

Helicopter control using linear control theory

Helicopter lab for TTK4115 at NTNU

Alexander Johansen - 759059

Lars C. M. van der Lee - 768644

Åshild Lien - 768635

Group 43

October 2017

Contents

1	Introduction	3
2	Theory	4
2.1	Part 1: Feed Forward	4
2.1.1	Problem 1	4
2.1.2	Problem 2	5
2.2	Part 2: Monovariabale Control	7
2.2.1	Problem 1	7
2.2.2	Problem 2	7
2.3	Part 3: Multivariable Control	8
2.3.1	Problem 1	8
2.3.2	Problem 2	8
2.3.3	Problem 3	10
2.4	Part 4: State Estimation	11
2.4.1	Problem 1	11
2.4.2	Problem 2	11
2.4.3	Problem 3	12
3	Results	13
3.1	Part 1	13
3.1.1	Problem 3	13
3.1.2	Problem 4	14
3.2	Part 2	14
3.2.1	Problem 1	14
3.2.2	Problem 2	15
3.3	Part 3	16
3.3.1	Problem 2	16
3.3.2	Problem 3	18
3.4	Part 4	19
3.4.1	Problem 1	19
3.4.2	Problem 2	21
3.4.3	Problem 3	23
A	Nomenclature	27
B	Matlab code snippets	29
B.1	Pole placement	29
B.2	Observability	29
	References	30

1 Introduction

This lab assignment aims to use linear control theory to allow the control of a helicopter using a joystick. Trying to control the helicopter, without a regulator to stabilize the system, is nearly impossible. The helicopter becomes hard to control, and will crash if the user makes a small mistake.

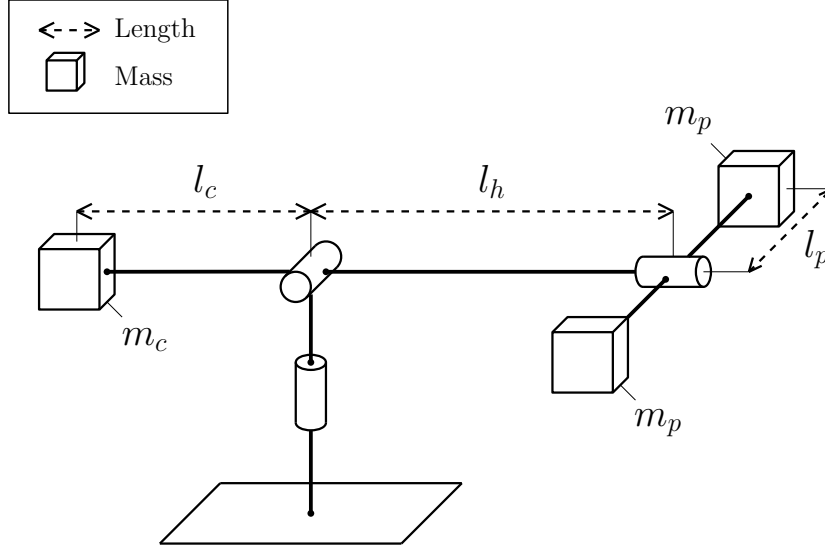


Figure 1: Helicopter setup, [3]

Figure 1 shows the helicopter setup used in the lab. The 2 propellers are balanced with a counterweight on the other side of the center joint. These are controlled by changing the voltage to each motor.

To complete the task, we have derived a model for the helicopter system. In the different sub problems of the lab assignment, this model will be used to test different regulators. During the lab, we are testing P, PD and multivariable controllers. In the last part of the lab we will also implement an estimator for the states, and use observability to estimate states that aren't measured. This will be combined with an LQR regulator to control the helicopter.

This rapport will have two main parts, theory and results. During the theory we will derive all the equations for our model and controllers. The results will showcase the helicopter behaviour, and compare it to other controllers. Here we also discuss why some controllers work better than others.

Tables with constants and parameters can be found in Appendix A. Appendix B contains Matlab code snippets. We used Matlab 2014a in the lab.

2 Theory

2.1 Part 1: Feed Forward

2.1.1 Problem 1

The relations between motor voltage, and actual force delivered are given by

$$F_f = K_f V_f \quad (1a)$$

$$F_b = K_f V_b \quad (1b)$$

where F_b and F_f are propeller forces, V_b and V_f are motor voltages and K_f is the motor force constant. Our regulator will create two outputs, V_s and V_d , which relate to the actual motor voltages like this:

$$V_s = V_f + V_b \quad (2a)$$

$$V_d = V_f - V_b \quad (2b)$$

We start with the momentum equations, derived from Newton's second law,

$$\sum M = J\alpha \quad (3)$$

where M is torque, J is moment of inertia and α is angular acceleration. The moment around the pitch joint, $J_p \ddot{p}$, is affected by four forces. The propeller forces generated by their blades, and their weight. Since our propellers have the same weight, and are mounted on each side of the joint, the weight forces cancel each other out. Therefore only the forces generated by the blades affect the moment. Propellers are mounted a distance l_p from the pitch joint, giving us the following expression for moment around the pitch joint:

$$J_p \ddot{p} = l_p F_f - l_p F_b \quad (4)$$

Substituting F_f and F_b with V_d given from eqs. (1) and (2b) gives

$$J_p \ddot{p} = l_p K_f V_d \quad (5)$$

By defining $L_1 = l_p K_f$, we find the equation of motion for pitch to be

$$J_p \ddot{p} = L_1 V_d \quad (6)$$

The moment around the elevation joint is affected by the three gravitational forces from the propellers and the counterweight, and can be described by the equation

$$J_e \ddot{e} = (F_{g,c} l_c - (F_{g,f} + F_{g,b}) l_h) \cos(e) + ((F_f + F_b) l_h) \cos(p) \quad (7)$$

where e is the elevation angle. Choosing $L_2 = F_{g,c} l_c - (F_{g,f} + F_{g,b}) l_h$, using eqs. (1) and (2a) to substitute F_f and F_b and then setting $L_3 = K_f l_h$, we get the second equation of motion on the desired form:

$$J_e \ddot{e} = L_2 \cos(e) + L_3 V_s \cos(p) \quad (8)$$

Only the propeller forces contribute to moment around the travel joint. The way the assignment defines direction for pitch and travel cause the need of a minus sign on the equation, given by

$$J_\lambda \ddot{\lambda} = -(F_f + F_b) l_h l_p \sin(p) \cos(e) \quad (9)$$

where λ is the travel angle. Again, we substitute $F_f + F_b = K_f V_s$ and define $L_4 = -K_f l_h$ and the equation of motion becomes

$$J_\lambda \ddot{\lambda} = L_4 V_s \sin(p) \cos(e) \quad (10)$$

Below are all the equations that make up our model of the helicopter, with their constants:

$$J_p \ddot{p} = L_1 V_d \quad (11a)$$

$$J_e \ddot{e} = L_2 \cos(e) + L_3 V_s \cos(p) \quad (11b)$$

$$J_\lambda \ddot{\lambda} = L_4 V_s \cos(e) \sin(p) \quad (11c)$$

with

$$L_1 = K_f l_p \quad (12a)$$

$$L_2 = F_{g,c} l_c - (F_{g,f} + F_{g,b}) l_h \quad (12b)$$

$$L_3 = K_f l_h \quad (12c)$$

$$L_4 = -K_f l_h \quad (12d)$$

2.1.2 Problem 2

As we can see from eq. (11), our system is not linear. The purpose of this lab is to control the system using linear control theory, which requires us to linearize the system. The system will be linearized around an equilibrium defined: $(p, e, \lambda)^\top = (p^*, e^*, \lambda^*)^\top$ and $(V_d, V_s)^\top = (V_d^*, V_s^*)^\top$ with $p^* = e^* = \lambda^* = 0$. Inserting $\ddot{p} = \ddot{e} = \ddot{\lambda} = 0$ in eq. (11) gives

$$V_d^* = 0 \quad (13a)$$

$$V_s^* = -\frac{L_2 \cos(e)}{L_3 \cos(p)} \quad (13b)$$

We linearize using

$$\tilde{\mathbf{A}} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}^*, \mathbf{u}^*}, \quad \tilde{\mathbf{B}} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{\mathbf{x}^*, \mathbf{u}^*} \quad (14)$$

where \mathbf{f} represents the equations in eq. (11) divided by J_p , J_e and J_λ , respectively [2].

Using eq. (14), we get the following matrices for our linearized system.

$$\tilde{\mathbf{A}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\frac{L_2 L_4}{L_3 J_\lambda} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (15)$$

and

$$\tilde{\mathbf{B}} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{L_1}{J_p} \\ 0 & 0 \\ \frac{L_3}{J_e} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (16)$$

By writing out the system equations, we get

$$\ddot{\tilde{p}} = \frac{L_1}{J_p} \tilde{V}_d \quad (17a)$$

$$\ddot{\tilde{e}} = \frac{L_3}{J_e} \tilde{V}_s \quad (17b)$$

$$\ddot{\tilde{\lambda}} = -\frac{L_2 L_4}{L_3 J_\lambda} \tilde{p} \quad (17c)$$

We assume the the moments of inertia for the helicopter are constant, meaning J_p , J_e and J_λ are given with these values:

$$J_p = 2m_p l_p^2 \quad (18a)$$

$$J_e = m_c l_c^2 + 2m_p l_h^2 \quad (18b)$$

$$J_\lambda = m_c l_c^2 + 2m_p (l_h^2 + l_p^2) \quad (18c)$$

These values come from the lab assignment. Combining (17) and (18) gives us the system on the following form.

$$\ddot{\tilde{p}} = K_1 \tilde{V}_d \quad (19a)$$

$$\ddot{\tilde{e}} = K_2 \tilde{V}_s \quad (19b)$$

$$\ddot{\tilde{\lambda}} = K_3 \tilde{p} \quad (19c)$$

with the constants K_1 , K_2 and K_3 given by

$$K_1 = \frac{K_f}{2m_p l_p} \quad (20a)$$

$$K_2 = \frac{K_f l_h}{m_c l_c^2 + 2m_p l_h^2} \quad (20b)$$

$$K_3 = -\frac{F_{g,c} l_c - (F_{g,f} + F_{g,b}) l_h}{L_3 m_c l_c^2 + 2m_p (l_h^2 + l_p^2)} \quad (20c)$$

2.2 Part 2: Monovariabe Control

2.2.1 Problem 1

The PD-controller is given by

$$\ddot{V}_d = K_{pp}(\tilde{p}_c - \tilde{p}) - K_{pd}\dot{\tilde{p}} \quad (21)$$

where K_{pp} and K_{pd} are controller gains, and \tilde{p}_c is the pitch angle reference. By substituting eq. (21) into the linearized model of elevation given by eq. (19a), we get

$$\ddot{\tilde{p}} = K_1 K_{pp}(\tilde{p}_c - \tilde{p}) - K_{pd}\dot{\tilde{p}} \quad (22)$$

By applying the Laplace transform to eq. (22), we obtain the transfer function given by

$$\frac{\tilde{p}}{\tilde{p}_c} = \frac{1}{1 + s \frac{K_{pd}}{K_{pp}} + \frac{s^2}{K_1 K_{pp}}} \quad (23)$$

In order to choose suitable values for K_{pp} and K_{pd} , we write the denominator on the form $1 + s \left(\frac{2\zeta}{\omega_0} \right) + s^2 \left(\frac{1}{\omega_0^2} \right)$. We want our system to be critically damped, which can be achieved when $\zeta = 1$. Using this, we can express K_{pp} and K_{pd} as functions of ω_0 in the following way:

$$K_{pp} = \frac{\omega_0^2}{K_1} \quad (24a)$$

$$K_{pd} = \frac{2\omega_0}{K_1} \quad (24b)$$

2.2.2 Problem 2

We want to control the travel with a P controller. Travel is controlled by changing the pitch. This make the helicopter rotate around the travel joint. The controller model is given by:

$$\tilde{p}_c = K_{rp}(\dot{\lambda}_c - \dot{\lambda}) \quad (25)$$

where $K_{rp} < 0$. Inserting this into the linearized equation of motion for travel, (19c) results in

$$\ddot{\lambda} = K_3 K_{rp} (\dot{\lambda}_c - \dot{\lambda}) \quad (26)$$

assuming that the pitch is perfectly controlled, i.e. $\tilde{p} = \tilde{p}_c$. By applying the Laplace transform, the transfer function from travel rate reference $\dot{\lambda}_c$ to travel rate $\dot{\lambda}$ is given by

$$\frac{\dot{\lambda}}{\dot{\lambda}_c} = \frac{K_3 K_{rp}}{s + K_3 K_{rp}} \quad (27)$$

2.3 Part 3: Multivariable Control

2.3.1 Problem 1

We want to put eq. (19a) and (19b) in state-space form,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (28)$$

Given $\mathbf{x} = [\tilde{p}_c, \dot{\tilde{p}}_c, \dot{\tilde{e}}_c]^\top$ and $\mathbf{u} = [\tilde{V}_s, \tilde{V}_d]^\top$ we get

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (29)$$

and

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \end{bmatrix} \quad (30)$$

2.3.2 Problem 2

We first want to check the controllability of the system in eq. (28) with \mathbf{A} and \mathbf{B} given by (29) and (30). In order to check if the system is controllable, we need to compute the controllability matrix $\mathcal{C} = [\mathbf{B} \ \mathbf{A}\mathbf{B} \ \dots \ \mathbf{A}^{n-1}\mathbf{B}]$. In this case, to obtain the correct dimensions $n \times qn$, where n is the number of rows in \mathbf{A} and q is the number of columns in \mathbf{B} , \mathcal{C} is found to be

$$\mathcal{C} = \begin{bmatrix} 0 & 0 & 0 & K_1 & 0 & 0 \\ 0 & K_1 & 0 & 0 & 0 & 0 \\ K_2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (31)$$

In order to be controllable, \mathcal{C} has to have rank equal to n . As our controllability matrix \mathcal{C} has $\text{rank}(\mathcal{C}) = 3 = n$, the system is controllable.

