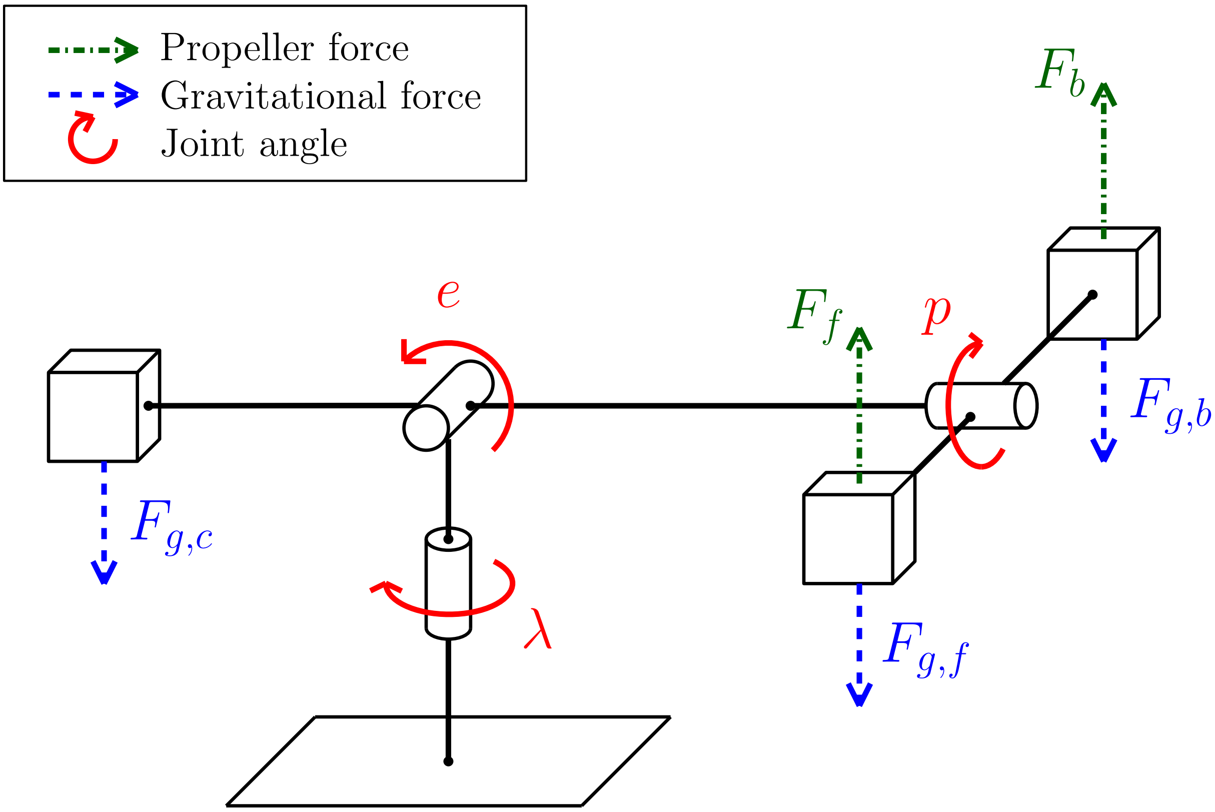

Figure 1: Model of the helicopter used in the lab. Shows forces and joint angles.

Problem 5.1.1

To compute the equations of motion, torque equilibrium with respect to each axis is used.

Torque equilibrium with respect to the pitch angle:

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

This leads to the following equation for the moment of inertia about the pitch axis:

$$J_p\ddot{p} = L_1 V_d \tag{1a}$$

where L_1 is defined in eq. (2a) and $V_d = V_f - V_b$.

Torque equilibrium with respect to the elevation angle:

$$J_e\ddot{e} = \underbrace{m_c g}_{F_{g,c}} \cdot l_c \cos e - (\underbrace{m_p g}_{F_{g,b}} + \underbrace{m_p g}_{F_{g,f}}) \cdot l_h \cos e + (\underbrace{K_f V_f}_{F_f} + \underbrace{K_f V_b}_{F_b}) \cdot l_h \cos p,$$

which leads to the following equation for the moment of inertia about the elevation axis:

$$J_e\ddot{e} = L_2 \cos e + L_3 V_s \cos p, \tag{1b}$$

where L_2 and L_3 are defined in eq. (2b) and eq. (2c), and $V_s = V_f + V_b$.

In the case of the travel angle, only the propeller force has an impact. Torque equilibrium with respect to the travel angle:

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

This leads to the following equation for the moment of inertia about the travel axis:

$$J_\lambda\ddot{\lambda} = L_4 V_s \cos e \sin p \tag{1c}$$

where L_4 is defined in eq. (2d).

$$L_1 = l_p K_f \tag{2a}$$

$$L_2 = m_c l_c g - 2 l_h m_p g \tag{2b}$$

$$L_3 = l_h K_f \tag{2c}$$

$$L_4 = -l_h K_f \tag{2d}$$

Problem 5.1.2

To find V_s^* and V_d^* we set $\ddot{p} = \ddot{e} = \ddot{\lambda} = 0$. Because we linearize around the point $(p, e, \lambda)^T = (p^*, e^*, \lambda^*)^T$, with $p^* = e^* = \lambda^* = 0$, we also set $p = e = \lambda = 0$. Inserting this into eq. (1a) and eq. (1b) and solving gives the following expressions for these two voltages:

$$L_1 V_d^* = 0 \quad \Longrightarrow \quad V_d^* = 0 \tag{3a}$$

$$L_3 V_s^* = -L_2 \quad \Longrightarrow \quad V_s^* = -\frac{L_2}{L_3} \tag{3b}$$

Applying eqs. (1a) and (1c), the non-linearized equation can be written as

$$\begin{bmatrix} \ddot{p} \\ \ddot{e} \\ \ddot{\lambda} \end{bmatrix} = \begin{bmatrix} \dfrac{L_1 V_d}{J_p} \\ \dfrac{L_2 \cos e + L_3 V_s \cos p}{J_e} \\ \dfrac{L_4 V_s \cos e \sin p}{J_\lambda} \end{bmatrix} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{4}$$

where the moments of inertia are defined as

$$J_p = 2m_p l_p^2 \qquad J_e = m_c l_c^2 + 2m_p l_h^2 \qquad J_\lambda = m_c l_c^2 + 2m_p(l_h^2 + l_p^2)$$

Given

$$\begin{bmatrix} V_s \\ V_d \end{bmatrix} = \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} + \begin{bmatrix} V_s^* \\ V_d^* \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} p \\ e \\ \lambda \end{bmatrix} = \begin{bmatrix} \tilde{p} + p^* \\ \tilde{e} + e^* \\ \tilde{\lambda} + \lambda^* \end{bmatrix}, \quad \text{where } p^* = e^* = \lambda^* = 0,$$

and

$$\begin{aligned} \begin{bmatrix} \tilde{p} & \tilde{e} & \tilde{\lambda} \end{bmatrix}^T &= \begin{bmatrix} p - p^* & e - e^* & \lambda - \lambda^* \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\ \begin{bmatrix} p^* & e^* & \lambda^* \end{bmatrix}^T &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\ \begin{bmatrix} \tilde{V}_s & \tilde{V}_d \end{bmatrix}^T &= \begin{bmatrix} 0 & 0 \end{bmatrix}^T \end{aligned}$$

Linearizing eq. (4) to find the matrices $\mathbf{A}$ and $\mathbf{B}$ for the linearized model $\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\tilde{\mathbf{u}}$ gives

$$\mathbf{A} = \frac{\partial}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{u})\Big|_{\mathbf{x}^*, \mathbf{u}^*} = \begin{bmatrix} 0 & 0 & 0 \\ -\frac{L_3(\tilde{V}_s + V_s^*)\sin p^*}{J_e} & -\frac{L_2 \sin e^*}{J_e} & 0 \\ \frac{L_4(\tilde{V}_s + V_s^*)\cos e^* \cos p^*}{J_\lambda} & -\frac{L_4(\tilde{V}_s + V_s^*)\sin e^* \sin p^*}{J_\lambda} & 0 \end{bmatrix} \tag{5}$$

$$\mathbf{B} = \frac{\partial}{\partial \mathbf{u}} \mathbf{f}(\mathbf{x}, \mathbf{u})\Big|_{\mathbf{x}^*, \mathbf{u}^*} = \begin{bmatrix} 0 & \dfrac{L_1}{J_p} \\ \dfrac{L_3 \cos p^*}{J_e} & 0 \\ \dfrac{L_4 \cos e^* \sin p^*}{J_\lambda} & 0 \end{bmatrix} \tag{6}$$

We set V_s^* as given by eq. (3b), $\tilde{V}_s = 0$ and $p^* = e^* = \lambda^* = 0$ and get

$$\begin{bmatrix} \ddot{\tilde{p}} \\ \ddot{\tilde{e}} \\ \ddot{\tilde{\lambda}} \end{bmatrix} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\tilde{\mathbf{u}} = \begin{bmatrix} 0 \\ 0 \\ -\dfrac{L_2 L_4}{J_\lambda L_3}\tilde{p} \end{bmatrix} + \begin{bmatrix} \dfrac{L_1}{J_p}\tilde{V}_d \\ \dfrac{L_3}{J_e}\tilde{V}_s \\ 0 \end{bmatrix} = \begin{bmatrix} \dfrac{L_1}{J_p}\tilde{V}_d \\ \dfrac{L_3}{J_e}\tilde{V}_s \\ -\dfrac{L_2 L_4}{J_\lambda L_3}\tilde{p} \end{bmatrix} = \begin{bmatrix} K_1 \tilde{V}_d \\ K_2 \tilde{V}_s \\ K_3 \tilde{p} \end{bmatrix} \tag{7}$$

where

$$K_1 = \frac{L_1}{J_p} \quad \text{(8a)} \qquad K_2 = \frac{L_3}{J_e} \quad \text{(8b)} \qquad K_3 = -\frac{L_4 L_2}{J_\lambda L_3} \quad \text{(8c)}$$

Considerations In addition to the linearization, some forces are excluded from the model to make it less complex. These forces include drag, centripetal acceleration, joint friction and the forces applied to the system as a consequence of the arm's weight. Also, the motors are assumed to have a linear relationship between force and voltage, which is just an approximation. With that said, the advantage of the linear model is it's simplicity and the fact that it is easy to control. As long as the angles in pitch and elevation are small, the model is sufficiently accurate for our purposes.

Problem 5.1.3

Looking at the linearized model in eq. (7) we see that acceleration in pitch angle should be proportional to the joystick input $\tilde{V_d}$. This reflects what we observed in the lab and makes sense as it is also given by the non-linear equation eq. (4).

According to eq. (7), the elevation acceleration is proportional to $\tilde{V_s}$. This is only true for small pitch angles, as large pitch angle will lead to less lift from the propellers and less acceleration in elevation. In reality, elevation is dependant on both $\tilde{V_s}$ and the pitch angle.

