TTK4115 – Linear System Theory

Group 3
Henrik Dobbe Flemmen - 477564
Jan Fijalkowski - 478412
Kevin Kaldvansvik - 478460

October, 2018

Helicopter lab assignment

Department of Engineering Cybernetics

Abstract

This report is dedicated to the lab in TTK4115, Linear System Theory. We have used Simulink, MATLAB and QuaRC to simulate and control a helicopter.

Contents

1 Introduction

This report is divided into four parts. In the first part we will derive a non-linear model of the helicopter and linearize it. In the second part we derive and apply a mono- variable control for pitch and travel, and we will discuss how we designed the P and PD controller used to control the system outputs. In the third part we controlled several system outputs simultaneously using a multi-variable control, also referred to in this report as a linear quadric regulator (LQR). In the last part we developed an observer, and used it to estimate the non measured states.

2 Part I - Mathematical modeling

In this part we are introduced to the helicopter, and we want to make a mathematical model, linearize it, and implement it in Simulink.

2.1 Problem 1

To study and control our system we have to make a mathematical model. Our helicopter has three theoretical point masses. The two points, m_p, are equal and they will be representing the two motors. They are connected to the propellers and placed symmetrically in relation to the cylindrical pitch axis and connected to two rods. The two rods have a length of l_p, and connected perpendicular to the horizontal rod with length l_h from the elevation axis. The point mass m_c is connected with the length l_c from the elevation axis, and represents the counterweight. The cylindrical elevation joint is connected perpendicular to to a rod which is connected to the travel joint. In our model p denotes the pitch angle of the helicopter, e denotes the elevation angle of the head, and λ denotes the travel angle of the helicopter (See fig. 1). In fig. 1 all joint angles are zero.

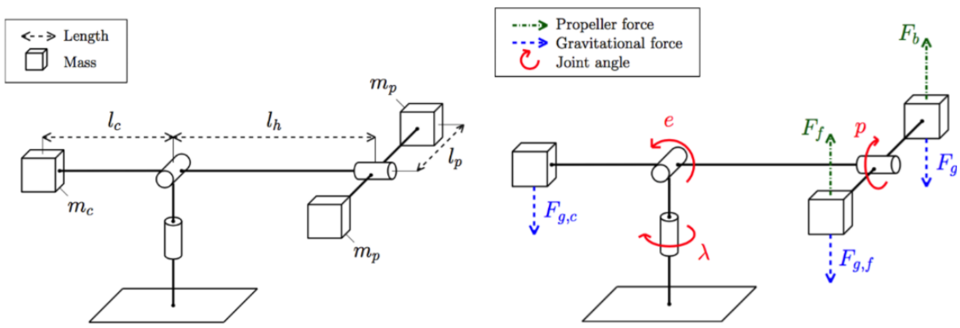


Figure 1: The helicopter model with the masses, distanced and forces.

In our mathematical model we have assumed that there is a linear relation between the supplied voltages V_b and V_f to the motors, and the forces generated by the propellers which yields

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

where K_f is the motor force constant.

To describe the rigid body motion of the helicopter we need to derive a set of equations. First we look at the helicopters ability to manoeuvre.

The lift is determined by the sum of the forces F_f and F_b. Its ability to rotate back and forth is determined by the difference between F_f and F_b. If we instead use the voltages V_f and V_b to describe the helicopters ability to move we get:

$$V_s = V_f + V_b \tag{2a}$$
$$V_d = V_f - V_b \tag{2b}$$

where V_s describes the voltage used for lift, and V_d describes the voltage for rotation. Now we can look at Newtons second law of rotation

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

where the sum of τ describes the net torque, or the angular acceleration about a single principle axis. Now if we assume that the moments of inertia are constant we have

$$J_p = 2m_p{l_p}^2 \tag{4a}$$
$$J_e = m_c{l_c}^2 + 2m_p{l_h}^2 \tag{4b}$$
$$J_\lambda = m_c{l_c}^2 + 2m_p({l_h}^2 + {l_p}^2) \tag{4c}$$

If we use the properties we described earlier to find a mathematical model for our helicopter we get the equation

$$\sum \tau = F_f l_p - F_b l_b = l_p K_f (F_f - F_b) \tag{5a}$$
$$J_p \ddot{p} = l_p K_f V_d \tag{5b}$$

for motion around the pitch. In order to have a lift, the force from the rotors have to be greater than the gravitational pull. While the gravitational force will always point down, the force given from the rotors will be dependent on the elevation angle and the pitch angle. In order find net torque around the elevation axis we simply balance the torque around the axis and get

$$\sum \tau = [F_{g,c} l_c - (F_{g,f} + F_{g,b}) l_h] \cos e + (F_f - F_b) l_h \cos p \tag{6a}$$
$$J_e \ddot{e} = g(m_c l_c - 2m_p l_h) \cos e + K_f V_s l_h \cos p \tag{6b}$$

where $F_{g,c}$ is the gravitational pull on m_c and $F_{g,f}$ and $F_{g,b}$ is the gravitational pull on m_p respectively. The travel angel is defined around the zaxis and as we see in fig. 1 that the travel angel is dependent on both the pitch and elevation. If we decompose the forces in the horizontal direction we get that the net torque around the z- axis has to be

$$\sum \tau = (F_f + F_b) l_h \cos e \sin(-p) \tag{7a}$$

$$J_\lambda \ddot{\lambda} = -K_f V_s l_h \cos e \sin p \tag{7b}$$

If we now substitute the constants in eq. (5b), eq. (6b) and eq. (7b) with

$$L_1 = l_p K_f \tag{8a}$$

$$L_2 = g(m_c l_c - 2m_p l_h) \tag{8b}$$

$$L_3 = K_f l_h \tag{8c}$$

$$L_4 = -K_f l_h \tag{8d}$$

we get the equations

$$J_p \ddot{p} = L_1 V_d \tag{9a}$$

$$J_e \ddot{e} = L_2 \cos(e) + L_3 V_s \cos(p) \tag{9b}$$

$$J_\lambda \ddot{\lambda} = L_4 V_s \cos(e) \sin(p) \tag{9c}$$

2.2 Problem 2

Now as we have our mathematical model eq. (9), we can linearize the equations of motion around the point $(p, e, \lambda)^T = (p^*, e^*, \lambda^*)^T$, with $p^* = e^* = \lambda^* = 0$. In order to do so we first need to determine V_s^* and V_d^*, such that for all time $(\dot{p}, \dot{e}, \dot{\lambda})^T = (0,0,0)^T$ and $(V_s, V_d)^T = ({V_s}^*, {V_d}^*)^T$ which will imply that $(\ddot{p}, \ddot{e}, \ddot{\lambda})^T = (0,0,0)^T$. Now we can find ${V_d}^*$ simply by using eq. (9a)

$$0 = L_1 {V_d}^* \Rightarrow {V_d}^* = 0 \tag{10}$$

In order to find ${V_s}^*$ we will use eq. (9b)

$$0 = L_2 \cos 0 + L_2 {V_d}^* \cos 0 \Rightarrow {V_s}^* = -\frac{L_2}{L_3} \tag{11}$$

Our zero condition matrix yields

$$\begin{bmatrix} V_s^* \\ V_d^* \end{bmatrix} = \begin{bmatrix} -\dfrac{L_2}{L_3} \\ 0 \end{bmatrix} \tag{12}$$

Now we can write the coordinate transformation as

$$\begin{bmatrix} V_s \\ V_d \end{bmatrix} = \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} + \begin{bmatrix} V_s^* \\ V_d^* \end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix} p \\ e \\ \lambda \end{bmatrix} = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} + \begin{bmatrix} p^* \\ e^* \\ \lambda^* \end{bmatrix} \tag{13}$$