We use the controller

$$\mathbf{u} = -\mathbf{K}\mathbf{x} + \mathbf{P}\mathbf{r} \quad (32)$$

where \mathbf{K} is the linear quadratic regulator. \mathbf{K} is defined so that $\mathbf{u} = -\mathbf{K}\mathbf{x}$ optimizes the cost function

$$J = \int_0^\infty (\mathbf{x}^\top(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^\top(t)\mathbf{R}\mathbf{u}(t)) dt \quad (33)$$

The cost function calculates a cost for the system. The cost increases when the error for the states increases, and when input is applied. When tuning the regulator we aim to minimize the system cost, thus having a small error using little input.

\mathbf{Q} and \mathbf{R} are diagonal weighting matrices. Along the diagonal are values for each of the states or inputs. By increasing the values in \mathbf{Q} , the cost for the error increases. This will force the regulator to minimize the error. Increasing the values in \mathbf{R} increases the cost for using input, forcing the regulator to be more careful on input usage. So by tuning individual values in these matrices, we can control which aspects of the system the regulator prioritizes.

After deciding what the \mathbf{Q} and \mathbf{R} matrices should be, we use the matlab command `lqr(A, B, Q, R)` which returns the corresponding \mathbf{K} matrix.

Now that we have the \mathbf{K} matrix, we can substitute \mathbf{u} from eq. (32) into eq. (28), giving us the following system:

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{BK})\mathbf{x} + \mathbf{BPr} \quad (34)$$

We want to track the reference $\mathbf{r} = [\tilde{p}_c, \dot{\tilde{e}}_c]^\top$. That is, we want our output \mathbf{y} , given as

$$\mathbf{y} = \mathbf{Cx} + \mathbf{Du} \quad (35)$$

to follow the reference \mathbf{r} . The output is $\mathbf{y} = [\tilde{p}, \dot{\tilde{e}}]^\top$, and we find the matrices \mathbf{C} and \mathbf{D} to be

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (36)$$

$$\mathbf{D} = 0 \quad (37)$$

We want to choose \mathbf{P} such that $\lim_{t \rightarrow \infty} \tilde{p} = \tilde{p}_c$ and $\lim_{t \rightarrow \infty} \dot{\tilde{e}} = \dot{\tilde{e}}_c$, with fixed values for $\tilde{\mathbf{p}}_c$ and $\dot{\tilde{\mathbf{e}}}_c$. This implies $\lim_{t \rightarrow \infty} \dot{\tilde{p}} = 0$, thus $\dot{\mathbf{x}} = 0$.

Setting $\dot{\mathbf{x}} = 0$ in eq. (28) gives the following equation

$$\mathbf{0} = (\mathbf{A} - \mathbf{BK})\mathbf{x}_\infty + \mathbf{BPr} \quad (38)$$

where \mathbf{x}_∞ is the value of \mathbf{x} when $\dot{\mathbf{x}} = 0$. Solving for \mathbf{x}_∞ and combine with eq. (35) allows us to find an equation for \mathbf{P} .

$$\mathbf{y}_\infty = \mathbf{r} = \mathbf{Cx}_\infty = \mathbf{C}(\mathbf{BK} - \mathbf{A})^{-1}\mathbf{BPr} \quad (39)$$

We solve for \mathbf{P} :

$$\mathbf{P} = (\mathbf{C}(\mathbf{BK} - \mathbf{A})^{-1}\mathbf{B})^{-1} \quad (40)$$

2.3.3 Problem 3

We now want to add integral effect to our system. Given $\mathbf{x}_i = [\tilde{p}_c, \dot{\tilde{p}}_c, \dot{\tilde{e}}_c, \dot{\gamma}, \dot{\zeta}]^\top$ where γ and ζ are given by

$$\dot{\gamma} = \tilde{p} - \tilde{p}_c \quad (41)$$

$$\dot{\zeta} = \dot{\tilde{e}} - \dot{\tilde{e}}_c \quad (42)$$

and subscript i denote integral effect, we get a new state space equation

$$\dot{\mathbf{x}}_i = \mathbf{A}_i \mathbf{x}_i + \mathbf{B}_i \mathbf{u}_i + \mathbf{F}_i \mathbf{r}_i \quad (43)$$

with

$$\mathbf{A}_i = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (44)$$

$$\mathbf{B}_i = \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (45)$$

$$\mathbf{F}_i = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (46)$$

With this new model, we also need to calculate new matrices for the controller. This means generating a new \mathbf{K} and \mathbf{P} matrix. We get \mathbf{K} from the LQR matlab command. \mathbf{P} is calculated with eq. (40). Because our system has a lot of zeroes, calculating \mathbf{P} with out system matrices doesn't work. We get a singular matrix, which cannot be inverted.

During the Linear systems lecture, we learned to ignore the integral states when calculating \mathbf{P} to avoid this problem, as they are not physical states in our system but aid our mathematical model. We therefor sliced the 5x5 matrices to 3x3 matrices, and used this to calculate the controller matrices.

2.4 Part 4: State Estimation

2.4.1 Problem 1

We start by deriving a state space model on the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (47)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \quad (48)$$

based on eq. (19), the linearized model of the system, where $\mathbf{x} = [\tilde{p}, \dot{\tilde{p}}, \tilde{e}, \dot{\tilde{e}}, \tilde{\lambda}, \dot{\tilde{\lambda}}]^\top$.

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ K_3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (49)$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ 0 & 0 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (50)$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (51)$$

2.4.2 Problem 2

We begin by examining the observability of the system. We look at the observability matrix \mathcal{O} , defined as the $nq \times n$ matrix

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \vdots \\ \mathbf{C}\mathbf{A}^{n-1} \end{bmatrix} \quad (52)$$

where n is the number of columns in \mathbf{A} and q is the number of rows in \mathbf{C} . With \mathbf{A} and \mathbf{C} from eqs. (49) and (51), \mathcal{O} becomes a 18×6 matrix. We used the matlab command `obsv()` (see appendix B.2) to obtain the following:

$$\mathcal{O} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ K_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (53)$$

We see from (53) that $\text{rank}(\mathcal{O}) = 6$, and the system is thus observable. We can now create a linear system on the form

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{L}(\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}) \quad (54)$$

where $\hat{\mathbf{x}}$ is the observer and \mathbf{L} is the observer gain matrix. We use pole placement to place the poles, done with the matlab command `poles(A', C', poles)'`, where `poles` is a list of desired poles and `'` represent the matrix transpose.