In the lab, we also observe that the acceleration in travel seems proportional to the pitch angle. This has a good correspondence with eq. (7), but according to eq. (4) this is not entirely true. The travel also depends on elevation. Large elevation angles will lead to a smaller lever arm to rotate around the travel axis, thus we get less acceleration in travel.

Problem 5.1.4

The joystick is used as feed-forward to keep the helicopter at a horizontal position. From fig. 2 we consider it reasonable to set $V_s^* = 6.8$ V. In our initial calculations given below, V_s^* was rounded up to 7V, but later in the lab we changed it to 6.8V. This was because 7V turned out to be slightly too high. Ideally K_f should also have been changed, but due to the fact that it would result in a change of less than 3%, we did not see it as necessary.

Using eq. (3b) we get the following:

$$V_s^* = -\frac{L_2}{L_3} = \frac{2l_h m_p g - m_c g l_c}{K_f l_h}$$

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

$$V_s^* = 7\text{V} \quad \Longrightarrow \quad K_f \approx 0,143$$

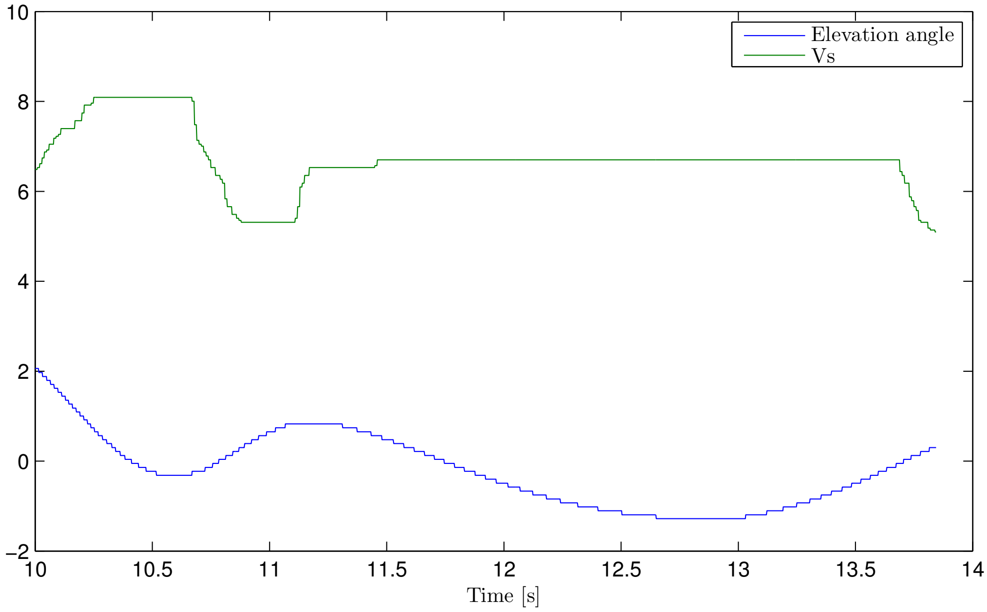

Figure 2: Plot of a manual steering towards 0° in elevation angle with the joystick. The blue line is elevation angle in degrees, and the green line is the corresponding voltage V_s.

3 Part 2 - Monovariable control

Problem 5.2.1

This part of the lab focuses on control of the pitch angle. The following PD controller for the pitch is given in the assignment:

$$\tilde{V_d} = K_{pp}(\tilde{p_c} - \tilde{p}) - K_{pd}\dot{\tilde{p}} \tag{9}$$

Using the expression for $\ddot{\tilde{p}}$ from eq. (7) in addition to eq. (9) yields the following expression:

$$\ddot{\tilde{p}} = K_1(K_{pp}(\tilde{p_c} - \tilde{p}) - K_{pd}\dot{\tilde{p}}) \tag{10}$$

To ease readability two new constants, K_p and K_d, are defined:

$$K_p = K_1 K_{pp} \tag{11a}$$

$$K_d = K_1 K_{pd} \tag{11b}$$

The Laplace transform is applied to eq. (10) to find the desired transfer function $\dfrac{\tilde{p}}{\tilde{p}_c}(s)$.

$$\begin{aligned}\mathcal{L}\{\ddot{\tilde{p}}(t)\}(s) &\implies s^2\tilde{p} = K_p(\tilde{p_c} - \tilde{p}) - K_d s\tilde{p} \\ &\implies \frac{\tilde{p}}{\tilde{p}_c}(s) = \frac{K_p}{s^2 + K_d s + K_p}\end{aligned} \tag{12}$$

Next, the transfer function in eq. (12) is compared with the general form of a second order ordinary differential equation.

$$h(s) = \frac{K\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{13}$$

Comparing the numerators in eq. (12) and eq. (13), the constant K has to be equal to 1. $\zeta = 1$ is inserted to achieve critical damping. This is desirable because a critical damped system has the fastest possible response without any oscillation. The expressions for the unknown constants are:

$$K_p = \omega_0^2 \qquad\qquad K_d = 2\sqrt{K_p}$$

Inserting the values from table 1 in (8a) gives $K_1 = 0,5688$. Putting these expressions back in eq. (11a) and eq. (11b) gives the following equations:

$$K_{pp} = \frac{\omega_0^2}{K_1} \tag{14a}$$

$$K_{pd} = \frac{2\omega_0}{K_1} \tag{14b}$$

Figure 19 shows the implementation of the PD controller for the pitch in Simulink, and fig. 20 shows the complete implementation in Simulink. To find reasonable numerical values for K_{pp}

and K_{pd}, several attempts were conducted. In the first attempt ω_0 was set equal to 4. The corresponding numerical values are K_{pp} = 28,13 and K_{pd} = 14,06. With these values for K_{pp} and K_{pd}, we observed a fast and stable control of the pitch.

Next, the value of K_{pp} was set to 30 and K_{pd} was set to 3, without choosing a corresponding ω_0. This was done in order to see how the damping part of the regulator affected this closed-loop system in particular. The pitch behaviour in this case was practically unstable.

In the last attempt, ω_0 was set to 3. Using this value in eq. (14a) and eq. (14b) give K_{pp} = 15,82 and K_{pd} = 10,55. These values led to robust and fast control of the pitch and a time constant of about 1 second, fig. 3 shows the step response with these values. The stationary deviation in this plot is due to the lack of integral effect in the controller.

Since the stability margins are unknown when using this trial and error tuning method, we chose to use the lowest value of ω_0 (= 3) that gave a stable and fast response. The reason being that a high gain may lead to small margins and a less robust system with regards to disturbances. Several assumptions taken in the assignment, in addition to the linearization, leads to an invalid mathematical model when the helicopter is far away from the equilibrium point. Thus, we chose $K_{pp} = 15,82$ and $K_{pd} = 10,55$ as the parameter values.

The eigenvalues that give the best response for a system, may vary depending on the individual system even though the mathematical model is identical. We acknowledge the fact that the trial and error tuning method used in the section above, is far from optimal. With this method both the gain and phase margins are unknown, thus the stability and robustness of the system is difficult to determine.

A more methodical approach to tuning is e.g. finding the closed-loop system's transfer function and using Bode plots to determine the controller parameters. In addition, the phase and gain margins may easily be determined this way. We chose not to use this method due to the fact that Bode plots only give us theoretically good margins. This may not be the best values for our helicopter due to individual properties in addition to the simplifications made in modelling.

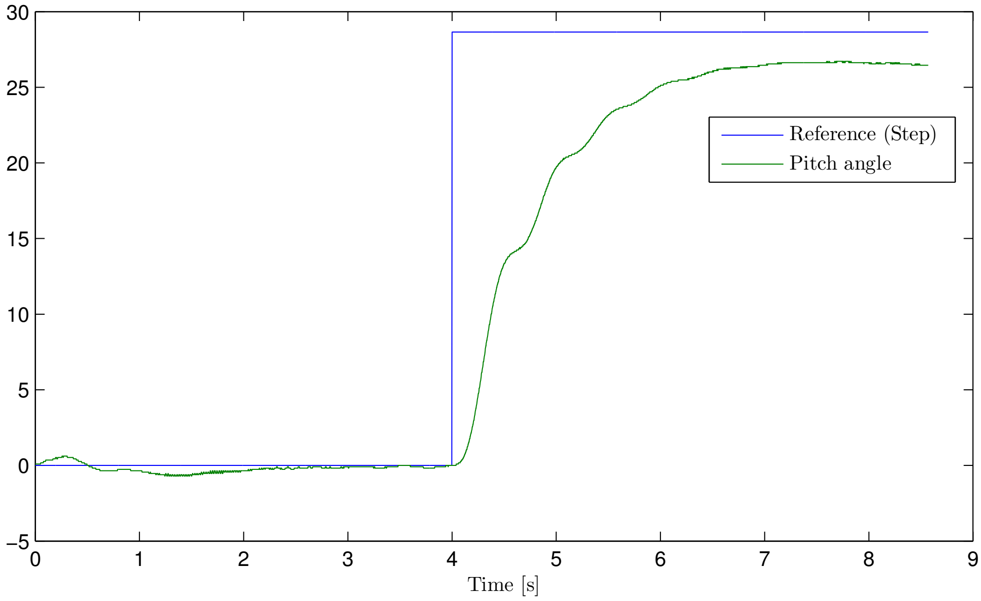

Figure 3: The pitch angle response, controlled with a PD controller, when a step function was set as the pitch reference. The y-axis is given in degrees, and the controller parameters equal K_{pp} = 15,82 and K_{pd} = 10,55.

Problem 5.2.2

The next task was to add a P contoller to control the travel rate. Assuming that the pitch is controlled perfectly, i.e. $\tilde{p} = \tilde{p}_c$, and using the following P controller:

$$\tilde{p_c} = K_{rp}(\dot{\tilde{\lambda}}_c - \dot{\tilde{\lambda}}), \qquad \ddot{\tilde{\lambda}} = K_3\tilde{p} \qquad \text{and} \qquad \tilde{p_c} = \tilde{p}$$

we get the following expression:

$$\ddot{\tilde{\lambda}} = K_3K_{rp}(\dot{\tilde{\lambda}}_c - \dot{\tilde{\lambda}}) \tag{15}$$

Laplace-transforming eq. (15) gives the transfer function:

$$\frac{\dot{\tilde{\lambda}}}{\dot{\tilde{\lambda}}_c}(s) = \frac{K_3K_{rp}}{s + K_3K_{rp}} = \frac{\rho}{s+\rho} \tag{16}$$

After using eq. (8c) and inserting values from table 1, we get $K_3 = -0,6117$. It follows that $\rho = -0,6117K_{rp}$. Tuning the controller now depends on the value K_{rp}. We know that a simple P controller utilizes the reference deviation multiplied with a gain to reach the desired reference value. To get a fast controller, it is reasonable to increase the gain, but the controller should also be stable and not overshoot. Figure 21 shows the implementation of the P controller for the travel rate, and fig. 22 shows the complete implementation of the system in Simulink. A scaling of the joystick output was added to limit the joystick sensitivity, respectively a gain of 0,7.

We tried to set $\rho = 2$, corresponding to $K_{rp} = -3,27$. The corresponding response is shown fig. 4. The behaviour has practically no stationary deviation but much oscillation, reaching almost 18° when the pitch reference is 12°. The latter indicating that the value for K_{rp} is a bit too high.

