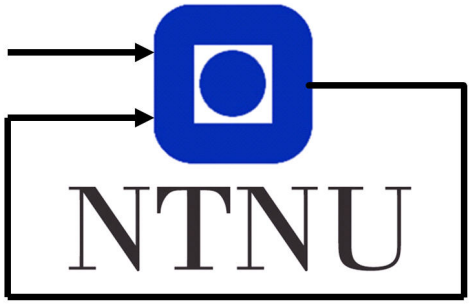
TTK4115 Helicopter Lab Report

Group 52
Ola Moen 999733
Mats Storebø 478388
Morten Fredriksen 768705

October 21, 2018



Department of Engineering Cybernetics

Abstract

This helicopter lab assignment revolves around controlling a helicopter model using different controller designes. First, a mathmatical model was devolped, and then linearized. Based on this linearized model, both monovariable and multivariable controller schemes were implemented. Finally, a state estimator was introduced, and the impact of reducing the number of measured states was investigated.

Contents

System Description

The system to be evaluated consists of a helicopter model bolted to a table. The helicopter consists of two propellers with weight m_p and a counterweight m_c, connected by a rod as illustrated in figure [1]. The figure also gives us the length between the counterweight and the central pivot point l_c, the length between the central pivot point and the propeller pivot point l_h, aswell as the distance from the propellers to the propeller pivot point l_p.

By inspection of figure [2], it is clear that the system has 3 degress of freedom with regards to motion, which is annotated by travel λ, elevation e and pitch p. The gravitational forces on the motors $F_{g,f}$ and $F_{g,b}$, aswell as their lifting forces F_f and F_b, together with the gravitational force on the counterweight $F_{g,c}$ is also illustrated.

The lab assignment [1] provides a number of properties for the system. These will be needed in future computations, and will be restated here. First, the correlation between voltage to the propeller motors and the lifting force provided is as follows:

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

Next, the sum and the difference in voltage supplied to the two propeller motors are defined as:

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

Finally, the moments of inertia used in the computation of pitch, elevation and travel are as follows:

$$J_p = 2m_p l_p^2 \tag{3a}$$
$$J_e = m_c l_c^2 + 2m_p l_h^2 \tag{3b}$$
$$J_\lambda = m_c l_c^2 + 2m_p (l_h^2 + l_p^2) \tag{3c}$$

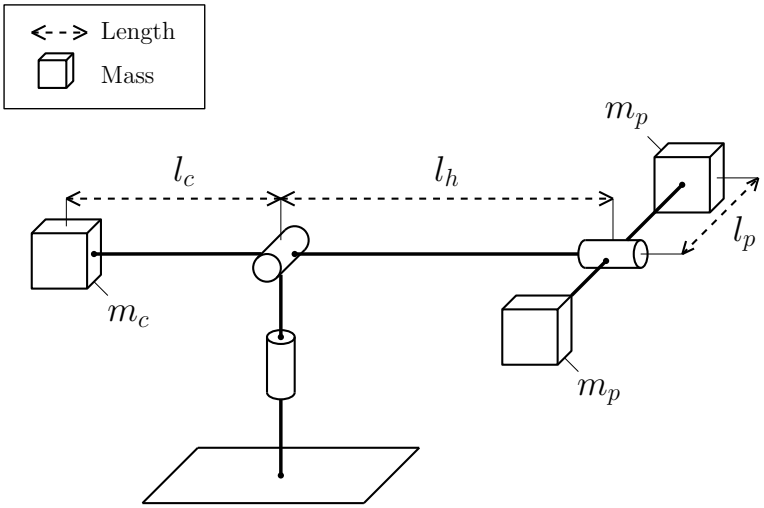


Figure 1: Helicopter system with weights and lengths. Source: [1].

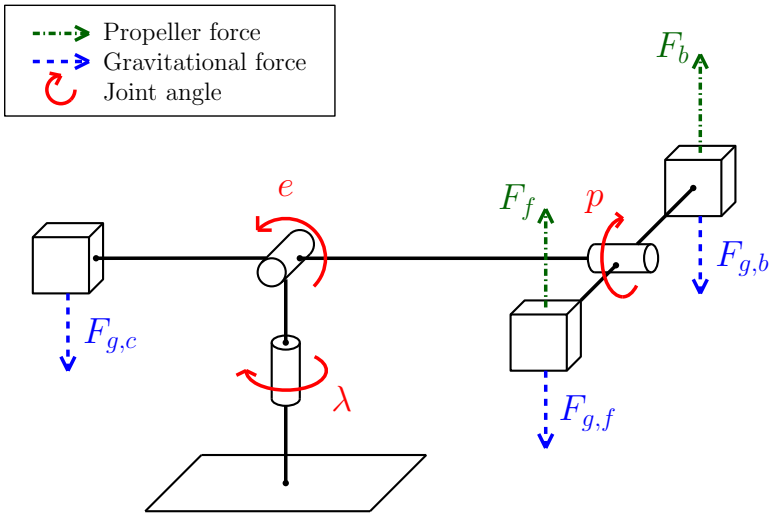
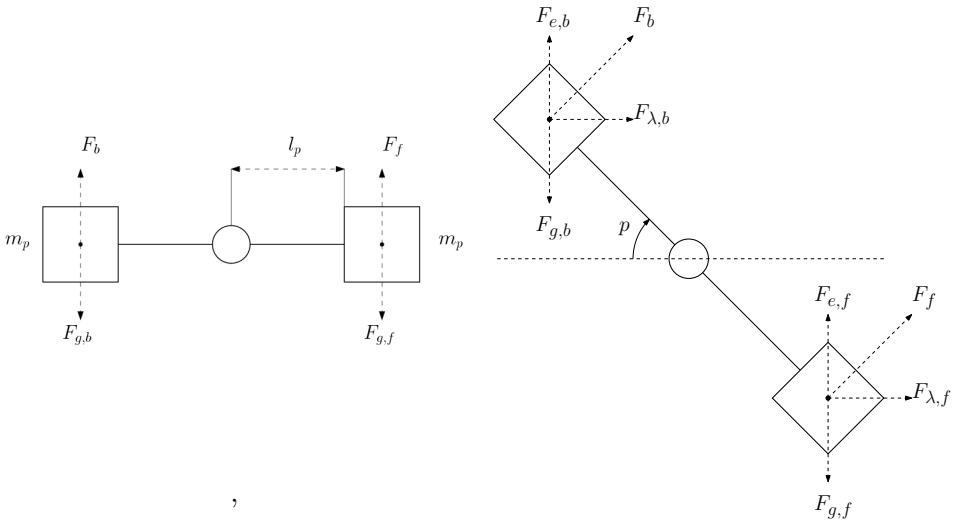


Figure 2: Helicopter system with rotational axes and forces. Source: [1]



(a) Propeller subsystem in equilibrium (b) Propeller subsystem at an angle

Figure 3: Horizontal sketch of propeller subsystem, based on illustrations from helicopter lab assignment [1].

1 Part 1 - Mathematical modeling

1.1 Problem 1 - Equations of motion

In this section, the equations of motion for the helicopter system will be computed. The equations will be derived with regards to pitch p, elevation e and travel λ, respectively.

1.1.1 Pitch

Figure [3a] presents a sketch of the propeller subsystem from a horizontal viewpoint. In this subsection, we will regard the propeller subsystem as independent from the system as a whole, and thus the pivot point will be viewed as fixed. Because the two propellers are connected by a pivot point, any force applied to one propeller has an equal and opposite effect on the other. Thus, by assuming the two motors have equal mass m_p, the two gravitional forces $F_{g,b}$ and $F_{g,f}$ cancel eachother out. The only forces affecting pitch is therefore the two motor forces F_b and F_f.

Newton's second law of rotation yields:

$$\begin{aligned} J\alpha &= \Sigma\tau \\ J_p\ddot{p} &= \tau_f - \tau_b \\ J_p\ddot{p} &= l_pF_f - l_pF_b \end{aligned} \tag{4}$$

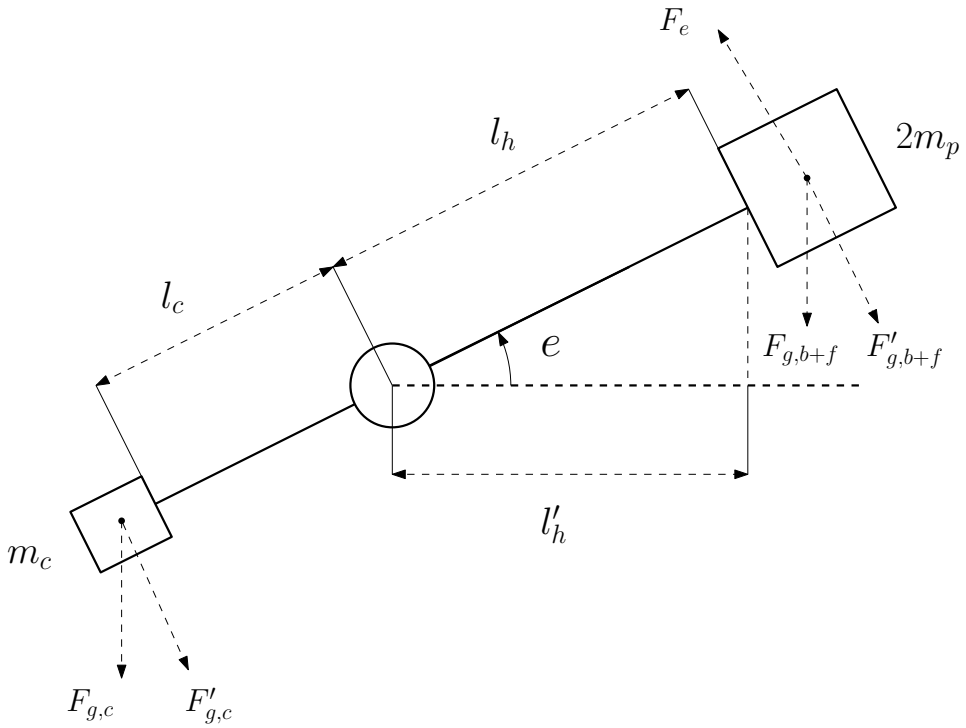


Figure 4: Helicopter system from a horizontal viewpoint, based on illustrations from [1].