2.4.3 Problem 3

We now wish to examine the observability of the system for $\mathbf{y} = [\tilde{e}, \tilde{\lambda}]^\top$ and $\mathbf{y}_2 = [\tilde{p}, \tilde{e}]^\top$. This gives us new \mathbf{C} matrices

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (55)$$

and

$$\mathbf{C}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (56)$$

We can use the same matlab command as in part 4, problem 2, to examine the observability of the systems. The observability matrices are

$$\mathcal{O} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ K_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (57)$$

and

$$\mathcal{O}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (58)$$

We see from (57) that $\text{rank}(\mathcal{O}) = 6$ and from (58) that $\text{rank}(\mathcal{O}_2) = 4$, that is, the system is not observable when only the pitch and elevation is measured. However if we measure elevation and travel, the system is observable. Since \mathbf{y} and \mathbf{C} have new dimensions, we need a new \mathbf{L} . This can be obtained by the same `place()` command as before.

3 Results

3.1 Part 1

3.1.1 Problem 3

When steering the helicopter using only feed forward control from the joystick, it is extremely difficult to control. The mathematical models seem to represent the physical system quite well although forces from friction, air resistance, unlinearity in the motors etc. are not included. Also our propellers seemed to have a slight difference in weight causing the pitch to increase without any differential input.

We observed that the linearized model, as expected, did not represent the system very well outside of the linearized area. For example the helicopter needed a higher voltage in order to maintain its elevation the higher it was.

3.1.2 Problem 4

Considering that our linearized model works best around the equilibrium, 0 elevation, we need to alter the encoders in our system to count the equilibrium as 0. We did this by adding constants to the encoder outputs, so that the equilibrium matched 0 on the output.

We also had to find V_s^* , which is the V_s input required to keep the helicopter at an elevation that matched the new encoder 0. V_s^* is added to the output from the controller, and was found by measuring the required voltage.

Table 1: Measured values.		
Variable	Value	Unit
Elevation encoder offset	-32	°
Pitch encoder offset	-4.1	°
V_s^*	6.7	V

Combining the equations for motor voltage and momentum around the elevation joint, and inserting V_s^* gives the motor force constant, K_f

$$K_f = \frac{(m_c l_c - 2m_p l_h)g}{V_s l_h} \quad (59)$$

Inserted values give $K_f = 0.149N/V$.

3.2 Part 2

3.2.1 Problem 1

We added a PD controller for the pitch to Simulink, as shown in Figure 2 and 3.

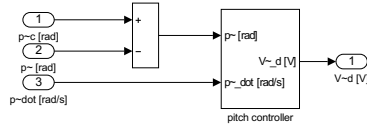


Figure 2: Simulink diagram for pitch controller

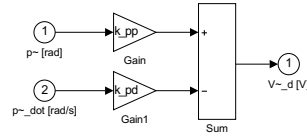


Figure 3: Simulink diagram for pitch controller subblock

By tuning the value of ω_0 in eqs. (24a) and (24b), we can achieve different responses for the pitch controller.

By testing we found the system to be optimally tuned when $\omega_0 = \pi$, this is the value we continued using. The pitch response to step inputs can be seen in Figure 4. To illustrate the response when choosing ω_0 too low or too high, we have included the responses for $\omega_0 = \pi/2$ and $\omega_0 = 2\pi$ in Figures 5 to 6.

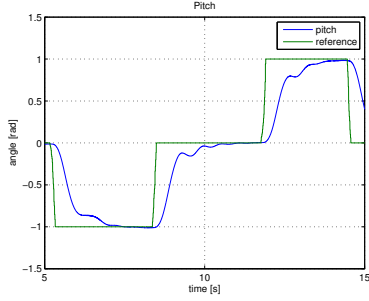


Figure 4: Pitch with $\omega_0 = \pi$.

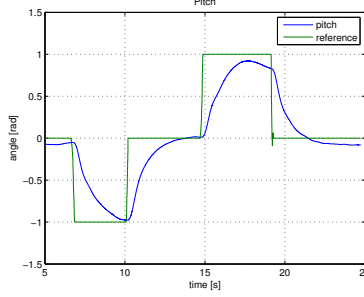


Figure 5: Pitch with $\omega_0 = \pi/2$.

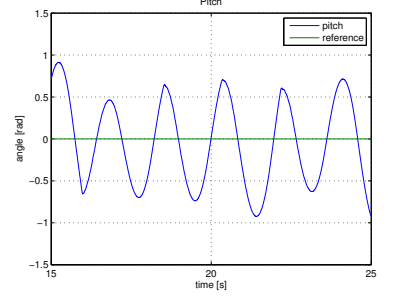


Figure 6: Pitch with $\omega_0 = 2\pi$.

As we see, the response is fast and accurate for $\omega_0 = \pi$ in Figures 4. When $\omega_0 = \pi/2$, the pitch angle is slower and does not reach the reference, and as we see from Figure 5. $\omega_0 = 2\pi$ gives unstable oscillations, these would increase if pitch did not have a maximum physical value.

By inserting $\omega_0 = \pi$ into eq. (24), we get $K_{pp} = 16.70$ and $K_{pd} = 10.63$.

3.2.2 Problem 2

We implemented a P controller for travel rate in Simulink as shown in Figure 7.

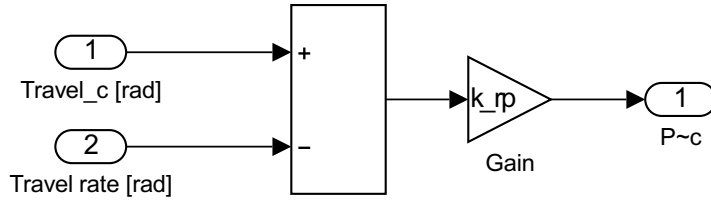


Figure 7: Simulink diagram for travel rate controller

We tuned the controller by trying different values of K_{rp} until we found a satisfying response. The plots in Figures 8-10 shows two different choices of K_{rp} . Choosing $K_{rp} = -1.1$ gave the faster and more accurate response.

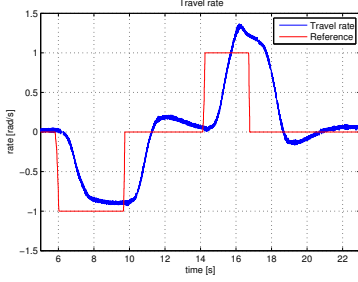


Figure 8: Travel rate, $K_{rp} = -1.1$.

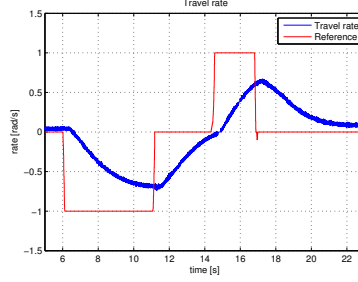


Figure 9: Travel rate, $K_{rp} = -0.5$.