Next, we set ρ equal to 1, which gave a $K_{rp} = -1,635$. With this value for the controller parameter the travel rate reacted fast with no overshoot, but had a stationary deviation. See fig. 5 for the step response. We chose $\rho = 1$. The reason for this choice is that we prioritize to not have any overshoot. We prioritize this because an overshoot may lead to the helicopter moving further away from the equilibrium point.

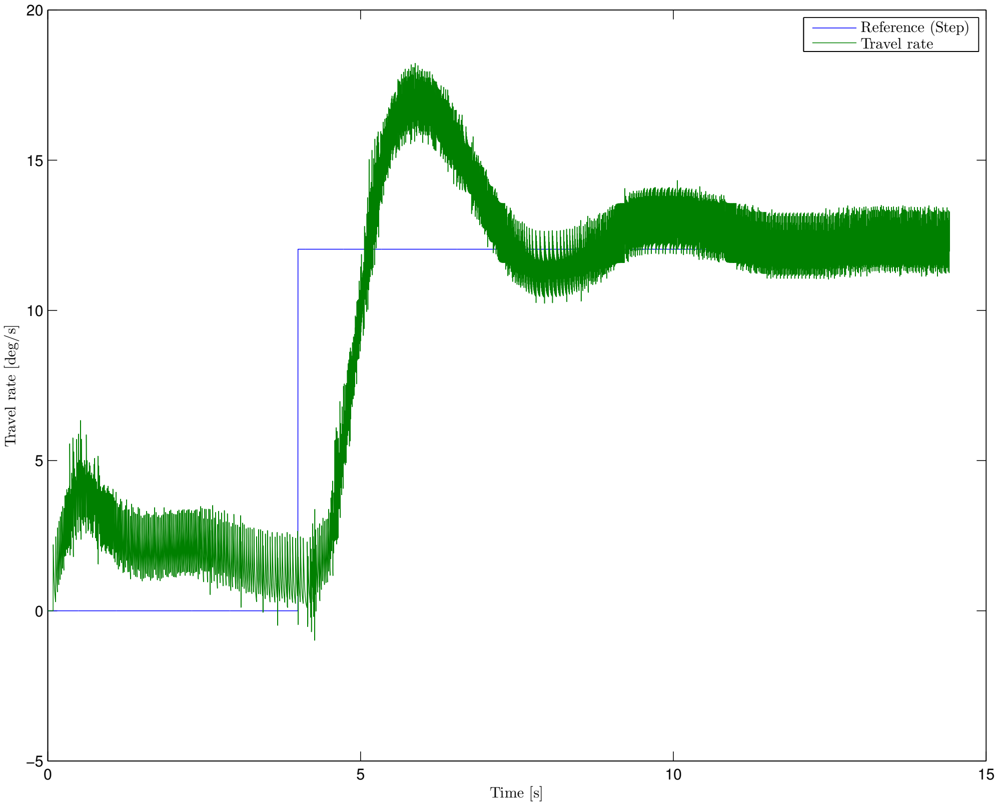

Figure 4: The step response to the travel rate with a P controller, with K_{rp} = -3,27.

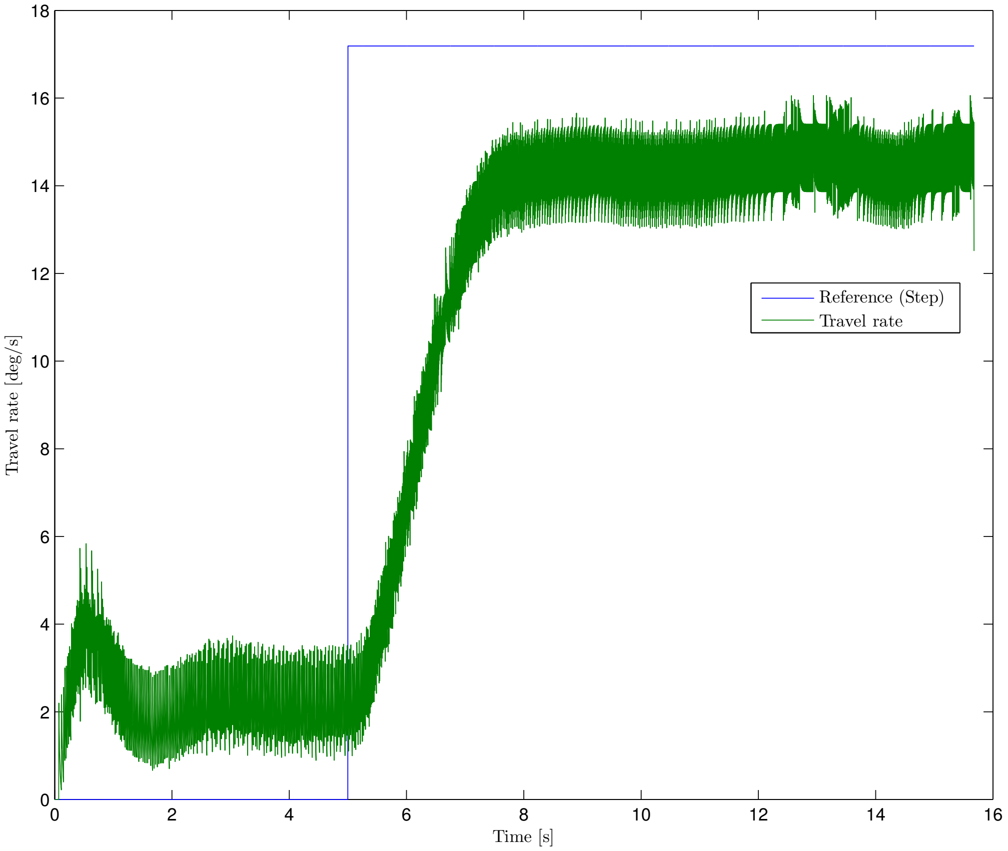

Figure 5: The step response to the travel rate with a P controller, with K_{rp} = -1,635.

4 Part 3 - Multivariable control

Problem 5.3.1

The general state-space formulation is given by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{17a}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \tag{17b}$$

Given the relationship between the system's pitch, pitch rate and elevation rate we can derive a state space model on the form given in eq. (17a)–(17b). This will be our first step towards making a multivariable controller for the helicopter. From the assignment, it is specified that:

$$\mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \qquad \ddot{\tilde{p}} = K_1 \tilde{V}_d \qquad \ddot{\tilde{e}} = K_2 \tilde{V}_s$$

From this we can rewrite the relationship as a system of equations

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = K_1 \tilde{V}_d$$

$$\dot{x}_3 = K_2 \tilde{V}_s$$

Given this system, the matrices in eq. (17a) will be the following:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \end{bmatrix} \tag{18}$$

Problem 5.3.2

The next task is to control the helicopter with a controller on the form

$$\mathbf{u} = \mathbf{Pr} - \mathbf{Kx} \tag{19}$$

where $\mathbf{r} = \begin{bmatrix}\tilde{p}_c & \dot{\tilde{e}}_c\end{bmatrix}^T$. The reference for the pitch angle and elevation rate comes from the joystick output. However, in the lab it seems like the reference from the joystick output y is elevation angle. In practice the helicopter goes to a constant angle and not a constant rate if we move the joystick and keep it in a fixed position. We have gone through the calculations and the implementation in Simulink, but can not find the reason for this error.

The first step towards implementing this controller is to verify that the system is controllable. This was done by computing the controllability matrix $\mathcal{C}$.

We found $\mathcal{C}$ by manual derivation as well as using MATLAB's function `ctrb(A,B)`. Both methods gave the same result.

$$\mathcal{C} = \begin{bmatrix}\mathbf{B} & \mathbf{AB} & \mathbf{A^2B}\end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & K_1 & 0 & 0 \\ 0 & K_1 & 0 & 0 & 0 & 0 \\ K_2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{20}$$

The systems controllability matrix has a rank of 3, which corresponds to the system having full rank. Thus the system is controllable.

The matrix $\mathbf{K}$ is the result of the Linear Quadratic Regulator (LQR) where $\mathbf{u} = -\mathbf{Kx}$ maximizes the following cost function

$$J = \int_0^\infty \left(\mathbf{x}^T(t)\mathbf{Qx}(t) + \mathbf{u}^T(t)\mathbf{Ru}(t)\right) dt \tag{21}$$

To calculate the $\mathbf{K}$ we utilized the MATLAB command `K=lqr(A,B,Q,R)`. As arguments to this command we used the matrices $\mathbf{A}$ and $\mathbf{B}$ from eq. (18) and matrices $\mathbf{Q}$ and $\mathbf{R}$. The matrices $\mathbf{Q}$ and $\mathbf{R}$ are diagonal matrices where the diagonal values tune the regulator. $\mathbf{Q}$ gives the relationship between the states and how much it costs to have a deviation from the reference value in the respective state. A higher value in $\mathbf{Q}_{ii}$ forces the regulator to keep the corresponding deviation low. The matrix $\mathbf{R}$ corresponds to how much a control output costs. A higher value in $\mathbf{R}_{ii}$ corresponds to the controller being more conservative using control output.

To find $\mathbf{Q}_{ii}$ and $\mathbf{R}_{ii}$ we had to test different values, analyze the effect on the system and synthesize what the optimal value should be. We started with the identity matrix and saw the differences in response for variations of values. For instance, a $\mathbf{Q}_{33} = 0.1$ gave an oscillatory response in elevation, while $\mathbf{Q}_{33} = 100$ gave a extremely slow response in elevation. This "trial and error" method resulted in the following matrix $\mathbf{Q}$ which gave a fast and accurate response, but with a noticeable stationary deviation, see fig. 6. This deviation is due to the absence of an integral effect in the controller. We chose to prioritize to avoid overshoot and have a slightly slower response, rather than a faster but oscillatory response.

$$\mathbf{Q} = \begin{bmatrix} 50 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} 0.1 & 0 \\ 0 & 1 \end{bmatrix}$$

The next step towards controlling the system is to calculate the P matrix. We insert eq. (19) in for $\mathbf{u}$ in $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ and assume that $lim_{t\to\infty}\dot{\tilde{e}}(t) = \dot{\tilde{e}}_c$ and $lim_{t\to\infty}\tilde{p}(t) = \tilde{p}_c$. By this, the states goes to zero and the output goes to the reference value as time goes to infinity. From this we get the following expression for $\mathbf{P}$.

$$\begin{aligned} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x}_\infty + \mathbf{B}\mathbf{u} = \mathbf{0}, &\qquad \mathbf{u} = \mathbf{P}\mathbf{r_0} - \mathbf{K}\mathbf{x}_\infty \quad \Longrightarrow \\ -\mathbf{B}\mathbf{P}\mathbf{r_0} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}_\infty, &\qquad \mathbf{y} = \mathbf{C}\mathbf{x} \quad \Longrightarrow \\ \mathbf{y}_\infty = \mathbf{C}(\mathbf{B}\mathbf{K} - \mathbf{A})^{-\mathbf{1}}\mathbf{B}\mathbf{P}\mathbf{r_0}, &\qquad \mathbf{P} = (\mathbf{C}(\mathbf{B}\mathbf{K} - \mathbf{A})^{-\mathbf{1}}\mathbf{B})^{-\mathbf{1}} \end{aligned}$$

A possible consequence of assuming a fixed reference might be that the model is insufficient if we continuously change the reference by moving the joystick.

