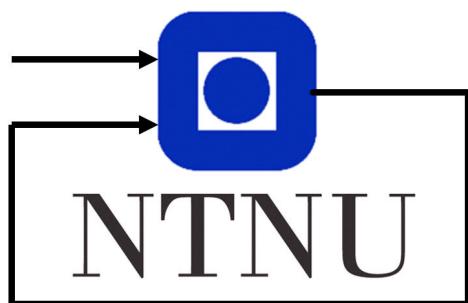


# Helicopter lab assignment

Group 5

Tarek El-Agroudi, 528146  
Finn Gross Maurer, 528070  
Lisa Blaser, 523544

November 23, 2021



Departement of Engineering Cybernetics

## **Abstract**

The goal of the this assignment is to test linear controller design and state estimation in practice. We start by implementing a PD controller for pitch on a linearised model. We continue by implementing a LQR state feedback controller for pitch and elevation rate, for a more intuitive method of tuning. We also study the effect of integral action. In the second part of the assignment, the focus is shifted to state estimation, and the measurements are extracted from a noisy IMU as opposed to an encoder. State estimation is performed with a Luenberger observer and a Kalman filter. Results are discussed and the differences between estimators are highlighted.

# Contents

<b>1</b>	<b>Introduction and Problem Description</b>	<b>1</b>
<b>2</b>	<b>Monovariable Control</b>	<b>3</b>
2.1	Hypothesis and test plan . . . . .	4
2.1.1	Hypothesis . . . . .	4
2.1.2	Test plan . . . . .	6
2.2	Results and Discussion - Monovariable control . . . . .	6
2.2.1	Real poles . . . . .	6
2.2.2	Complex conjugated poles . . . . .	10
2.3	Conclusion, Monovariable control . . . . .	12
<b>3</b>	<b>Multivariable control</b>	<b>14</b>
3.1	Hypothesis and test plan . . . . .	16
3.1.1	Tuning weighing matrices, no integral effect . . . . .	16
3.1.2	Integral action and feedforward matrix . . . . .	18
3.2	Results and Discussion . . . . .	18
3.2.1	Tuning weighing matrices . . . . .	18
3.2.2	Integral effect . . . . .	22
3.2.3	Tuning the feedforward matrix $\mathbf{F}$ . . . . .	24
3.2.4	Optimising the integral action LQR controller . . . . .	26
3.3	Conclusion . . . . .	27
<b>4</b>	<b>Luenberger observer</b>	<b>29</b>
4.1	Hypothesis and test plan . . . . .	31
4.2	Results and Discussion . . . . .	32
4.2.1	Gyroscope errors due to IMU placement . . . . .	32
4.2.2	Testing observer poles . . . . .	33
4.2.3	Minimal Luenberger observer . . . . .	37
4.3	Conclusion . . . . .	37
<b>5</b>	<b>Kalman filter</b>	<b>44</b>
5.1	Hypothesis and test plan . . . . .	46
5.2	Results and Discussion . . . . .	47
5.2.1	Tuning the covariance matrix $\mathbf{Q}$ . . . . .	48
5.2.2	Comparison to Luenberger observer . . . . .	49
5.3	Conclusion . . . . .	57
<b>Appendix</b>		<b>58</b>
<b>A Parameters and constants</b>		<b>58</b>
<b>References</b>		<b>59</b>

# 1 Introduction and Problem Description

Helicopters and drones have with the advancement of embedded technology and control theory become essential tools for society. As applications become autonomous, it is of paramount importance that the control systems for motion are robust and reliable. This report documents our response to the Helicopter Lab assignment in the course TTK4115 Linear System Theory, designed to give an introduction to such control.

We begin by giving a brief introduction to the helicopter, its modelling and any important preparatory model-related work. Figure 1 shows the helicopter setup. The three rotational axes  $p$ ,  $e$  and  $\lambda$  are pitch, elevation and travel, respectively. The configuration with zero pitch and horizontal elevation depicted in the figure is chosen as the equilibrium point for later linearisation since it can be viewed as the state the helicopter most often will reside in during flight time.

The helicopter is driven by two motors attached to the end of the arm. The motors were in the provided Matlab code set to saturate at  $\pm 5$  volts, and the two voltages that generate the forces  $F_f$  and  $F_b$  are denoted  $V_f$  and  $V_b$ . Since we in this lab will be controlling the helicopters pitch and elevation, it is useful to remap the voltage inputs into a differential input  $V_d = V_b - V_f$  and a summed input  $V_s = V_f + V_b$ . This way, the input  $V_d$  induces a torque about the pitch axis while  $V_s$  induces a torque about the elevation axis. It is evident from the equations of  $V_d$  and  $V_s$  that they will saturate at  $\pm 10$  volts.

When neglecting friction, drag forces and centripetal acceleration, modelling the helicopter becomes a straightforward implementation of Newtons second laws of linear and rotational motion. The resulting nonlinear model is given by

$$J_p \ddot{p} = L_1 V_d \quad (1a)$$

$$J_e \ddot{e} = L_2 \cos(e) + L_3 V_s \cos(p) \quad (1b)$$

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

where  $J_p$ ,  $J_e$  and  $J_\lambda$  are the moments of inertia about the pitch, elevation and travel axis respectively.  $L_1$ ,  $L_2$ ,  $L_3$  and  $L_4$  are real constants. For completeness, all model constants, both here and later in the report, are given in appendix A.

One final important model-related preparatory step was to alter the inputs such that zero input corresponded to the helicopter hovering at its equilibrium point with zero pitch in the horizontal position. For this, we defined

a control input  $\tilde{V}_s = V_s - V_{s0}$ . We then chose  $V_{s0}$  such that  $\tilde{V}_s = 0$  at the equilibrium point. This could be found experimentally by testing how much summing voltage  $V_s$  was needed for the helicopter to hover horizontally.

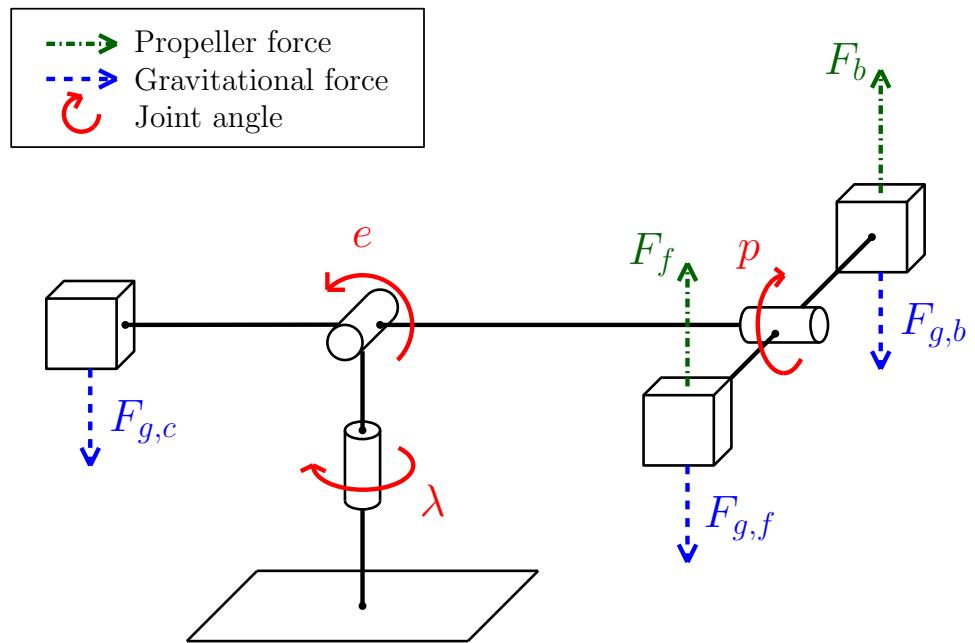


Figure 1: This figure shows the helicopter in its equilibrium point. Illustration is taken from the lab assignment [1].

## 2 Monovariable Control

In this section we present the first part of the project, monovariable control of the helicopters pitch. Since controlling only the pitch of the helicopter is impractical if it constantly collides with the table, a elevation controller was already implemented. This part focuses on implementing the pitch controller on top of this. Since it is much easier to study and control linear than nonlinear systems, the model from (1) was linearised around the equilibrium point shown in figure 1. The linearised model is given by

$$\ddot{p} = K_1 V_d \quad (2a)$$

$$\ddot{e} = K_2 \tilde{V}_s \quad (2b)$$

$$\ddot{\lambda} = K_3 p. \quad (2c)$$

where  $K_1$ ,  $K_2$  and  $K_3$  are real constants given by the model parameters and are summarised in appendix A. We make the observation that the differential equation describing the pitch angle is not affected by this linearisation. Thus the model is valid for any particular pitch, assuming that the nonlinear model was accurate. However, the elevation and travel dynamics are altered significantly. The effect of this is discussed in the hypothesis in section 3.

To control the pitch, a simple PD controller was implemented in simulink. PD controllers are often desirable for their stability properties, because the derivative term introduces a damping effect on the closed loop dynamics. Later sections of this report also study integral effect. Since  $V_D$  is the input that produces torque about the pitch axis, it is the best choice of control input for pitch control. The standard PD controller is then given by

$$V_D = K_{pp}(p_c - p) - K_{pd}\dot{p} \quad (3)$$

where  $p_c$  is the desired reference for the pitch angle. Intuitively, the parameter  $K_{pp}$  yields a control input proportional to the pitch error. The parameter  $K_{pd}$  yields a control input negatively proportional with the pitch rate, because the desired pitch rate is zero (we want the pitch to be stationary at its reference). This term can hence be viewed as controlling the damping in the closed loop system. Substituting the control law (3) into the model dynamics in (2a) gives the final closed loop system:

$$\ddot{p} = K_1(K_{pp}(p_c - p) - K_{pd}\dot{p}) \quad (4)$$

For experimenting with different configurations of pole values for the closed loop system, we used the Laplace transform to derive the transfer function from reference to pitch:

$$G(s) = \frac{p(s)}{p_c(s)} = \frac{K_1 K_{pp}}{s^2 + K_1 K_{pd}s + K_1 K_{pp}} \quad (5)$$

Since they allow for highly intuitive discussions, we derive expressions for  $K_{pp}$  and  $K_{pd}$  in terms of the poles of the closed loop system, given by the roots of the denominator of (5). Rearranging these pole expressions gives the following controller gains

$$K_{pp} = \frac{\lambda_1 \lambda_2}{K_1} \quad K_{pd} = -\frac{\lambda_1 + \lambda_2}{K_1}, \quad (6)$$

where  $\lambda_1$  and  $\lambda_2$  are the poles. The remainder of this section studies the effect of the closed loop poles  $\lambda_1$  and  $\lambda_2$  on the pitch control, a technique known as pole placement.

## 2.1 Hypothesis and test plan

### 2.1.1 Hypothesis

The complex plane can be split into three regions of interest where the poles of the closed loop system (4) can be placed: the left half-plane, the right half-plane and the imaginary axis. The poles  $\lambda_1, \lambda_2$  can either be real or complex valued, but if they are complex valued, they have to exist as a conjugated pair. This is because of the real physical nature of the system. We formulated a hypothesis for five general pole configurations.

#### Real-valued poles in the left half-plane

We observe from (6) that negative real poles results in positive  $K_{pp}$  and  $K_{pd}$ . For the system (4) this means that we both try to suppress  $\dot{p}$  to 0 as well as pushing the system towards the desired pitch reference. Thus this is a stable system, which agrees with the theory that poles in the left half-plane are stable. The ratio between  $K_{pp}$  and  $K_{pd}$  determines the compromise between letting the pitch rate remain zero and the pitch approach its reference. Since  $K_{pp}$  grows faster than  $K_{pd}$  for poles further in the left half plane, as is evident from (6), we expect a faster response towards the reference for more negative poles. Since the poles are real, we expect the pitch response to go exponentially towards the reference.

#### One pole in the left-half plane and one in the right half-plane

From (6) we can observe that  $K_{pp}$  will be negative. The negative gain implies the incitement of movement in the opposite direction of the reference. Because it will never reach its reference this way, the pitch control will be unstable and shoot to infinity in the opposite direction of the reference. Because the poles are real, the instability will be exponential and not oscillatory. We can through pole placement choose the sign of  $K_{pd}$ , but the system will nonetheless display instability due to the sign of  $K_{pp}$ .

#### Real-valued poles in the right half-plane

For right half plane poles, we expect the system to be unstable and the pitch

to shoot to infinity exponentially.  $K_{pd}$  will be negative, thus enhancing velocity. This time, however,  $K_{pp}$  is positive and the pitch shoots to infinity in the same direction as the reference, given zero initial conditions.

### Poles on the origin

If we have one pole, say  $\lambda_1$ , on the origin, equation (6) becomes

$$K_{pd} = -\frac{\lambda_2}{K_1}. \quad (7)$$

In the case where  $\lambda_2 > 0$ , we have  $K_{pd} < 0$  and the pitch control is unstable, because the system has a pole in the right half-plane and amplifies any error in velocity. From equation (3), if  $\lambda_2 = 0$ , the system becomes as if there were no controller at all. The helicopter is open loop unstable, and hence this case is also unstable. Finally, if  $\lambda_2 < 0$ , we have  $K_{pd} > 0$ , and the pitch control is marginally stable. This is because one of the poles is on the imaginary axis and the other in the negative half plane. Intuitively we expect the system to tend towards zero velocity, but not really care about the actual pitch control.

### Complex poles

The system consists of two poles  $\lambda_2 = \lambda_1^*$ . Because the response now includes the exponential of complex poles, which will include oscillatory terms, we expect the pitch to display oscillatory behavior. From (6) we see that the imaginary components only affect  $K_{pp}$ , which grows if they grow. Thus, larger imaginary parts are expected to produce faster control and more oscillations. As for the effect of the real part of the poles, it is closely related to the discussion of purely real poles:

- $\Re(\lambda_1, \lambda_2) < 0$ : the system is expected to be stable and will oscillate towards the reference.
- $\Re(\lambda_1, \lambda_2) > 0$ : the system is unstable, and the pitch is expected to violently oscillate towards infinity.
- $\Re(\lambda_1, \lambda_2) = 0$ : we expect the system to undergo periodic oscillations with constant amplitude. This means the pitch will oscillate back and forth around the reference value

### Theory vs practice

In all cases we expect the physical system to act not too far from the theoretical response. There are still a few things to consider in practise:

- Giving the poles large values results in large values for the control parameters. (3) shows that this will result in large voltage values. The physical motors have voltage limits, which means we expect the voltage to saturate. This can give weird behaviour like unexpected oscillations and potential instability.

- Air resistance, frictional effects and other disturbances will affect the helicopter. We do not have an integral effect in the controller, so we might expect a stationary offset from the reference.
- Other inaccuracies in the model can give unexpected behaviour.
- When we in the hypothesis say that the pitch will "shoot to infinity", we will observe the pitch increasing until it reaches the physical limit, which in the case of the helicopter is slightly past the 90 degrees mark.

### 2.1.2 Test plan

As we did not have the constants or motor information necessary to make specific plans for concrete pole values, we planned in terms of the general qualitative behaviour we wanted to test. The plan was to test at least 3 pole configurations (where applicable) for each of the theoretical cases in the hypothesis. These configurations will be selected with the purpose of testing both stable, unstable and marginally stable cases.

In order to select suitable ranges for pole selections, we will run a simulation of the system in Matlab to view the theoretical step response. Although not directly representative of reality, this will make it a lot easier to find good values to test as opposed to pure trial and error on the actual helicopter. The actual values tested are presented in section 2.2.

We wanted to test the different poles for a series of different reference behaviours. We therefore controlled the helicopter using a joystick for each pole configuration. However, in order to compare pole configurations in an unbiased way, we designed a reference signal which consisted of different sized steps in both directions, as well as testing edge cases like an almost 90 degrees reference for the pitch. The following results and discussion are based on these step responses.

## 2.2 Results and Discussion - Monovariable control

Table 1 summarises the poles tested as well as the figures in which their respective step response is plotted.

### 2.2.1 Real poles

For the case of two negative real poles, we systematically attempted the set of poles plotted in figure 2a. The resulting step responses are shown in figure 2b. More negative values of  $\lambda$  do as expected correspond to faster trajectories towards the reference. Because the eigenvalues are real, these trajectories do not oscillate but look like they approach the reference exponentially, which

Table 1: Pole placements and their corresponding figures.