This transformation simplifies the further analysis, as the equilibrium point of this new coordinate system is located at the origin. Now we can rewrite eq. (9) and set in eq. (13) which yields

$$\ddot{\tilde{p}} = \frac{L_1}{J_p}\tilde{V}_d \tag{14a}$$

$$\ddot{\tilde{e}} = \frac{L_2}{J_e}\cos(\tilde{e}) + \frac{L_3}{J_e}[\tilde{V}_s - \frac{L_2}{L_3}]\cos(\tilde{p}) \tag{14b}$$

$$\ddot{\tilde{\lambda}} = \frac{L_4}{J_\lambda}[\tilde{V}_s - \frac{L_2}{L_3}]\cos(\tilde{e})\sin(\tilde{p}) \tag{14c}$$

The transformed system $\mathbf{f}(\mathbf{x}, \mathbf{u}) = (\ddot{\tilde{p}}, \ddot{\tilde{e}}, \ddot{\tilde{\lambda}})^T$ can be written as a linearized state- space equation:

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\tilde{\mathbf{u}} \tag{15}$$

where we have

$$A = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\bigg|_{\mathbf{x_0},\mathbf{u_0}} \qquad \text{and} \qquad B = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\bigg|_{\mathbf{x_0},\mathbf{u_0}} \tag{16}$$

where $\mathbf{x_0} = (p^*, e^*, \lambda^*)^T$ and $\mathbf{u_0}$ equals the zero condition matrix eq. (12). By setting eq. (16) into eq. (15) we have

$$\begin{bmatrix} \ddot{\tilde{p}} \\ \ddot{\tilde{e}} \\ \ddot{\tilde{\lambda}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\frac{L_4}{J_\lambda}\frac{L_2}{L_3} & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} + \begin{bmatrix} 0 & \frac{L_1}{J_p} \\ \frac{L_3}{J_e} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{17}$$

We can now see that the linearized equations of motion can be written as

$$\ddot{\tilde{p}} = \frac{L_1}{J_p}\tilde{V}_d = K_1\tilde{V}_d \tag{18a}$$

$$\ddot{\tilde{e}} = \frac{L_3}{J_e}\tilde{V}_s = K_2\tilde{V}_s \tag{18b}$$

$$\ddot{\tilde{\lambda}} = -\frac{L_4}{J_\lambda}\frac{L_2}{L_3}\tilde{p} = K_3\tilde{p} \tag{18c}$$

2.3 Problem 3

Comparing the physical behaviour of the system with the theoretical models in eq. (9) and eq. (18) it is clear that the linearized equations of motion are extremely simplified around the linearized equilibrium. Information regarding elevation acceleration being dependent on the elevation angle and the pitch angle is lost. The same applies to the travel acceleration being dependent on the elevation. The discrepancies can therefore be explained by that larger deviations from the equilibrium angles will result in larger deviations from the physical model.

In addition to this, we have done a lot of assumptions in our mathematical model of the helicopter, and when we compare the model eq. (9) with the physical behaviour of the helicopter we clearly see some deviation in the response. Our model assumes that there is no friction, vibrations due to rotation of the rotors, ground effect, time delays and aerodynamic effects.

Controlling the system by only using feed forward was therefore hard. We have previously discussed that we want a proportional response in pitch and elevation, with the voltage applied eq. (18) to the system. To realise this we carefully chose the gain from the x and y output to have a good response when using the joystick, used to control the helicopter.

2.4 Problem 4

In order to implement a controller that is based on eq. (9) of motion, we will have to find the motor force constant K_f. First we obtained the $V_s^* = 6{,}5$ V by measurement in the helicopters equilibrium state fig. 1. We can use the eq. (6), and set the values which makes the helicopter continuously maintain the equilibrium point. The initial values are $p = 0$, $\lambda = 0$ and $e = 0$, which means that $\ddot{e} = 0$. And we will have

$$J_e\ddot{e} = 0 = g(m_c l_c - 2m_p l_h)\cos 0 + K_f V_s l_h \cos 0 \tag{19a}$$

$$K_f = -\frac{g(m_c l_c - 2m_p l_h)}{V_s l_h} \tag{19b}$$

$$K_f \approx 0.154 \tag{19c}$$

We can now also find the value of K_1, K_2 and K_3 in eq. (18)

$$K_1 = \frac{L_1}{J_p} = 0.611 \tag{20a}$$

$$K_2 = \frac{L_3}{J_e} = 0.0983 \tag{20b}$$

$$K_3 = -\frac{L_4}{J_\lambda}\frac{L_2}{L_3} = 0.6117 \tag{20c}$$

3 Part II - Mono-variable control

In this part we will implement a PD controller for the pitch angle $\tilde{p}$ and a P controller for the travel rate $\dot{\tilde{\lambda}}$.

3.1 Problem 1

The pitch angle p can be controlled by the PD controller in eq. (21) where the values for K_{pp} and K_{pd} need to satisfy $K_{pp}, K_{pd} > 0$.

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

inserting the equation for the PD controller in eq. (21) in to the equation for the pitch dynamics in eq. (18a) yields the equation

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

Taking the Laplace transform of eq. (22) and finding the transfer function from the pitch reference $\tilde{p_c}$ to the pitch angle $\tilde{p}$ yields

$$\frac{\tilde{p}}{\tilde{p_c}}(s) = \frac{K_1 K_{pp}}{s^2 + K_1 K_{pd} s + K_1 K_{pp}} \tag{23}$$

The values K_{pp} and K_{pd} in eq. (23) clearly effects the eigenvalues of the system. When tuning the PD controller we would like the output to rapidly follow the reference pitch, as well as not giving rise to excessive oscillations. Choosing K_{pp} and K_{pd} in a way that keeps the eigenvalues of the closed loop system real should keep the system from oscillating.

Recalling from Trond Andresens lectures we know that a second order transfer function $H(s)$ can be written as

$$H(s) = \frac{K\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{24}$$

Comparing eq. (23) and eq. (24) we get the following equations for the natural frequency ω_0 and the relative damping constant ζ

$$\omega_0 = \sqrt{K_1 K_{pp}} \tag{25}$$

$$\zeta = \frac{K_1 K_{pd}}{2\omega_0} \tag{26}$$

If we want no oscillations we need to choose $\zeta \geq 1$. Since we want the fastest possible system that does not oscillate we select the relative damping constant ζ to be equal to 1. This makes the values for K_{pp} and K_{pd} in the equations 25 and 26 solely depend on the selected natural frequency ω_0.

In Table 1 the resulting PD parameters and eigenvalues are given with varying natural frequency. From this table it is clear that an increasing ω_0 with a critically damped system yields increasing values for the PD parameters. This means that the more negative eigenvalues the faster the system.

Table 1: Varying the ω_0 with the resulting PD parameters and closed loop eigenvalues

ω_0	K_{pp}	K_{pd}	Eigenvalues
$\frac{\pi}{3}$	1.80	3.43	-1.05 , -1.05
$\frac{\pi}{2}$	4.04	5.14	-1.57 , -1.57
$\frac{3\pi}{4}$	9.08	7.71	-2.36 , -2.36
π	16.15	10.28	-3.14 , -3.14

The PD controller was implemented in Simulink as seen in fig. 13. Testing the different values for ω_0 for the helicopter we noticed that the helicopter struggled to maintain the reference pitch and a stationary deviation was more visible for low ω_0. To get more qualitative data for the different values of natural frequency we put a step response on the pitch reference and logged the measured pitch at the output. The step response where the reference pitch $\tilde{p}_c$ is set to 25° is plotted in figure 2.

From the plot, the hypothesis about a larger stationary deviation for low ω_0 seems to be correct. One can observe that the system overshoots the reference value and this indicates that the theoretical assumption that setting $\zeta = 1$ removes any oscillations does not hold for physical systems. The resulting physical system is a under damped system with a single overshoot before the system decays and a stationary deviation builds up.