As a result of the calculations above we get the following $\mathbf{P}$ matrix:

$$\mathbf{P} = \begin{bmatrix} 0 & 4.472 \\ 7.071 & 0 \end{bmatrix}$$

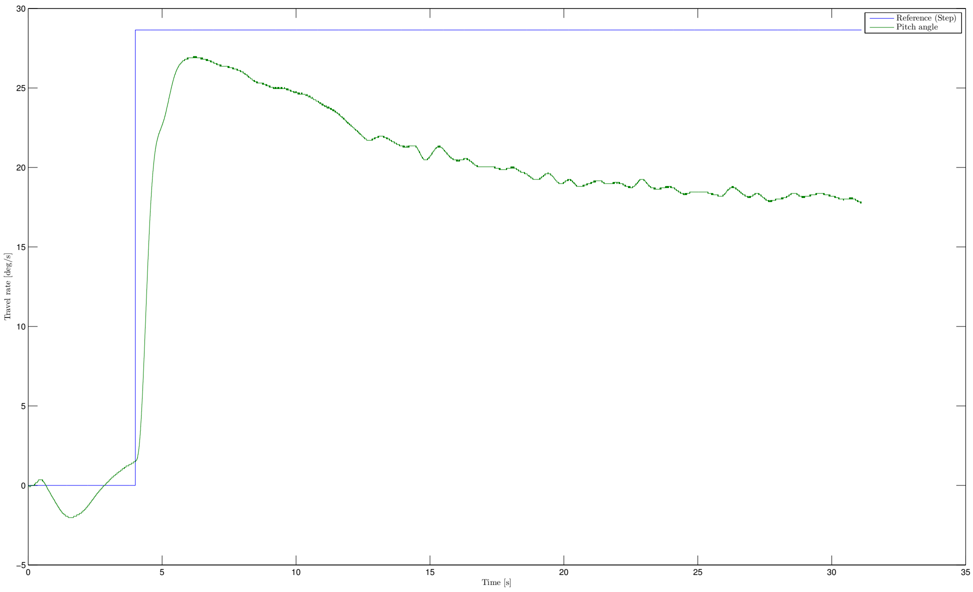

Figure 6: Step response on pitch angle, with a P controller.

Problem 5.3.3

To further improve our chance of making a fast and accurate controller for the helicopter we want to modify the former controller by adding integral effect for elevation rate and pitch angle. This can be done by adding two new states, ζ and γ, where their differential equations are:

$$\dot{\gamma} = \tilde{p} - \tilde{p}_c \tag{22a}$$

$$\dot{\zeta} = \dot{\tilde{e}} - \dot{\tilde{e}}_c \tag{22b}$$

For controlling the system, we continued with a controller on the form given in eq. (19). Expanding the matrices **A**, **B** and **C** to fit the addition of our new states gives us:

$$\mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \\ \gamma \\ \zeta \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & K1 \\ K2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

As a result of adding the states specified in eq. (22a) and eq. (22b) for integral effect, we have to rescale the matrix **Q**. As before, we used the trial and error method to determine the values $\mathbf{Q}_{ii}$ and $\mathbf{R}_{ii}$. This resulted in the **Q** and **R** matrix shown in eq. (23).

The pitch response, with a step function on the pitch reference $\tilde{p}_c$, is plotted in fig. 7. Comparing this plot with the one in fig. 6, we see that the stationary deviation is far less in fig. 7. In theory, this stationary deviation should be equal to zero. In practice it's less than 1°. To summarize, the step response is improved a lot with integral effect in the controller.

$$\mathbf{Q} = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 50 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{23}$$

When adding integral effect to the controller, the **P** matrix becomes singular and non-invertible. Hence, we set the values of the **P** matrix to:

$$\mathbf{P} = \begin{bmatrix} 0 & 10 \\ 5 & 0 \end{bmatrix}$$

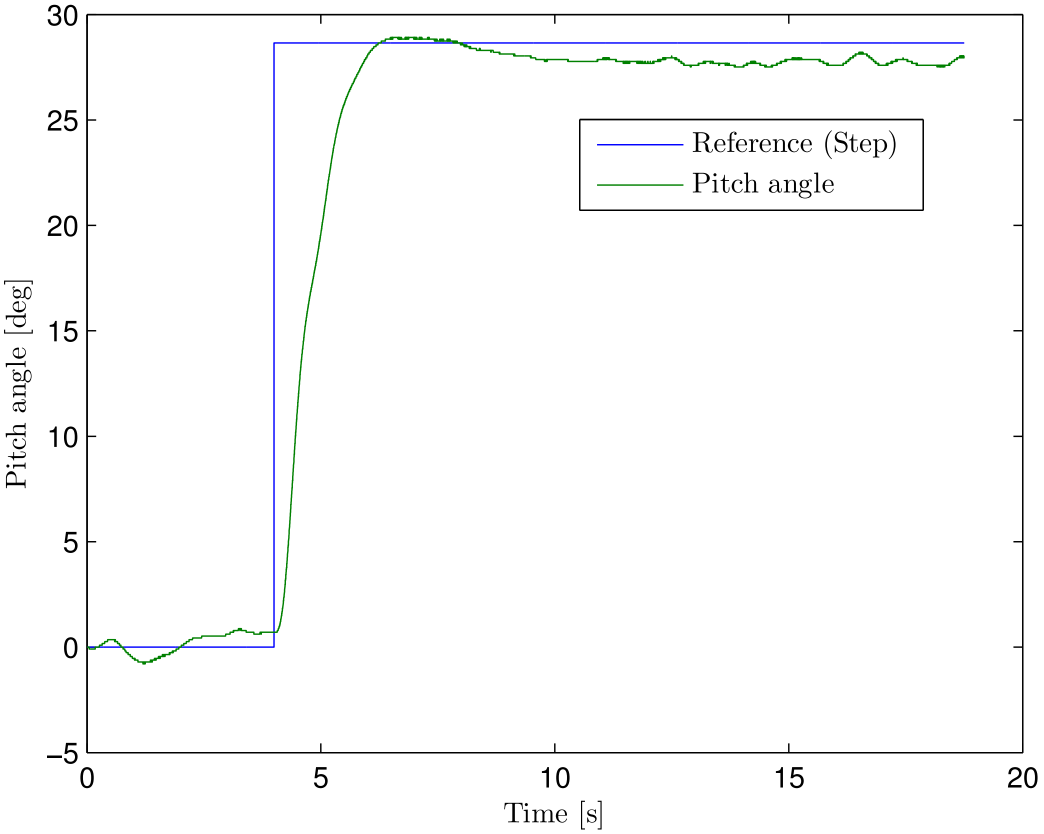

Figure 7: Step response on the pitch angle, with a PI controller.

5 Part 4 - State estimation

Problem 5.4.1

In this part of the assignment, state estimation will be used to estimate the angular velocities $\dot{\tilde{p}}$, $\dot{\tilde{e}}$ and $\dot{\tilde{\lambda}}$ instead of computing them with numerical differentiation. The first task is to derive a state-space model of the system. The following vectors are defined in the assignment:

$$\mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}, \qquad \mathbf{u} = \begin{bmatrix} \tilde{V_s} \\ \tilde{V_d} \end{bmatrix} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix}$$

Deriving a state-space model from these vectors and eq. (7) produces the following matrices:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ K_3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ 0 & 0 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{24}$$

when the system is on the general form given in eq. (17a) and (17b).

Problem 5.4.2

Next, the task is to examine the observability of the state-space model in eq. (24), and then create a linear observer. The general equation for the observability matrix is given by

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA^2} \\ \vdots \\ \mathbf{CA^{n-1}} \end{bmatrix} \tag{25}$$

Using the matrices **A** and **C** from eq. (24) and checking the rank of the resulting $\mathcal{O}$ using the MATLAB commands `rank(obsv(A, C))`, the matrix is found to have full rank, thus the system is observable. This implies that all the initial states can be reconstructed from the measurements **y**.

A linear observer for the system is given by eq. (26). In general, this can be used when the full state is not available for state feedback, e.g. when it's not possible to measure all the states. The observer $\hat{\mathbf{x}}$ is defined as $\hat{\mathbf{x}} = \mathbf{x} - \mathbf{e}$, where $\mathbf{e}$ is the error. The implementation of the linear observer in Simulink is shown in fig. 23.

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{L}(\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}) \tag{26}$$

We used the MATLAB code given below to obtain the observer gain matrix L, which in turn places the poles in the observer $\mathbf{A} - \mathbf{LC}$ where we want.

```
L = real(transpose(place(transpose(A), transpose(C)))
```

The plots in figures 9 – 14 compare the measurements of the six states with their respective observer, using the P controller. The spike in our observer at the start of the plots of p, $\dot{p}$, e and $\dot{e}$ might be a consequence of the observer needing a couple of measurements to determine the values. The observer manages to estimate both the pitch angle and travel angle well, see fig. 9 and fig. 13. The elevation angle is also estimated well, although the plot shows a stationary deviation of approximately 2°, see fig. 11. This stationary deviation could be the result of a wrong value of the elevation constant added to elevation angle, see fig. 20 for the implementation of this constant.

When it comes to the derivatives, the estimated elevation rate in fig. 12 is estimated well. The pitch rate shown in fig. 10 is estimated well most of the time, the noise in the measured state $\dot{p}$ is most likely due to the numerical differentiation. At last, fig. 14 shows the estimated and measured travel rate. This response indicates that the poles of the observer could be placed in a different way if travel rate was a more critical state, but since the five other states are estimated well and the travel rate is not used in the controller, this was not a priority.

The most noticeable difference between the behaviour of the system using the observer in contrast to the measurements with a PI controller, is that the pitch reacts with a more oscillatory response when it's subject to small disturbances. We tried to reduce $\mathbf{Q}_{44}$ in eq. (23) from 100 to 10, which gave a less oscillatory pitch response. The way we interpreted the assignment, the values in the LQR stays the same as in the previous task and we then compare the behaviour. Thus the value of $\mathbf{Q}_{44}$ remained 100.

A rule of thumb is to have an observer that is 2 – 20 times faster than the system itself, meaning that the eigenvalues of the matrix $\mathbf{A} - \mathbf{LC}$ should have an absolute value 2 – 20 times bigger than the gratest eigenvalue to $\mathbf{A} - \mathbf{BK}$. Placing the poles to the matrix $\mathbf{A} - \mathbf{LC}$ as in fig. 8 with a small r, leads to the observer being too slow. If the absolute value of the poles equals 4, the system becomes oscillatory, see fig. 15. If r is set to 20 and the poles are placed as in fig. 8, the system is stable and the observer is fast enough, see fig. 16. We also tried to increase the radius even more, this led to a sensitivity to noise in the observer. Thus, we chose to set the radius equal 20.