<b>Real, negative poles</b>		
-0.5	-0.5	Figure 2
-1	-1	Figure 2
-0.5	-0.75	Figure 2
-10	-10	Figure 2
<b>Real, negative and positive poles</b>		
-0.5	0.5	Figure 4
-0.75	0.15	Figure 4
-0.75	0.0015	Figure 4
<b>Real positive poles</b>		
0.15	0.15	Figure 5
0.5	0.5	Figure 5
<b>Real single-zero poles</b>		
0	0.15	Figure 6
0	-0.5	Figure 6
<b>Complex poles with negative real part</b>		
-0.5+1.5i	-0.5-1.5i	Figure 7
-1+1.5i	-1-1.5i	Figure 7
-0.25+0.5i	-0.25-0.5i	Figure 7
-0.5+i	-0.5-i	Figure 7
<b>Complex poles with positive real part</b>		
0.5+1.5i	0.5-1.5i	Figure 8
0.15+1.5i	0.15-1.5i	Figure 8
0.015+1.5i	0.015-1.5i	Figure 8
<b>Complex poles with zero real part</b>		
0.15i	-0.15i	Figure 9
0.5i	-0.5i	Figure 9
0.3i	-0.3	Figure 9

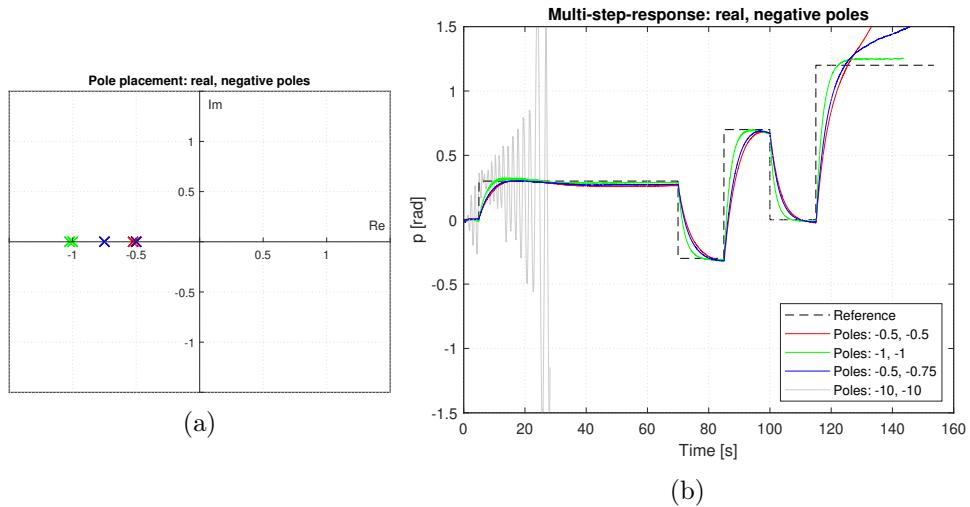


Figure 2: Pole placement (a) and step-response (b) for real, negative poles.

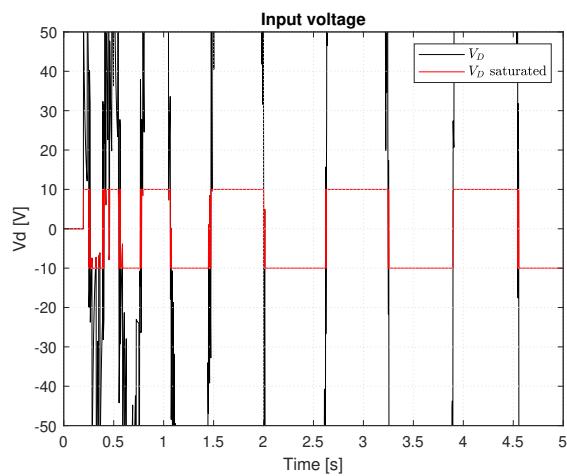


Figure 3: Input voltage,  $V_D$ , and saturated  $V_D$  for poles -10 and -10.

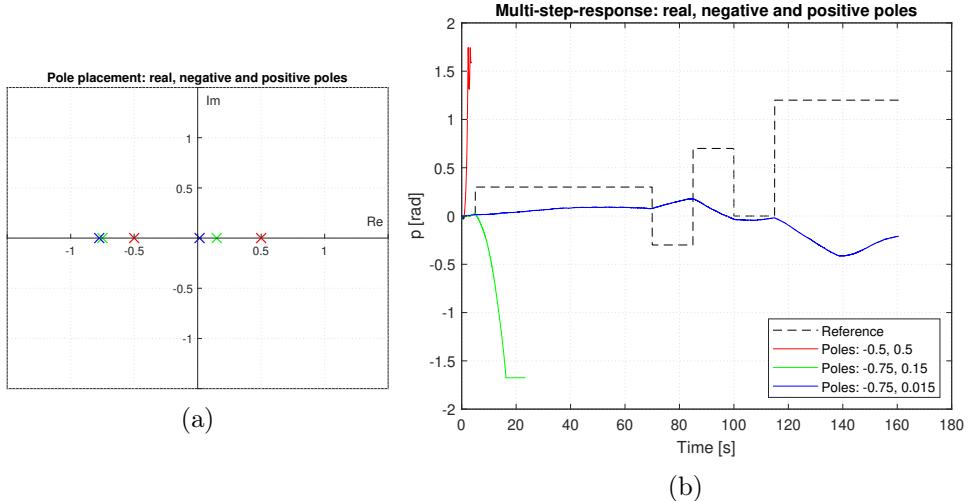


Figure 4: Pole placement (a) and step-response (b) for real, negative and positive poles.

again supports the theory. Looking closely at the response to the first step, we also observe a stationary offset from the reference. This was expected, and is likely due to disturbances or physical phenomena like friction that the model does not account for.

Another interesting observation is behaviour for the poles  $\lambda_1, \lambda_2 = -10$ , which corresponds to a controller gain of  $K_{pp} \approx 3165$ . The reason for the seaming instability can be seen from the plot of the input voltage  $V_D$  in figure 3. It is clear that the input voltage saturates due to high controller gain. The helicopter is not able to generate large enough thrust but the controller ignores this, resulting in growing oscillations.

One observation we did not expect is that two of the pole configurations fail at a pitch reference of 1.2 radians. We know that the helicopter can handle this case from the case of repeated poles at -1, and we know that the inputs did not saturate by looking at these. A reason for the overshoot could be model inaccuracies or disturbances, and the reason for two of the controllers not recovering could simply be that they are not tuned strongly enough.

As expected, moving one of the poles into the right half plane,  $\lambda_1 < 0 < \lambda_2$ , brings the system into instability, as can be seen from figure 4. This instability is more and more pronounced, as the unstable pole travels further into the right half plane. When the right half-plane pole is close to zero, the trajectory almost looks stable, though it does not follow the reference. This phenomenon is discussed in the next paragraph. Note that unstable trajec-

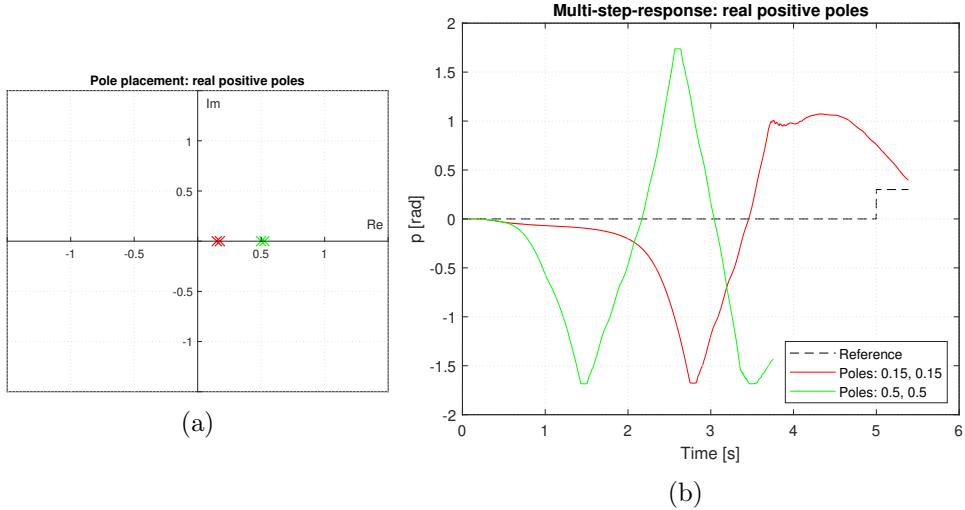


Figure 5: Pole placement (a) and step-response (b) for real positive poles.

tories stop when the helicopter reaches its physical limit or is about to hit the table. Similarly, when both poles are in the right half plane,  $\lambda_1, \lambda_2 > 0$ , the system is highly unstable from the start, as we can see from figure 5.

A rather more interesting case occurs when we set one of the eigenvalues, say  $\lambda_1 = 0$ , to zero and is plotted in figure 6. This automatically sets the controller gain  $K_{pp}$  to 0. Our choice of the second pole then only affects the derivative term of the controller equation. For  $\lambda_2 < 0$  we have  $K_{pd} > 0$ , and the controller will attempt to bring the derivative of the pitch,  $\dot{p}$ , to zero, independent of the actual value of the pitch. We can see this happening in figure 6 for the case of  $\lambda_2 = -0.5$ , where the pitch is held relatively around zero, without caring about the pitch signal. In contrast, the unstable case with  $\lambda_2 = 0.15 > 0$  produces a positive  $K_{pd}$  which attenuates pitch rate, and causes the pitch to violently shoot towards its physical limit and then bounces back before we stopped the simulation.

### 2.2.2 Complex conjugated poles

We then moved on to study complex poles, of the form  $\lambda_1, \lambda_2 = \alpha \pm \beta i$ .

For poles with negative real parts,  $\alpha < 0$ , the results are plotted in figure 7. Firstly, we notice that moving  $\alpha$  further into the left half plane for fixed  $\beta$ , for example moving from the green to the blue curve, reduces oscillations. In addition, a more negative  $\alpha$  generally gives a faster and less sluggish response, just like for the real case. An increase in  $\beta$  for fixed  $\alpha$ , for example moving from red to the green curve, aggressively increases oscillations. This fits with the hypothesis.

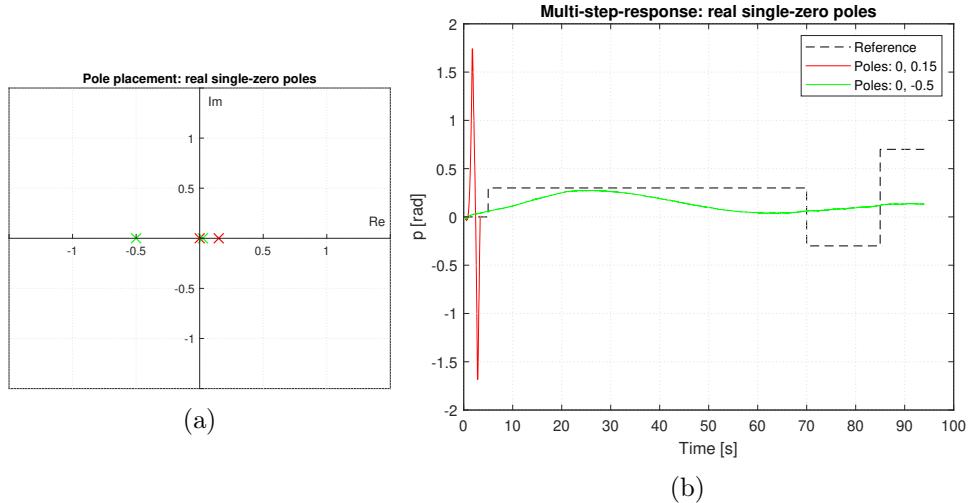


Figure 6: Pole placement (a) and step-response (b) for real single-zero poles.

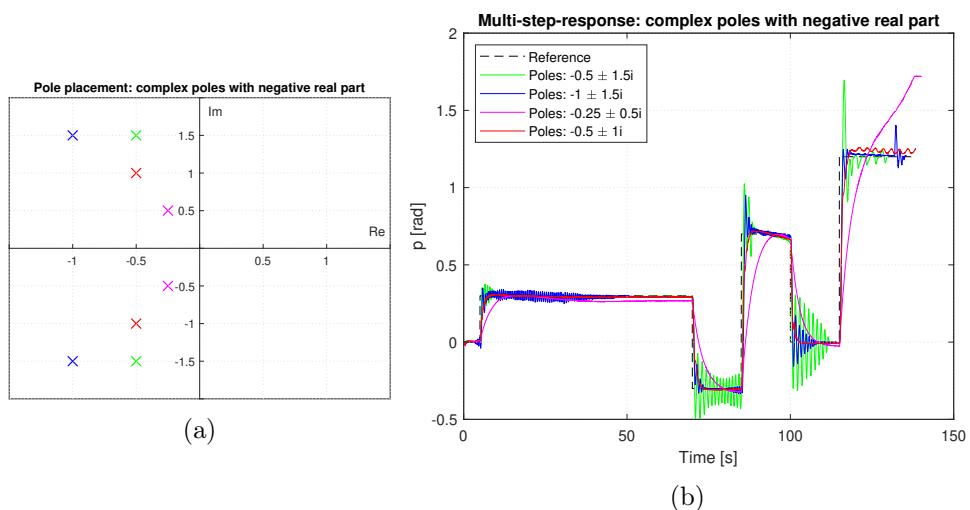


Figure 7: Pole placement (a) and step-response (b) for complex poles with negative real parts.

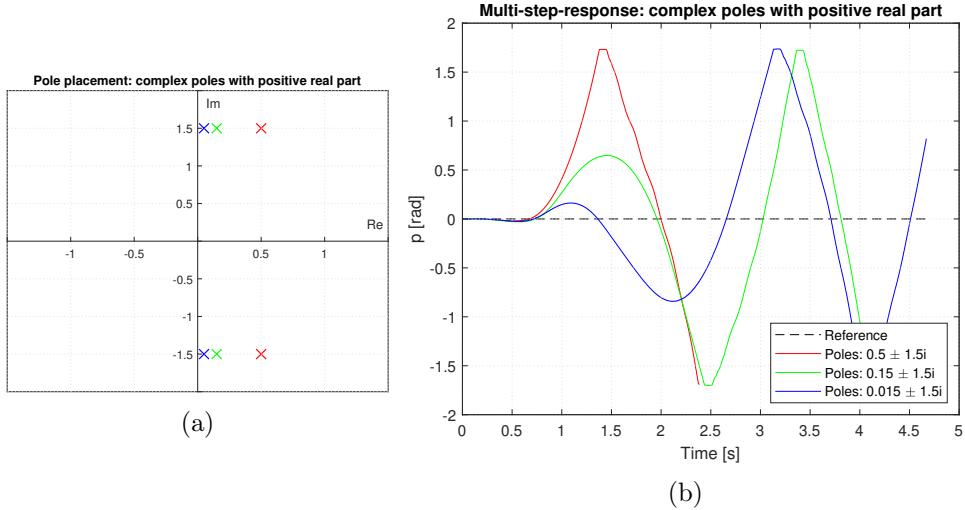


Figure 8: Pole placement (a) and step response (b) for complex poles with positive real parts.

Moving the real part of the poles into the right half plane,  $\alpha > 0$ , introduces instability to the system, just like before. However, this instability is now oscillatory in nature due to the complex poles. From figure 8, we clearly see that the helicopter undergoes a few oscillations, more and more powerful in nature, until the pitch reaches its physical limitations.

Finally, for the special case when the real part of the poles is zero,  $\alpha = 0$ , the results in figure 9 show oscillations as expected. Also, it is evident that, for small values of  $\beta$ , the helicopter does not produce sufficient torque to combat the gravitational pull, such that the helicopter falls into the vertical position. On the other extreme, high values of  $\beta$  result in the input voltage to saturate, causing instability as before. This contrasts the theory which predicted periodic oscillations with constant amplitude.