Using (1a) and (1a)

$$\begin{aligned} J_p\ddot{p} &= l_p(K_f V_f - K_f V_b) \\ J_p\ddot{p} &= l_p K_f (V_f - V_b) \end{aligned} \tag{5}$$

Substituting by (2a), equation (5) now yields

$$\begin{aligned} J_p\ddot{p} &= l_p K_f V_d \\ J_p\ddot{p} &= L_1 V_d \end{aligned} \tag{6}$$

(6) is the equation of motion for pitch p, where

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

1.1.2 Elevation

Figure [4] presents the helicopter system from a horizontal viewpoint and at an angle. The propeller subsystem in figure [3] is now regarded as a black box, where F_e is the combined lifting force of the propellers.

We derive an expression for F_e by inspecting figure [3b].

$$\begin{aligned} F_e &= F_{e,b} + F_{e,f} \\ F_e &= F_b cos(p) + F_f cos(p) \\ F_e &= (F_b + F_f) cos(p) \\ F_e &= K_f V_s cos(p) \end{aligned} \tag{8}$$

The equation of motion for elevation follows from Newton's second law of rotation (4) aswell. Inspection of the system in figure [4] yields:

$$\Sigma\tau = \tau_{g,c} + \tau_{g,b+f} + \tau_e \tag{9}$$

where $\tau_{g,c}$ is the torque applied to the system by the gravitational force on the counterweight, $\tau_{g,b+f}$ is the torque of the gravitational force on the combined weight of the two propellers, and τ_e is the torque of the lifting force of the propellers. Applying the second law of rotation (4) and using (8) and (9) we obtain:

$$\begin{aligned} \ddot{e}J_e &= \Sigma\tau \\ \ddot{e}J_e &= \tau_{g,c} + \tau_{g,b+f} + \tau_e \\ \ddot{e}J_e &= F'_{g,c} l_c + F'_{g,b+f} l_h + F_e l_h \\ \ddot{e}J_e &= F_{g,c} cos(e) l_c + F_{g,b+f} cos(e) l_h + F_e l_h \\ \ddot{e}J_e &= g m_c cos(e) l_c + 2 g m_p cos(e) l_h + K_f V_s cos(p) l_h \\ \ddot{e}J_e &= g cos(e)(m_c l_c + 2 m_p l_h) + K_f V_s cos(p) l_h \\ \ddot{e}J_e &= L_2 cos(e) + L_3 V_s cos(p) \end{aligned} \tag{10}$$

(10) is the equation of motion with regards to elevation e, where

$$L_2 = g(m_c l_c + 2 m_p l_h) \tag{11}$$

$$L_3 = K_f l_h \tag{12}$$

1.1.3 Travel

Figure [5] Shows the helicopter system from a top-down perspective, where the propeller subsystem is once again regarded as a black box, and F_λ is the combined lateral force of the propellers. This is also the only force directly affecting the travel of the helicopter system, seeing as the travel direction is perpendicular to the direction of the gravitational force. The gravitational force and propeller lift does however affect the length l'_h, which again influences the torque τ_λ of the lateral force, as will be derived later.

We derive an expression for F_λ by inspecting figure [3b]:

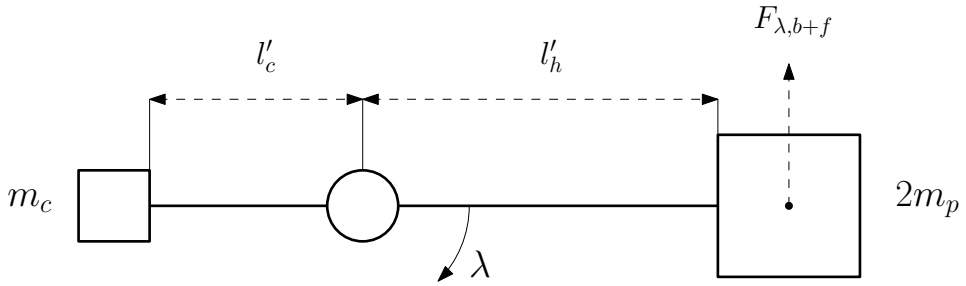


Figure 5: Helicopter system from a top-down viewpoint, based on illustrations from [1].

$$
\begin{aligned}
F_\lambda &= F_{\lambda,b} + F_{\lambda,f} \\
F_\lambda &= F_b sin(p) + F_f sin(p) \\
F_\lambda &= (F_b + F_f) sin(p) \\
F_\lambda &= K_f V_s sin(p)
\end{aligned} \tag{13}
$$

By comparing figure [4] and figure [5], it is clear that l'_h is a variable depending on the angle of elevation e.

$$l'_h = l_h cos(e) \tag{14}$$

Combining (13) and (14), and applying Newton's second law of rotation (4) we have:

$$
\begin{aligned}
\ddot{\lambda} J_\lambda &= \Sigma\tau = \tau_\lambda \\
\ddot{\lambda} J_\lambda &= F_\lambda l'_h \\
\ddot{\lambda} J_\lambda &= K_f V_s sin(p) l_h cos(e) \\
\ddot{\lambda} J_\lambda &= L_4 V_s cos(e) sin(p)
\end{aligned} \tag{15}
$$

(15) is the equation of motion with regard to travel λ, where

$$L_4 = K_f l_h \tag{16}$$

1.2 Problem 2 - Linearization

The system will be linearized around the point $[p, e, \lambda]^T = [p^*, e^*, \lambda^*]^T$ where $p^* = e^* = \lambda^* = 0$. First, we will determine the voltages V_s^* and V_d^* that keeps the system at the equilibrium point.

By inspection of figure [3] we extrapolate that in order for the pitch of the propellers to be horizontal, the difference in motor voltage must be zero, assuming equal voltage-to-force characteristics for the two motors. Thus,

$$V_d^* = V_f^* - V_b^* = \frac{F_f - F_b}{K_f} = 0 \tag{17}$$

because $F_f = F_b$ at the equilibrium.

Similarly, by inspection of figure [4] its clear that some force F_e has to be applied to elevate the system to a horizontal position, given that $\tau_{g,b+f}$ is inherently larger than $\tau_{g,c}$. This force is exactly the force that cancels out the combined torque of the two gravitational forces:

$$\begin{aligned} \tau_{eq} &= \tau_{g,b+f} - \tau_{g,c} \\ F_{eq} l_h &= F_{g,b+f} l_h - F_{g,c} l_c \\ F_{eq} &= \frac{2 m_p g l_h - m_c g l_c}{l_h} \end{aligned}$$

Substituting in (1a) and (1b) we obtain the voltage V_s^* that drives the elevation to equilibrium:

$$V_s^* = \frac{2 m_p g l_h - m_c g l_c}{l_h K_f} \tag{18}$$

Next, we utilize the given coordinate transformation for the system:

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

Inserting the calculated values from above we obtain:

$$\begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} = \begin{bmatrix} p \\ e \\ \lambda \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \tilde{V_s} \\ \tilde{V_d} \end{bmatrix} = \begin{bmatrix} V_s - V_s^* \\ V_d \end{bmatrix} \tag{19}$$

where V_s^* is expressed by (18).

The system can now be linearized around (19). First we express the system in its nonlinear form with regards to the states $\mathbf{x}$ and input vetor $\mathbf{u}$:

$$\mathbf{x} = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \\ \dot{\tilde{\lambda}} \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{20a}$$

$$\dot{\mathbf{x}} = \mathbf{h}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{\tilde{p}} \\ \dot{\tilde{e}} \\ \dot{\tilde{\lambda}} \\ \frac{L_1}{J_p}\tilde{V}_d \\ \frac{L_2}{J_e}cos(\tilde{e}) + \frac{L_3}{J_e}(\tilde{V}_s + V_s^*)cos(\tilde{p}) \\ \frac{L_4}{J_\lambda}(\tilde{V}_s + V_s^*)cos(\tilde{e})sin(\tilde{p}) \end{bmatrix} \tag{20b}$$

The linearized system around $\mathbf{x}_0 = [0, 0, 0]^T$ and $\mathbf{u}_0 = [0, 0]$ can be expressed as $\mathbf{Ax} + \mathbf{Bu}$, where $\mathbf{A} = \frac{\partial \mathbf{h}(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}|_{\mathbf{x}_0,\mathbf{u}_0}$ and $\mathbf{B} = \frac{\partial \mathbf{h}(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}|_{\mathbf{x}_0,\mathbf{u}_0}$. Appliying partial derivatives and inserting $\mathbf{x}_0$ and $\mathbf{u}_0$ yields:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{L_4}{J_\lambda}V_s^* & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{L_1}{J_p} \\ \frac{L_3}{J_e} & 0 \\ 0 & 0 \end{bmatrix}$$

Solving the state-space equation with $\mathbf{A}$ and $\mathbf{B}$ yields the linearized equations of motion:

$$\ddot{\tilde{p}} = K_1\tilde{V}_d \tag{22a}$$

$$\ddot{\tilde{e}} = K_2\tilde{V}_s \tag{22b}$$

$$\ddot{\tilde{\lambda}} = K_3\tilde{p} \tag{22c}$$

where,

$$K_1 = \frac{L_1}{J_p} \tag{23a}$$

$$K_2 = \frac{L_3}{J_e} \tag{23b}$$

$$K_3 = -\frac{L_4 V_s^*}{J_\lambda} \tag{23c}$$

1.3 Problem 3 - Controlling helicopter with joystick

The first controller scheme for the system used a simple feed forward from the input joystick, as seen in the Simulink diagram [15]. In order for the feed forward input gain to affect the elevation of the system, a constant gain had to be applied to the input signal. A satisfactory gain was found by simple trial and error, starting with a low-valued gain and incrementally increasing it until acceptable response was achieved. Final value for the gain was $K = 7$.