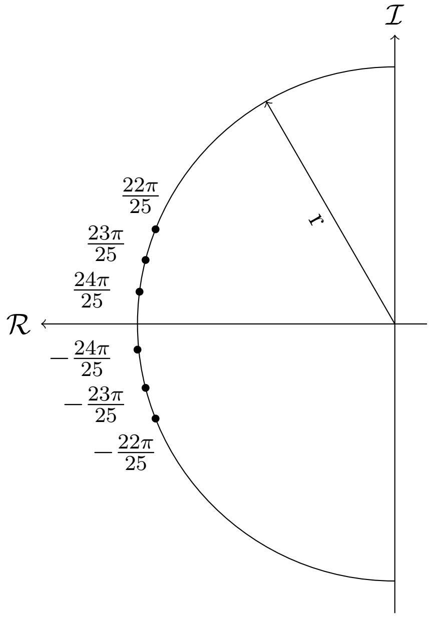

Figure 8: Pole placement of the matrix $\mathbf{A} - \mathbf{LC}$.

Problem 5.4.3

In this task, the goal is to check the observability of the system derived in eq. (24) using two different measurement vectors, and then create and test a linear observer.
If only the elevation $\tilde{e}$ and the travel $\tilde{\lambda}$ is measured, the matrix $\mathbf{C}$ in eq. (24) changes to the following matrix:

$$\mathbf{y} = \begin{bmatrix} \tilde{e} \\ \tilde{\lambda} \end{bmatrix}, \quad \mathbf{C_1} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Inserting $\mathbf{C_1}$ and the matrix $\mathbf{A}$ from eq. (24) into eq. (25) gives an observability matrix with full rank. Thus the system is observable when the measurement matrix equals $\mathbf{C_1}$.
On the other hand, if only the pitch $\tilde{p}$ and the elevation $\tilde{e}$ is measured, the measurement matrix changes to

$$\mathbf{y} = \begin{bmatrix} \tilde{p} \\ \tilde{e} \end{bmatrix}, \quad \mathbf{C_2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Inserting this and the matrix $\mathbf{A}$ into eq. (25), the observability matrix gets a rank of 4. This is not full rank, thus the system is not observable when the measurement matrix equals $\mathbf{C_2}$.

After having created a linear observer based on the measurement vector $\mathbf{y} = \begin{bmatrix} \tilde{e} & \tilde{\lambda} \end{bmatrix}^T$, we placed the poles in the fan given by fig. 8 and tried different radii. Appendix B.1 shows the MATLAB code to place the poles as shown in fig. 8, and fig. 23 shows the implementation of the linear observer used in this task. The plots in fig. 17 and fig. 18 show the measured states e, p and λ together with their respective observer. We chose to omit the plots of the derivatives of the states in the report because of noise in the plots.

In fig. 17, the travel angle $\tilde{\lambda}$ is well estimated. The estimated elevation angle $\hat{e}$ lags noticeably behind the corresponding measured state, probably due to a slow observer, but the estimated value tracks the measurement well. In fig. 18, both $\tilde{\lambda}$ and $\tilde{e}$ are well estimated, with only a spike at the very start of the plot. In both plots, the pitch angle is estimated poorly. The reason for this is that we don't measure $\tilde{p}$ directly, but instead have to estimate it from $\tilde{\lambda}$. As we have to differentiate the $\tilde{\lambda}$ two times to estimate $\tilde{p}$, the estimate for pitch becomes highly sensitive to the noise from $\tilde{\lambda}$. Thus we get a poor estimate for $\tilde{p}$. Several different placements of the poles were tested, e.g. all on the real axis, different radii with the same distribution of the poles as in fig. 8 and placing two and two complex conjugated pairs, where the pairs were placed at the same angle but at different radii. Nevertheless, the pitch remained unstable.

Even though a system is observable in theory, it may be difficult to estimate all the states in practice. Our system seems to be close to unobservable. The reason may be the estimation of a nonlinear system using the linear observer, the simplifications made or the mentioned noise in the measurements. To conclude, tuning an observer by placing the poles of $\mathbf{A} - \mathbf{LC}$ is far from easy in many cases.

A Plots of estimated and measured states

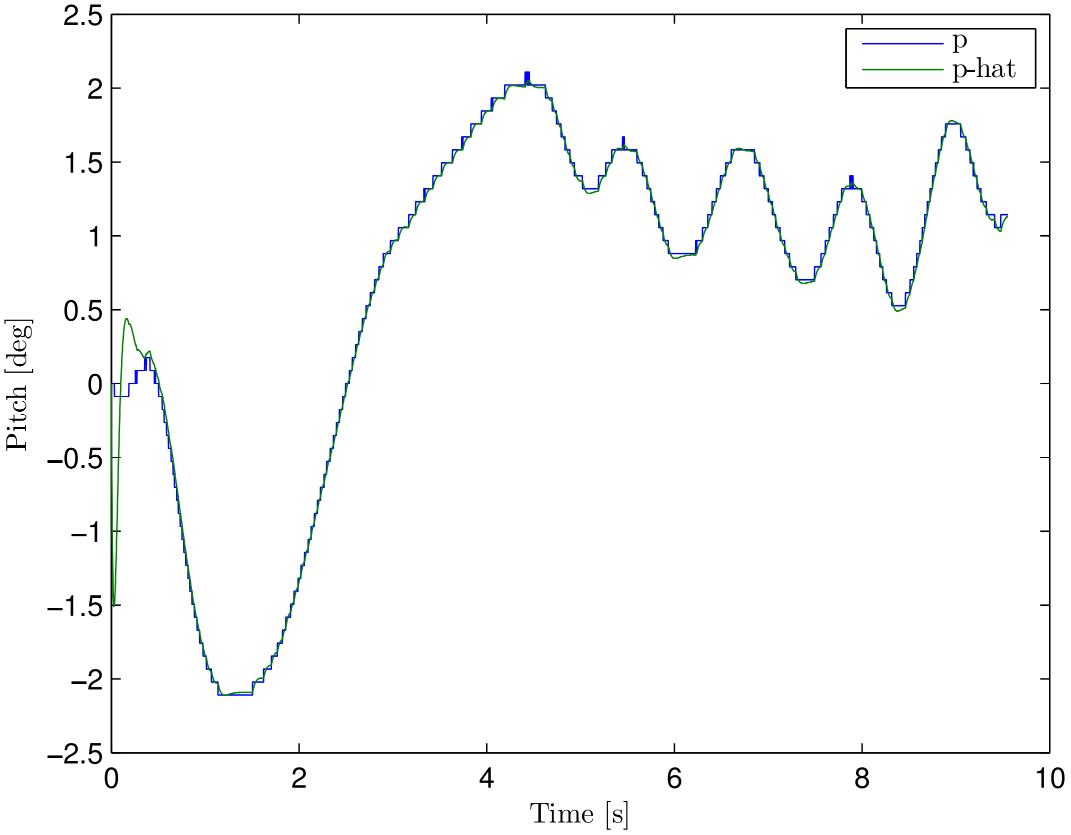

Figure 9: Plot of the measured $\tilde{p}$ against the estimated state $\hat{p}$.

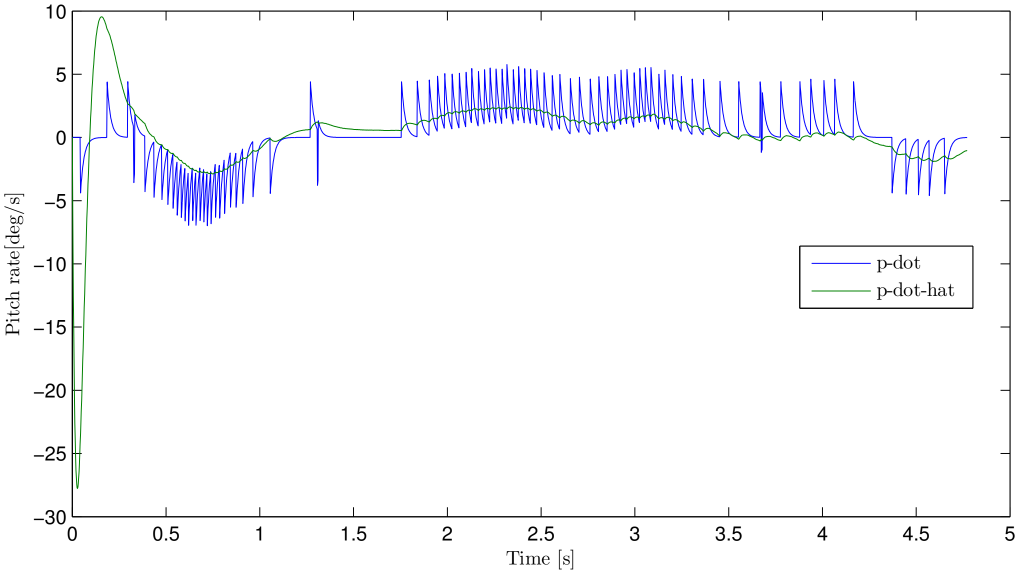

Figure 10: Plot of the measured state $\dot{\tilde{p}}$ against the estimated state $\dot{\hat{p}}$.

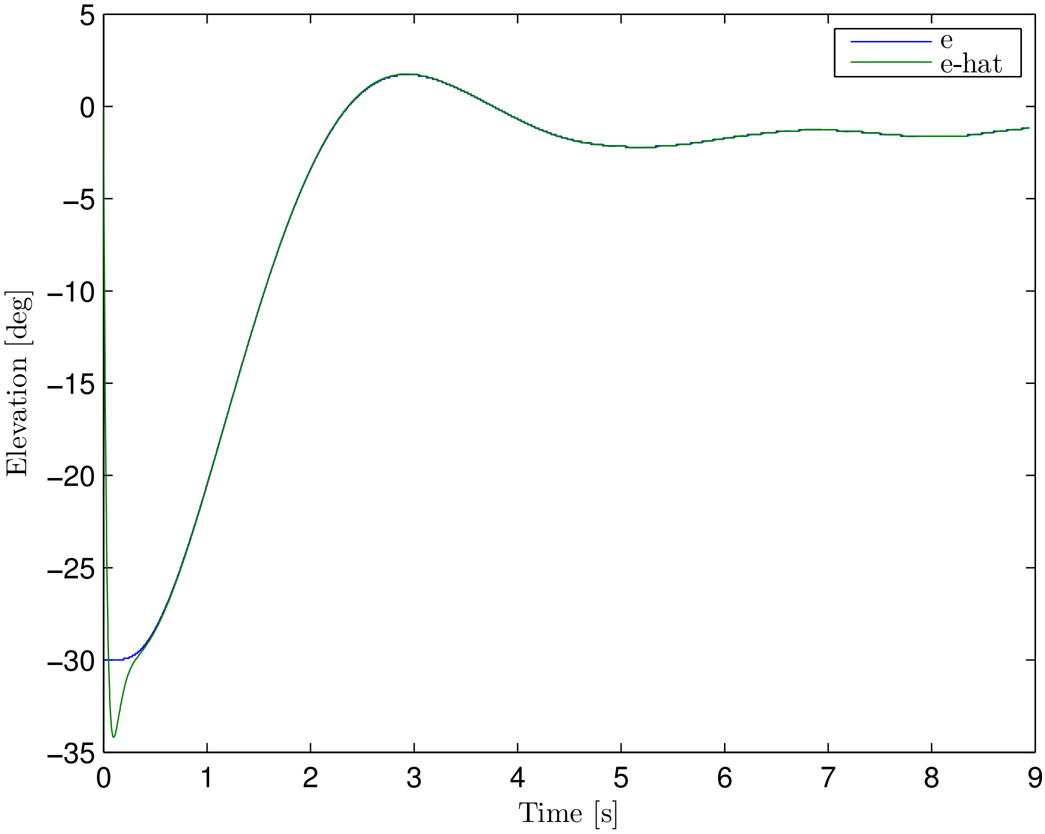

Figure 11: Plot of the measured state $\tilde{e}$ against the estimated state $\hat{e}$.