### 2.3 Conclusion, Monovariable control

The results confirmed the theory in the sense that left half plane poles are stable, and imaginary parts of poles introduce oscillations. The biggest differences in practice as opposed to the theory relate to the instability due to input voltage saturation at large controller gains. This occurs both when the real part of poles becomes very negative and when the imaginary part becomes large. There is also a stationary offset and a small overshoot due to model inaccuracies and disturbances, which don't occur in theory.

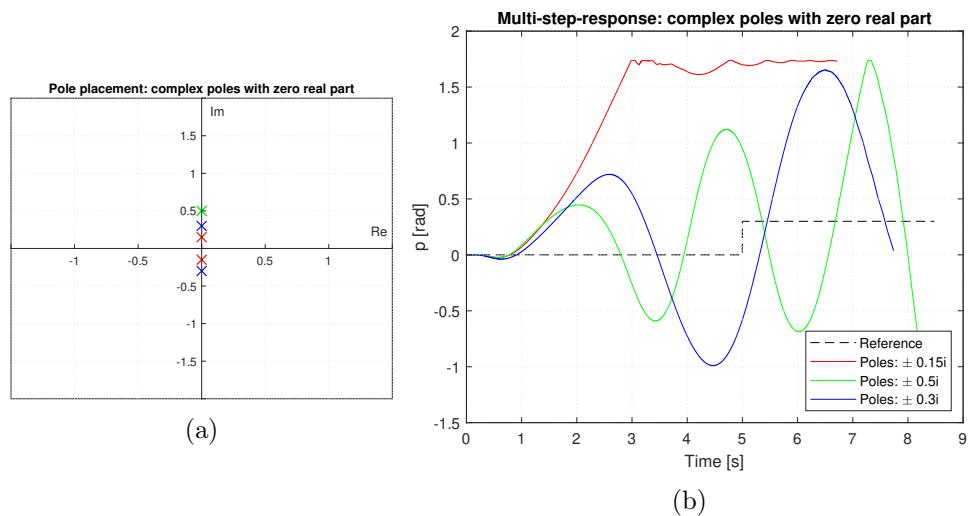


Figure 9: Pole placement (a) and step-response (b) for complex poles with zero real part.

### 3 Multivariable control

In the second part of the project we extend our controller to the multivariable case and control pitch and elevation rate, instead of only pitch. The motivation behind this is to test multivariable control laws. Also, instead of tuning the controller by pole-placement, we will apply the Linear Quadratic Regulator or LQR. The motivation behind this is to apply a more intuitive tuning method than the pole-placement method. With the LQR method we can directly prioritize which state deviations we want to reduce the most and how strong inputs we want the controller to use.

For multivariable control, it is useful to write the system in state space form. This is easily derived from (2) and is shown in (8).

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 0 & \frac{L_1}{J_p} \\ \frac{L_3}{J_p} & 0 \end{bmatrix} \mathbf{u} \quad (8)$$

where

$$\mathbf{x} = \begin{bmatrix} p \\ \dot{p} \\ \dot{e} \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} \tilde{V}_s \\ V_D \end{bmatrix}.$$

Whether such a state space system is controllable can be identified by checking whether the controllability matrix of (8) has full rank. This is easily checked with the matlab function `ctrb()`, which showed that the system is controllable.

To control this system, we implemented a state feedback control law, given by:

$$\mathbf{u} = -\mathbf{K}\mathbf{x}, \quad (9)$$

where  $\mathbf{K}$  is a  $2 \times 3$  gain matrix. By including pitch rate in the state space, we can design  $\mathbf{K}$  such that the control law acts like a PD controller for the pitch and a P controller for the elevation rate, both with a reference set to 0. To be able to follow a reference, we augment the controller (9) to include a reference feedforward term  $\mathbf{Fr}$ , where  $\mathbf{r}$  is the system reference  $[p_c \ \dot{e}_c]^T$  and  $\mathbf{F}$  is a  $2 \times 2$  matrix. This term depends on the state feedback and is designed such that the equilibrium point of the system is moved to the reference. We derived the equation for  $\mathbf{F}$  with respect to  $\mathbf{K}$  by substituting steady state values into the state space equation (8). That is, we let  $\dot{x} = 0$ ,  $\lim_{t \rightarrow \infty} p(x) = p_c$  and  $\lim_{t \rightarrow \infty} e(x) = \dot{e}_c$ . This gives the following equality:

$$\mathbf{F} \mathbf{r} = \begin{bmatrix} k_{11} & k_{13} \\ k_{21} & k_{23} \end{bmatrix} \begin{bmatrix} p_c \\ \dot{e}_c \end{bmatrix} \quad (10)$$

where  $\mathbf{F}_0$  is the optimal  $\mathbf{F}$ . The new control law is given by

$$\mathbf{u} = \mathbf{Fr} - \mathbf{Kx} \quad (11)$$

As mentioned in the introduction, we will tune the state feedback gain  $\mathbf{K}$  with the LQR method. The LQR defines a cost  $J$ , given by

$$J = \int_0^\infty \mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) + \mathbf{u}^T(t) \mathbf{R} \mathbf{u}(t) dt, \quad (12)$$

where the matrices  $\mathbf{Q}$  and  $\mathbf{R}$  assign weights to individual states and inputs, respectively. Depending on the choice of the weighing matrices, the LQR algorithm generates the state feedback matrix  $\mathbf{K}$  in (11) that minimises the cost  $J$  by setting its derivative to zero.

Finally, this part introduces integral action to the controller. Integral action involves controlling not only the error but also the integral of the error to zero. This can easily be done by augmenting the state space equation with two new integral states  $\zeta$  and  $\gamma$  to be the integral of the pitch error and elevation rate error, respectively:

$$\dot{\zeta} = p - p_c \quad \dot{\gamma} = \dot{e} - \dot{e}_c \quad (13)$$

The function of integral action is to remove stationary errors due to disturbances, and gradually increase the control input in the correct direction if a state error prevails over time. The new state space system with the augmented integral states is given by

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 0 & \frac{L_1}{J_p} \\ \frac{L_3}{J_p} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{u} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \mathbf{r} \quad (14)$$

where

$$\mathbf{x} = \begin{bmatrix} p \\ \dot{p} \\ \dot{e} \\ \gamma \\ \zeta \end{bmatrix} \quad \mathbf{u} = \mathbf{Fr} - \mathbf{Kx} \quad \mathbf{r} = \begin{bmatrix} p_c \\ \dot{e}_c \end{bmatrix}.$$

The value of the feedforward matrix  $\mathbf{F}$  must again be derived by setting in steady state values. Let  $\zeta_s$  and  $\gamma_s$  be the unknown stationary integral states. The resulting equality is:

$$\begin{bmatrix} 0 \\ \frac{L_1}{J_p}(p_c k_{21} + \dot{e}_c k_{23} + \gamma_s k_{24} + \zeta_s k_{25}) \\ \frac{L_3}{J_e}(p_c k_{11} + \dot{e}_c k_{13} + \gamma_s k_{14} + \zeta_s k_{15}) \\ -p_c \\ -\dot{e}_c \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{L_1}{J_p}(F_{21}p_c + F_{22}\dot{e}_c) \\ \frac{L_3}{J_e}(F_{11}p_c + F_{12}\dot{e}_c) \\ -p_c \\ -\dot{e}_c \end{bmatrix} \quad (15)$$

We particularly notice that optimal value of  $\mathbf{F}$  now depends on the steady state integral values, which are unknown! This is discussed in detail in the hypothesis section.

We then finalise by studying the interaction between the reference feedforward term and the integral action.

### 3.1 Hypothesis and test plan

#### 3.1.1 Tuning weighing matrices, no integral effect

The purpose of this part of the testing is to uncover the effect of the weighing matrices  $\mathbf{Q}$  and  $\mathbf{R}$  on controller performance. Since the feedforward matrix is not the variable of interest for this, the feedforward matrix was kept constant at its optimal value  $\mathbf{F}_0$  (10).

The constraints on  $\mathbf{Q}$  and  $\mathbf{R}$  are that they must be real, diagonal, symmetric, positive definite matrices. So we must choose the diagonal elements in  $\mathbf{Q}$  and  $\mathbf{R}$  positive and real.

Our hypothesis will, as in section 2 of this assignment, be of a qualitative nature. Our initial discussion is a general comparison between the effects of increasing/decreasing  $\mathbf{Q}$  or  $\mathbf{R}$ . We therefore need an initial configuration from which we can increase/decrease the matrices. We had two ideas for such initial configurations. These were to initialise them as identity matrices, or to use Bryson's Rule (16) to find appropriate values.

The second method was chosen because it takes into consideration the relative size of states and inputs, whereas the identity method does not. Bryson's parameters for  $\mathbf{Q}$  and  $\mathbf{R}$ , henceforth denoted by  $\mathbf{Q}_b$  and  $\mathbf{R}_b$ , depends of the maximum acceptable value of the states and inputs, which can be hard to set right. Table 2 shows our educated guesses for these values and the corresponding Bryson parameters.

Table 2: Brysons parameters

$x_i$	Max. acceptable value	$Q_{ii}$
$p$	$\frac{\pi}{2}$ [rad]	0.405
$\dot{p}$	$\frac{\pi}{2}$ [rad/s]	0.405
$\dot{e}$	$\frac{\pi}{3}$ [rad/s]	0.912
$u_i$	Max. acceptable value	$R_{ii}$
$\tilde{V}_S$	10 [V]	0.01
$V_D$	10 [V]	0.01

$$Q_{b,ii} = \frac{1}{\text{maximum acceptance value of } x_i^2} \quad (16a)$$

$$R_{b,ii} = \frac{1}{\text{maximum acceptance value of } u_i^2} \quad (16b)$$

From this initial configuration, we hypothesise on the effect of changing the matrices relative to each other. Note that it is not the absolute value, but the ratio of the matrix elements that determines their priority relative to each other.

### Elements of $\mathbf{Q}$ and $\mathbf{R}$ as given by Brysons

We now allocate comparable importance/priority to both the states and inputs. The response will thus compromise between fast convergence and low input use.

### Increasing $\mathbf{Q}$ , decreasing $\mathbf{R}$

We allocate greater importance/priority to state convergence and thus expect faster convergence at the expense of greater inputs. Practically, we should not exaggerate this as the inputs will saturate.

### Increasing $\mathbf{R}$ , decreasing $\mathbf{Q}$

We allocate greater importance/priority to keeping the inputs low, and expect this to happen at the expense of slower convergence. This should not be exaggerated as the inputs may become too low to stabilize the pitch in time.

But the weighing matrices are not adjusted as a whole. Each element in  $\mathbf{Q}$  relates to a specific state, and each element in  $\mathbf{R}$  relates to a specific input (this is because the matrices are diagonal). We can thus alter the relative importance of states or inputs amongst themselves.

The effect is again expected to be that the states corresponding to larger elements in  $\mathbf{Q}$  will converge faster, and that inputs corresponding to larger elements in  $\mathbf{R}$  will be kept lower. Extreme differences in priority will again showcase themselves with the state/input of highest priority dominating the others, which may lead to saturation in the other inputs or slow convergence in other states.

### 3.1.2 Integral action and feedforward matrix

The purpose of this testing is to see how introduction of integral action affects the controller performance and how it can be tuned using the weighing matrices, as well as how the integral action and reference feedforward interact.

We now have the option to weight the new integral states  $\gamma$  and  $\zeta$  by changing their respective weights in the diagonal matrix  $\mathbf{Q}$ . Increasing these would correspond to generating an input that gives faster convergence if  $\gamma$  and  $\zeta$  to steady state. Quite counter-intuitively, this corresponds to a greater integral effect because deviations of the integral states from zero are "punished" more aggressively in order to keep these deviations low.

We see from equation (15), that without feedforward (setting  $\mathbf{F} = 0$ ), the steady state equality holds only if the steady state integral states fully compensate the reference state in lines 2 and 3 of the vectors. If, conversely, the integral steady states are 0, the feedforward matrix is the same as if there were no integral effect. Our hypothesis is that the integral states work to even out the final equality, such that they work harder if  $\mathbf{F}$  is smaller than the zero-integral  $\mathbf{F}_0$  and harder, but in the opposite direction, if  $\mathbf{F}$  is greater than  $\mathbf{F}_0$ . Ideally, we want the integrals to do as little work as possible. This is because the feedforward term takes instant effect while the integral effect takes time to act. We therefore expect the optimal  $\mathbf{F}$  to be the zero-integral one,  $\mathbf{F}_0$ .

## 3.2 Results and Discussion

Table 3 summarises the tests conducted as well as their corresponding figures.

### 3.2.1 Tuning weighing matrices

Figures 12a and 11a shows the response of the pitch and elevation rate, respectively, for three configurations of the  $\mathbf{Q}$  and  $\mathbf{R}$  matrices. Since the elevation rate signal was quite noisy, a low pass filtered version of the elevation rate is plotted. This was only done in this section of the report, since it

Table 3: Table showing different weighing of matrices.  $\mathbf{Q}_b$  and  $\mathbf{R}_b$  are given by Brysons rule.

<b>Weighing Q and R</b>		
$1\mathbf{Q}_b$	$1\mathbf{R}_b$	Figure 12, Figure 11, Figure 14
$10\mathbf{Q}_b$	$1\mathbf{R}_b$	Figure 12, Figure 11
$1\mathbf{Q}_b$	$10\mathbf{R}_b$	Figure 12, Figure 11
$100\mathbf{Q}_b$	$1\mathbf{R}_b$	Figure 13
$1\mathbf{Q}_b$	$100\mathbf{R}_b$	Figure 13
individual tuning		Figure 15
<b>Integral action</b>		
$Q_{44} = 1$	$Q_{55} = 1$	Figure 17, Figure 16
$Q_{44} = 0.1$	$Q_{55} = 0.1$	Figure 17, Figure 16
$Q_{44} = 10$	$Q_{55} = 10$	Figure 17, Figure 16
$Q_{44} = 0.1$	$Q_{55} = 10$	Figure 18
<b>Feed forward</b>		
$\mathbf{F} = \mathbf{F}_0$		Figure 19, Figure 20
$\mathbf{F} = 0.5\mathbf{F}_0$		Figure 19
$\mathbf{F} = 2\mathbf{F}_0$		Figure 19

does not affect the discussion. The default case was given by Brysons rule, and the results are most clearly seen from the pitch response. Although the response reaches its reference in the time allocated by the step signal, there is both a small overshoot and a significant stationary error. The elevation rate response, however, is not as good. Instead of approaching the reference and staying in its vicinity, the elevation rate briefly shoots in the direction of the reference, before approaching zero with significant overshoot. A reason for this might be model inaccuracies due to the linearisation from (1) to (2). Far from the equilibrium, this model is less accurate and the controller may therefore not be able to generate the correct torque. Another reason may be physical limitations of keeping the helicopter rising at high elevations.

More specifically, one such physical factor could be the model inaccuracy provided by the attachment of the helicopter arm itself, highlighted in red in figure 10. This attachment means that, at high elevation, the arm of the helicopter is actually longer than what it is in the model. This means the helicopter, using the now faulty model, generates a torque about the elevation axis that is too small to combat the gravitational torque, which is increased due to the longer arm. The remedy for this error, as well as the one originating from linearisation, is introduced in section 3.2.2, with integral action.

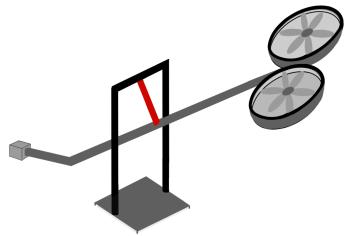


Figure 10: Sketch of the helicopter attachment, highlighted in red.