This controller scheme is however a bit crude. For instance, in order to bring the system back to equilibrium point after an input, an equal and opposite input has to manually be fed to the system from the joystick, as can be seen in figure [6]. In other words, the operator is the sole factor affecting the state of the system, which again leads to a system that is rather hard to control.

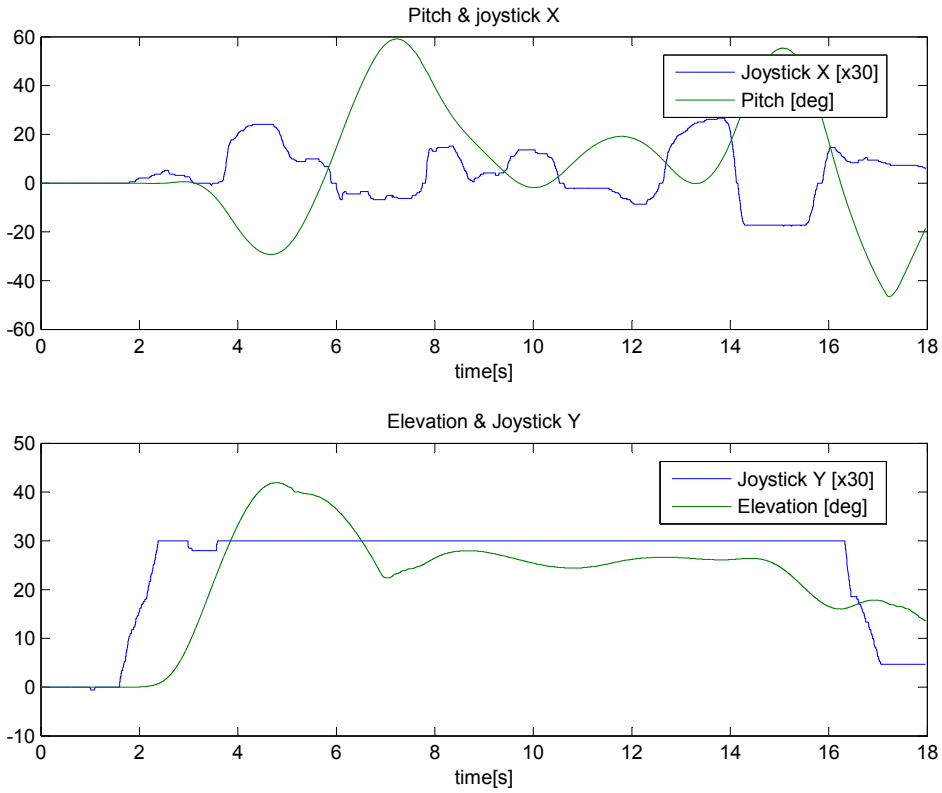


Figure 6: Plot of pitch and elevation response versus the joutstick X- and Y input, respectively. Please note that the joystick X input is flipped in magnitude, due to a plotting oversight.

Making matters worse, the operator also has to compensate for the inaccuracies of the system implementation. One such inaccuracy we observed

was a constant pitch, even with joystick input to V_d being zero. This discrepancy might be caused by an inequality in the voltage-to-force characteristics of the two motors, which in the theoretical models have been assumed equal. Another possibilty might be that the inital resting position of the propeller subsystem is not perfectly horizontal on startup, thus leading to a pitch, or that the weight of the propellers is not perfectly balanced. Another factor the theoretical models do not consider is the time delay between the joystick input and actual force applied by the rotors, caused by the dc-motor characteristics, among other things.

Finally, the theoretical models do not consider motor performance irregularaties, as was experienced a few times during testing of the problems to come. The motor would, across all the different controller designes, sometimes output a much lower lifting force than what the controller voltage output would imply. A restart was required during these outlier events.

In general, when linearizing a non-linear system, some loss of model accuracy is to be expected. The consequence of this is simply that the model we end up with, while simpler to design controllers for, does not reflect the actual system all that well. When we later design controllers for this system based on the linearized model, we expect the speed and accuracy of the linearized controller to be subpar compared to that of a controller designed for a nonlinear system. Also, when estimating the states of this system, we expect to use a high gain to compensate for the inaccuracy of the linearized model.

1.4 Problem 4 - Offset and the motor force constant, K_f

The helicopter system is equipped with a decoder to measure the angles of pitch, elevation and travel. This encoder is however not nullset in compliance with the point (19) in which we linearized the system, so an offset was included in the Simulink diagram [16]. The value of this offset was obtained by measuring the encoder output at the linearization point. We observed that the pitch matched the linearization point value quite accuratly, but had to offset due to a constant travel. Offsetting the equilibrium of the pitch by 8 degrees was enough to cancel out the propeller inequalities, as discussed in section 1.3. The elevation decoder output was offset by $\Delta_e = -30$. The travel of the system is defined as the displacement from the point of initialization, and is thus always zero at initialization and not in need of an offset. The elevation and pitch offset gain, as well as a degree-to-radian convertion block can be observed in the updated Simulink diagram [16].

Next, the value of K_f will be computed. By measuring the voltage V_s^* necessary to keep the system at equilibrium elevation, an expression for K_f can be obtained from (18). The procedure for measuring V_s^* was simply to hold the helicopter at the equilibrium elevation, and then incrementally increasing a constant input feed to V_s until the helicopter held itself. The resulting constant was $K = V_s^* = 7.3[V]$. Using (18) and inserting numerical values we obtain:

$$K_f = \frac{2m_p g l_h - m_c g l_c}{l_h V_s^*} = 0.137[N/V] \tag{24}$$

2 Part 2 - Monovariable controll

In this section, a PD-controller for the pitch and a P-controller for the travel rate is introduced. Following is a discussion regarding implementation and tuning of these controllers. We use an elevation controller to control V_s throughout this section.

2.1 Problem 1 - PD controller for pitch

The first controller to be introduced is the following PD-controller for pitch:

$$\tilde{V_d} = K_{pp}(\tilde{p_c} - \tilde{p}) - K_{pd}\dot{\tilde{p}} \tag{25}$$

where $K_{pp} > 0$ and $K_{pd} > 0$. This controller implementation can be seen in Simulink diagram [17] and [18].

Substitution of (25) in (22c) yields a new linearized equation of motion:

$$\ddot{\tilde{p}} = K_1(K_{pp}(\tilde{p_c} - \tilde{p}) - K_{pd}\dot{\tilde{p}}) \tag{26}$$

The laplace transform of (26) results in the following transfer function:

$$\begin{aligned} s^2\tilde{p}(s) &= K_1K_{pp}(\tilde{p_c}(s) - K_1K_{pp}\tilde{p}(s)) - sK_1K_{pd}\tilde{p}(s)) \\ \frac{\tilde{p}(s)}{\tilde{p_c}(s)} &= \frac{K_1K_{pp}}{s^2 + K_1K_{pd}s + K_1K_{pp}} \end{aligned} \tag{27}$$

This transfer function is a second-order differential equation, similiar to that of a mass-spring-damper.

To make the system respond as fast as possible, it has to be critically damped. A critically damped system has equal poles, and its transfer function is on the form:

$$\frac{K}{(s+a)^2} = \frac{K}{s^2 + 2as + a^2} \tag{28}$$

By comparing (27) and (28), we find:

$$\begin{aligned} K &= K_1K_{pp} \\ a^2 &= K_1K_{pp} \\ 2a &= K_1K_{pd} \end{aligned}$$

This lets us extract a relation between K_{pp} and K_{pd}:

$$K_{pd} = \sqrt{\frac{4K_{pp}}{K_1}} \tag{30}$$

A second property the system must exhibit, is stability. A BIBO stable system has its poles within the unit circle in the complex plane. This property produces a necessary relation between the roots of the quadratic

equation (28) in order for the system to be stable. If we let $2a = a_1 = K_1 K_{pd}$ and $a^2 = a_2 = K_1 K_{pp}$, we obtain the useful relation (see:[3])

$$|a_2| < 1$$
$$|K_1 K_{pp}| < 1$$

A critically damped system is on the verge of stability, so by assuming a positve P-gain we have:

$$|K_1 K_{pp}| = 1$$
$$K_{pp} = \frac{1}{K_1} \tag{31}$$

Substituting (31) into (30), we obtain the second controller gain:

$$K_{pd} = \sqrt{\frac{4\frac{1}{K_1}}{K_1}} = \frac{2}{K_1} \tag{32}$$

Figure [7] shows a plot of the system response to these controller gains. This response was not satisfactory, seeing as the system never reached the reference pitch, as well as being unbearably slow. It is however not surprising that the calculated controller gains were suboptimal. They are after all calculated from a linearized system, which we discussed earlier has many discrepencies compared to the real system the controller acts upon. Some trial and error had us increase K_{pp} by a factor of 4, resulting in the response seen in figure [8]. Even though the system experiences a slight overshoot, the response is much faster. Thus, the calculated gains provided a fair starting point for the tuning process.

An interesting relation to consider is how the controller gain influences the closed-loop system poles. From (28) it is clear that the poles, or the solution to the quadratic equation, are functions of K_{pp} and K_{pd}. We can also see from both the response [8] and directly from the transfer function (27) that an increase in K_{pp} will lead to poles with a more negative real part, and thus a more aggressive system response. To counteract the overshoot caused by a larger K_{pp}, an increase of K_{pd} could also be applied. We could also choose K_{pd} and K_{pp} in such a way that the poles become complex conjugate. This could lead to even faster response, however the pitch would be prone to oscillations. This is highly undesirable for our system, due to the fact that oscillations in pitch also leads to oscillations in travel.

Overall, the system is much easier to control with the help of a PD-controller for the pitch compared to feed-forward control. The PD-controller returns the pitch to equilibrium point by feeding an appropriate V_d to the motors, a task not easily done by a human operator.