When testing smaller values for the pitch reference on the system for a fixed value of natural frequency it appeared as if the system could keep track of $\tilde{p}_c$ with small oscillations. This is show in Figure 3 with $\tilde{p}_c$ set equal to 10°.

Controlling the helicopter with the joystick is much easier now that we included a PD controller for the pitch. However, a stationary deviation builds up that makes the helicopter unable to maintain the desired pitch. Including an integrator in the system, that is using a PID instead of a PD controller seems like a good idea to get rid of the stationary deviation.

3.2 Problem 2

The travel rate $\dot{\lambda}$ can be controlled by the P controller in equation 27 where $K_{rp} < 0$. Adding the P controller to the Simulink diagram we got fig. 12.

$$\tilde{p}_c = K_{rp}(\dot{\tilde{\lambda}}_c - \dot{\tilde{\lambda}}) \tag{27}$$

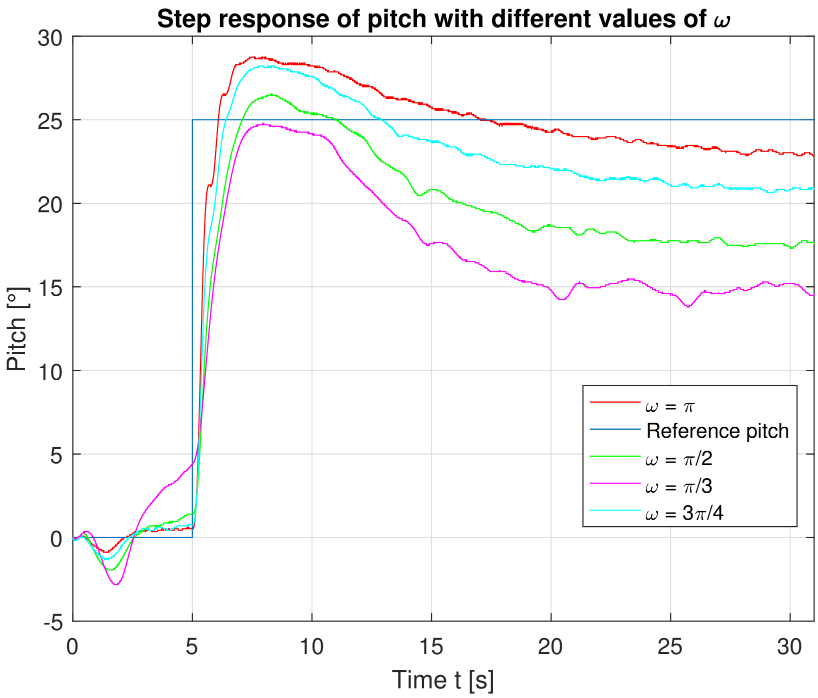


Figure 2: Pitch step response with varying natural frequency ω

Assuming that the pitch angle can be controlled perfectly with $\tilde{p} = \tilde{p}_c$, we can insert the P controller in eq. (27) in to the equation for the travel rate dynamics in eq. (18c) which yields the equation

$$\frac{\ddot{\tilde{\lambda}}}{K_3} = K_{rp}(\dot{\tilde{\lambda}}_c - \dot{\tilde{\lambda}}) \tag{28}$$

Taking the Laplace transform of eq. (28) and finding the transfer function from the travel rate reference $\dot{\tilde{\lambda}}_c$ to the resulting travel rate $\dot{\tilde{\lambda}}$ yields the equation

$$\frac{\dot{\tilde{\lambda}}(s)}{\dot{\tilde{\lambda}}_c(s)} = \frac{K_3 K_{rp}}{s + K_3 K_{rp}} \tag{29}$$

The P controller was implemented in Simulink as seen in fig. 14. Introducing the constant $\rho = K_3 K_{rp}$ we can write equation 29 on the form

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

Noting that the derived eq. (30) is a first order low pass filter with a cut-off frequency of ρ and a pole at $s = -\rho$ we can make the system faster

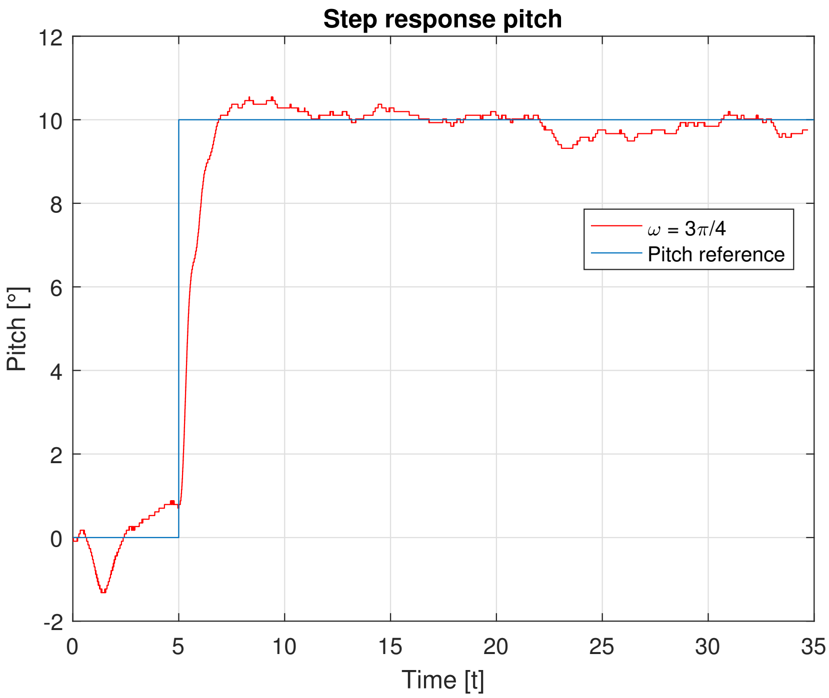


Figure 3: Pitch step response with natural frequency ω set to $\frac{3\pi}{4}$

by decreasing K_{rp} and thus decreasing the value of ρ.

We tried different values of K_{rp}, starting with -1 which happened to be the best value for the travel rate P-controller gain. Using K_{rp} = -2 made the controller give a too high pitch reference to the pitch controller and thus made the helicopter hit the table. Using K_{rp} = -1.5 made the helicopter occasionally hit the table. The result can be seen in figure 4, where we test different values of K_{rp} with a step response on the travel rate reference. We get a fast and accurate response with K_{rp} = -1 where the helicopter does not hit the table. One can also notice that there is a minimal overshoot of the travel rate before it converges to the reference travel rate, since we want a combination of a fast and accurate system, this overshoot is reasonable.

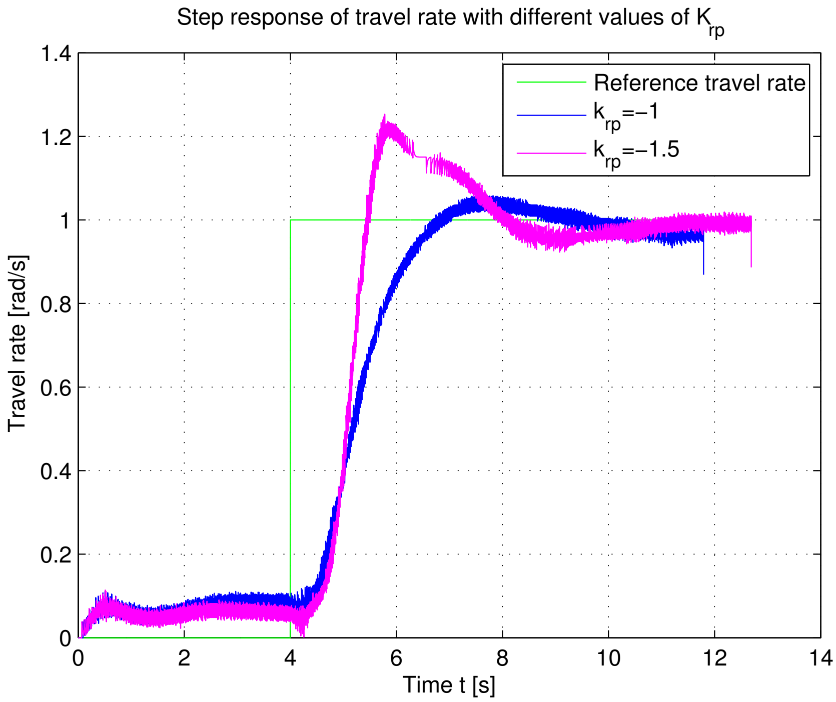


Figure 4: Travel rate step response with varying values of the P travel controller gain K_{rp}

4 Part III - Multivariable control

In this part we want to control the pitch angle $\tilde{p}$ and the elevation rate $\dot{\tilde{e}}$ with a multivariable controller using the joystick as reference for the pitch and elevation.