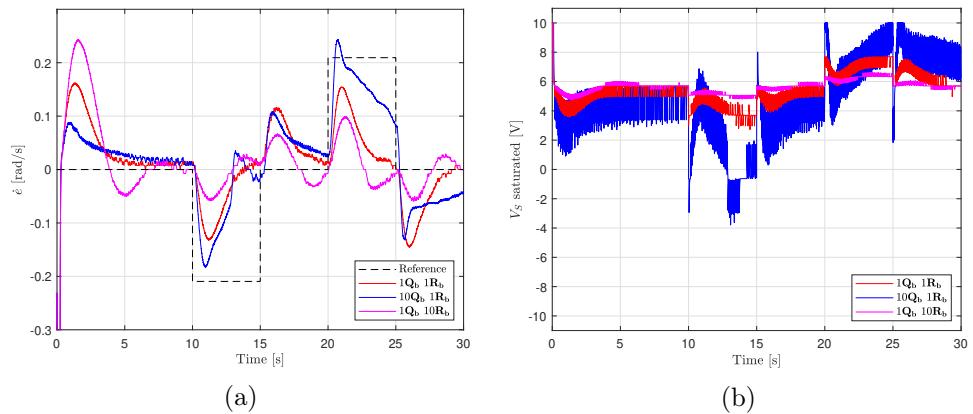


Figure 11: Elevation rate response (a) and saturated input  $V_S$  (b) for different  $\mathbf{Q}$  and  $\mathbf{R}$ .

The other two configurations of the weighing matrices, besides Brysons rule, where respectively increasing  $\mathbf{Q}$  or  $\mathbf{R}$  by a factor of 10, keeping the other matrix unchanged. Here our results clearly support the hypothesis. Looking at the pitch response in figure 12a again, it is clear that increasing the relative size of  $\mathbf{R}$  leads to worse control. The overshoot is greater, and so is the stationary error. This is because the control law now punishes large inputs more than state deviations, relatively, and hence no longer produces high enough inputs to counteract the overshoot quickly or to combat the stationary error. This is also confirmed by the plot of the input differential voltage in 12b. When, instead, the  $\mathbf{Q}$  matrix was amplified relative to  $\mathbf{R}$ , control turned much better. Both overshoot and stationary offset are massively reduced, because the controller now generates greater input voltages to combat these phenomena. The difference in input can again be seen from 12b. For the case of elevation rate control, this trend is again confirmed in figure 11a. For the case of larger  $\mathbf{Q}$ , the elevation rate is even held in the correct sign for almost the entire step period. However, elevation rate control is still relatively poor.

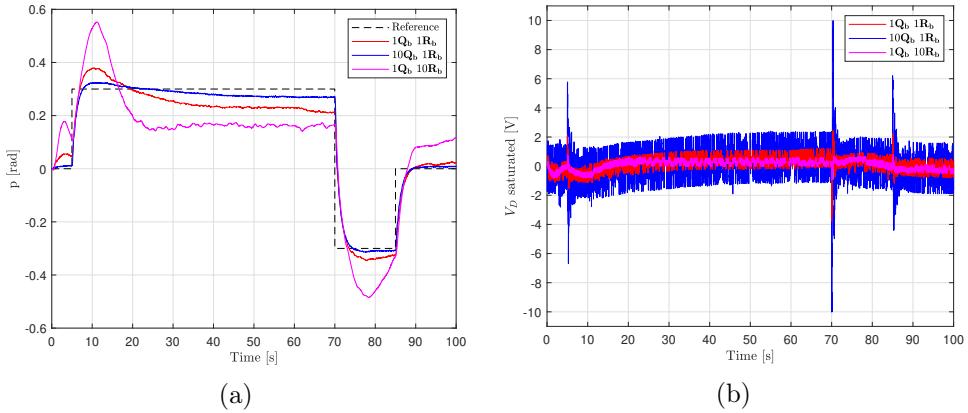


Figure 12: Pitch response (a) and saturated input  $V_D$  (b) for different  $\mathbf{Q}$  and  $\mathbf{R}$ .

Figure 13 shows the pitch response when the relative sizes of  $\mathbf{Q}$  and  $\mathbf{R}$  are further increased. While increasing the  $\mathbf{Q}$  matrix by another factor of 10 seems to keep the controller good, the plot of input voltage clearly shows very large inputs. This was shown in the first part of this report to be problematic in terms of stability, but it seems this case is still fine. If energy, power or cost considerations were important, this controller would be too aggressive and one might opt for a more moderate  $\mathbf{Q}$  matrix. The converse case where  $\mathbf{R}$  is amplified even more does not work at all. The pitch simply plummets until the helicopter reaches its physical limits. This is because the  $\mathbf{R}$  matrix punishes large inputs so much that the controller no longer can

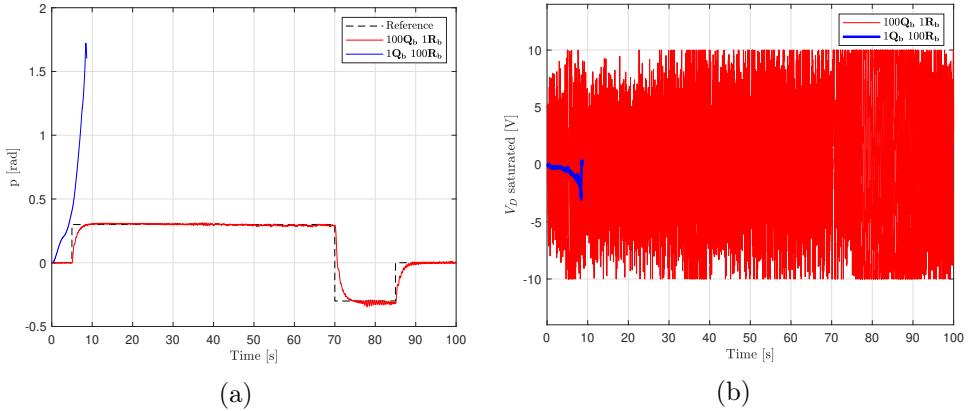


Figure 13: Pitch response (a) and saturated input  $V_D$  (b) for different weightings of  $\mathbf{Q}$  and  $\mathbf{R}$ .

supply the needed voltage to combat the gravitational pull on the helicopter. Since a real helicopter would be flown with simultaneous references in both pitch and elevation rate, tests were conducted where both states of interest were given references at the same time. The result for the default case of  $\mathbf{Q}$  and  $\mathbf{R}$ , from Brysons rule, is displayed in figure 14. Initially, it is evident that the neither pitch nor elevation rate following is ideal, and that pitch following is significantly worse in the case where the elevation rate also has a reference to follow. Furthermore, elevation rate control is catastrophically bad, with almost no indication of following the reference. The reason for this is likely a conflict of interest between the pitch and elevation rate control inputs.

However, an analysis can still be made by increasing the individual weights connected to the pitch and elevation rate, shown in figure 15. It is clear that prioritising the pitch, which was done by increasing the pitch term in the  $\mathbf{Q}$  matrix by 10, and leaving the other entries of  $\mathbf{Q}$  and  $\mathbf{R}$  unchanged, results in much better pitch following response. However, prioritising elevation rate produced little notable change in the elevation rate response, apart from following the zero reference better. However, the pitch response is slower in the case where more weight is assigned to the elevation rate, confirming that it is no longer as important in the control law.

### 3.2.2 Integral effect

The analysis of adding integral states and the effect of their corresponding terms in  $\mathbf{Q}$  was, for the sake of consistency, analysed from the basis of Brysons values for the other terms in  $\mathbf{Q}$  and  $\mathbf{R}$ . Since using Brysons rule to initialise the integral state weights  $Q_{44}$  and  $Q_{55}$  is less intuitive, these were

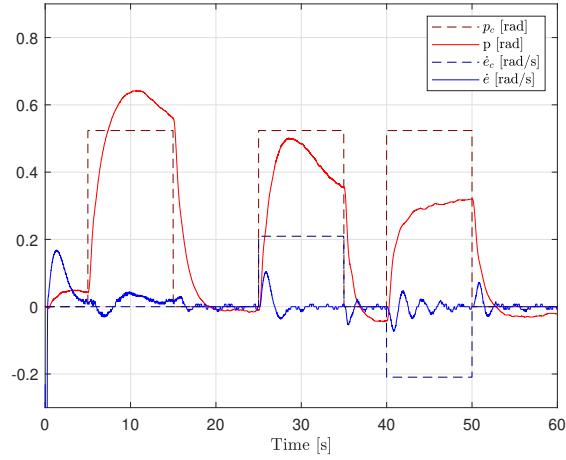


Figure 14: Pitch and elevation rate response for simultaneously testing, with **Q** and **R** as brysons.

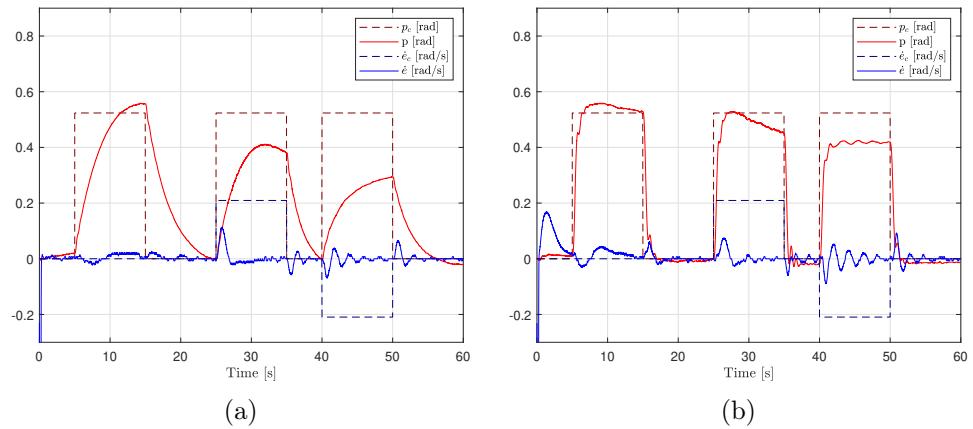


Figure 15: Pitch and elevation rate response with elevation rate prioritized (a) and pitch prioritized (b).

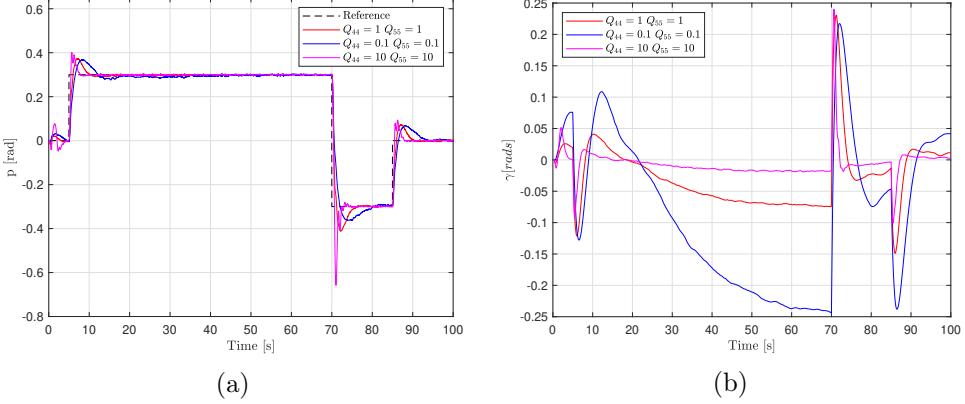


Figure 16: Pitch response (a) and gamma (pitch error integral) (b) for different weighings of  $\mathbf{Q}$  and  $\mathbf{R}$ .

initialised to be 1. Figures 16 and 17 show the responses for pitch and elevation rate along with their respective integral states, for three configurations of  $Q_{44}$  and  $Q_{55}$ . In the case of pitch, we firstly notice the disappearance of the stationary error. This is the most important feature of adding integral action to the controller. The effect of different  $Q_{44}$  on the controller is best seen by looking at the integral state, which grows largest for smallest  $Q_{44}$ . Quite counter-intuitively this corresponds to less integral effect, because the controller does not punish deviations from zero integral as violently. For the pitch response, this is showcased by faster responses and more aggressive overshoots for larger  $Q_{44}$  and opposite for smaller. Also for the elevation rate plots, we see a great improvement. Control was so much better, in fact, that we had to change the step response such that the helicopter would not collide with the table on its way downwards. We see that the stationary error is reduced the fastest for larger values  $Q_{55}$ , at the expense of more overshoot. This behaviour is confirmed by the plots of the integral state, which converge the fastest towards zero.

Finally, it was again important to observe the case were the pitch and elevation rate get references simultaneously. The result is shown in 18. When only the pitch has a nonzero reference, control is quite good, and the integral effect manages to compensate for the stationary offset. However, once the elevation rate and pitch rate have a nonzero reference at the same time, control fails. Again, this is attributable to a conflict of interest between the two controller inputs.

### 3.2.3 Tuning the feedforward matrix $\mathbf{F}$

We then proceeded to analyse the impact of the feedforward matrix  $\mathbf{F}$ . For this, Bryson's weighing matrices, with the integral state weights  $Q_{44}$  and  $Q_{55}$

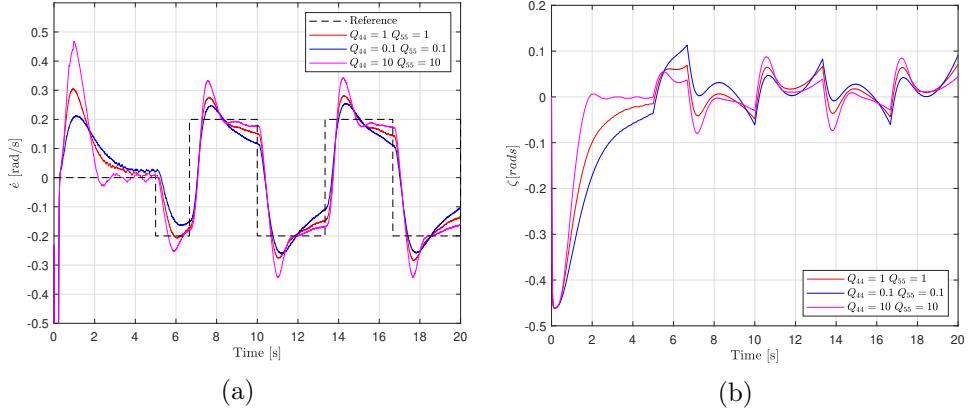


Figure 17: Elevation rate (a) and and input  $V_s$  (b) for different weighings of  $\mathbf{Q}$  and  $\mathbf{R}$ .

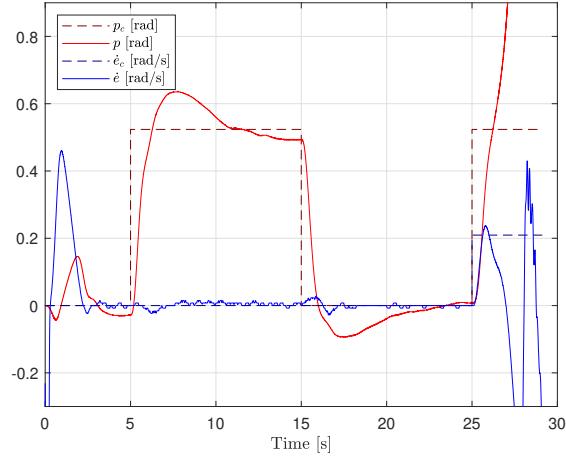


Figure 18: Pitch and elevation rate response for  $Q_{44} = 0.1$  and  $Q_{55} = 10$  with simultaneous testing.

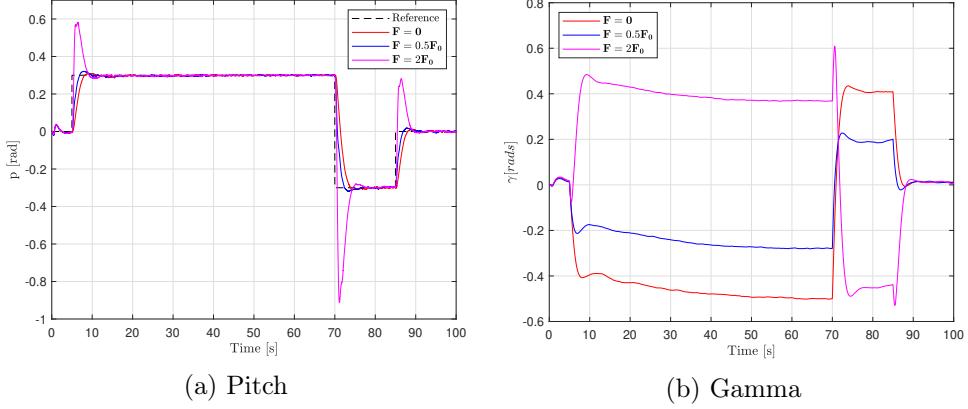


Figure 19: Pitch (a) and gamma (b) response for different  $\mathbf{F}$ .

set to 1 was used as default. Figure 19 shows the resulting pitch response as well as the pitch integral state for a series of  $\mathbf{F}$ . The best results are obtained for  $\mathbf{F} = \mathbf{F}_0$ . Doubling  $\mathbf{F}$  gives the system an aggressive overshoot, since we now are adding a too large constant to the input. This is equivalent to doubling the reference in the feedforward term. The feedforward term pushes the pitch towards twice the reference and, as a consequence, the integral state grows heavily in the opposite direction than without feedforward term, in order to bring the system to equilibrium. With  $\mathbf{F}$  halved, the response is slower, but much more acceptable than that of doubling. This is because, for a helicopter, it is much better to have a slower approach towards the reference than a massive overshoot. These observations confirm the hypothesis that the integral effect works to compensate the final error, after the feedforward term  $\mathbf{F}$  has acted.