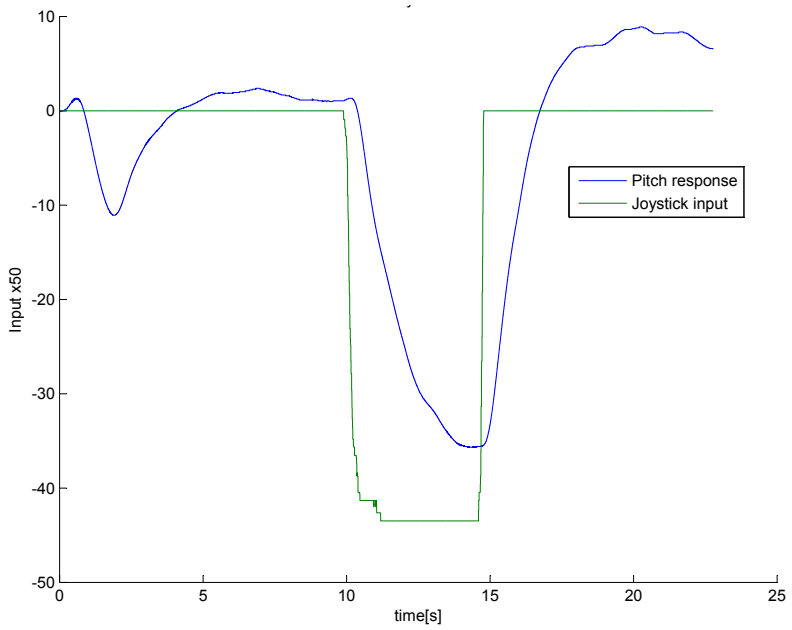


Figure 7: Pitch response with estimated controller gains: $K_{pp} = \frac{1}{K_1}$ and $K_{pd} = \frac{2}{K_1}$. Joystick input is multiplied by 50.

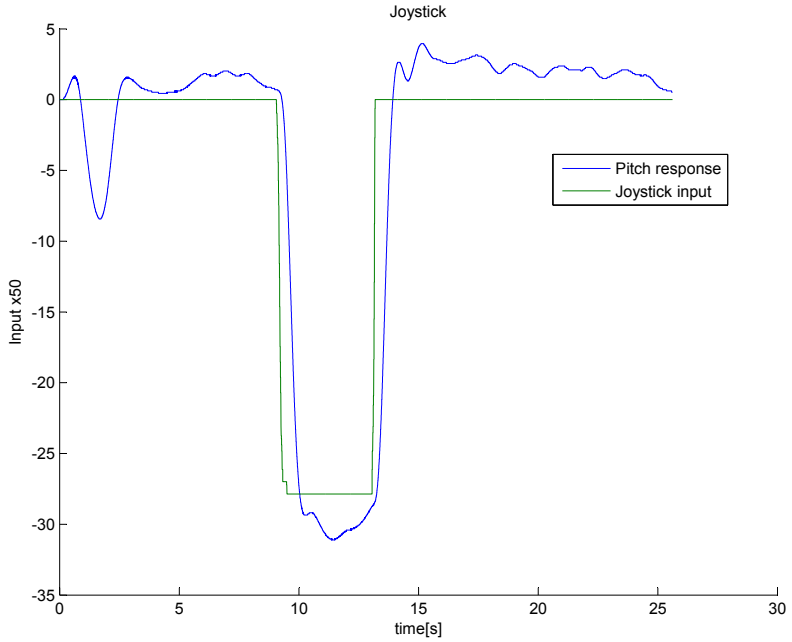


Figure 8: Pitch response with ajusted controller gains: $K_{pp} = \frac{4}{K_1}$ and $K_{pd} = \frac{2}{K_1}$. Joystick input is multiplied by 50.

2.2 Problem 2 - P-controller for travel

The travel rate will now be controlled by a simple P-controller on the form:

$$\tilde{p_c} = K_{rp}(\dot{\tilde{\lambda}}_c - \dot{\tilde{\lambda}}) \tag{33}$$

Where $K_{rp} < 0$. Implimentation can be seen in Simulink diagram [19] and [20].

We will assume that the pitch is controlled perfectly $\tilde{p} = \tilde{p_c}$. This assumption enables us to substitute (33) into the linearized equation of motion for travel (22c) to obtain

$$\ddot{\tilde{\lambda}} = K_3 K_{rp}(\dot{\tilde{\lambda}}_c - \dot{\tilde{\lambda}}) \tag{34}$$

By applying the Laplace transform to (34), we obtain a a transfer function between the travel rate and the travel rate reference:

$$\begin{aligned} s(\dot{\tilde{\lambda}}) &= K_3 K_{rp}(\dot{\tilde{\lambda}}_c(s) - \dot{\tilde{\lambda}}(s)) \\ \frac{\dot{\tilde{\lambda}}(s)}{\dot{\tilde{\lambda}}_c(s)} &= \frac{K_3 K_{rp}}{s + K_3 K_{rp}} = \frac{\rho}{s + \rho} \end{aligned} \tag{35}$$

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

In order for the system to have a fast and accurate response, we need to choose an appropriate controller gain K_{rp}. Analysis of (35) reveals that the steady state of this system is

$$\frac{\dot{\tilde{\lambda}}(0)}{\dot{\tilde{\lambda}}_c(0)} = \frac{\rho}{\rho} = 1$$

Therefore, any gain K_{rp} chosen will still lead to the response reaching the reference, and accuracy of the controller is not an issue. The remaining problem is to choose the gain such that the response is fast enough, without introducing too much noise or oscillations. Since the pole of the system is defined by $s = -K_{rp}K_3$, we figured a negative pole of magnitude of $|s| \approx 1$ would be a good strating point. With $K_3 = -0.6117$, we chose $K_{rp} = -1$ such that $s \approx 0.6$. This response was as expected not the best, so we approached $|s| \approx 1$ by increasing the gain in small increments. $K_{rp} = -1.6$ and $s \approx -1$ led to a good response, as seen in figure [9]. Traversing the plot from left to right, we see that the travel rate tracks the reference fairly well, with a slight overshoot when a step response is applied to stop counterclockwise rotation at $t = 8$. The same step response to stop clockwise rotation however looks close to critically damped, as can be seen around $t = 16$. Essential to obtaining this travel rate response was a nicely tuned pitch controller, seeing as the travel rate controller design was based upon the assumption that pitch was tracked perfectly. The corrosponding pitch response to correct the travel rate can also be observed in figure [9].

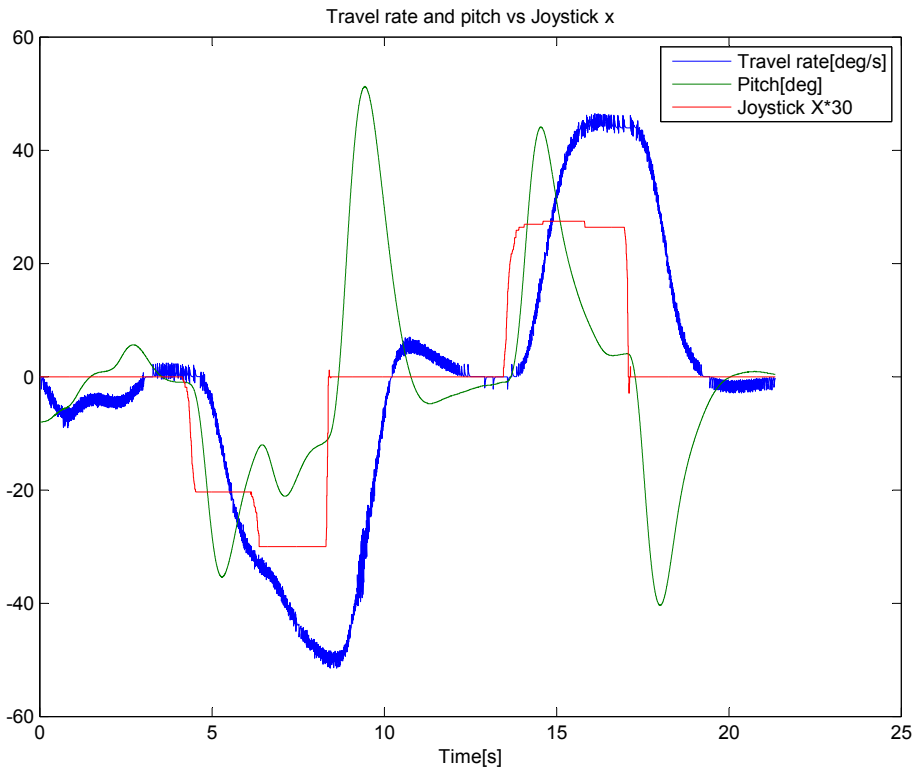


Figure 9: Travel rate and pitch response to input on the joystick X-axis.

3 Part 3 - Multivariable control

The controller scheme for the system will now be implemented using multivariable control. First, a LQR with feed-forward gain will be added. Next, the controller will be expanded to also include an integral effect.

3.1 Problem 1 - State-space formulation

The states to be controlled from now on will be pitch $\tilde{p}$ and elevation rate $\dot{\tilde{e}}$. The state-space equation for the multivariable controller must then be represented using these states.

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \qquad \text{with} \qquad \mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \end{bmatrix} \qquad \text{and} \qquad \mathbf{u} = \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{36}$$

From the linearized equations (22a - 22c) we obtain:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \text{and} \qquad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \end{bmatrix} \tag{37}$$

3.2 Problem 2 - LQR with PD effect

Using the Matlab command `rank = rank(ctrb(A,B))`, the controllability of the system was evaluated. The controllability matrix was found to have full rank, $\text{rank}(\mathcal{C}) = 3$, meaning the system is controllable.

We then implemented the controller

$$\mathbf{u} = \mathbf{Pr} \text{ - } \mathbf{Kx} \qquad \text{with} \qquad \mathbf{r} = \begin{bmatrix} \tilde{p_c} \\ \dot{\tilde{e}}_c \end{bmatrix} \tag{38}$$

where $\mathbf{K}$ corresponds to the LQR and $\mathbf{P}$ is the feed-forward gain, as seen in the Simulink diagram [21]. Furthermore, the references $\tilde{p_c}$ and $\dot{\tilde{e}}_c$ are given by the joysticks x-axis and y-axis respectively.