4.1 Problem 1

In order to control the system using state feedback we need to derive a state space model of our system on the form shown in eq. (31). The desired state vector $\mathbf{x}$ and input vector $\mathbf{u}$ is given in eq. (32)

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{31}$$

$$\mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \end{bmatrix} \qquad \text{and} \qquad \mathbf{u} = \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{32}$$

From the differential equations in eq. (18) we can see that our system becomes

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \end{bmatrix} \mathbf{u} \tag{33}$$

Thus $\mathbf{A}$ and $\mathbf{B}$ is given by

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \text{and} \qquad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \end{bmatrix} \tag{34}$$

We have now found the model that we will use for our system. Next we will control the system by determining the input vector $\mathbf{u}$.

4.2 Problem 2

We will now implement a LQR-controller for our system to track the reference $\mathbf{r}$ given in eq. (37). Firstly, we examine the controlability of the system to check if it is possible to control the states. We therefore calculate the controlability matrix:

$$\mathcal{C} = \begin{bmatrix} \mathbf{B} & \mathbf{A}\mathbf{B} & \mathbf{A}^2\mathbf{B} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & K_1 & 0 & 0 \\ 0 & K_1 & 0 & 0 & 0 & 0 \\ K_2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{35}$$

It is trivial to see that the matrix in eq. (35) has full rank, which again implies that the system is in fact controllable.

Using the LQR for state feedback as well as a feed forward from the reference yields the input vector in

$$\mathbf{u} = \mathbf{Pr} - \mathbf{Kx} \tag{36}$$

where $\mathbf{P}$ are defining the feed forward matrix gain and $\mathbf{K}$ the feedback matrix gain. $\mathbf{r}$ is our reference and are defined as

$$\mathbf{r} = \begin{bmatrix} \tilde{p}_c \\ \dot{\tilde{e}}_c \end{bmatrix} \tag{37}$$

We will use an optimal controller, the Linear Quadratic Controller(LQR). The implementation of the LQR in Simulink can be seen in fig. 15 and fig. 16. The LQR provides us with the $\mathbf{K}$ that minimises the cost function

$$J = \int_0^\infty \mathbf{y}^T(t)\mathbf{Q}\mathbf{y}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)dt \tag{38}$$

where J is the cost for the given set of parameters. To calculate the exact $\mathbf{K}$ we utilised the MATLAB command *lqr* which gave us the $\mathbf{K}$ gain matrix. We want to determine $\mathbf{P}$ such that the state converges on the reference given infinite time. When the states converges to a fixed value, the derivatives of the states converges to $\mathbf{0}$. Therefore we can set eq. (31) equal to zero. During the same conditions, we want the output $\mathbf{y}$ to converge to the reference $\mathbf{r}$. Under static conditions will therefore

$$\mathbf{r} = \mathbf{y} = \mathbf{Cx} \tag{39}$$

where $\mathbf{C}$ is the output matrix

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{40}$$

Inserting eq. (36) into eq. (31) and setting $\dot{\mathbf{x}} = 0$ gives us:

$$\mathbf{0} = \mathbf{Ax} + \mathbf{B}(\mathbf{Pr} - \mathbf{Kx}) = \mathbf{Ax} + \mathbf{BPCx} - \mathbf{BKx} = (\mathbf{A} + \mathbf{BPC} - \mathbf{BK})\mathbf{x} \tag{41}$$

Since we can not generally assume that $\mathbf{x}$ is zero, we must choose $\mathbf{P}$ such that the other factor is zero, namely that:

$$\mathbf{A} + \mathbf{BPC} - \mathbf{BK} = \mathbf{0} \tag{42}$$

which again implies that

$$\mathbf{BPC} = \mathbf{BK} - \mathbf{A} \tag{43}$$

which leads us to the conclusion that the feed forward gain matrix $\mathbf{P}$ is given by

$$\mathbf{P} = \mathbf{B}^{-1}(\mathbf{BK} - \mathbf{A})\mathbf{C}^{-1} = (\mathbf{C}(\mathbf{BK} - \mathbf{A})^{-1}\mathbf{B})^{-1} \tag{44}$$

4.2.1 Tuning the controller

As we know $\mathbf{K}$, and thereby $\mathbf{P}$, depends on two cost matrices $\mathbf{Q}$ and $\mathbf{R}$. $\mathbf{Q}$ assigns an associated cost with error between the actual state and the reference and $\mathbf{R}$ assigns a cost associated with the use of gain. The LQR tries to minimise the total cost J of the system with the decided weights from $\mathbf{Q}$ and $\mathbf{R}$ and it is our job to choose the $\mathbf{Q}$ and $\mathbf{R}$ so that the system acts reasonably. Due to our incomplete model, limited gain and nonlinearities(saturation, vibrations) we can not just increase the cost of everything and make a perfect behaving system. In fact the cost is relative to the cost of the other parameters, so increasing one is equivalent to decreasing all the others. To tune the $\mathbf{Q}$ and $\mathbf{R}$ matrix, we started out with setting them to the identity matrix and experimented on the system altering one value at the time. First we used step functions to change the reference, which enabled us to make comparable responses from each attempt to the next. After utilising this approach to get a nice response we tried to control it using the joystick. With the joystick it felt somewhat slow. We therefore continued to tune it using the joystick as reference. Using the joystick instead of the step functions gives us a better impression of whether the helicopter is fast and accurate, but reduces the ability to compare subsequent attempts.

We afterwards tried to adjust the $\mathbf{R}$ and $\mathbf{Q}$ matrices to impose a higher cost for error in pitch, and lower cost for the gain V_d. In the end we ended up with the response fig. 5, which we accepted as the best we could achieve without an integrator.

Our final $\mathbf{R}$ and $\mathbf{Q}$ matrices are given by

$$\mathbf{R} = \begin{bmatrix} 0.1 & 0 \\ 0 & 1 \end{bmatrix} \tag{45a}$$

$$\mathbf{Q} = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 30 & 0 \\ 0 & 0 & 1000 \end{bmatrix} \tag{45b}$$

Using the MATLAB command *lqr(A,B,Q,R)* with the obtained $\mathbf{Q}$ and $\mathbf{R}$ gave the following feedback matrix $\mathbf{K}$

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 100 \\ 10 & 7.92 & 0 \end{bmatrix} \tag{46}$$