### 3.2.4 Optimising the integral action LQR controller

Having analysed a multitude of cases, it was desired to determine a good compromise. This was achieved by the following  $\mathbf{Q}$ ,  $\mathbf{R}$  and  $\mathbf{F}$  matrices:

$$\mathbf{Q} = \begin{bmatrix} 0.805 & 0 & 0 & 0 & 0 \\ 0 & 0.605 & 0 & 0 & 0 \\ 0 & 0 & 0.912 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 5 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} \quad \mathbf{F} = 0.85 \begin{bmatrix} k_{11} & k_{13} \\ k_{21} & k_{23} \end{bmatrix}$$

The resulting step responses are plotted in figure 20. We have here tuned experimentally based on our findings as initial guesses. The  $\mathbf{F}$  matrix was chosen moderately smaller than  $\mathbf{F}_0$  value to reduce overshoot.  $\mathbf{R}$  was chosen small compared to  $\mathbf{Q}$  as to allow aggressive inputs, and integral state weights

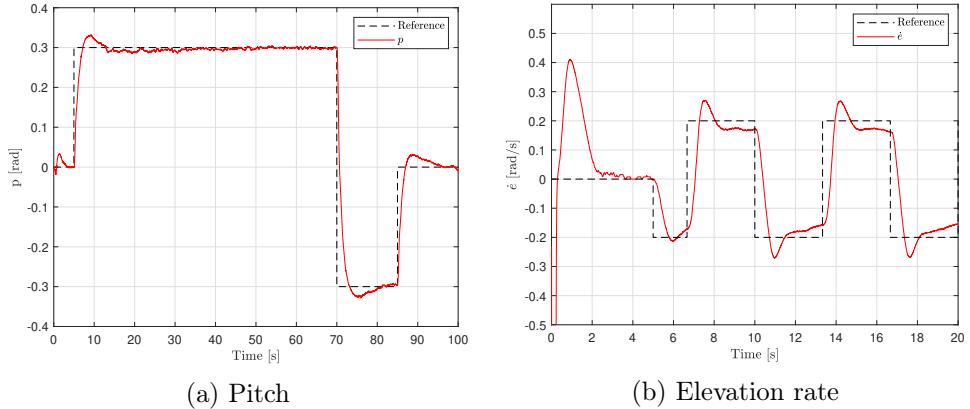


Figure 20: Pitch (a) and elevation rate (b) response for the optimal  $\mathbf{F}$ .

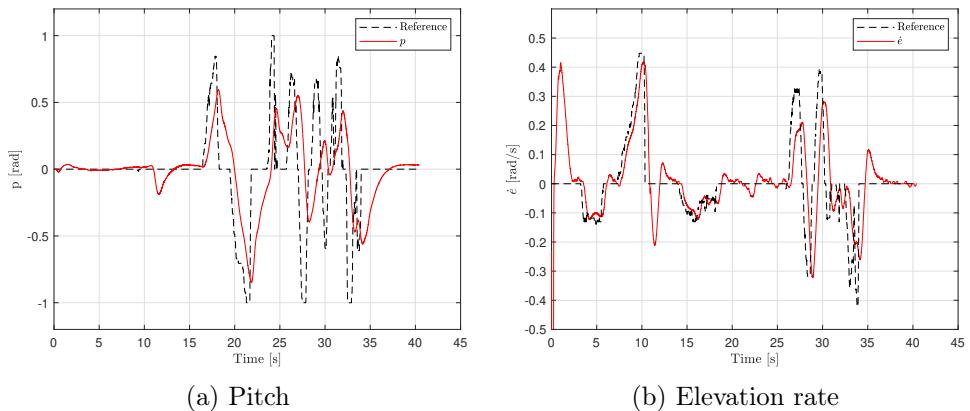


Figure 21: Pitch (a) and elevation rate (b) response for joystick input as reference and the optimized LQR controller.

were kept small such that their main function is removing stationary offset without causing too much overshoot. In order to test whether this controller behaves well for non-step responses, it was tested with simultaneous pitch and elevation references given by a joystick. The results are shown in figure 21 and show decent control except when the reference varies very quickly.

### 3.3 Conclusion

The experimentation has confirmed that the size of the elements in  $\mathbf{Q}$  of LQR determine the importance or priority assigned to their respective states in the resulting state feedback controller. Increasing these entries produces faster responses at the expense of larger inputs, which eventually leads to saturation and instability. Larger  $\mathbf{R}$  entries in turn prioritises keeping respective inputs low, at the expense of slower control and instability, in the case where  $\mathbf{R}$  becomes very large.

LQR with no integral effect did not produce satisfactory regulation of pitch and elevation rate due to large stationary offsets and failure of regulating elevation rate at large elevation. These issues were remedied by the integral action.

Finally, the feedforward matrix  $\mathbf{F}$  helps the system quickly approach the reference as it acts immediately. However, it must not be increased too much as this will bring a large overshoot. For this reason, the optimal LQR controller implements a slightly lower value than the theoretical ideal  $\mathbf{F}$ .

## 4 Luenberger observer

In the previous tasks we have been using encoders to measure the states. They give very precise measures of the states, and can directly be used in the previous controllers. The main disadvantage to use the encoders is that the helicopter needs to be stuck to the ground, which is not what helicopters are made for. Therefore we use an Inertial Measurement Unit (IMU), consisting of a gyroscope and an accelerometer, to measure the states instead. This introduces significant measurement noise. In the third part of the project, we therefore shift our focus from synthesising and tuning the controller to instead estimating the system states in the presence of fewer and/or noisy measurements. In this section, state estimation is done using the Luenberger observer.

The controller used in this section will be the optimal LQR controller found in section 3. However, since estimation is in focus, we also include estimates of elevation and travel rate, even though these are not used in the controller. We don't include the travel itself, since there is no way of estimating it from the IMU measurements. The new state space system is given by

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ K_3 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ 0 & 0 \\ K_2 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{u} \quad (17a)$$

$$\mathbf{y} = \mathbf{Cx} + \mathbf{n}, \quad (17b)$$

where

$$\mathbf{x} = \begin{bmatrix} p \\ \dot{p} \\ e \\ \dot{e} \\ \dot{\lambda} \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} \tilde{V}_s \\ V_D \end{bmatrix}.$$

$\mathbf{n}$  is the measurement noise and  $\mathbf{C}$  contains which states are included in the output.

An important question in state estimation is how to make the system observable with the least amount of measurements. To make the system observable, minimally  $\dot{\lambda}$  and  $e$  have to be measured. The resulting  $\mathbf{C}$  matrix is shown in (18). It can be noticed from (2) that  $p$  can be obtained by differentiating  $\dot{\lambda}$ , which again can be differentiated to obtain  $\ddot{p}$ . By differentiating  $e$  we can obtain  $\dot{e}$ . Thus by measuring those two states, all the other states can be calculated indirectly. The observability matrix will have full rank, which means that the system is observable.

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (18)$$

Because there is no interconnection between  $e, \dot{e}$  and  $p, \dot{p}, \dot{\lambda}$ , it will be impossible to obtain an observable system with only a single measurement. There exists no other combination of two states that will make the system observable, since elevation rate and pitch rate don't uniquely determine elevation and pitch. Also, pitch only determines the derivative of travel rate, which in turn does not determine travel rate uniquely.

A Luenberger observer estimates states by blending the prediction of the model in (17) with the measurement  $\mathbf{y}$ , according to the following dynamics:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{L}(\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}), \quad (19)$$

where  $\dot{\hat{\mathbf{x}}}$  contains the estimates of the states  $\mathbf{x}$  in (17). First all states will be measured, by choosing  $\mathbf{C}$  as the identity matrix. Afterwards, we will try to implement a controller by measuring only the two necessary states.

As for how we will tune the blending factor  $\mathbf{L}$ , this will be done by pole placement. More specifically, it will be determined by placing the poles of the closed loop estimate error dynamics. These are simply derived by defining an estimate error  $\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}$ , differentiating it, and substituting the state dynamics (17) and estimate dynamics (19). The results are shown in (20).

$$\dot{\mathbf{e}} = (\mathbf{A} - \mathbf{LC})\mathbf{e} - \mathbf{Ln} \quad (20)$$

Note that the dynamics contain a term where the blending factor  $\mathbf{L}$  is multiplied with the measurement noise. The effects of this are discussed in the hypothesis and discussion sections. If we choose  $\mathbf{A} - \mathbf{LC}$  to have real negative poles,  $\mathbf{L}$  can always be chosen such that the system matrix becomes diagonal. This means each pole will relate to one specific state estimation error. This can be useful for a manual tuning of the observer gain. With fewer measurements this is not true, and manual tuning becomes a greater challenge. Furthermore, picking  $\mathbf{A} - \mathbf{LC}$  diagonal means there is a direct relation between the size of the diagonal elements of  $\mathbf{L}$  and the poles of  $\mathbf{A} - \mathbf{LC}$ , namely that increasing these elements gives more negative poles and vice versa.

A final important implementational detail when using an IMU instead of an encoder is that there is a small, almost constant offset or bias between IMU measurements and the encoder. This bias can be directly measured by taking the difference of the two signals when the helicopter is stationary on

the ground, and can simply be added to the IMU measurement. This process was done for each state, every day the IMU was used, since the offsets may vary slightly between each day. Section 5 introduces a much better way of treating these offsets.

#### 4.1 Hypothesis and test plan

Since pole placement was discussed in great detail in section 2.1.1, the hypothesis for observer pole placement will be kept briefer and focus on stable poles.

Like previously, we need a default configuration of poles for the error dynamics (20). For this purpose we used the following rule of thumb: select poles roughly twice as negative as the desired poles of the controlled closed loop system (14). The reason we want more aggressive poles for the estimate than the system dynamics is such that the estimate always has time to adjust to rapid system fluctuations. As discussed previously, we choose real poles in order to give us individual control over each estimation state. The most aggressive pole from the LQR controller was -3. Our initial poles are hence  $2[-3 - 3 - 3 - 3 - 3] \equiv 2\mathbf{p}_0$ . Further we tested both less aggressive and more aggressive poles for this.

For the less aggressive poles, i.e. less negative poles, we expect the estimate to be slow. This is because the estimate error dynamics are slower, which means the error goes more slowly to zero. By the definition of the estimation error, this means that the state estimates approach the true estimates more slowly. For the more aggressive poles further in the left half plane, we expect the estimate to be faster but also more noisy. This is a direct result from the fact that more aggressive poles correspond to a larger  $L$ . Hence, the term  $\mathbf{L}\mathbf{n}$  in the error dynamics (20) has a larger impact, passing more of the noise  $\mathbf{n}$  to the estimate. It is also important to note that this term, if the norm of  $\mathbf{L}$  exceeds 1, can amplify the measurement noise, as opposed to attenuating it.

Because we have measurements of each state, we will also try to directly feedback the measured signal without using a Luenberger observer. Our hypothesis is that all noise will pass through, and that the estimates will be far too noisy to use.

In theory, the system is observable with only two measurements. Whether this works or not will probably depend on the tuning of the observer poles and the accuracy of the model.

Table 4: Estimate error dynamics poles and reduced measurements tested.

<b>Observer poles</b>	
$2\mathbf{p}_0$	Figure 23
$100\mathbf{p}_0$	Figure 25
$0.5\mathbf{p}_0$	Figure 26
$5 \times [-3 - 6 - 1.5 - 3 - 3]$	Figure 28
<b>Reduced measurements</b>	
3 states	Figure 30
2 states	Figure 29

## 4.2 Results and Discussion

Table 4 summarises the tests conducted as well as their corresponding figures.

### 4.2.1 Gyroscope errors due to IMU placement

An important observation was that the gyroscope outputs were incorrect for certain configurations of the helicopter in pitch, elevation and travel. To study how the individual gyroscope outputs differ based on the geometric configuration of the helicopter, we performed four tests and plotted gyroscope values against encoder values. First, we manually moved the helicopter about the travel axis with the pitch kept at a constant angle of about 90 degrees. The results are shown in figure 22a. In the plots, it almost seems like elevation rate and travel rate have switched places! Physically this has to do with the IMU being located on the part of the helicopter that turns with pitch, which means that a movement about the travel axis results in the gyroscope observing an elevation (upwards/downwards) movement in its local frame. For the same fixed pitch, figure 22b shows the outputs when we manually moved the helicopter along the elevation axis. A similar switching of states is observed here, showing that the gyroscope outputs are incorrect for nonzero pitch.

For the next two tests, the helicopter was kept at constant positive elevation. A manual movement of the helicopter back and forth about the travel axis and pitch axis are shown in figures 22c and 22d, respectively. For the both cases, there are no major unexpected oscillations of gyroscope values. Furthermore, the gyroscope output almost perfectly follows the encoder outputs for the variable that is being manually changed. The small oscillations about zero for the other variables are probably caused by small actual changes in the values, which is confirmed by the encoder values that also oscillate.

The analysis of the IMU gyroscope outputs has shown that gyroscope outputs are incorrect for nonzero pitch. However, since this is directly rooted in the helicopters geometry, it can be compensated for and is thus not an issue. This transformation was already implemented in Matlab, and the outputs can be used further.

#### 4.2.2 Testing observer poles

The responses for the initial case of repeated real poles,  $2\mathbf{p}_0$ , is shown in figure 23, given a reference step signal in pitch. For each state, the estimate is plotted on top of the encoder value and the IMU measurement. We first notice that the actual response of the pitch and pitch rate is more oscillatory than the case of same LQR controller with encoder measurements. This might be due to the estimate lagging slightly behind the encoder values, causing the response to always come a bit after the measurement. Furthermore, the pitch rate estimate has the interesting property that the amplitude of the oscillations is smaller than those of the encoder. This can again be contributed to the estimation lag, resulting in the estimate not having enough time to reach its maximum value before the helicopter again changes direction. The travel rate estimate in turn looks really good, with no very obvious lag effects. Finally, we observe that the elevation estimate is noisy and actually drops below the encoder value around the ten second point.

Our theory for the cause of this last phenomenon is that the travel rate grows so large that centripetal forces, which are neglected by the model, become significant. Looking at the travel rate curve, we can see a clear peak around exactly the same time when the elevation estimate drops. More specifically, the acceleration component away from the centre of rotation,  $a_x$ , decreases because centripetal acceleration points towards the centre [1]. This leads the estimate to believe the elevation is lower than it actually is. To visualise this effect even better, we conducted a specific experiment where we, by keeping pitch constant, let the travel rate increase. The results are shown in figure 24, and clearly show a drop in the elevation estimate as the travel rate grows large. Since pitch rate and travel rate are intertwined with elevation, a similar drop is observed for these. However, it is worth noting that a real helicopter would not be flown at extremely large travel rate, such that this drop in elevation estimate is not as critical. It does however clearly show the importance of the model in estimation, and indicates that centripetal forces may have to be added for a better implementation.