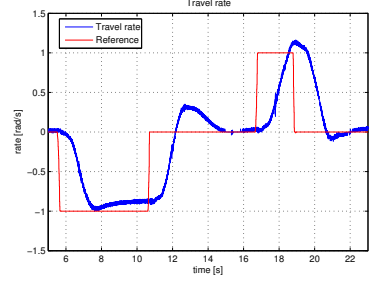


Figure 10: Travel rate, $K_{rp} = -1.5$.

3.3 Part 3

3.3.1 Problem 2

We now add an LQR controller to our system, aiming to control the pitch angle and elevation rate. The controller is given by eq. (32). The Simulink diagram can be seen in Figure 11.

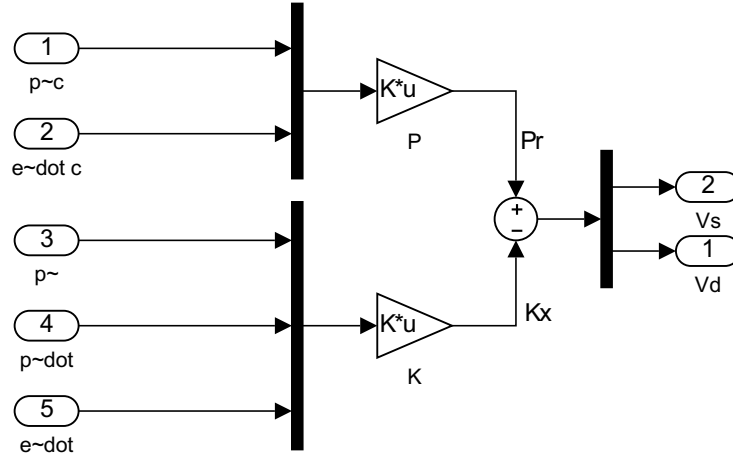


Figure 11: Simulink diagram for multivariable controller

When tuning the weighting matrices \mathbf{Q} and \mathbf{R} we have to keep the system as

close to the linearized model as possible or the assumptions made in (28) will no longer be valid. We started with 1 on all the values. We want to control pitch and elevation rate, so these values were increased if the system was too slow/far away from the references. When the system would overshoot, we increased the value in R , so that the regulator can't use as much input, and lowered them when they failed to reach the reference. After repeating these steps a few times we found our final values. The final weighting matrices are seen in (60) and (61).

$$\mathbf{Q} = \begin{bmatrix} 10000 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 100000 \end{bmatrix} \quad (60)$$

$$\mathbf{R} = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \quad (61)$$

With these weighting matrices, \mathbf{K} and \mathbf{P} become

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 31.62 \\ 10 & 5.828 & 0 \end{bmatrix} \quad (62)$$

$$\mathbf{P} = \begin{bmatrix} 0 & 31.62 \\ 10 & 0 \end{bmatrix} \quad (63)$$

The resulting responses for elevation rate and pitch angle can be seen in Figure 12 and 13.

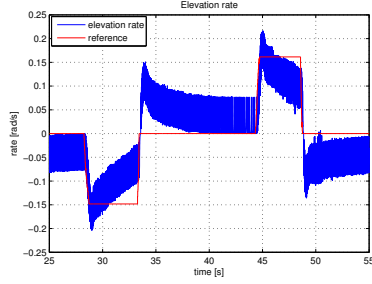


Figure 12: Elevation rate with LQR.

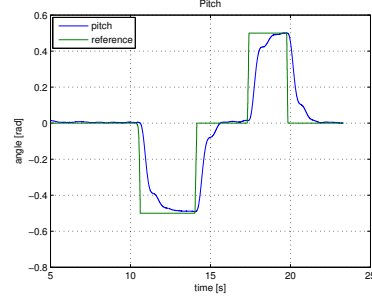


Figure 13: Pitch with LQR.

From Figure 12 we can see that the LQR controller alone is not enough to control the elevation rate. This is a result of having a linearized model. As the helicopter moves away from the zero state the nonlinearity in (11b) forces the helicopter back to its zero state.

3.3.2 Problem 3

We wanted to include the integral effect to mitigate the effects of linearizing the model, this is most significant for \dot{e} . This works by integrating the steady state error that occurs from the change of gravitational force on the helicopter when it moves away from the zero state. In effect this will set a reference elevation that the the system til try to return to, even when exposed to external physical disturbances.

The Simulink model for the multivariable controller with integral effect can be seen in Figure 14.

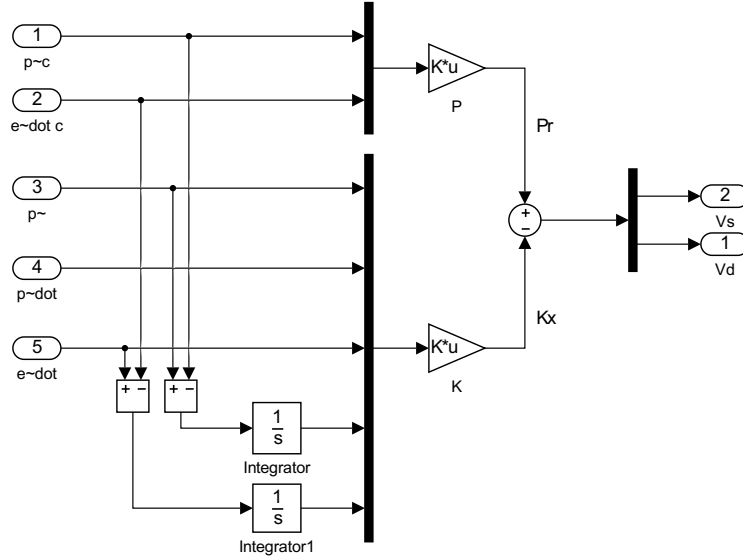


Figure 14: Simulink diagram for multivariable controller with integral effect.

When deciding how to weight $\dot{\gamma}$ and $\dot{\zeta}$ we knew that the pitch using the LQR controller already gave us good results, while the elevation rate would be more dependant on the integral effect. We therefore weighted $\dot{\gamma}$ lower than $\dot{\zeta}$ which we weigh quite high. After tuning the controller the final values for \mathbf{Q}_i and \mathbf{R}_i are

$$\mathbf{Q}_i = \begin{bmatrix} 10000 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 10000 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 10000 \end{bmatrix} \quad (64)$$

$$\mathbf{R}_i = \begin{bmatrix} 75 & 0 \\ 0 & 100 \end{bmatrix} \quad (65)$$

After optimizing the cost function (33) using the matlab command `lqr(A, B, Q, R)` - where A, B, Q and R are respectively \mathbf{A}_i , \mathbf{B}_i , \mathbf{Q}_i and \mathbf{R}_i , we acquire the following \mathbf{K}_i matrix

$$\mathbf{K}_i = \begin{bmatrix} 0 & 0 & 17.62 & 0 & 10 \\ 12.04 & 7.13 & 0 & 3.16 & 0 \end{bmatrix} \quad (66)$$

The resulting response can be seen in Figure 15 and 16. As we see, the elevation

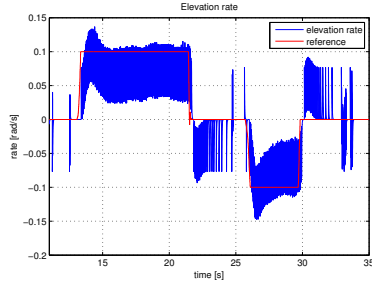


Figure 15: Elevation rate with LQR and integral effect.