Using the formula for the feed forward matrix $\mathbf{P}$ in eq. (44) yields

$$\mathbf{P} = \begin{bmatrix} 0 & 100 \\ 10 & 0 \end{bmatrix} \tag{47}$$

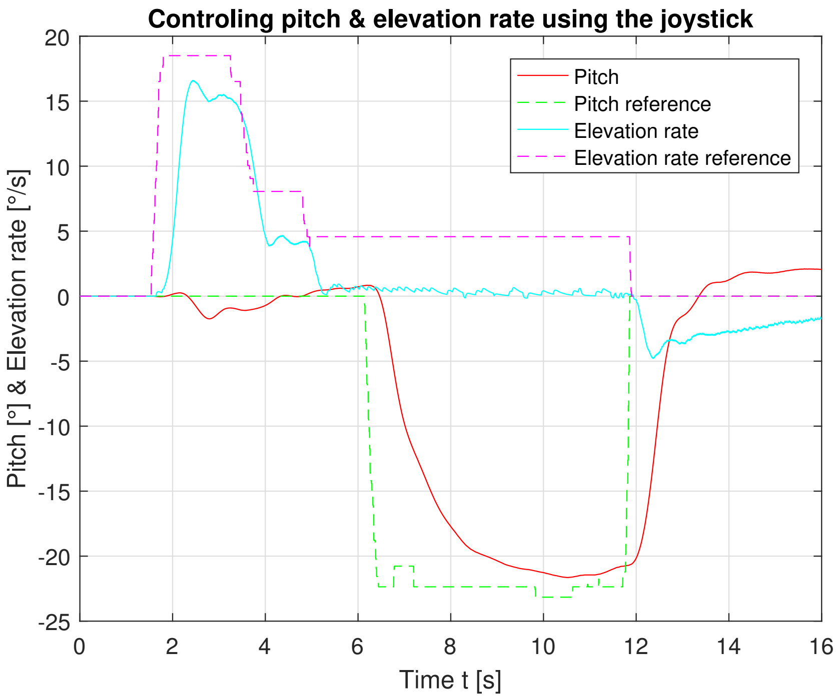


Figure 5: Controling the pitch and elevation rate using the LQR with the joystick as reference to the pitch and elevation rate.

4.3 Problem 3

We will now add an integral effect on the elevation rate and the pitch angle. We do this to eliminate the deviation from the reference and the actual state in static or slowly moving situations. To implement the integral effect, we modify our system by adding two new states defined by eq. (48).

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

If we denote the vector with the two new states as $\mathbf{x_i}$, we can express eq. (48) as a vector equation:

$$\dot{\mathbf{x}}_\mathbf{i} = \mathbf{C}\mathbf{x} - \mathbf{r} \tag{49}$$

Here $\mathbf{C}$ is the output matrix from eq. (40) and $\mathbf{r}$ is still the reference from eq. (37).

If we denote all our old vectors and matrices with subscript zero, we can use these to define the new system. Our new state vector $\mathbf{x}$ changes to:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x_0} \\ \mathbf{x_i} \end{bmatrix} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \\ \gamma \\ \zeta \end{bmatrix} \tag{50}$$

Then our system will be given by a sum of the equations eq. (49) and eq. (9):

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{x}}_\mathbf{0} \\ \dot{\mathbf{x}}_\mathbf{i} \end{bmatrix} = \begin{bmatrix} \mathbf{A_0} & \mathbf{0} \\ \mathbf{C_0} & \mathbf{0} \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{B_0} \\ \mathbf{0} \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{0} \\ -\mathbb{I} \end{bmatrix} \mathbf{r} \tag{51}$$

This implies that our new $\mathbf{A}$ and $\mathbf{B}$ is given as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_0} & \mathbf{0} \\ \mathbf{C_0} & \mathbf{0} \end{bmatrix} \tag{52a}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{B_0} \\ \mathbf{0} \end{bmatrix} \tag{52b}$$

$\mathbf{0}$ is a matrix width only zero entries with dimension 2x2 or 3x2 depending on the size of the expected block matrix.

We will use the same $\mathbf{P}$ matrix that we used in our system without the integral effect. With integral effect the states will converge on the reference independent of the feed forward.

4.3.1 Tuning the controller

After adding the integral effect to the Simulink fig. 17 without any further tuning we got the result shown in fig. 6.

We can see that the exact pitch struggles to keep up with the reference, which it should accomplish when utilising integral effect. As without integral effect it proved difficult to tune only with step responses. We therefore started to tune with the joystick as reference, which gave us a better ability to determine if the system was fast and precise. After some tuning, we got fig. 7.

We can see from fig. 7 that the states follows the respective states rather closely. This is a result of the integrator effect. We have eliminated the large constant deviation from the reference that we had without integral effect. In other words, we can say that the system is more precise with the integrator. The increase in accuracy have seemingly not come at the expense of the speed of the helicopter. Thus after a step in the reference, it requires roughly the same time to reach the new reference.

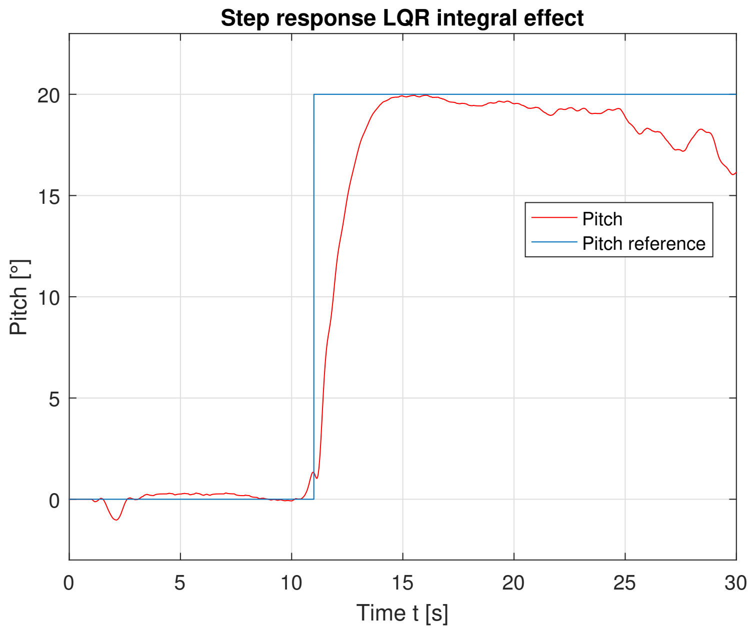


Figure 6: Initial response using the LQR with a unit step response on pitch.

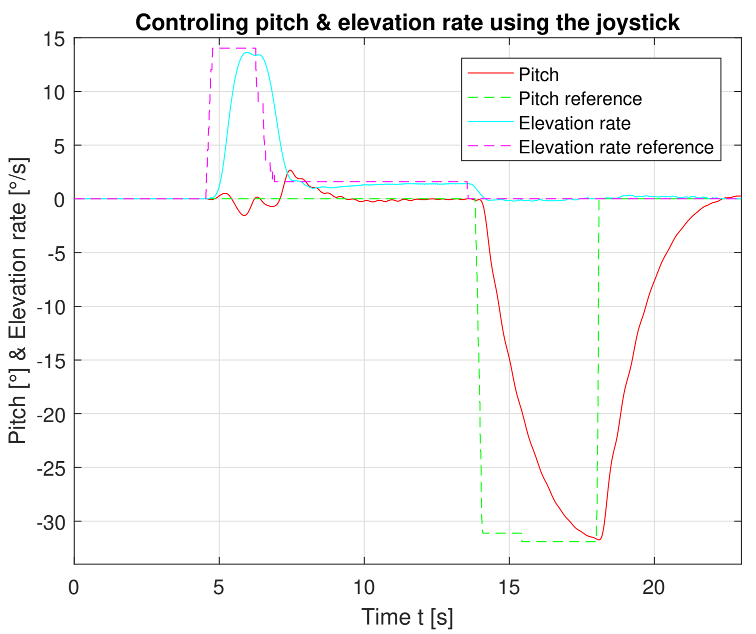


Figure 7: Tuned LQR with integrator effect with the joystick as reference on pitch and elevation rate

5 Part IV - State estimation

In the previous tasks, the angular velocities corresponding to these angles have been computed using numerical differentiation. In this part of the assignment, an observer will be developed in order to estimate these non measured states instead.