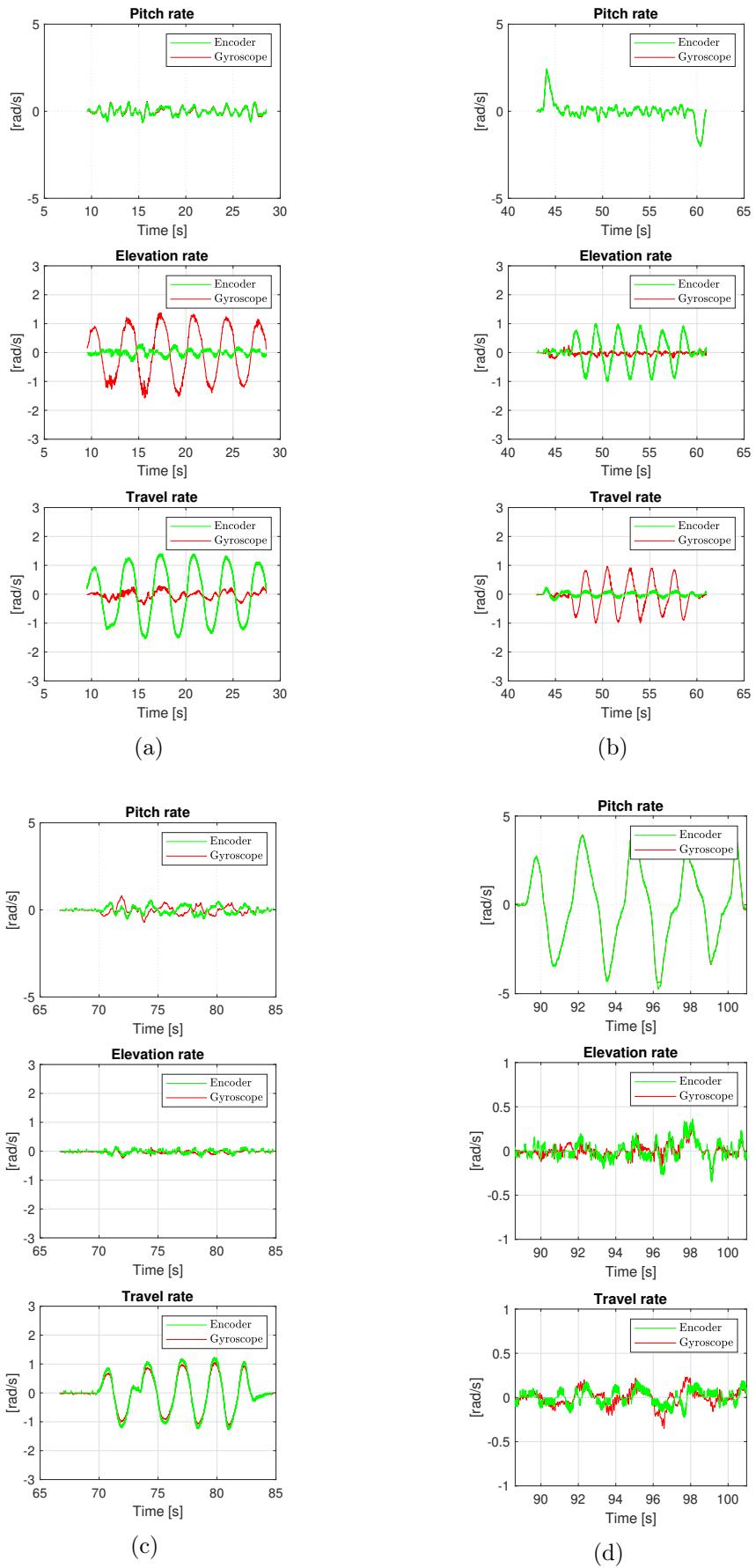


Figure 22: Gyro vs encoder without any corrections. The four cases are explained in 4.2.1.

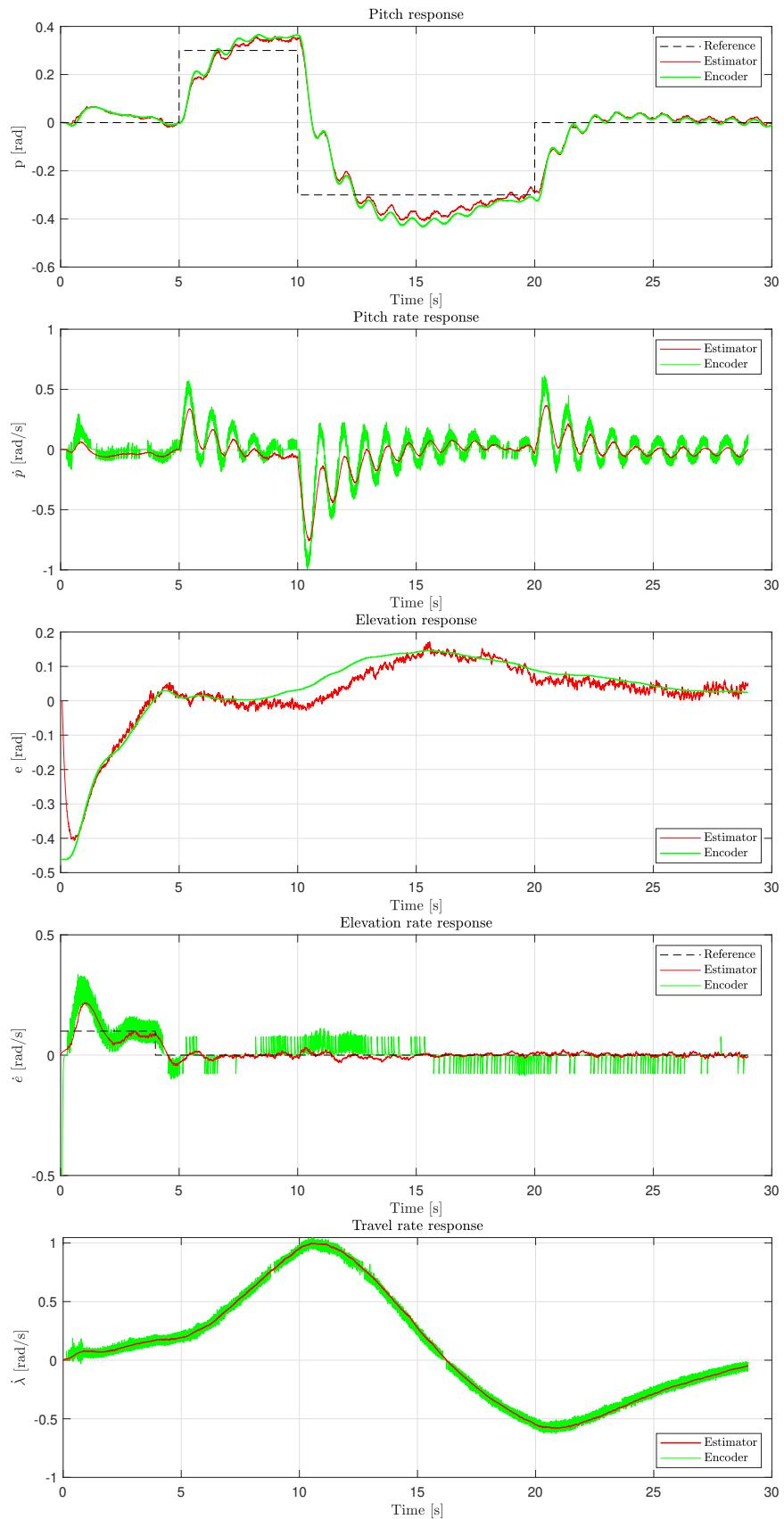


Figure 23: Encoder and estimator responses for  $2\mathbf{p}_0$ .

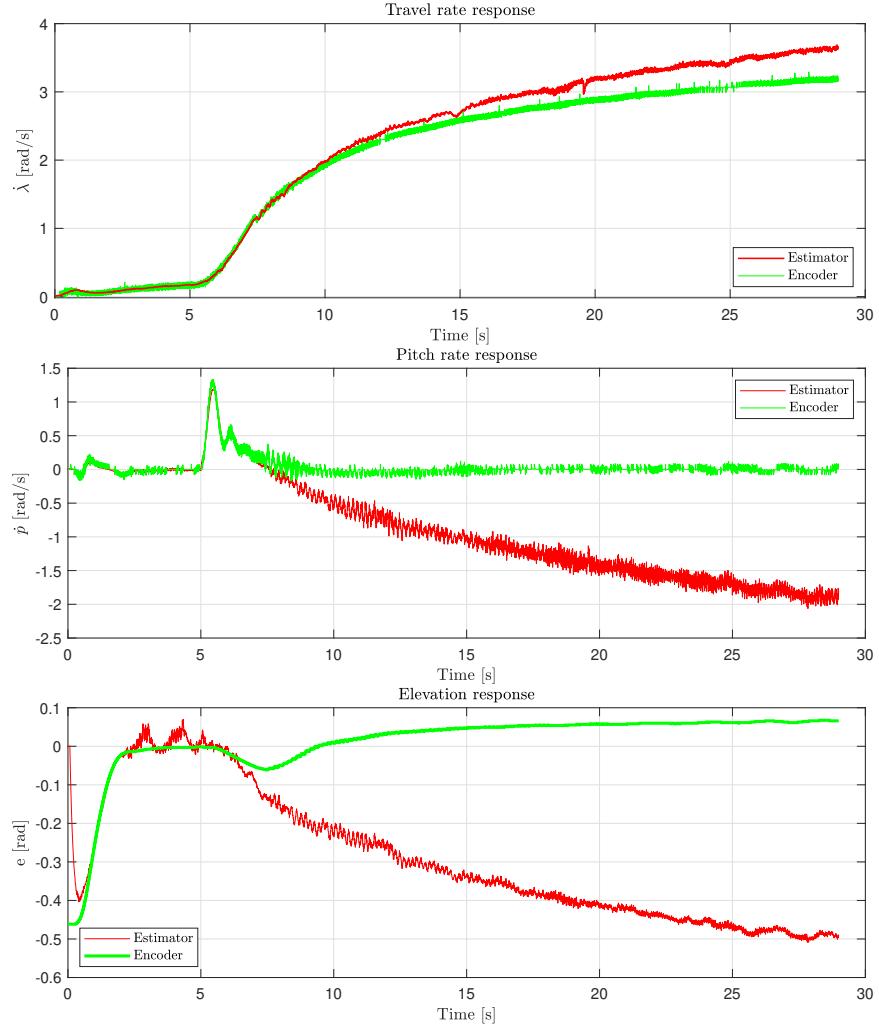


Figure 24: Effects of a large travel rate on elevation rate and pitch rate estimation.

Figures 25 and 26 display the same responses for repeated real poles  $100\mathbf{p}_0$  and  $0.5\mathbf{p}_0$  respectively. As expected, moving the observer poles closer to the right half plane yields a smoother but slower and less accurate response. Conversely, moving the poles more into the left half plane produces a faster, more noisy estimate. The difference between the two cases is very clear for the pitch responses, and the results correspond to the hypothesis.

As described in the test plan, we also conducted a test where we completely bypassed the Luenberger observer and controlled the helicopter with IMU measurement as feedback without estimation. The results are shown in 27 and display surprisingly good pitch control. Looking at the pitch response, the noise is smaller in this case than in the case of aggressive

Luenberger poles in figure 25. This demonstrates the possibility of the Luenberger observer to attenuate measurement noise.

We then proceeded to look for an optimal pole configuration. After some trial and error and educated guessing, the following configuration was arrived at:  $5 \times [-3 - 6 - 1.5 - 3 - 3]$ . The resulting responses are shown in figure 28. Comparing with the initial case of  $2\mathbf{p}_0$ , all poles have been moved further into the left half plane. However, this was done most aggressively with pitch rate, and least on elevation, in accordance with what the plots indicated was most needed.

#### 4.2.3 Minimal Luenberger observer

In the preparation it was proposed that, in theory, measuring only elevation and travel rate and letting the estimator provide the other states suffices for control. This was tested by running the helicopter with the augmented  $\mathbf{C}$  matrix (18). The result is displayed in figure 29. It is clear that this did not work. Firstly, we observe extremely large estimation noise in especially pitch and pitch rate, of the order of multiple times a full rotation. This is a clear result of the derivation needed in determining the states from the already noisy travel rate. However, this does not mean that all state measurements are needed for control. Figure 30 shows the responses if pitch is measured in addition to travel rate and elevation. Although pitch rate and elevation rate estimates are noisy, the helicopter manages to follow the pitch response quite accurately.

### 4.3 Conclusion

The Luenberger observer can be used to estimate the helicopter states in the presence of noisy measurements by compromising between how much weight it gives the measurement against the model. Initialising the poles as twice the most aggressive closed loop system pole proved to work fine, although estimates were lagging slightly behind the encoder values.

Upon moving the poles, it was confirmed that poles further in the left half plane produce faster responses at the expense of more noisy estimates. This cannot be exaggerated, as to not have the observer amplify the measurement noise. An optimal pole placement was sought and tended to have slightly more aggressive poles than the initial configuration.

The Luenberger observer allows us to control the helicopter with only measuring three states, but not the theoretical two states, because of ridiculously large noise. Potentially this could have been improved with some more trial and error.

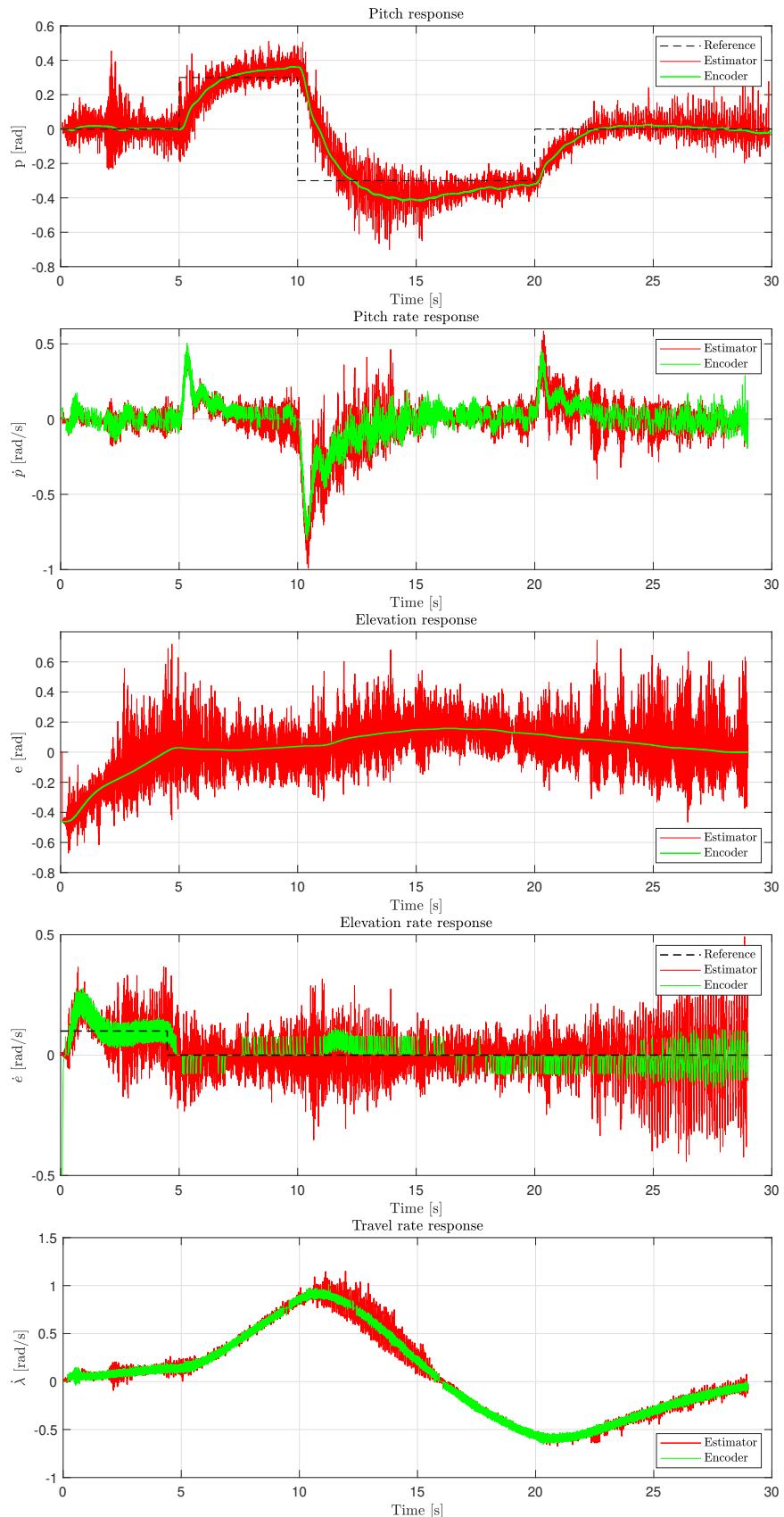


Figure 25: Encoder and estimator responses for  $100\mathbf{p}_0$ .