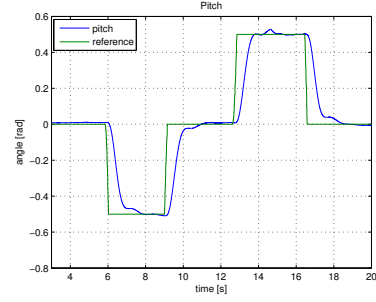


Figure 16: Pitch with LQR and integral effect.

rate does not go back to zero as it did with just the LQR in Figure 12. The pitch angle in Figure 16 is still fast and accurate, the response is about the same as we saw with LQR in Figure 13.

3.4 Part 4

3.4.1 Problem 1

Implemented our new model in Simulink.

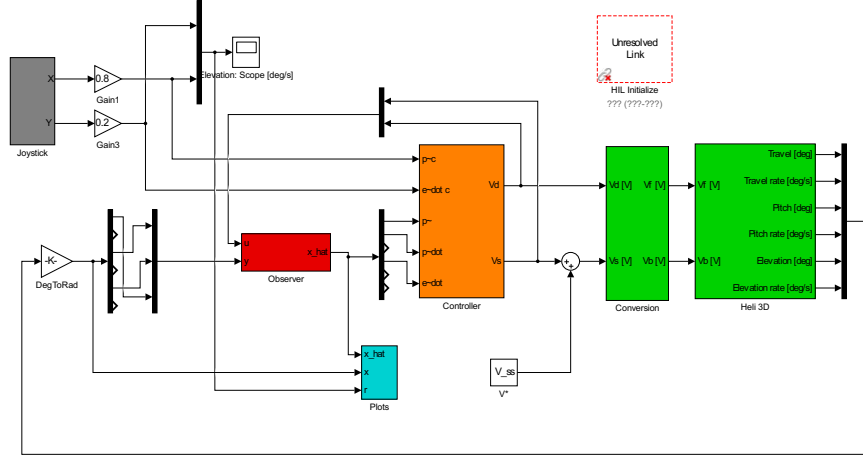


Figure 17: Simulink model with observer subsystem

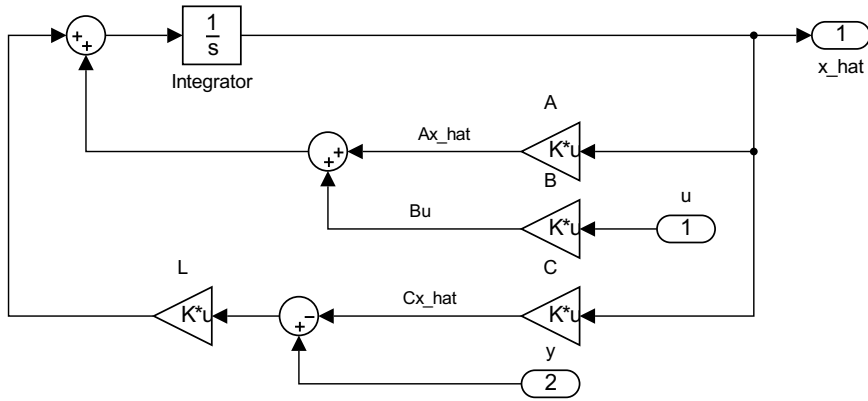


Figure 18: Simulink model of observer

The estimator is used as input to the controller, instead of the actual measurements y . To calculate the estimator, the matrices from the linearized system have been used. This means that the estimator, just like our modelled system, works best when close to the equilibrium. When the system moved away from this point, both the system and the estimator will differ from the real world, meaning that the regulator won't be able to fully regulate the system to the reference.

3.4.2 Problem 2

The estimator poles need to be placed in such a way that the estimator will be able to predict the error faster than the system. This means putting the estimator poles further out into the left plane. We used $r = 10 \cdot \max(\text{abs}(\text{eig}(A-BK)))$ which gave us an $r = 24$ as a base for our tuning, and adjusted the r value until the estimator followed the actual system without a lot of noise. Matlab code for computing the poles can be found in appendix B.1. Some example responses for different values of r can be seen in Figures 19, 20 and 21.

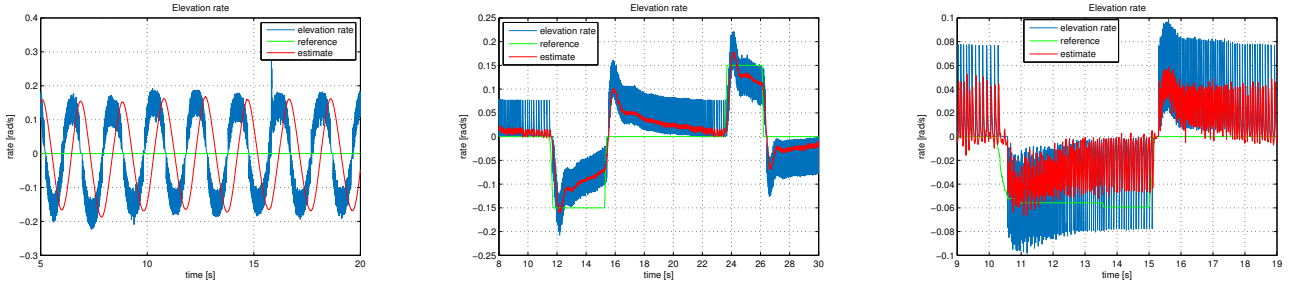


Figure 19: Elevation rate with $r = 5$. Figure 20: Elevation rate with $r = 27$. Figure 21: Elevation rate with $r = 60$.

These three figures demonstrate what happens when the poles are placed either too close to the origin, just right, and too far away. Figure 19 has the poles at $r = 5$. This is not big enough for the estimator to predict the system fast enough. As we see on the plot it creates a phase lag, and this results in the system not staying at the reference, which is 0. Our controller attempts to use the estimator to keep the system at the reference, when the estimator is too slow, the prediction will already have happened, meaning the controller is always reacting to late.

Figure 21 shows the opposite behaviour, with poles with a too large radius. This plot shows the behaviour for $r = 60$. Now the estimator is fast enough to predict the system. But where the small radius has a strong low filtering property, we see that this estimator doesn't. This causes a lot of noise on the estimator. A noisy estimator doesn't contribute anything useful to the system, and we didn't notice any big difference in the helicopter behaviour compared to the system without an estimator. Comparing the plots in fig. 15 and the blue line in fig. 21, also shows this behaviour.