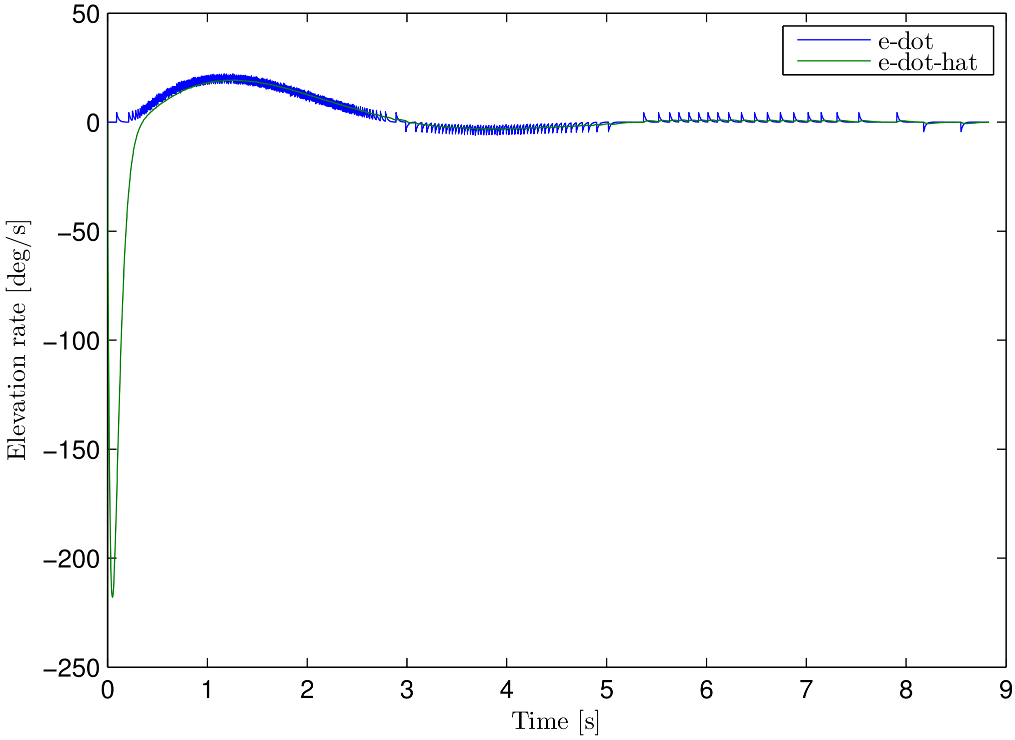

Figure 12: Plot of the measured state $\dot{e}$ against the estimated state $\dot{\hat{e}}$.

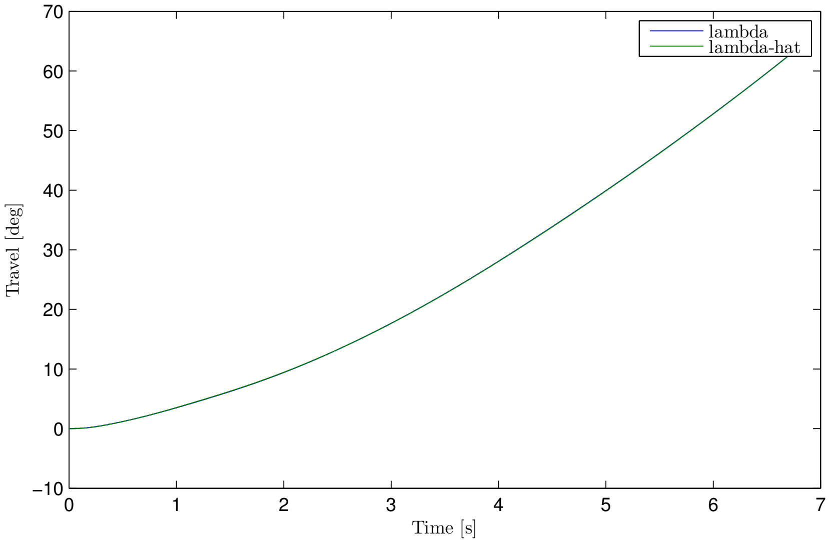

Figure 13: Plot of the measured state $\tilde{\lambda}$ against the estimated state $\hat{\lambda}$.

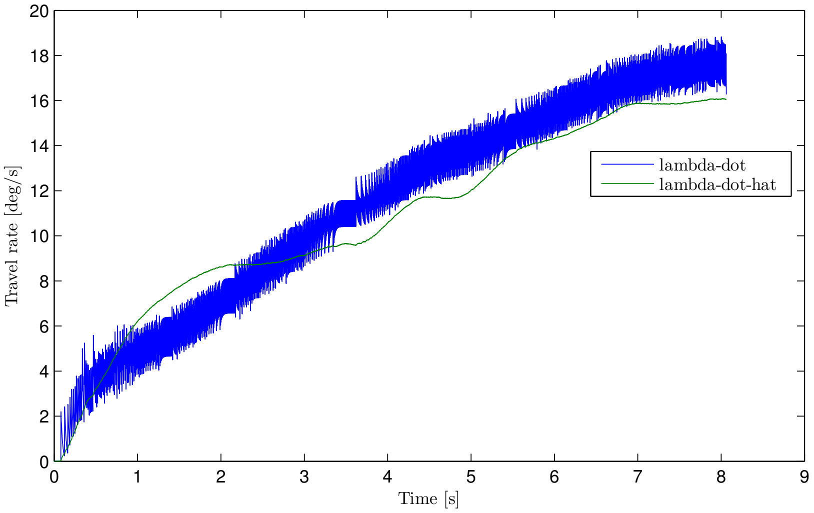

Figure 14: Plot of the measured state $\dot{\tilde{\lambda}}$ against the estimated state $\dot{\hat{\lambda}}$.

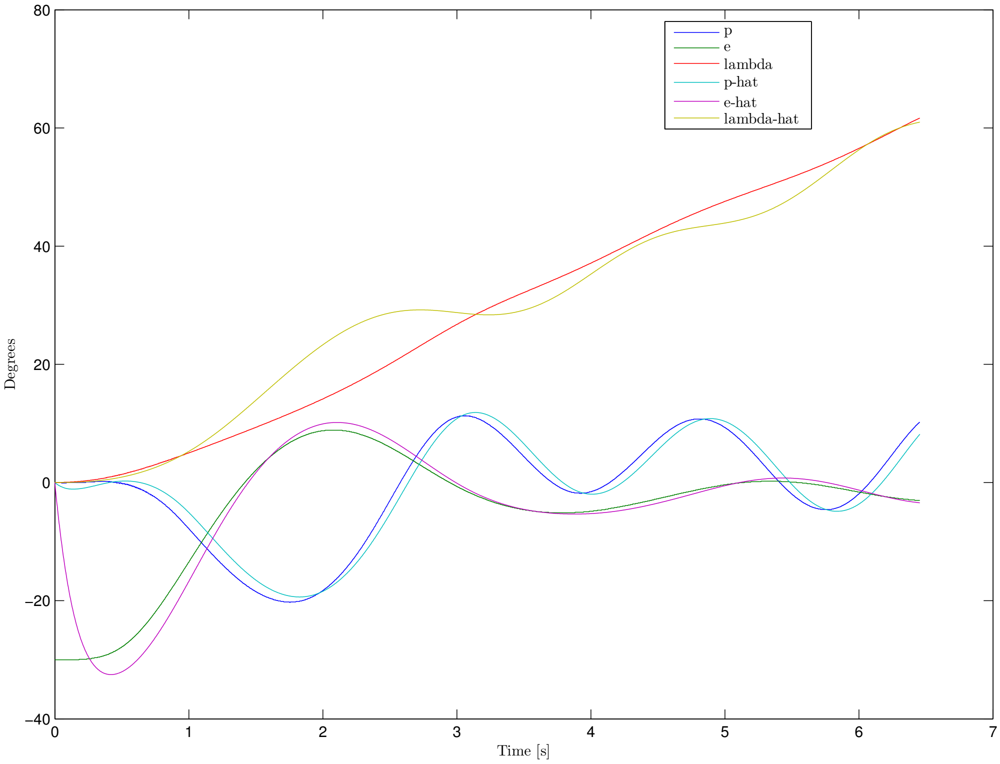

Figure 15: Plot of the measured and estimated states, with the absolute value of the poles equal 4 and measurements of $\tilde{p}$, $\tilde{e}$ and $\tilde{\lambda}$.

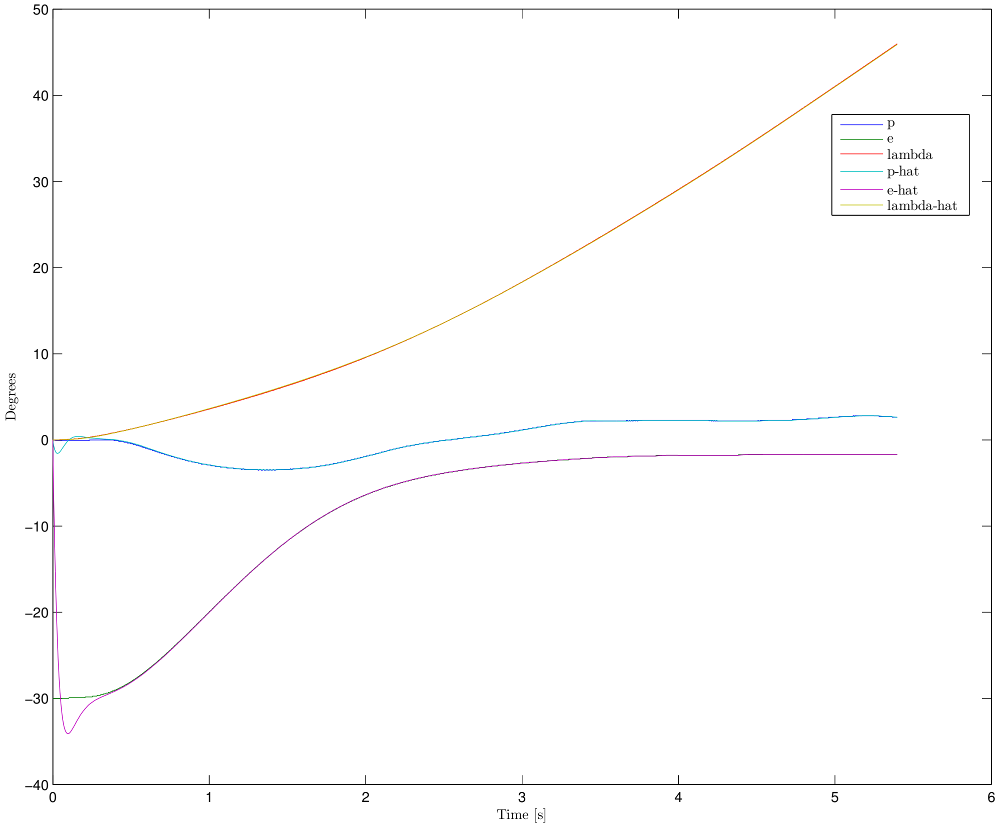

Figure 16: Plot of the measured and estimated states, with the absolute value of the poles equal 20 and measurements of $\tilde{p}$, $\tilde{e}$ and $\tilde{\lambda}$.