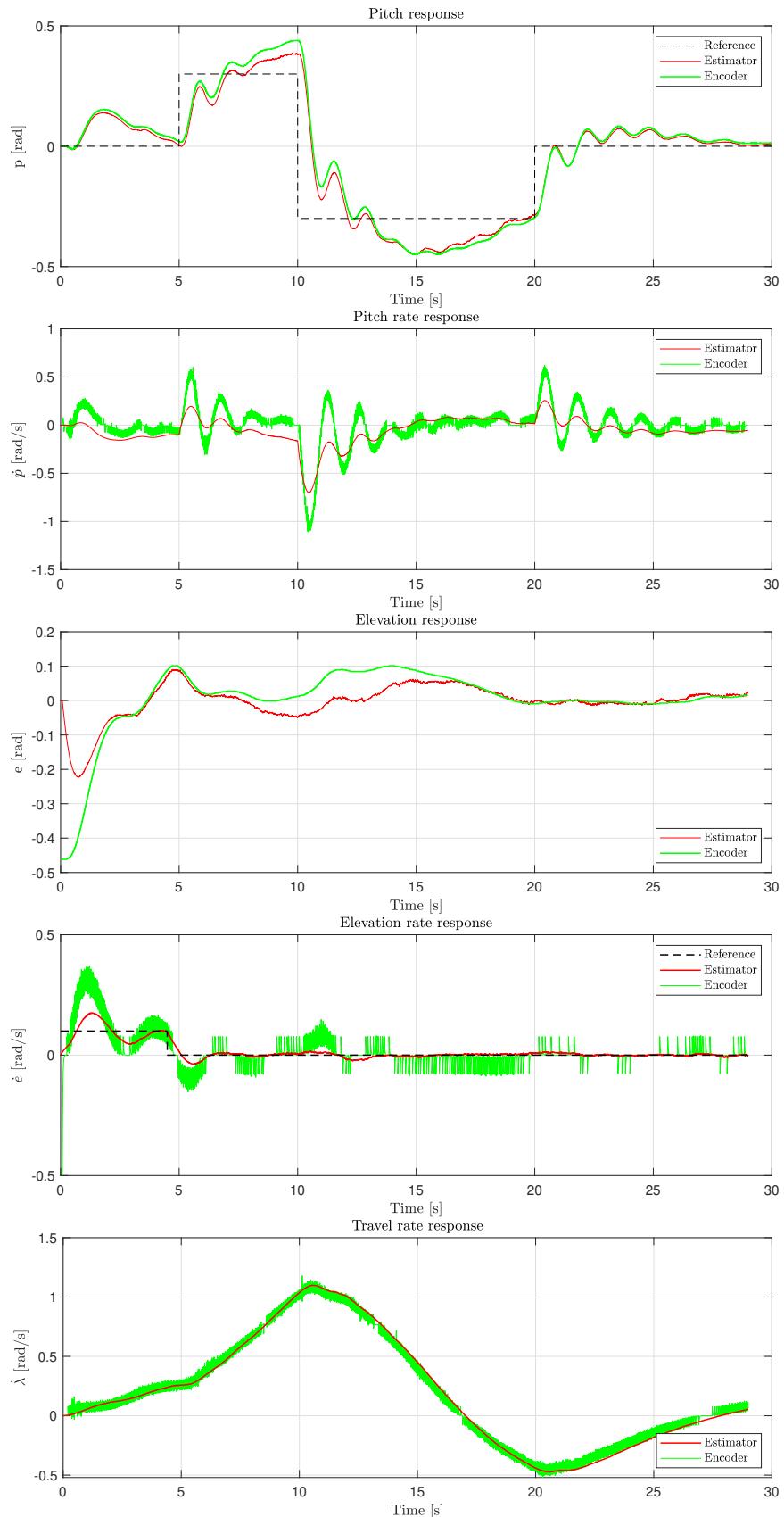


Figure 26: Encoder and estimator responses for  $0.5\mathbf{p}_0$ .

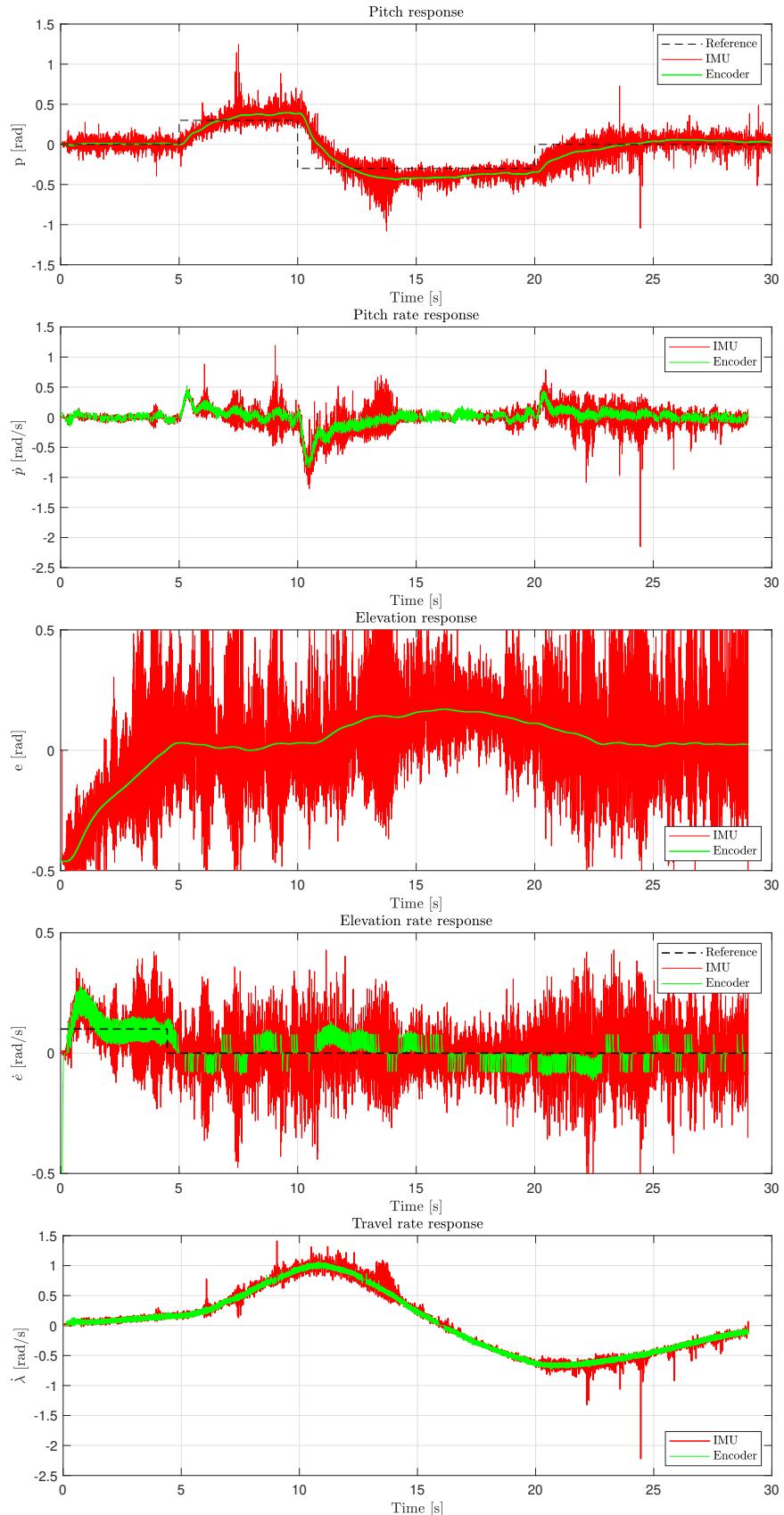


Figure 27: Encoder and estimator responses for direct measurement feed-back.

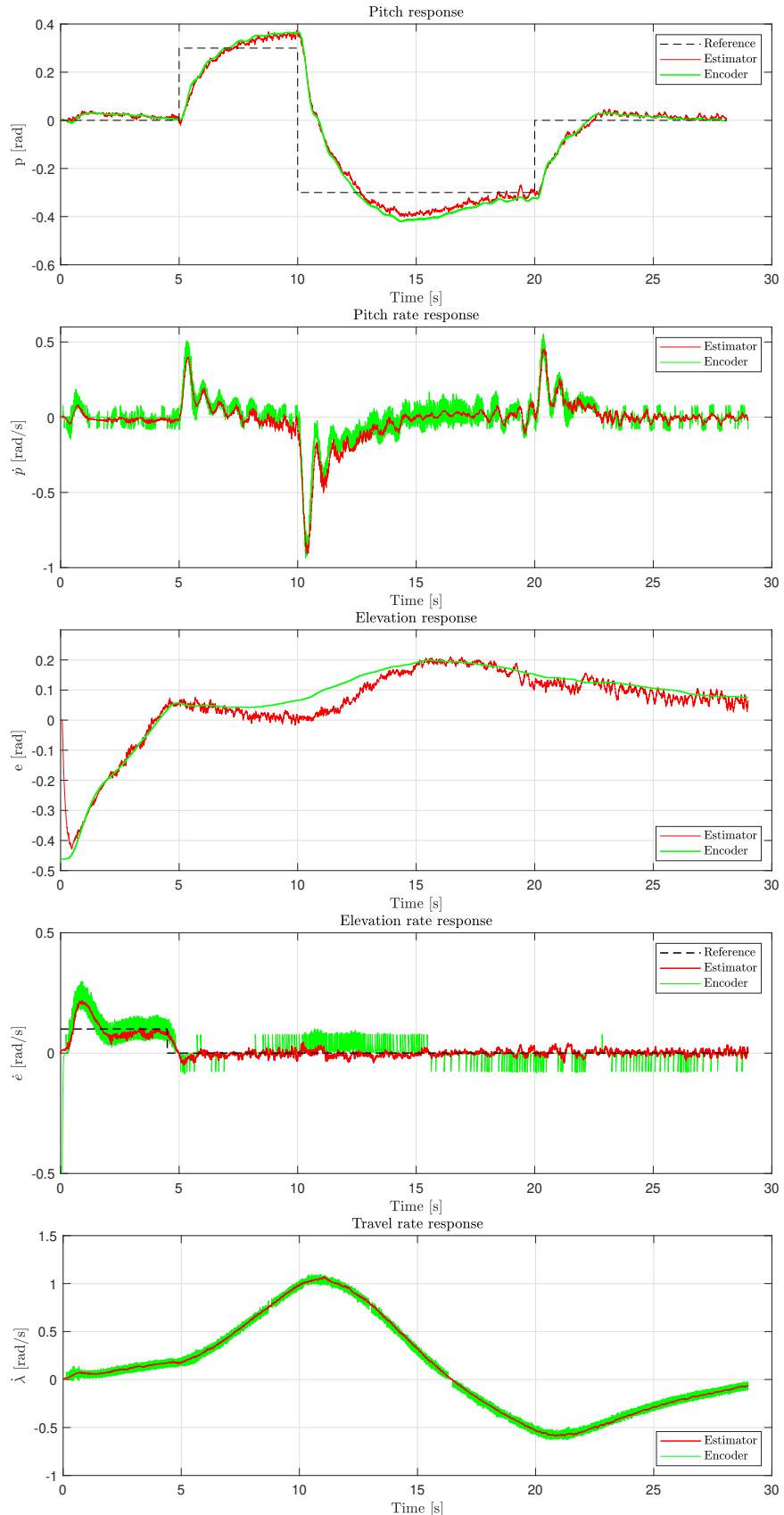


Figure 28: Encoder and estimator responses with optimal poles equal to  $5 \times [-3 - 6 - 1.5 - 3 - 3]$ .

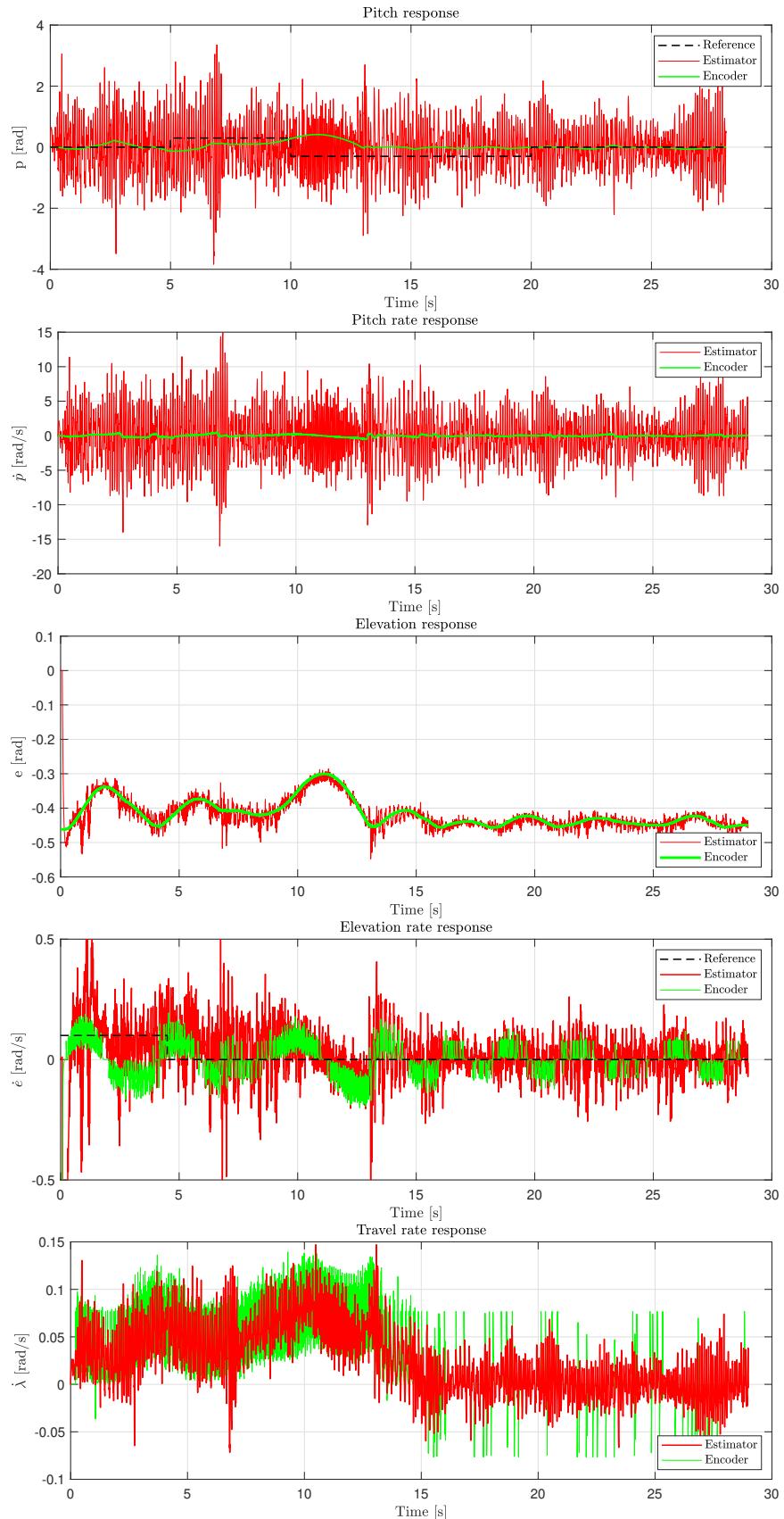


Figure 29: Encoder and estimator response with measurements of 2 states.