The figure in the middles, fig. 20 show our final result. Here the poles are placed as shown in fig. 22, with a radius $r = 27$. The estimator is fast enough to now predict the system, while still suppressing the noise from the elevation rate measurements by using its low pass characteristics.

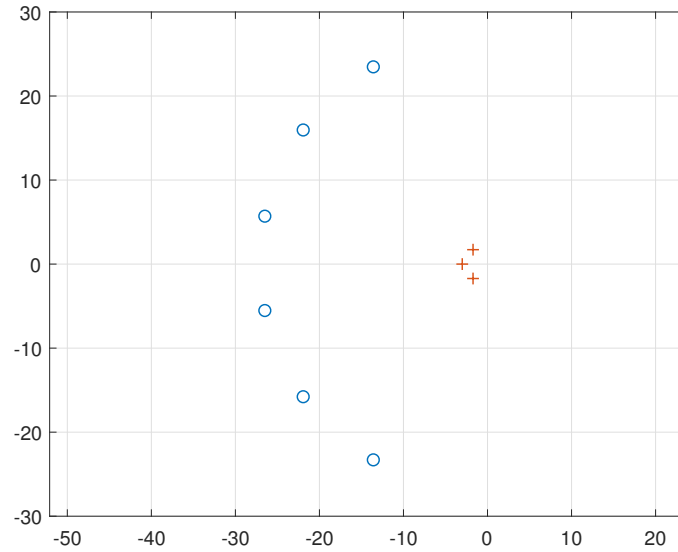


Figure 22: Pole placement for P controller. System poles: +, estimator poles: o

We can further improve the system by implementing an integral effect as we did in part 3 problem 3. We found we got better results here by increasing the radius of the poles slightly and the final result of this can be seen in Figure 23.

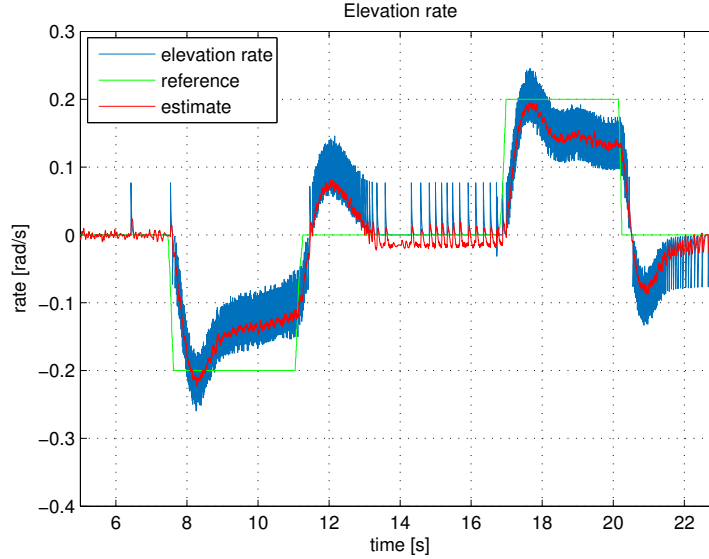


Figure 23: Elevation rate with $r = 42$ and controller with integral effect

3.4.3 Problem 3

In the final part of the lab assignment, our goal was to see whether we could use an estimator to control the system if we changed y . As shown in section 2.4.3, only measuring pitch and elevation meant the system wouldn't be observable. However, by only measuring the elevation and travel, the system could be observed. This is because of the connection between travel and pitch, given by eq. (19c). This allows the estimator to predict what the pitch is based on its knowledge of λ .

Even though the system is observable, it is harder to tune the controller. If the controller is to predict the pitch, it will need to differentiate the travel measurement, twice. It is a bad idea to differentiate a signal because differentiating noise creates even more, stronger noise. So differentiating noise twice isn't a good thing either.

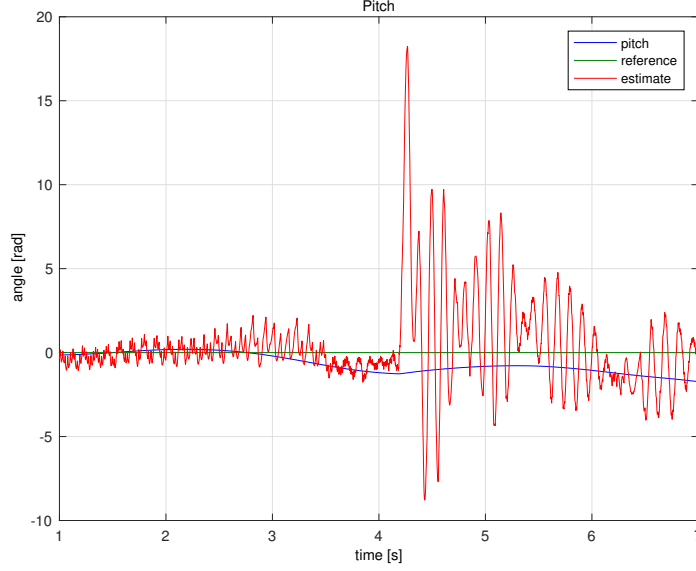


Figure 24: Pitch when pitch and elevation is measured.

Figure 24 shows this behaviour. The estimator thinks that the helicopter is swinging a lot around the p joint, while it in reality is just leaning to one side. The controller uses the estimated value for control, and because of this, the system is hard to tune for pitch. The elevation is passed into the observer block, and is therefore easier to control.

When deciding how to tune the system we had to reduce the weight of the integrating factors dramatically as these produced an uncontrollable amount of noise, but we could not remove the entirely as steady state errors are expected when only being able to observe the travel to calculate the pitch. We also reduced the rest of the values in the system overall. The system is therefore slower, but when regulating a signal with a high amount of noise it is favorable that the system is “sure” of a change in pitch before responding. As the elevation estimated quite well the cost of regulating it is cheaper than regulating the pitch. The final \mathbf{Q} and \mathbf{R} can be seen in equations

$$\mathbf{Q}_i = \begin{bmatrix} 150 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 150 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 10 \end{bmatrix} \quad (67)$$

$$\mathbf{R}_i = \begin{bmatrix} 2 & 0 \\ 0 & 20 \end{bmatrix} \quad (68)$$

The pole placement of the observer is also of paramount importance, and the system was highly sensitive to their placement. Poles at too close of a radius caused exponential oscillations, the observer is too slow to react to changes and the propellers simply vibrate at a high frequency. If the poles are too far away the integral effect causes the pitch and elevation to increase exponentially in one direction as the system is too slow to react to the estimated changes and the integral effect dominates. While tuning we found we got better responses by placing all poles close to each other on the real axis and settled on having the closest pole at -25.928^1 and the other poles at 0.1 increments behind. The poles can be seen in Figure 25.