The LQR was implemented using two diagonal matrices $\mathbf{Q}$ and $\mathbf{R}$, with each pivot in $\mathbf{Q}$ governing the cost of a state discrepency in $\mathbf{x}$, and each pivot in $\mathbf{R}$ governing the cost of the inputs in $\mathbf{u}$. The entries of $\mathbf{K}$ was generated using the Matlab command `K=lqr(A,B,Q,R)`.

In order to find the feed-forward gain $\mathbf{P}$, we evaluated the steady state response of the system. First, the state-space system had to be represented as a transfer matrix between the reference $\mathbf{r}$ and the response $\mathbf{y}$ as such [2]:

$$\begin{aligned} \mathbf{Y}(s) &= \mathbf{H}(s)\mathbf{R}(s) \\ \mathbf{H}(s) &= \mathbf{C}(\text{s}\mathbf{I}\text{-}(\mathbf{A}\text{-}\mathbf{BK}))^{-1}\mathbf{BP} \end{aligned} \tag{39}$$

In order for the system to be able to track a step response, the transfer matrix in (39) must satisfy the condition

$$\mathbf{H}(0) = \mathbf{C}(\mathbf{BK} - \mathbf{A})^{-\mathbf{1}}\mathbf{BP} = 1 \tag{40}$$

Using (40) we obtain an expression for $\mathbf{P}$ as a function of $\mathbf{K}$:

$$\mathbf{P} = \mathbf{(C(BK\text{-}A)^{-1}B)^{-1}} \tag{41}$$

Tuning of the controller is reduced to manipulating the matrices $\mathbf{R}$ and $\mathbf{Q}$ to assign a different "cost" to the variables in $\mathbf{x}$ and $\mathbf{u}$. Once again, the process of trial and error was utilized, starting with the identity matrices

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{42}$$

Unsurprisingly, the system did not exhibit an exceptable response to the reference input with these control parametres, so we went about tuning the system one state at a time. Some observations we made relating to the entries of $\mathbf{Q}$ and $\mathbf{R}$ and the system response was:

- The elevation rate had to be of utmost priority in order for the helicopter to stay airborne during pitch corrections.
- Pitch rate was not instrumental to the stability, nor the quality of the response, so it was left untouched. Indeed, when given a higher priority, it severly crippled the elevation rate.
- In order to avoid oscillations and generally unstable behaviour, pitch had to be prioritized to a certain degree.
- Again, to enable the helicopter to stay airborne, the cost of V_d had to be decreased to allow for more powerful response when at a large pitch.

With these observations in mind, the following control parametres were found:

$$\mathbf{Q} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1000 \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix} \tag{43}$$

The system response with these control parametres can be seen in figure [10]. Traversing the plots from left to right, we observe that in the first 3 seconds $t < 3$ the helicopter is excited by an input to the elevation rate. This is only done to elevate the helicopter to the linearization point faster, so the response should be analysed from the point $t = 3$ and onwards. Next, in the interval $4 < t < 12$, a pitch input is applied in both directions, which is

tracked decently. The pitch step response can be observed at $t = 12$. Please note that the scale of the input and the response do not correlate perfectly. It is of special interest to study the elevation response during the pitch corrections. We see that the system failes to keep a perfectly still elevation when a pitch reference is applied, but still within exceptable margins. From $t = 14$ and out, we see that an elevation reference is applied. The pitch response is completly still, while the elevation rate tracks the reference to a degree. Most notably, we observe a deviation leading to the helicopter rising when led far away from the linearization point. Nevertheless, given the current controller, the response was deemed acceptable.

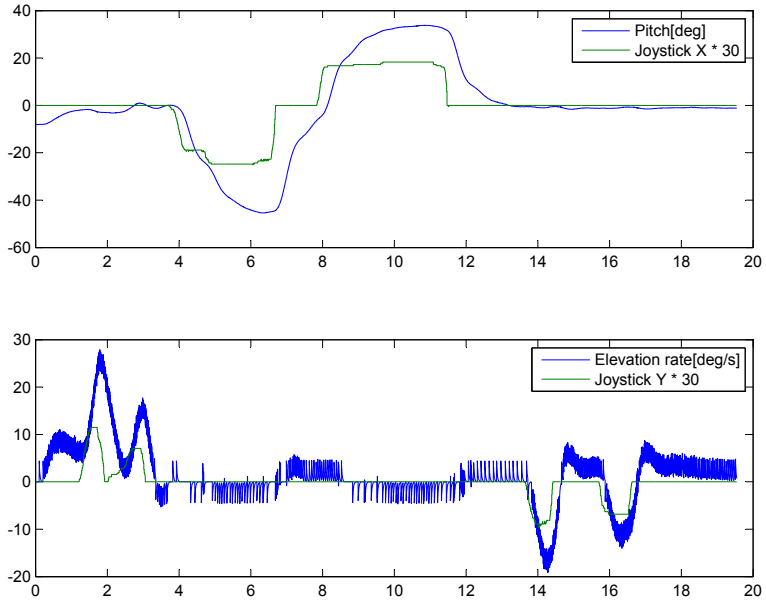


Figure 10: Pitch and elevation rate measured against their reference value over time, using a controller without integral effect.

3.3 Problem 3 - LQR with PID effect

To remove the stationary deviation from the LQR, we include an integral effect to the controller. This introduces two new states to our system:

$$\begin{aligned} \dot{\gamma} &= \tilde{p} - \tilde{p}_c \\ \dot{\zeta} &= \dot{\tilde{e}} - \dot{\tilde{e}}_c \end{aligned} \tag{44}$$

We observe that these new states depends on the reference signals, leading to the state-space model having to be expanded as such:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{F}\mathbf{r} \tag{45}$$

where,

$$\mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{\tilde{e}} \\ \gamma \\ \zeta \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \tag{46}$$

In this new system, we can regard **Fr** as a disturbance, and focus the controller design around **A** and **B** as previously.

Adding states to the system also changes the dimension of the **Q** matrix, due to the inherent realtion between **Q** and the states **x**. Implimentation of the new states in the Simulink diagram was done by integrating the error function $e = y - r$ for the two reference signals, as seen in figure [22].

An interesting property of this controller scheme is that the integral effect will push the system to the reference value, regardless of the feed-forward gain **P** and the "disturbance" **Fr**. Thus, it can be argued that the entries of **P** can be assigned arbitrarily without loss of system performance. However, by designing **P** in a specific manner, we can actually increase the controller performance by "pushing" the system towards the reference value before the integral effect even takes place. However, since the system requires tracking a highly variable reference, and not just simple step inputs, we found the algebraic computation of **P** to be slightly beyond the scope of our current abilities. The process would likely expand on the result found in (41), using the new **A** and **B**. Regardless, we chose to stick with the feed-forward gain computed previously, seeing as that gain will push the system to reference even with no integral effect, allowing us to place a lower priority on the integral states during the tuning process.

As previously, the tuning process utilized a trial and error approach, leading to observations that helped us shape the **Q** and **R** matrix. However, for this system we already had a good basis from the previous problem, so we concentrated our efforts on the new integral states introduced to the controller. The key observations were:

- A high priority on the integral states quickly "drowned" the gains from the rest of the system, resulting in volatile behaviour.
- The system had a harder time tracking elevation rate than before, but improved response was obtained through decreasing the cost of V_s.

Using the principles above in conjunction with the observations made previously, we found the following new control parametres:

$$\mathbf{Q} = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0.8 \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.1 \end{bmatrix} \tag{47}$$

Figure [11] shows the plot of the helicopter response with the new controller. We traverse the plot from left to right. Once again, the first 3 seconds $t < 3$ is used to elevate the helicopter to the linearization point faster. In the interval $6 < t < 14$ a pitch reference is applied, which is tracked well. The pitch step response can be observed at $t = 15$. The elevation is not perfectly still during the pitch corrections, but still fine. From $t = 16$ and onwards, an elevation reference is applied. The pitch stays perfectly still, while the elevation rate is tracked quite fast, but with some deviation when led far away from the linearization point. Some additional tuning could be done to improve the system response, but overall the response was deemed fine.

Comparing the response of the system with different controllers, some key characteristics of the controller designs are made apparent. Note the scale of the eleveation Y-axis in figure [10] and figure [11]. It is clear that while the elevation response looks quite similar, the difference in scale reveals a much smaller deviation in elevation rate for the controller with integral effect. This is of course the main feature of the integral controller scheme. Another interesting improvement is the pitch response to counterclockwise references, as can be seen by comparing the pitch response in both figures at around $t = 6$.

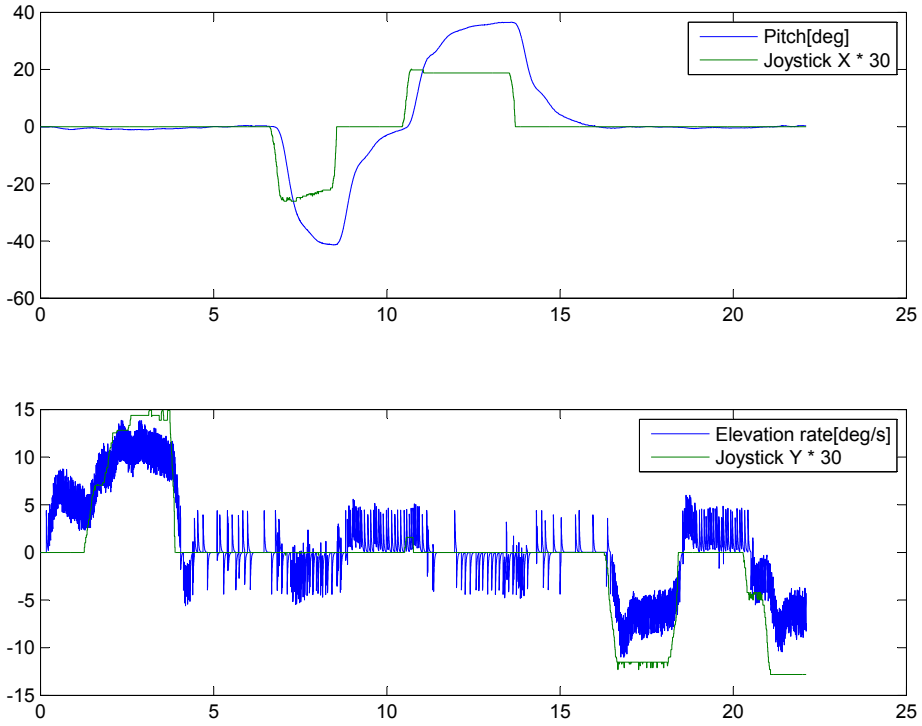


Figure 11: Pitch and elevation rate measured against their reference value over time, using a controller with integral effect.