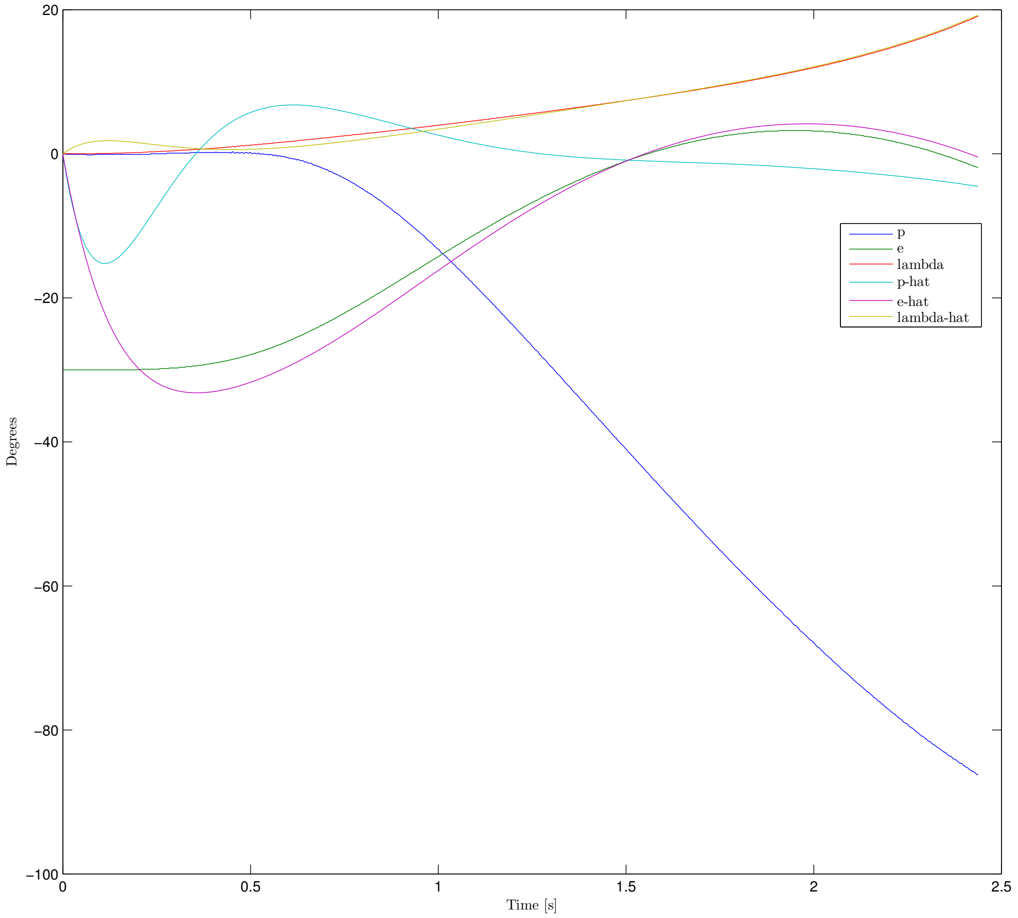

Figure 17: Plot of three estimated and measured states, when measuring $\tilde{e}$ and $\tilde{\lambda}$. The absolute value of the poles equals 5.

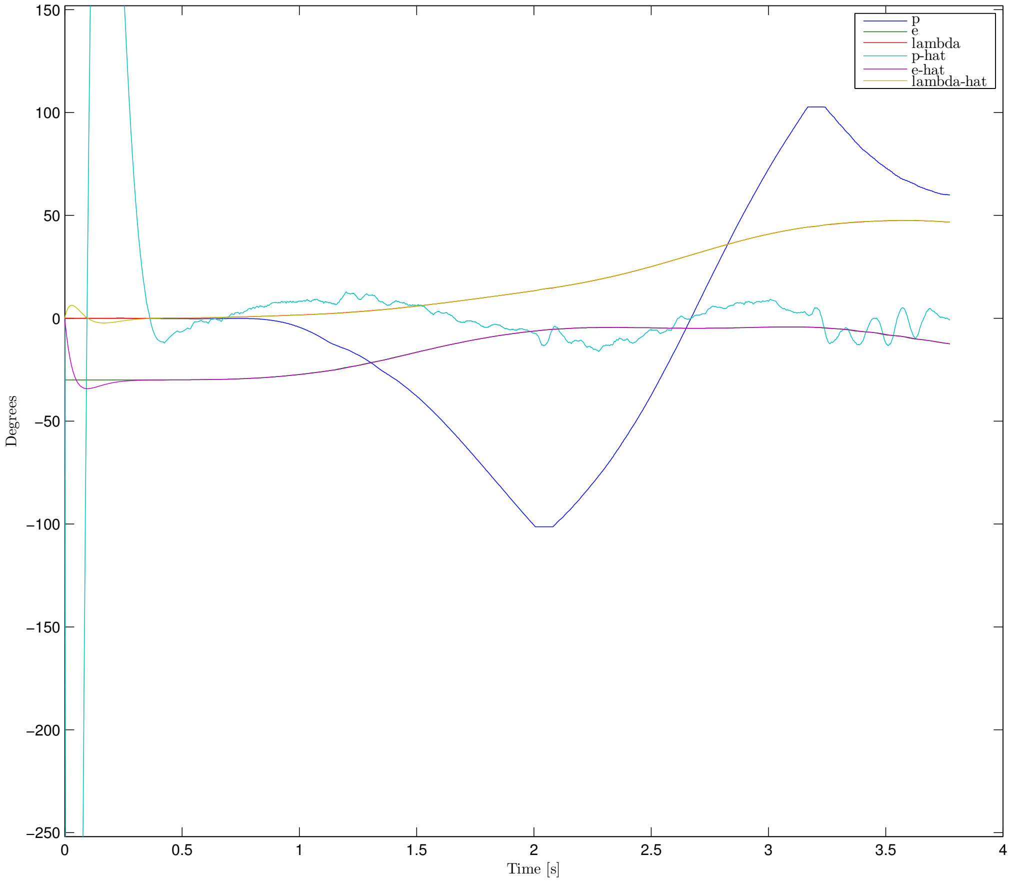

Figure 18: A zoomed in plot of three estimated and measured states when measuring $\tilde{e}$ and $\tilde{\lambda}$. The absolute value of the poles equals 20.

B MATLAB Code

B.1 P4p3_init

```
%%%%%%%%PI-reg from Part 3 problem 3
A = [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 = [0 0; 0 K1; K2 0; 0 0; 0 0];
C = [1 0 0 0 0; 0 0 1 0 0];
Q = [10 0 0 0 0; 0 1 0 0 0; 0 0 50 0 0; 0 0 0 100 0; 0 0 0 0 1];
R = [1 0; 0 1];

K = lqr(A,B,Q,R);
P = [0 10;
     5 0];

%%%%%%% A and B matrices part 4
A = [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;
     K3 0 0 0 0 0];

 B = [0 0;
     0 K1;
     0 0;
     K2 0;
     0 0
     0 0];

%% When not observable
C = [1 0 0 0 0 0;
     0 0 1 0 0 0];

O = obsv(A,C);
rank(O);

%% When observable
C = [0 0 1 0 0 0;
     0 0 0 0 1 0];

O = obsv(A,C);
rank(O);

r = 5;
theta = (pi/25)*[22 23 24 -24 -23 -22];
p = r*exp(i*theta);
L = real(transpose(place(transpose(A), transpose(C), p)));
```

C Simulink Diagrams

The following appendix contains some of the different Simulink diagrams that the group made.

C.1 Part 2 - Monovariable control, problem 1

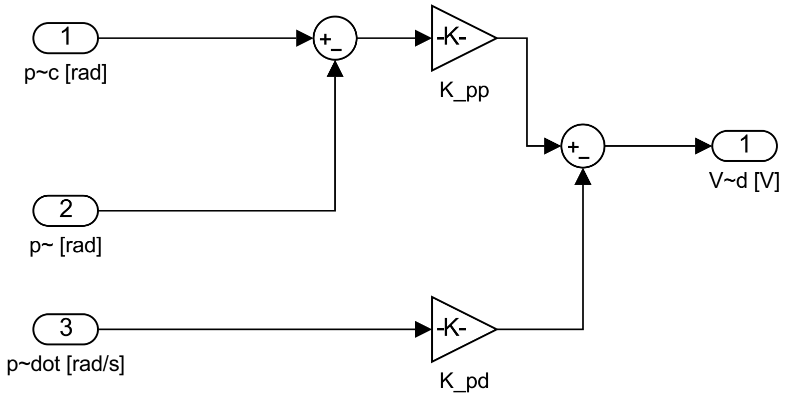

Figure 19: The pitch controller implemented in task 2, problem 1.