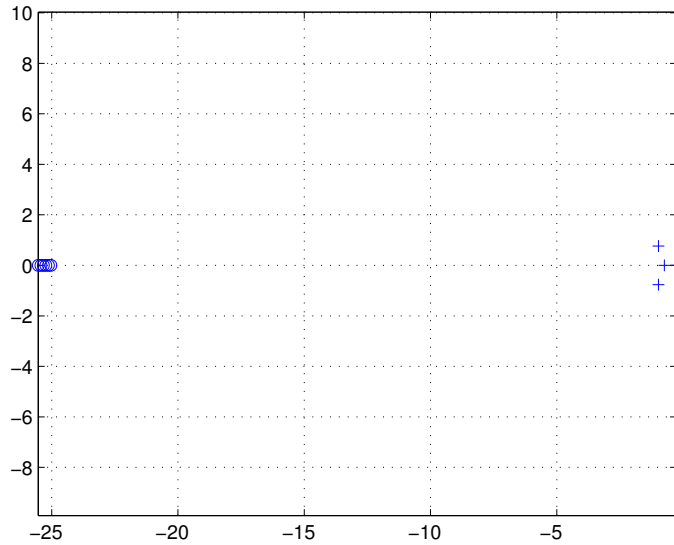


Figure 25: Poles of the stable observer system

The final response of the system can be seen in Figure 26.

¹the system was so sensitive that three decimal places were necessary

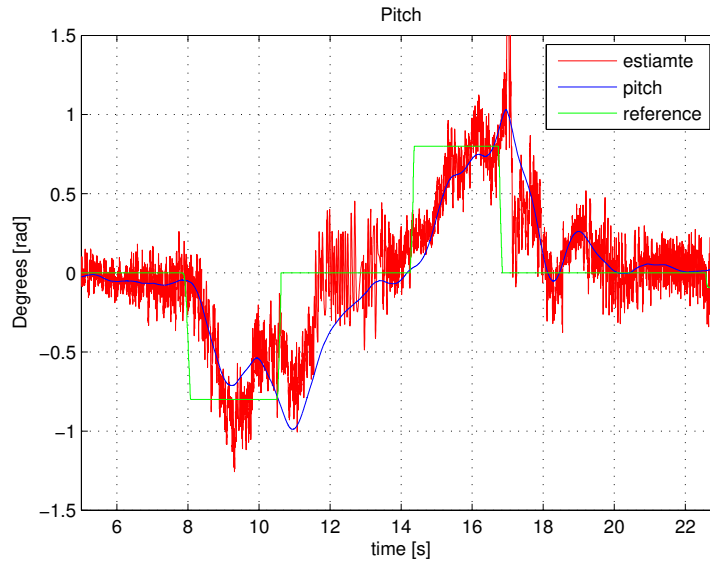


Figure 26: Pitch of the system using the bad observer

In conclusion; just because a system is observable for certain values it will not necessarily result in a well regulated system, it may be a good idea to apply a PBH test² when one only has access to a limited number of states [1].

²Popov-Belevitch-Hautus test. Used to determine how controllable or observable a system is.

A Nomenclature

Table 2: Parameters and values.

Symbol	Parameter	Value	Unit
e^*	Linearization point, elevation angle	0	rad
$F_{g,b}$	Gravitational force, back motor	7.063	N
$F_{g,c}$	Gravitational force, counterweight	18.84	N
$F_{g,f}$	Gravitational force, front motor	7.063	N
g	Gravitational acceleration	9.81	m/s ²
J_e	Moment of inertia for elevation	1.036	kg m ²
J_λ	Moment of inertia for travel	1.078	kg m ²
J_p	Moment of inertia for pitch	0.044	kg m ²
K_f	Motor force constant	0.149	N/V
l_c	Distance from elevation axis to counterweight	0.46	m
l_h	Distance from elevation axis to helicopter head	0.66	m
l_p	Distance from pitch axis to motor	0.175	m
m_p	Motor mass	0.72	kg
m_c	Counterweight mass	1.92	kg
p^*	Linearization point, pitch angle	0	rad
V_d^*	Linearization point, voltage difference	0	V
V_s^*	Linearization point, voltage sum	6.711	V
λ^*	Linearization point, travel angle	0	rad

Table 3: Parameters and values.

Symbol	Parameter	Unit
e	Elevation angle	rad
\tilde{e}	Elevation angle, after coordination transform	rad
\tilde{e}_c	Elevation angle reference	rad
F_b	Force from back propeller	N
F_f	Force from front propeller	N
J	Moment of inertia	kgm ²
\mathbf{J}	Cost function for linear quadratic regulator	—
K_{pd}	Controller gain	—
K_{pp}	Controller gain	—
K_{rd}	Controller gain	—
\mathbf{K}	Gain matrix for linear quadratic regulator	—
\mathbf{L}	Gain matrix for linear observer	—
M	Torque	Nm
p	Pitch angle	rad
\tilde{p}	Pitch angle, after coordination transform	rad
\tilde{p}_c	Pitch angle reference	rad
\mathbf{P}	Gain matrix	—
\mathbf{Q}	Weighting matrix for linear quadratic regulator	—
\mathbf{r}	Reference vector	—
\mathbf{R}	Weighting matrix for linear quadratic regulator	—
t	Time	s
\mathbf{u}	Input vector	—
V_b	Voltage back motor	V
V_d	Voltage difference	V
\tilde{V}_d	Voltage difference, after coordinate transform	V
V_f	Voltage front motor	V
V_s	Voltage sum	V
\tilde{V}_s	Voltage sum, after coordinate transform	V
\mathbf{x}	State vector	—
\mathbf{y}	Output vector	—
λ	Travel angle	rad
$\tilde{\lambda}$	Travel angle, after coordination transform	rad
$\dot{\lambda}_c$	Elevation rate reference	rad/s

B Matlab code snippets

B.1 Pole placement

```
1 r = 5;
2 theta = pi/3;
3
4 i = 1;
5
6 for p = (pi-theta:(2*theta)/5:pi+theta)
7     poles(i)=r*exp(p*1j);
8     i = i + 1;
9 end
10 poles = cplxpair(poles)
11
12 figure(8);
13 plot(real(poles),imag(poles), 'o');
14 hold on;
15 plot(real(sys_pol),imag(sys_pol), '+');
16 grid on;
17 axis equal;
18 hold off;
19 ylim([-30, 30])
20
21 L = place(A_hat', C_hat', poles)'
```

B.2 Observability

```
1 O = obsv(A_hat, C_hat);
2 rank(O);
```

References

- [1] Steve Brunton. *Control Bootcamp: Controllability and the PBH Test*. Youtube.
URL: <https://youtu.be/0XJHgLrcPeA>.
- [2] Chi-Tsong Chen. *Linear System Theory and Design*. Oxford University Press, Incorporated, 2014.
- [3] Department of Engineering Cybernetics. “Helicopter Lab Assignment”. 2015.