4 Part 4 - State estimation

In this section, our controller will no longer get its states from measurement and numerical derivation, but through an estimator. The estimator will initially be based on the full measurement vector, which later will be minimized to the least number of measured states possible.

4.1 Problem 1 - State-space formulation

The estimator is based on the system in its entirety, unlike the multivariable controller subsystem from section 3 which only utilized some states of the system. Thus, the input matrix $\mathbf{u}$ and state matrix $\mathbf{x}$ are the same as found in our initial system discussion in section 1.2, with some rearrangment. The measurement vector $\mathbf{y}$ will initially contain the full vector of measured states from the encoder.

$$\mathbf{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix}$$

A rearragement of the state-space equation found in section 1.2, coupled with the measurement vector yields the following state-space system:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$
$$\mathbf{y} = \mathbf{C}\mathbf{x}$$

where,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\frac{L_4}{J_\lambda} V_s^* & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{L_1}{J_p} \\ 0 & 0 \\ \frac{L_3}{J_e} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}^T \tag{50}$$

4.2 Problem 2 - Estimation with all measured states

In order to estimate the system, it has to be observable. To evaluate the observability of the system, it is possible to use the Matlab command `rank = rank(obsv(A,C))`. The output of this function is the rank of the observability matrix of the system. It has full rank if the rank of the observability matrix is equal to the number of states in the system, which is the case for our system (50).

Next, the estimator is introduced:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{L}(\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}) \tag{51}$$

The estimator was implemented as seen in the Simulink diagram [23], where the estimated states are fed into the controller from section 3.3, realized in simulink diagram [24]. Thus, if the estimator gain is correctly tuned, the system should behave as previously.

As with a multivariable controller system, the eigenvalues of the state estimator system can be arbitrarly placed, given that the system is observable. The challenge is to choose the correct poles to assign to the estimator. Unlike previously, we did not use a LQR to solve this problem. Instead, we noted that the most important property of the estimator is that it is faster than the controller it feeds into. Thus, by simply looking at the controller matrix $\mathbf{K}$ generated by the LQR and noting its leftmost pole, we had an estimate for the absolute theoretical minimum for the magnitude of the poles of the estimator. The leftmost pole of $\mathbf{K}$ with the current tuning of the system (45) was found to be $s \approx -28.5$. To give ourselves a solid margin, we decided to assign poles that were a magnitude of 5 to that of the leftmost pole of $\mathbf{K}$, leaving us with an initial guess of:

$$poles = [-150, -155, -130, -135, -140, -145] \tag{52}$$

The poles were placed in the estimator system by the Matlab command `L = place(A', C', poles)'`

Figure [12] shows the plots of the estimated states versus the measured and numerically computed states with the poles (52). Overall, we were quite pleased with the accuracy of the estimation. The measured states $\tilde{p}$, $\tilde{e}$ and $\tilde{\lambda}$ were, unsurprisingly, spot on, while the derivatives had some discrepancies. However, since discrete numerical derivation is not ideal, this discrepancy in the derivated states is to be expected. This also proves the advantage of state estimators, seeing as the steady state of the estimator always produces the actual state. That beeing said, our estimator still had room for improvement, especially with regard to noise. Shifting the closed-loop observer poles more to the right would reduce this noise, but the estimator speed would decrease as a consequence. Another improvement could have been to include lowpass filters on the derivated states, further reducing noise. Nevertheless, we settled with the current response in order to allocate more lab-time for the last problem.

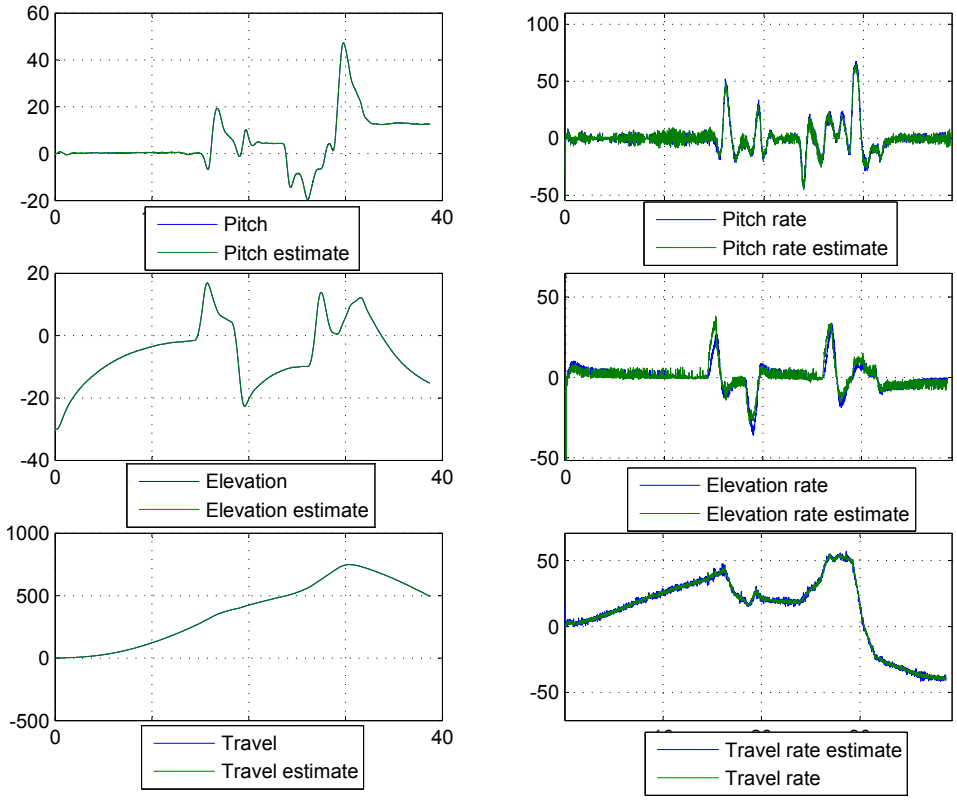


Figure 12: Estimated states plotted together with the measured, or numerically derivated values. Full measurement estimator poles = [-150, -155, -130, -135, -140, -145].

4.3 Problem 3 - Estimation with minimal amount of measured states

Often when considering real-world control systems, some states are simply not measurable, or too expensive to measure accurately. In these cases, a state estimator can once again be utilized. To illustrate this, our estimator system will have its measurement vector reduced to only elevation $\tilde{e}$ and travel $\tilde{\lambda}$. This yields a new measurement vector $\mathbf{y}$:

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

Naturally, this also leads to a change of dimensions of the matrices $\mathbf{C}$ and $\mathbf{L}$. This change is reflected only in the gain blocks of the Simulink diagram [23], so the diagram did not need to be redesigned in any way. The new $\mathbf{C}$ matrix is:

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{54}$$

This reduction in measurement vector is enabled by the inherent relation between pitch $\tilde{p}$ and travel $\tilde{\lambda}$, as described by (22c). A physical consideration of the system also yields the same result; by applying a pitch to the helicopter, the helicopter will start to move forward, which is the main principle of any helicopter propulsion system. Finally, this relation is also expressed in the observability of the system. By the definition of observability, if there exists a relation between the measurement vector $\mathbf{y}$ and all the states of the system, the system is observable. Since `rank = rank(obsv(A, C))` with the modified $\mathbf{C}$ (54) yields the same number as the number of states the system inhibits, the system is observable and the relation between $\tilde{p}$ and $\tilde{\lambda}$ is once again confirmed.

An interesting consideration to make is to use the measurement vector $\mathbf{y} = [\tilde{p}\ \tilde{e}]^T$ instead. Evaluating the observability of the system with this measurement vector, using the corrosponding $\mathbf{C}$, reveals that the observability matrix does not have full rank. This implies that the system is not observable, and a full set of explicit relations between $\tilde{p}$ or $\tilde{e}$ and the rest of the states is not present in the system.

As with the state estimator in section 4.2, the main problem is to place the poles of the closed-loop observer in a way that enables the estimator to be as fast and noise free as possible. However, with the lack of one of the measured states and the addition of a state to be estimated, this proved a lot more challenging than previously.

Using the poles (52) as in the first estimator resulted in disasterous response, with the helicopter not even beeing able to leave ground, as well as emitting a rather omnious electrical growl. Concluding that this was a very poor basis for further tuning, we decided to base our estimator poles in a different relation to the controller poles $\mathbf{K}$. Instead of placing the poles close

to eachother on the negative real axis, the poles were distributed along the negative half plane by multiplying all the poles of **K** by a constant. This at least got the helicopter up in the air, although in a rapid, uncontrolled fashion. The following tuning process from this basis was largely done by trial and error, evaluating the change in response to a change in pole gain, one pole at a time.

Using this process, we were able to tune the estimator to a point where some readable data was obtainable, as can be seen in figure [13]. At first glance, its clear that the system is unstable. The elevation response is fair, but the system has a hard time tracking pitch. That being said, being able to control height fairly well enabled us to use these plots for finer tuning, whereas previously the system was not stable over long enough periods to get coherent plots.