5.1 Problem 1

The state vector $\mathbf{x}$ should now also include the states we are trying to estimate. The output vector $\mathbf{y}$ should include the values that we measure. The new state vector, the measurement vector $\mathbf{y}$ and the input vector is now given as

$$\mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} \quad , \quad \mathbf{y} = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} \quad \text{and} \quad \mathbf{u} = \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{53}$$

Since we have changed the state vector, we need to find the new state space model given as

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \end{aligned} \tag{54}$$

Using the linearized equations of motion in eq. (18) the state space model takes the following form

$$\begin{aligned} \dot{\mathbf{x}} = \begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \dot{\tilde{e}} \\ \ddot{\tilde{e}} \\ \dot{\tilde{\lambda}} \\ \ddot{\tilde{\lambda}} \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ K_3 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ 0 & 0 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \\ \mathbf{y} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x} \end{aligned} \tag{55}$$

5.2 Problem 2

The observability matrix $\mathcal{O}$ for a system with n states is given by

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ . \\ . \\ \mathbf{CA}^{n-1} \end{bmatrix} \tag{56}$$

In our new state space model we have 6 states, which means that we have to compute $\mathbf{A}^5$ in order to check if the observability matrix has full column rank. That would result in an observable system where all the states are available through linear combinations of the output vector $\mathbf{y}$.

Using the MATLAB function $obsv(\mathbf{A}, \mathbf{C})$ we get the observability matrix $\mathcal{O}$. The rank can be checked using the function $rank(\mathcal{O})$. Putting in our new $\mathbf{A}$ and $\mathbf{C}$ matrix from eq. (55) we get the following observability matrix

$$\mathcal{O} = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ K_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{57}$$

which clearly has full column rank.

Since the system is observable we can reconstruct the states within the system from the measurements $\mathbf{y}$ by using the linear observer

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

This equation introduces a new observer matrix gain $\mathbf{L}$ which we can manipulate to place the closed loop observer poles. The equation can be rearranged in the following way

$$\dot{\hat{\mathbf{x}}} = (\mathbf{A} - \mathbf{L}\mathbf{C})\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{L}\mathbf{y} \tag{59}$$

and was implemented in Simulink as seen in fig. 19. We finally got the Simulink diagram seen in fig. 18. All the eigenvalues in $\mathbf{A} - \mathbf{LC}$ can be chosen arbitrarily by selecting $\mathbf{L}$ such that $\hat{\mathbf{x}}(t) \to \mathbf{x}(t)$ with a desired convergence rate. If all eigenvalues has real parts smaller than $-\sigma$, all the entries in $\hat{\mathbf{x}}$ will approach the actual values of $\mathbf{x}$ at a rate faster than $e^{-\sigma t}$[1, p.302]. The selection of eigenvalues in $\mathbf{A} - \mathbf{LC}$ does not affect the state feedback in any way.

As a rule of thumb the eigenvalues should be placed evenly along a circle inside the left half plane. The estimator eigenvalues should be faster than the feedback eigenvalues to make sure that the estimated states does not have a delay compared to the actual states. Too fast estimator eigenvalues will introduce amplification of noise and saturation problems. A physical result of this, is that the output signal $\mathbf{u}$ can deliver too high values to the motors and also an oscillating signal. An oscillating signal will make the motors vibrate and make a lot of noise.

This problem can be eliminated by moving the eigenvalues towards the origin. If one fails to remove the noise and the estimator eigenvalues gets slower than the feedback eigenvalues, one can reduce the eigenvalues in the state feedback. This should make room for reducing the estimator eigenvalues even further, and thus removing the noise problem.

Keeping in mind that the fastest eigenvalues in the state feedback with and without integrator being respectively -12.22 and -9.83, we started by placing the estimator eigenvalues evenly distributed along a circle in the left half plane with radius $\sigma = 30$. This gave rise to huge amplifications of noise and thus made the state estimations insufficient for controlling the helicopter.

Therefore we kept reducing the value of σ down to 20 which gave us the eigenvalues shown in figure 9a. These estimator eigenvalues suppressed the noise and made us able to control the helicopter using the regular LQR and the joystick as reference. The resulting estimators plotted against the measured states is shown in figure 8.

For the LQR with integrator effect, we kept reducing σ down to 18, but we were not satisfied by the response of the observer. Therefore we experimented by placing the poles evenly inside a sector shown in figure 9b with varying sector angle θ. Having a bigger θ results in a bigger overshoot and grouping the eigenvalues together usually makes the response slower [1, p.290].

Since we experienced some issues with overshoot we decided to place the eigenvalues closer together by choosing $\theta = 45°$. Slower eigenvalues means that the system works more like a low pass filter, filtering out the noise and

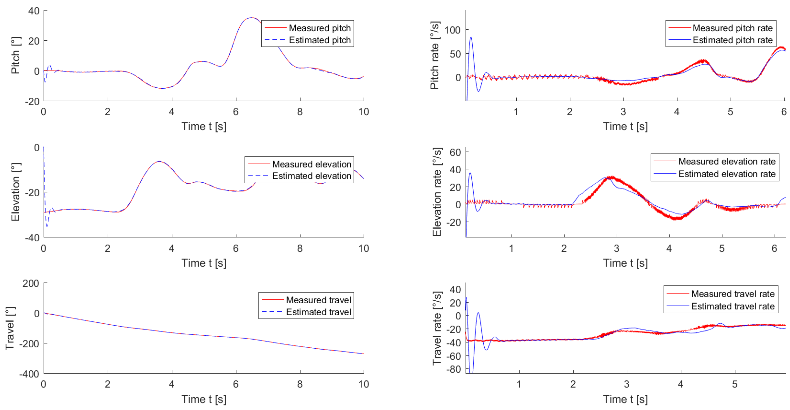
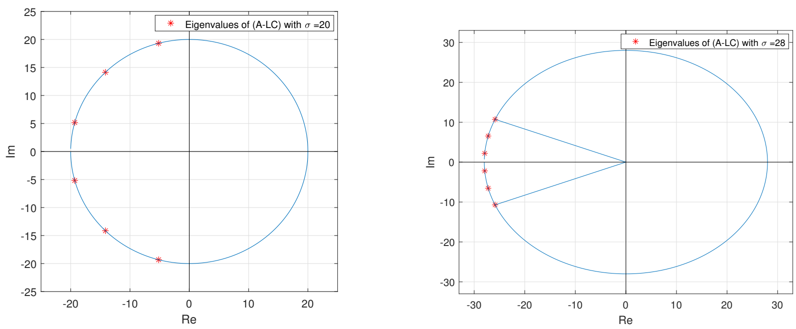


Figure 8: The estimated states plotted against the measured states using the LQR without integrator effect.



(a) Placement of observer eigenvalues using the LQR

(b) Placement of observer eigenvalues using the LQR with integrator effect

Figure 9: Placement linear observer eigenvalues for the two LQR controllers

follows the measured states more smoothly. The resulting eigenvalues that we found were the best for state estimation can be seen in figure 9b with $\sigma = 28$.

With these estimator eigenvalues we achieved the following response shown in figure 10. The pitch, elevation and travel angles seems to be estimated rather perfectly. The estimated pitch, elevation and travel rates follows the measured rates rather smoothly and does not show any signs of significant noise amplification, which was the goal of tuning the linear observer.