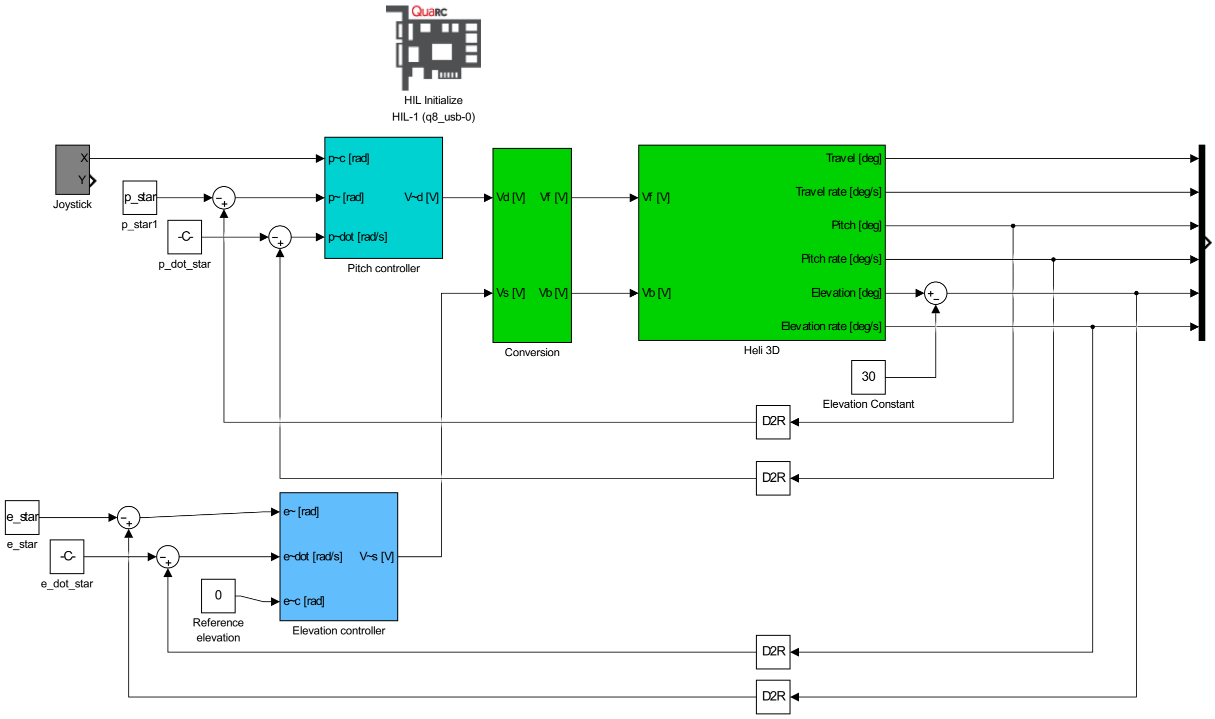

Figure 20: The Simulink diagram showing the handed out elevation angle controller and the PD controller for the pitch (in the turquoise subsystem) in task 2, problem 1.

C.2 Part 2 - Monovariable control, problem 2

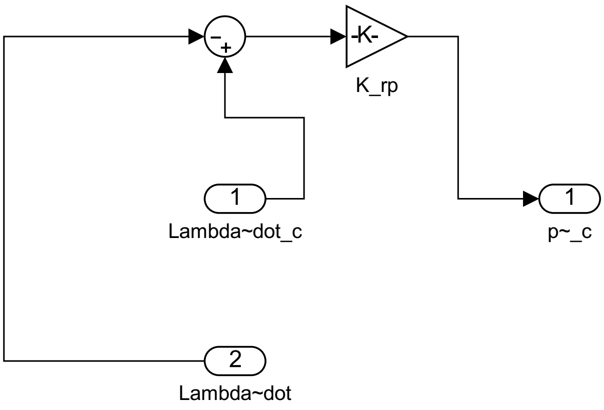

Figure 21: The P controller for travel rate implemented in task 2, problem 2.

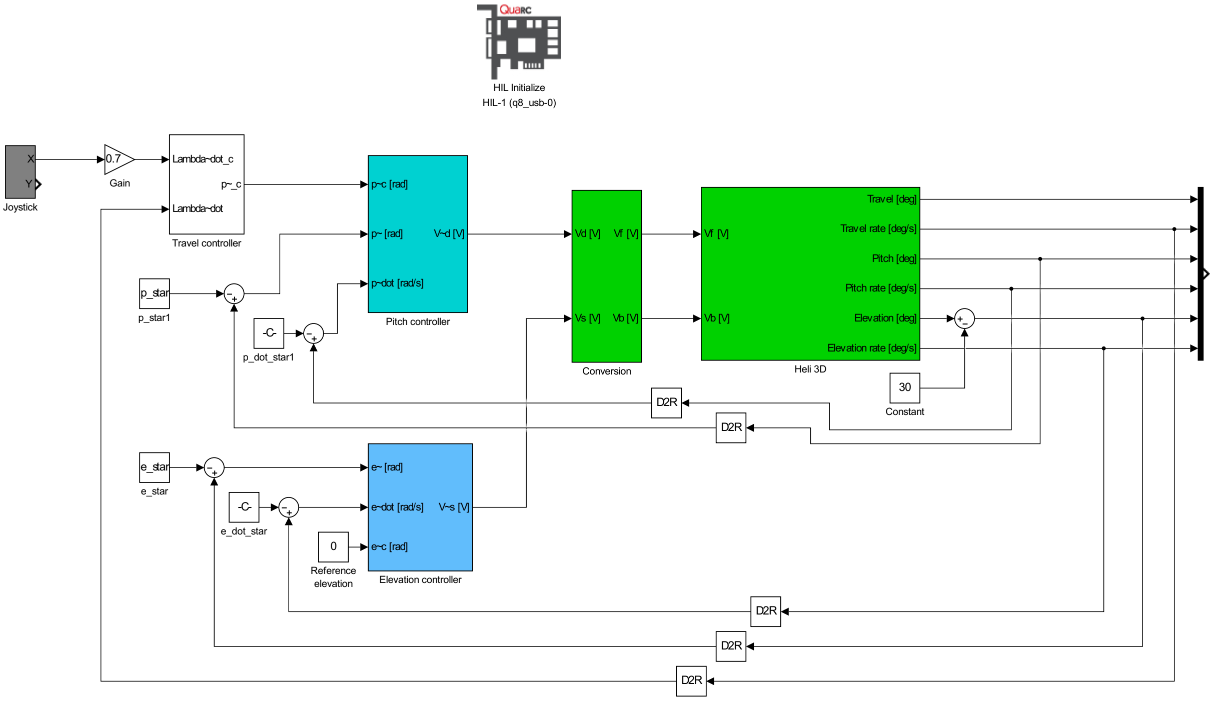

Figure 22: The Simulink diagram with the P controller for the travel rate added, in the white subsystem.

C.3 Part 4 - State estimation, linear observer

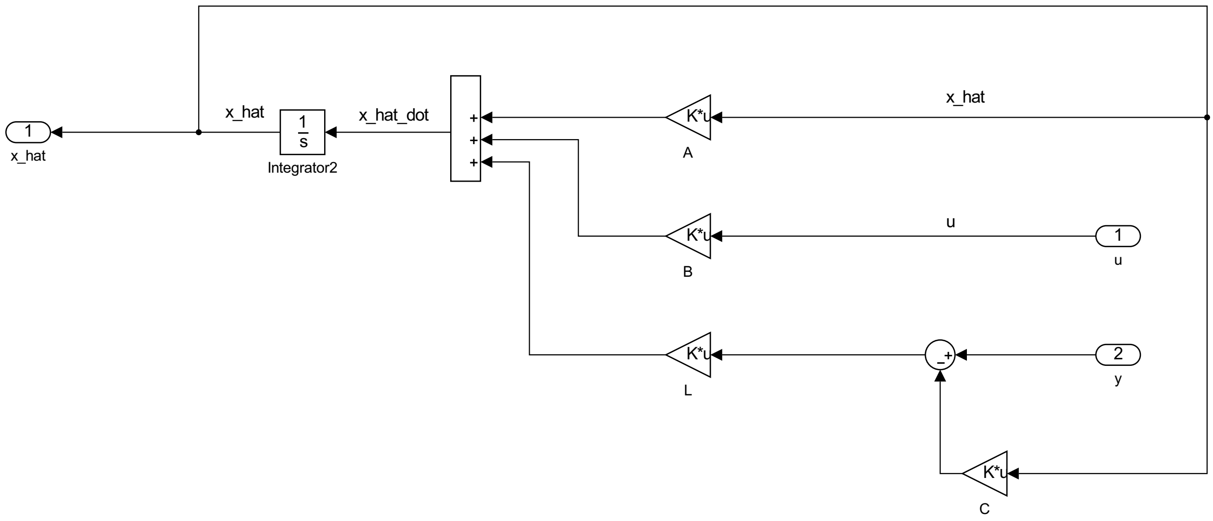

Figure 23: The linear observer implemented in Simulink in task 4, problems 2 and 3.

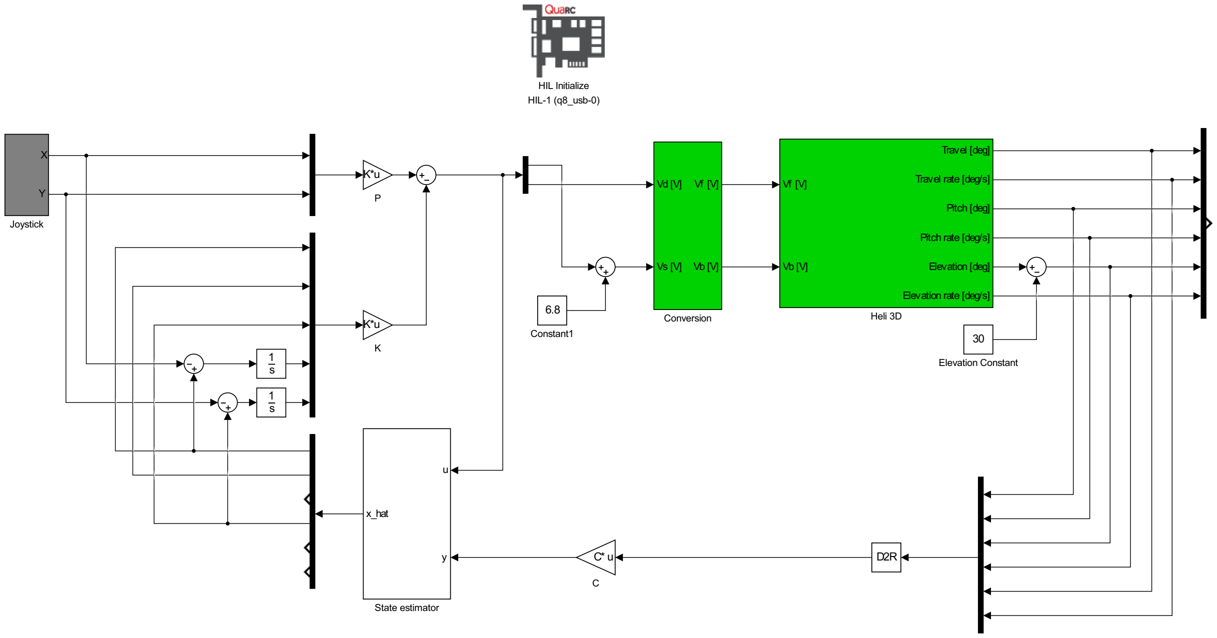

Figure 24: The Simulink diagram in task 4, problems 2 and 3 with a PI-controller.

D Table with constants

Table 1: Parameters and values. The numerical values are taken from the Matlab script that was handed out.

Symbol	Parameter	Numerical values
l_c	Distance from elevation axis to counterweight	0.46
l_h	Distance from elevation axis to helicopter head	0.66
l_p	Distance from pitch axis to motor	0.175
K_f	Force constant motor	0.143
J_e	Moment of inertia for elevation	1.034
J_λ	Moment of inertia for travel	1.078
J_p	Moment of inertia for pitch	0.044
m_p	Motor mass	0.72
m_c	Counterweight mass	1.92
g	Gravitational constant	9.81

References

[1] Chi-Tsong Chen, *Linear System Theory and Design*
Oxford University Press, International 4th edition, 2013