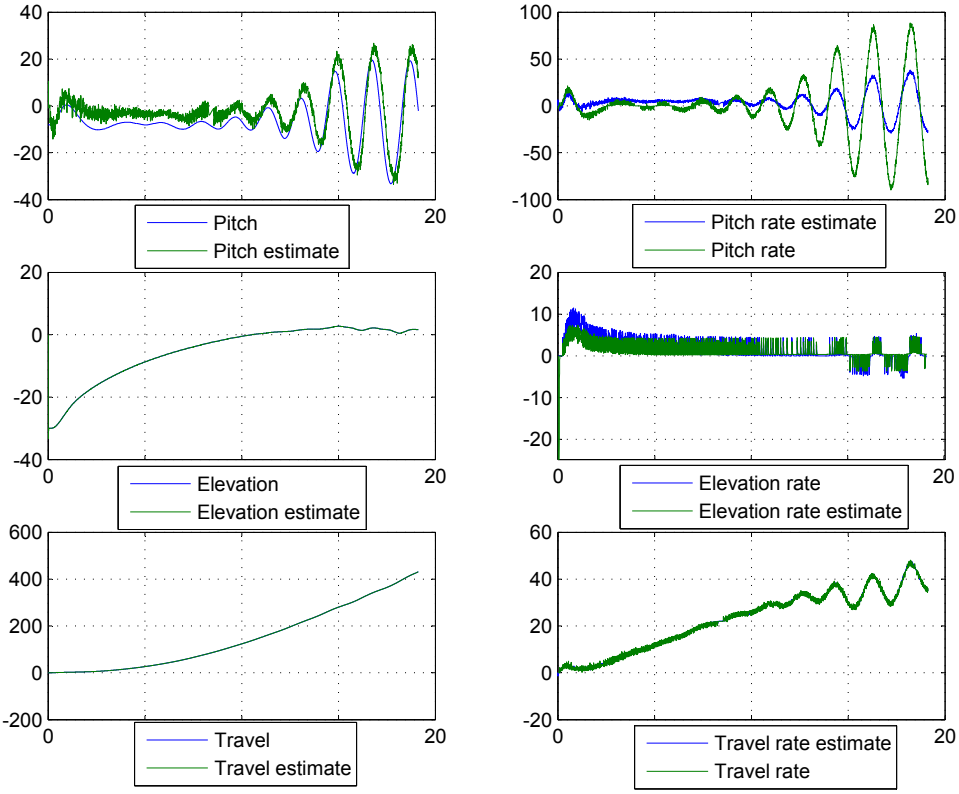


Figure 13: Estimated states plotted together with the measured, or numerically derivated values. Reduced measurement estimator poles = [-250, -70, -100, -50, -3, -1].

Figure [14] shows the result of finer tuning of the state estimator. While the pitch response is far from fast and accurate, the system is now stable,

if only barely. Comparing these plots with the ones in figure [13], some interesting relations are made clear. For instance, we observe an increase in the amount of noise in the pitch estimate for the stable system, however the pitch estimator is faster. This leads to the controller marginally being able to regulate the pitch rate.

Further tuning might have allowed for a more clean response, but seeing as we were able to grasp the main concepts behind the reduction of the measurement vector and its impact on the system response, we decided to end the tuning process here.

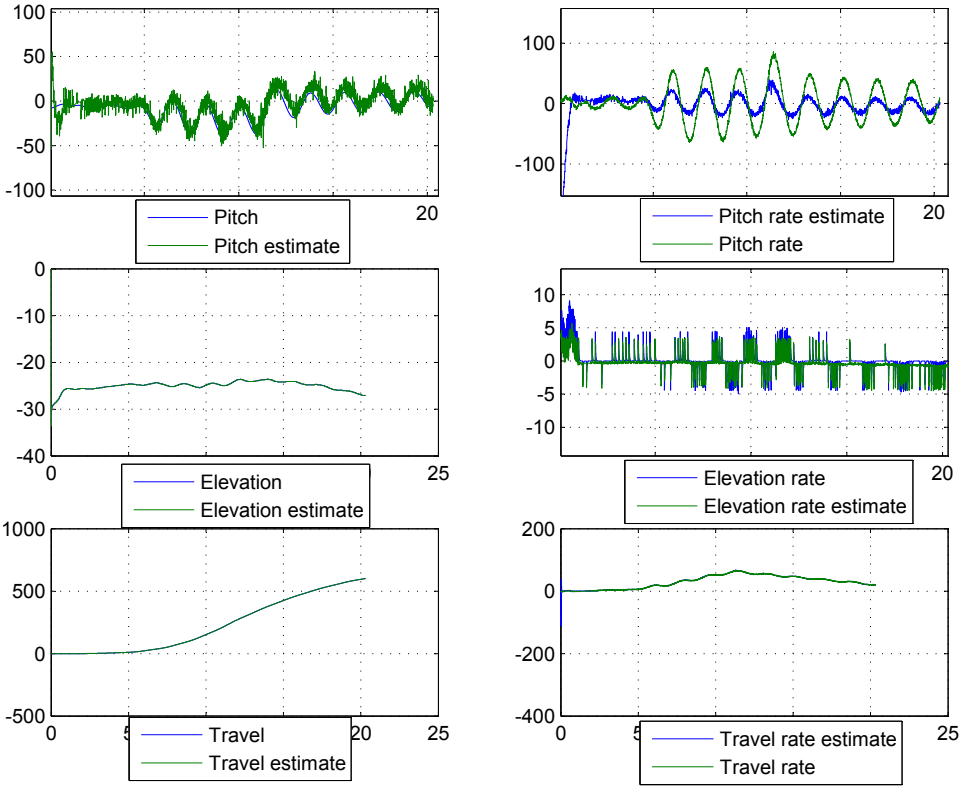


Figure 14: Estimated states plotted together with the measured, or numerically derivated values. Reduced measurement estimator poles = [-130, -150, -100, -60, -7, -0.5].

Another interesting comparison to make is between the response of the reduced measurement estimator in figure [14] and the full measurement estimator in figure [12]. There is a slight improvement in travel rate for the reduced estimator versus the full estimator, most likely due to finer tuning of the former. It is however abundantly clear that the pitch response is of a drastically lower quality in the estimator with reduced measurement vec-

tor. This can be caused by several factors. We know that the equations of motions the estimator system matrix $\mathbf{A}$ is based upon, are linearized around (19). During use, the helicopter system frequently exhibits a pitch far from the linearization point, hinting to that the linearized equation of motion for pitch might not be adequate for this type of performance requirement. Another consideration to make is that the pitch estimate is based upon the travel acceleration rate $\ddot{\tilde{\lambda}}$, as seen in (22c). The travel acceleration rate is an estimate of the travel rate $\dot{\tilde{\lambda}}$, which again is an estimate based on the measurement $\tilde{\lambda}$. Thus, we have two degrees of estimation for pitch $\tilde{p}$, and three degrees of estimation for pitch rate $\dot{\tilde{p}}$. With this in mind, it is not very surprising that the pitch response is rather poor. That being said, numerically derivating the travel measurement $\tilde{\lambda}$ three times to obtain travel rate $\dot{\tilde{p}}$ would probably be even worse as we could expect a lot more noise on the signals. Overall, we were able to obtain reasonable response for the controller-estimator configuration.

Conclusion

In this project, we have been able to use different controller designs to regulate a helicopter model. A mathmatical model for the helicopter system was developed, and the implications of linearizing this model were observed. Both an algebraic and trial-and-error approach to tuning have been utilized on monovariable and multivariable controller schemes, and the importance of model accuracy in calculating the controller gains was made clear. Generally, the challenges associated with tuning a multivariable system were experienced. Finally, we investigated the use of measured states versus estimated states for a controller, and observed the complications associated with reducing the measurement vector of the state estimator.

A MATLAB Code

A.1 Simulink Diagram

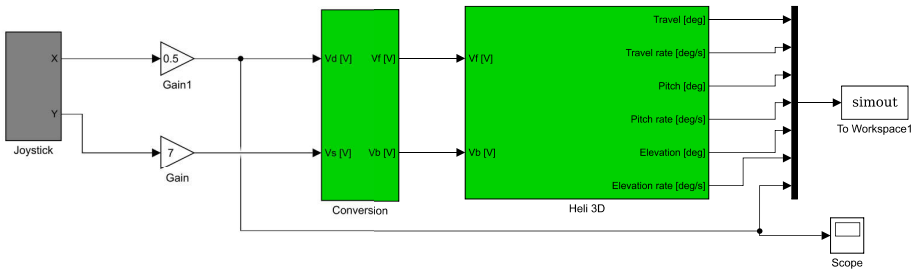


Figure 15: Problem 1.3: System with feed-forward gain

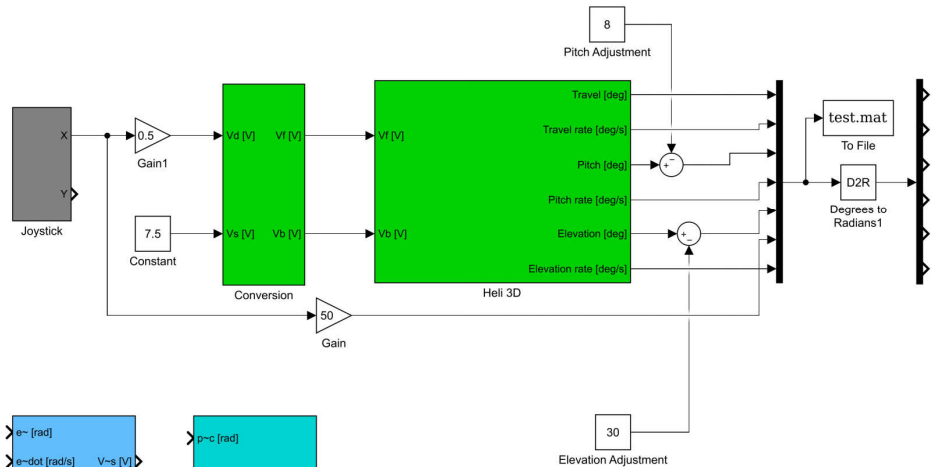


Figure 16: Problem 1.4: Added offset to the elevation and pitch. A converter from degrees to radians on all outputs is also included.