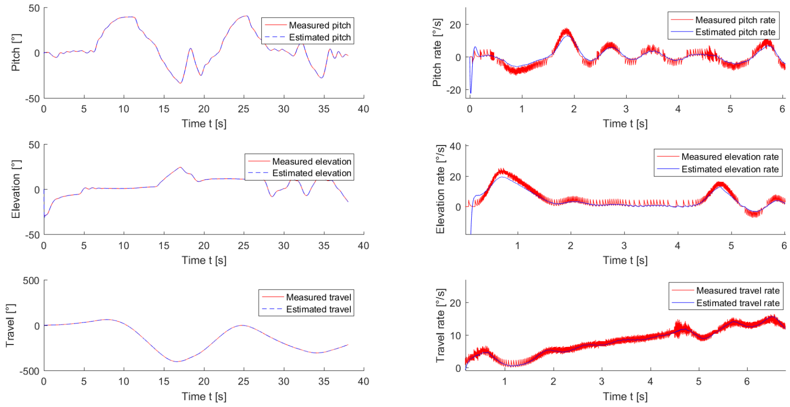


Figure 10: The estimated states plotted against the measured states using the LQR with integrator effect

5.3 Problem 3

We will now consider the possibility of controlling the system with only two measurements. We start by only measuring $\tilde{e}$ and $\tilde{p}$. If we change the definition of the output vector $\mathbf{y}$ from eq. (53) to eq. (60) we will get the new output matrix $\mathbf{C}$ given by eq. (61).

$$\mathbf{y} = \begin{bmatrix} \tilde{e} \\ \tilde{p} \end{bmatrix} \tag{60}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{61}$$

This will impose a new observatiblity matrix, given as

$$\mathcal{O} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{62}$$

which is clearly not full column rank. This means that the system is not observable when we only measure $\tilde{e}$ and $\tilde{p}$. If we instead were to measure $\tilde{e}$ and $\tilde{\lambda}$ the output vector $\mathbf{y}$ would be given as

$$\mathbf{y} = \begin{bmatrix} \tilde{e} \\ \tilde{\lambda} \end{bmatrix} \tag{63}$$

This would infer that the output matrix $\mathbf{C}$ would be given as

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{64}$$

The new observability matrix $\mathcal{O}$ will be given by

$$\mathcal{O} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ K_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{65}$$

Here we have $n = 6$ independent columns which indicates that the system is observable, and that we should be able to reconstruct all states with just these two measurements.

5.3.1 Implementing the new measurement restrictions

We created a new linear observer after the same pattern that we made the first one, but with the new measurement vector $\mathbf{y}$ and the new output matrix $\mathbf{C}$. This changed the $\mathbf{L}$ matrix and thus made it necessary to tune the observer eigenvalues once more. We embedded the new observer into the existing system, replacing the old observer. We then tried to run the system and tune it so that it would be fast and precise. This proved much harder than what we experienced with three measurements.

This can be explained using the linearized model in eq. (18). First off, one can notice that it is possible to obtain the pitch angle $\tilde{p}$ by differentiation of $\tilde{\lambda}$ twice using the linearized model. This makes the estimated pitch angle pick up and amplify any noise in the travel angle not once, but twice. The pitch rate could also be derived from differentiating the estimated pitch angle. One can therefore say that it is not strange that the estimated pitch angle and pitch rate is of low quality compared to the measured signal.

Since the pitch angle is related to the voltage difference $\tilde{V}_d$ it is reasonable to expect the helicopter tilting to 90° and then not being able to hover. This was also the case when we tried to control the helicopter using these estimated states. We tried to tune the estimators to make a best possible estimate of the travel and travel rate, giving a small hope about being able to correctly estimate the pitch and pitch rate. We were unfortunately not able to tune the estimators good enough to being able to control the helicopter. The final estimated states plotted against the measured states is shown in figure 11.

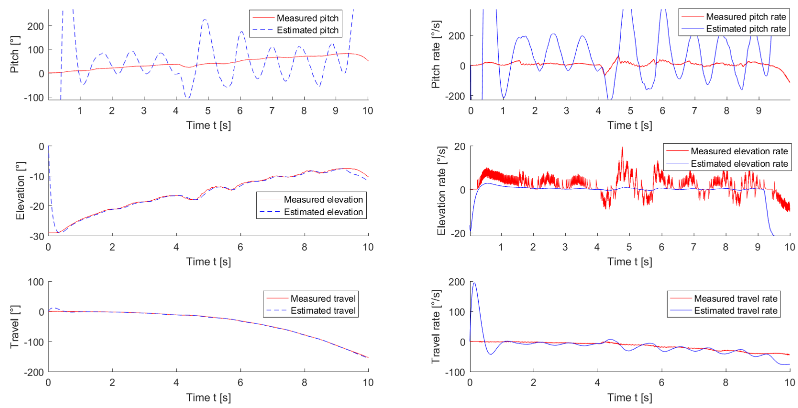


Figure 11: The estimated states and the measured states using the joystick as reference.

6 Conclusion

In this lab we have created a mathematical model of a helicopter and implemented a classical control system to regulate the system. We have also implemented a linear quadric regulator. Without an integral effect we had an constant deviation, but adding an integral effect we eliminated the constant deviation from the reference. Comparing the classical control system with the LQR, we can say that our system was most precise using the LQR with the integrator effect. In addition, we have also experienced that an estimator provides a better input value to the regulator than just differentiating the measured states.

Appendix

A Simulink Diagrams

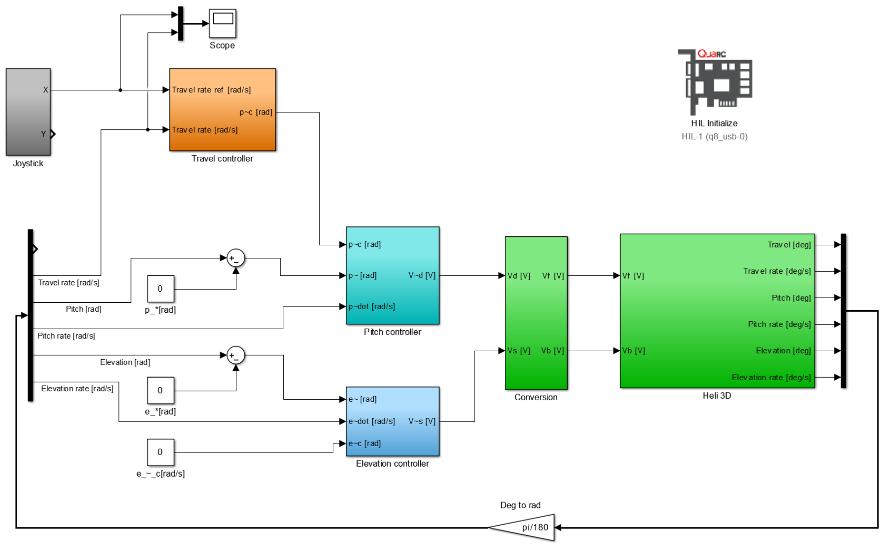


Figure 12: Simulink overview of the system with the implemented pitch and travel controller.

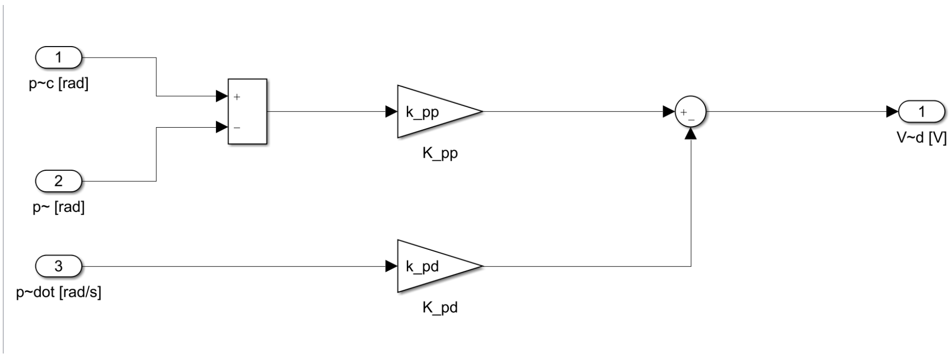


Figure 13: The implementation of the PD pitch controller.

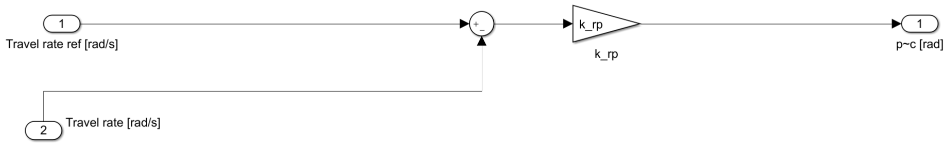


Figure 14: The implementation of the P travel controller.

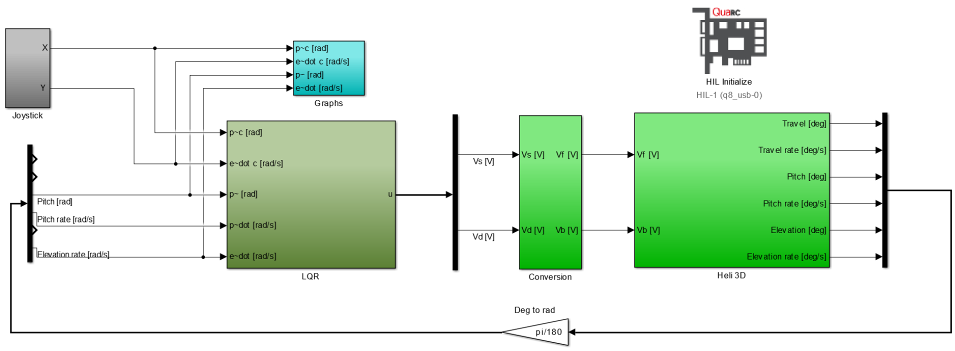


Figure 15: Simulink overview of the system with the implemented linear-quadratic regulator.

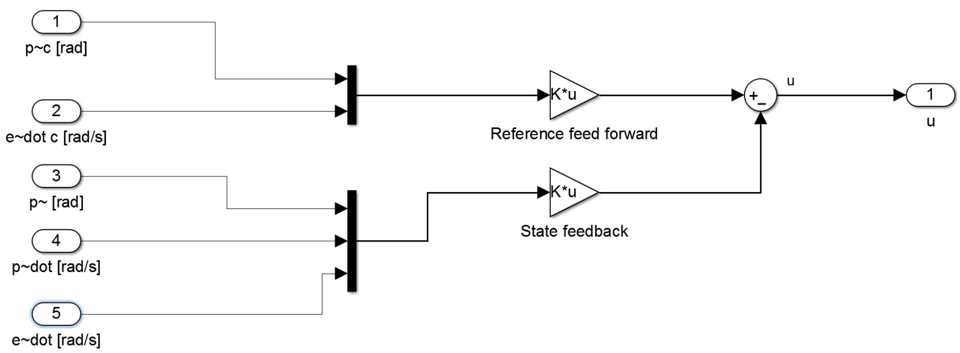


Figure 16: The implementation of the LQR.

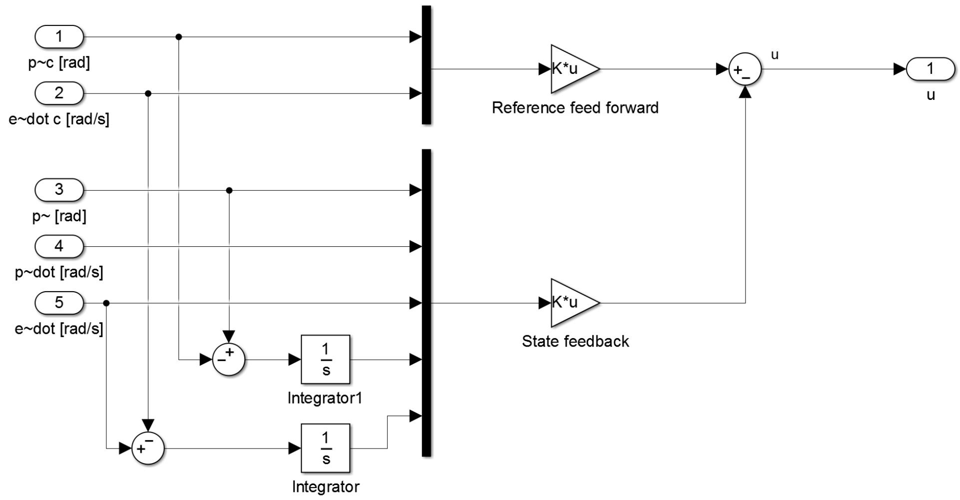


Figure 17: The implementation of the LQR with integrator effect.

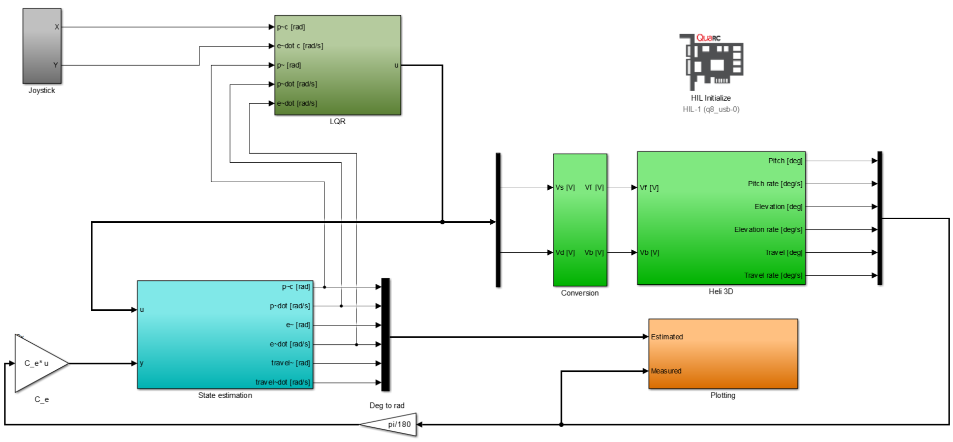


Figure 18: Simulink overview of the system with the implemented linear observer.

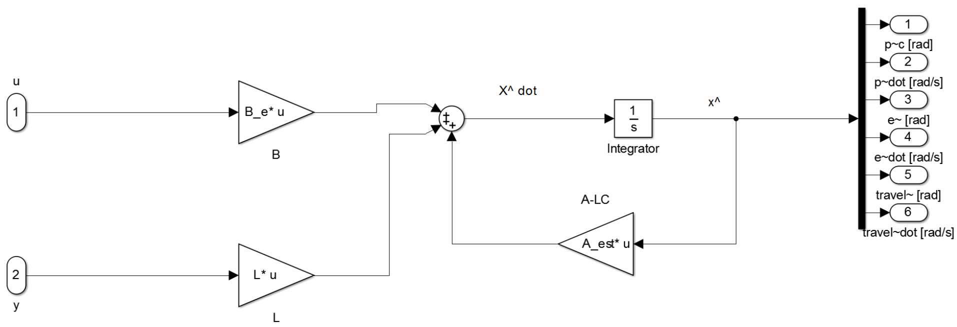


Figure 19: The implementation of the linear observer.

B 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	1.034	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	0.154	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	6.5	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	–
τ	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
$\dot{\tilde{\lambda}}_c$	Elevation rate reference	rad/s
$\mathbf{A}$	State matrix	–
$\mathbf{B}$	Input matrix	–
$\mathbf{C}$	Output matrix	–
$\mathcal{O}$	Observability matrix	–
$\mathcal{C}$	Controlability matrix	–

References

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

[2] *Helicopter_lab_assignment.pdf*. https://ntnu.blackboard.com/bbcswebdav/pid-420173-dt-content-rid-17161604_1/courses/194_TTK4115_1_2018_H_1/194_TTK4115_1_2018_H_1_ImportedContent_20180815022347/Helicopter_lab_assignment.pdf. Accessed: 2018-10-10.