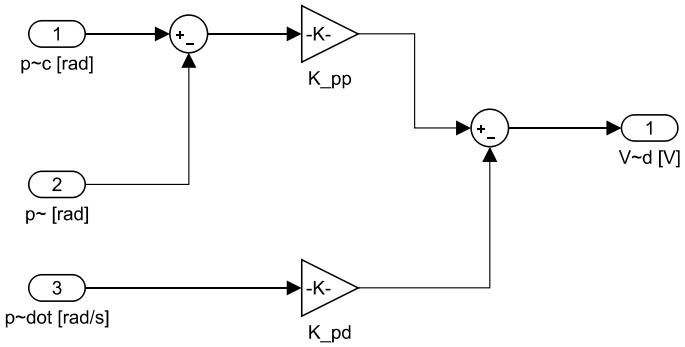


Figure 17: Problem 2.1: The PD-controller implementation for pitch.

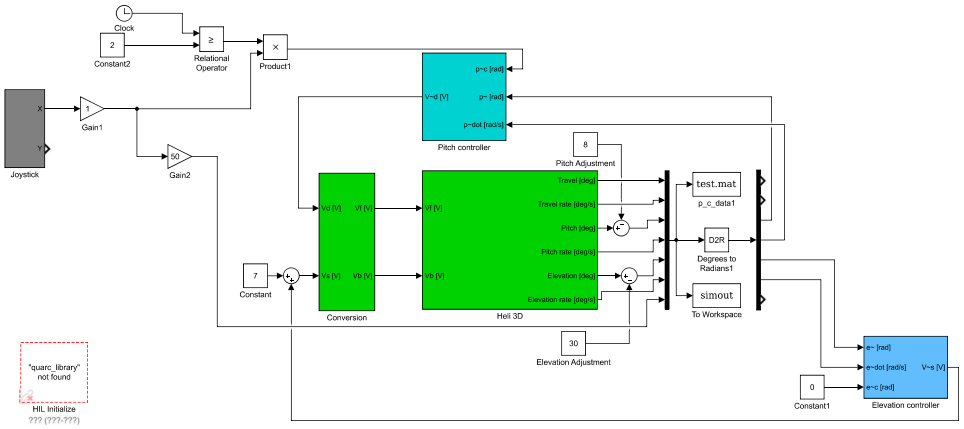


Figure 18: Problem 2.1: The PD-controller for pitch connected to joystick-x.

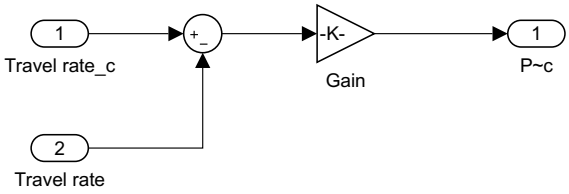


Figure 19: Problem 2.2: The P-controller implementation for travel rate.

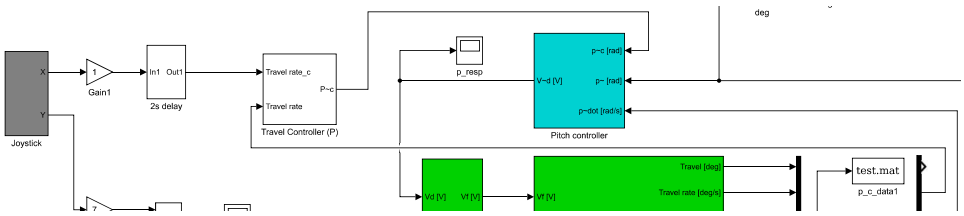


Figure 20: Problem 2.2: P-controller for travel included.

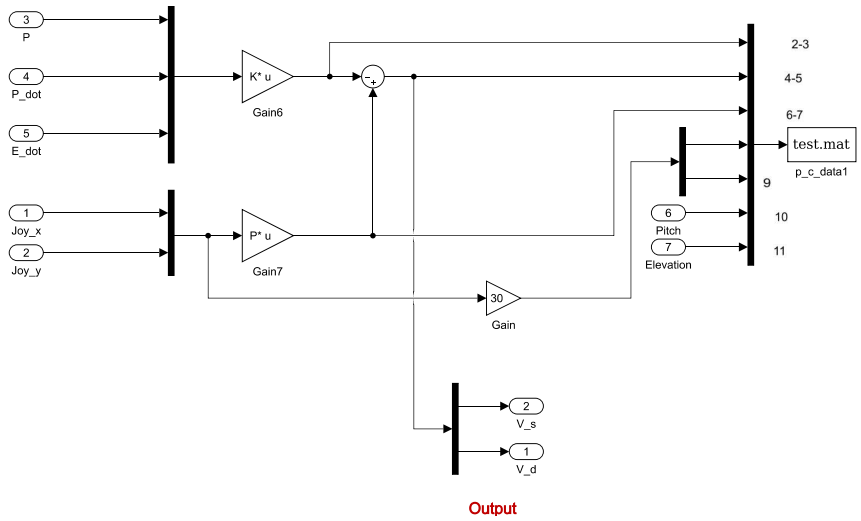


Figure 21: Problem 3.2: A LQR PD-controller for pitch and elevation rate. The gain blocks uses matrix multiplication.

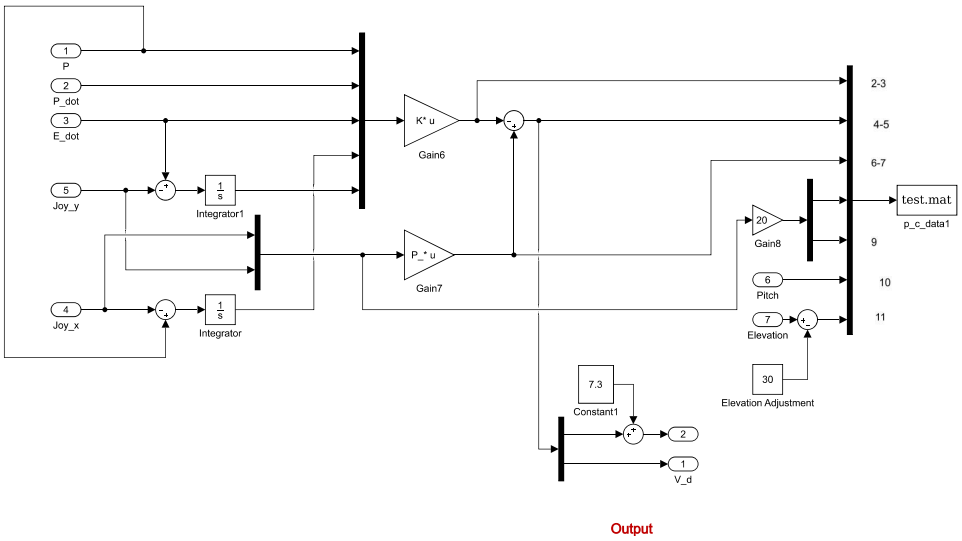


Figure 22: Problem 3.3: LQR PID-controller for pitch and elevation rate. The vector $\mathbf{x}$ in this case has two states extra in comparison to problem 3.2

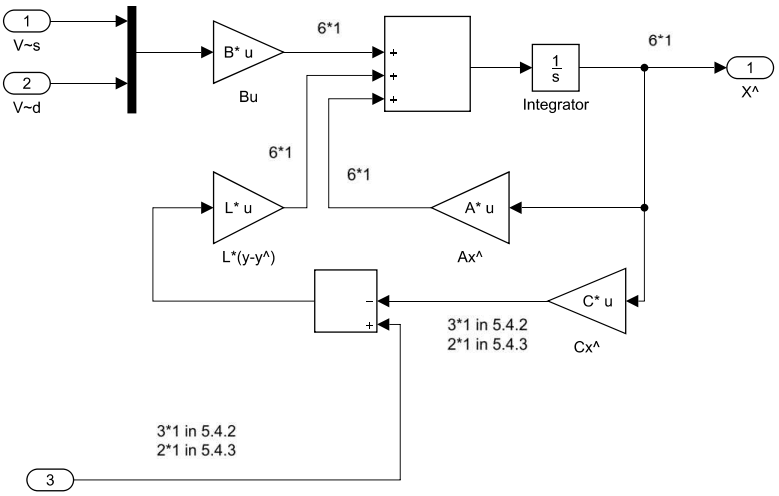


Figure 23: Problem 4: Design of the estimator.

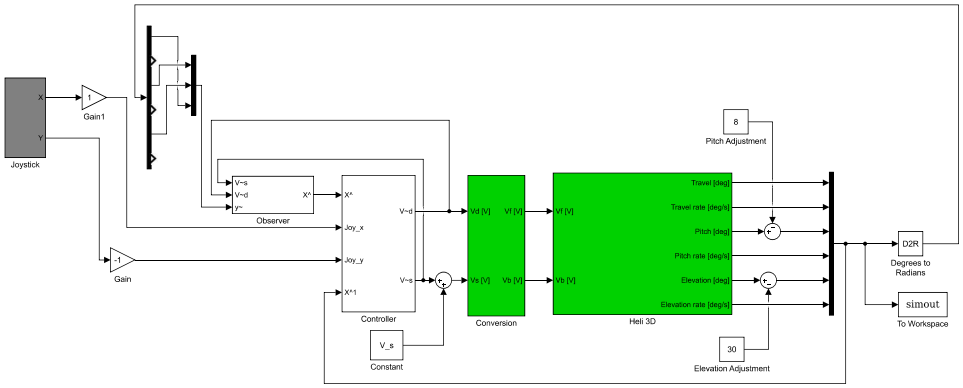


Figure 24: Problem 4.2: Estimator fed into the controller from problem 3.3

References

[1] Department of Engineering Cybernetics NTNU version 4.5 2015. *Helicopter lab assignment*. `https://ntnu.blackboard.com/webapps/blackboard/execute/content/file?cmd=view&content_id=_420173_1&course_id=_10887_1`. 2018.

[2] Chi-Tsong Chen. *Linear System Theory and Design*. Oxford University Press, Incorporated, 2013.

[3] John G. Proakis and Dimitris K. Manolakis. *Digital Signal Processing*. Pearson Education Limited, 2014.