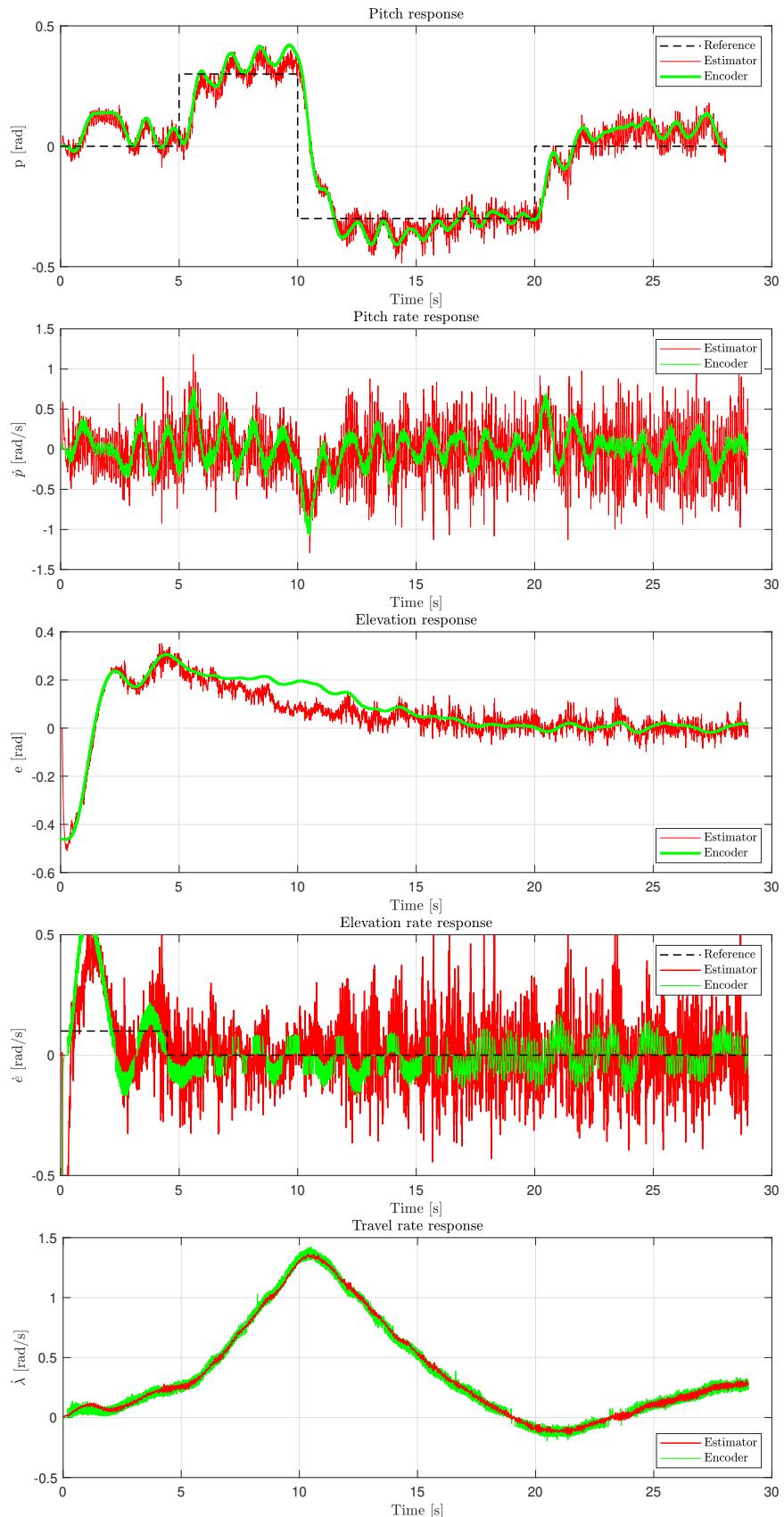


Figure 30: Encoder and estimator response with measurements of 3 states.

## 5 Kalman filter

In this section, we replace the Luenberger observer with the well known Kalman filter. The Kalman filter is based upon modelling the model disturbance and measurement noise as white random signals, and tuning it simply involves specifying the covariance matrices of these white signals. More specifically, the model disturbance contains the effect of model inaccuracies and is added to the state equations, while the measurement noise is added to the output. Assuming these are set correctly, the algorithm computes the optimal blending factor between model prediction and output. Tuning covariance matrices is significantly easier than the pole placement involved in the Luenberger observer, and the Kalman filter is therefore often preferable.

We will use the following discrete-time model:

$$\mathbf{x}[k+1] = \mathbf{A}_d \mathbf{x}[k] + \mathbf{B}_d \mathbf{u}[k] + \mathbf{w}_d[k] \quad (21a)$$

$$\mathbf{y}[k] = \mathbf{C}_d \mathbf{x}[k] + \mathbf{v}_d[k], \quad (21b)$$

where

$$\begin{aligned} \mathbf{A}_d &= \begin{bmatrix} 1 & 0.002 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.002 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \approx 0 & \approx 0 & 0 & 0 & 1 & 0.002 \\ \approx 0 & \approx 0 & 0 & 0 & 0 & 1 \end{bmatrix} & \mathbf{B}_d &= \begin{bmatrix} 0 & \approx 0 \\ 0 & 0.0009 \\ \approx 0 & 0 \\ 0.0002 & 0 \\ 0 & \approx 0 \\ 0 & \approx 0 \end{bmatrix} \\ \mathbf{C}_d = \mathbf{C} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} & \mathbf{D}_d = \mathbf{D} = 0 & \mathbf{x} = \begin{bmatrix} p \\ \dot{p} \\ e \\ \dot{e} \\ \lambda \\ \dot{\lambda} \end{bmatrix} & \mathbf{u} = \begin{bmatrix} \tilde{V}_s \\ V_D \end{bmatrix}, \end{aligned}$$

and

$$\mathbf{w}_d \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_d), \mathbf{v}_d \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_d),$$

are the model disturbance and measurement noise, respectively.  $\mathbf{Q}_d$  and  $\mathbf{R}_d$  are the covariance matrices. Note that we can not measure travel,  $\lambda$ , with the IMU, which is the reason for the  $5 \times 6$   $\mathbf{C}$  matrix. The discretised matrices were computed using the Matlab function `c2d(sysc, Ts)`, where  $Ts = 0.002$  and `sysc` is the continuous time system derived from (2). For the tuning of the covariance matrices, it is significantly easier to experimentally determine the measurement covariance matrix  $\mathbf{R}_d$  than the model disturbance covariance matrix  $\mathbf{Q}_d$ . This was done by simply recording the measurements for a period

of time, and using Matlab to calculate the covariance of the sequence. This procedure was done both for the helicopter lying stationary on the ground with the control inputs disconnected, and for the helicopter hovering at its linearisation point. The resulting matrix for the hovering measurement is given in (22). The resulting matrix for the ground measurement did not include values greater than  $10^{-4}$  and will for that reason not be included.

$$\mathbf{R}_d = \begin{bmatrix} 0.0021 & -0.0017 & -0.0013 & -0.002 & 0 \\ -0.0017 & 0.0252 & 0.0132 & -0.0047 & 0 \\ -0.0013 & 0.0132 & 0.0080 & -0.0031 & 0 \\ -0.002 & -0.0047 & -0.0031 & 0.0088 & 0 \\ 0 & 0 & 0 & 0 & 0.0110 \end{bmatrix} \quad (22)$$

It is clear that the estimated covariance matrix  $\mathbf{R}_d$  is much larger for the case of the helicopter hovering. This is likely because vibrations, electromagnetic noise and other disturbances. Since the Kalman filter estimates become critical once the helicopter lifts off, the  $\mathbf{R}_d$  in (22) was used. It is here worth noting that, if the noise were perfectly white and the covariance measurements were exact,  $R_d$  would have been diagonal. The fact that it is not may indicate that the white noise assumption is not completely accurate.

When it comes to the covariance matrix of the model disturbance ( $\mathbf{Q}_d$ ), this is not possible to measure directly. We hence have to tune it like if it were a control parameter.

The Kalman filter will then provide the optimal feedback gain by minimizing the variance of the estimation error. This is completely analog to the way LQR estimation minimises the cost function based on the weighting matrices for the states and inputs. The Kalman filter algorithm can be described by the following equations.

Correction with new data:

$$\mathbf{L}[k] = \bar{\mathbf{P}}[k] \mathbf{C}_d^T (\mathbf{C}_d \bar{\mathbf{P}}[k] \mathbf{C}_d^T + \mathbf{R}_d)^{-1} \quad (23a)$$

$$\hat{\mathbf{x}}[k] = \bar{\mathbf{x}}[k] + \mathbf{L}[k](\mathbf{y}[k] - \mathbf{C}_d \bar{\mathbf{x}}[k]) \quad (23b)$$

$$\hat{\mathbf{P}}[k] = (\mathbf{I} - \mathbf{L}[k] \mathbf{C}_d) \bar{\mathbf{P}}[k] (\mathbf{I} - \mathbf{L}[k] \mathbf{C}_d)^T + \mathbf{L}[k] \mathbf{R}_d \mathbf{L}^T[k] \quad (23c)$$

Predicting ahead:

$$\bar{\mathbf{x}}[k+1] = \mathbf{A}_d \hat{\mathbf{x}}[k] + \mathbf{B}_d \mathbf{u}[k] \quad (23d)$$

$$\bar{\mathbf{P}}[k+1] = \mathbf{A}_d \hat{\mathbf{P}}[k] \mathbf{A}_d^T + \mathbf{Q}_d, \quad (23e)$$

where  $\mathbf{P}$  is the covariance matrix for the estimation error, bars indicate model based estimates and hats indicate corrected estimates.

The Kalman filter can, in addition to estimating the states, be used to estimate the constant or slowly varying biases in the IMU measurements, introduced in section 4. This is preferable to hardcoding since these biases may change over time. The pitch measurement will be used as an example. Until now the pitch measurement  $y_p$  has been assumed to be a measurement of the pitch and some white noise:

$$y_p = p + n$$

The fault in this assumption lies in the noise term  $n$ , because this will include a bias, thus making the white noise assumption wrong. A better assumption of  $y_p$  will be

$$y_p = p + b + n, \quad (24)$$

where  $b$  is the bias. The bias can be modelled as a random walk:

$$\dot{b} = w$$

where  $w$  is assumed to be white noise. The state space can be augmented to include this bias, like shown below:

$$\dot{\mathbf{x}}' = \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{b} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ 0 & 0 \end{bmatrix} \mathbf{x}' + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \mathbf{u} + \mathbb{W}' \quad \mathbf{y} = \mathbf{C}' \mathbf{x}',$$

$\mathbf{C}'$  will result in (24) for  $y_p$ . Running the Kalman filter algorithm on this augmented state space will allow us to estimate the pitch bias. It is important to note that we can only measure the biases of redundant states, which for our system are pitch, pitch rate and elevation rate. What is meant by redundancy more specifically is that we need at least two independent ways to derive a state variable in order to estimate a bias, which basically is a difference.

## 5.1 Hypothesis and test plan

Since it is the only degree of freedom, the hypothesis focuses on the effect of the model disturbance covariance matrix  $\mathbf{Q}$ . It contains the variance of the states in the model, and thus summarises the effect of model inaccuracies, disturbances and fluctuations. Being diagonal, the  $i$ th element in  $\mathbf{Q}$  corresponds to the variance of the disturbance of state  $x_i$ .

The impact of  $\mathbf{Q}_d$  is easiest analysed via its relation to the blending factor, or Kalman gain ( $\mathbf{L}$  in (23)). A larger  $\mathbf{Q}_d$  yields a larger Kalman gain, and hence is expected to result in more weight being given to the measurement. This means that more noise from the measurement passes to the estimate, and that it will follow the measurement more than the model. On the other hand, a lower  $\mathbf{Q}_d$  is expected to result in more weight being given to the

Table 5: Connecting different weighing of  $\mathbf{Q}$  and their corresponding figures.

Values for $\mathbf{Q}$	
$\mathbf{Q} = 0.01\mathbf{I}$	Figure 31
$\mathbf{Q} = \mathbf{0}$	Figure 34
$\mathbf{Q} = 1000\mathbf{I}$	Figure 33
$\mathbf{Q} = 0.0001\mathbf{I}$	Figure 32
optimal $\mathbf{Q}$	Figure 38

model prediction, such that the effect of measurement noise is reduced, but the effect of the model disturbances is increased.

Just like the tuning of the LQR matrices, the tuning of  $\mathbf{Q}_d$  can also be performed by altering the variances of individual states. The expectations is again that larger entries in  $\mathbf{Q}_d$  for a state will make its estimate follow the noisy measurement more, while smaller entries will produce smoother estimates that stick more to the model prediction. Individual tuning should be performed when finding a best possible  $\mathbf{Q}_d$ .

For some constant  $\mathbf{R}_d$ , we expect infinite  $\mathbf{Q}_d$  to correspond to full trust in the measurement and no correction from model. This is expected to yield estimates that equal the measurement and hence have a lot of noise. On the other extreme,  $\mathbf{Q}_d = 0$  corresponds to complete trust in the model and virtually no weighting of the measurement. This is likely an unstable configuration, since the model only is an approximation, and any systematic disturbance or error will cause the estimation error to grow without ever being corrected.

As for the bias estimates, these are expected to approach constant values. The speed at which this conversion occurs depends on the covariance given to the bias state disturbances in  $\mathbf{Q}_d$ , just as it does for the regular states.

Finally, since it is easier to mentally visualise continuous disturbances, we plan to tune the  $\mathbf{Q}$  matrix, and let Matlab convert it to  $\mathbf{Q}_d$  using the Van Loan method.

## 5.2 Results and Discussion

Table 5 summarises the different model disturbance covariance matrices tested as well as their corresponding figures.

### 5.2.1 Tuning the covariance matrix $\mathbf{Q}$

Figure 31 shows the responses when  $\mathbf{Q}$  is identity with entries 0.01. This initial value was chosen experimentally. For each of the six states, we have plotted the accelerometer output, estimate and encoder output, where the latter is interpreted to be the actual helicopter response. We first notice the lack of accelerometer data and the massive estimation error for the travel response. This is a direct consequence of the travel being unobservable, and is a depiction of what happens when you try to estimate unobservable states. For the other responses, the estimate seems to, generally, resemble a low pass filtered version of the IMU measurement. However, there is a significant estimation error for the pitch response. This indicates that the estimate of  $\mathbf{Q}$  may not be perfect.

The responses for much smaller diagonal entries in  $\mathbf{Q}$ , 0.0001, and much larger entries, 1000, are shown in figures 32 and 33 respectively. It is evident that decreasing the covariance matrix reduces the amount of noise passed through the filter, seen by the much smoother estimate. Also, the estimate appears to follow the encoder value less precisely in general. The difference is particularly evident when comparing the pitch rate and elevation plots with those in the initial case. In contrast, increasing  $\mathbf{Q}$  leads to a faster, but much more noisy estimate. This makes sense, as we here trust the noisy measurement more and reduce the low pass effect with respect to it. Although pitch regulation still is fine, there is extremely much noise in the elevation rate estimate, such that the controller is less likely to work robustly for elevation rate control.

If we were to increase  $\mathbf{Q}$  even more, we would probably observe a more and more noisy measurement until we virtually only trust the measurement. While we cannot set a matrix to infinity, we can set it to zero. With  $\mathbf{Q} = \mathbf{0}$ , the control fails, as can be seen in figure 34. This is because the estimate now only cares about the model and does not incorporate the measurement at all. Hence, since the model is only an approximation, any systematic modelling errors cause the estimation error to grow.

To investigate this behaviour even further, we connected signals from a joystick button to the Simulink diagram in order to, in real time, be able to block the measurement to the Kalman filter. We then simulated the standard pitch reference step signal, but arbitrarily blocked the measurement for short periods of time. The resulting response is plotted in figure 35, and the plot of the diagonal elements in the error covariance matrix  $\mathbf{P}$ , representing the variances of the individual state estimates, is shown in figure 36. We can see that the error variances immediately begin to grow once the measurements are turned off. However, once we allow the measurements

again, the Kalman filter quickly brings the error to zero because the Kalman gain has been growing while the measurement was disconnected. Practically this means that even if the measurement is briefly lost, the helicopter would maintain control and quickly re-stabilise once the measurement returns.

Finally, by studying the individual estimates and adjusting them as described in the hypothesis, we arrived at an "optimal" configuration of  $\mathbf{Q}$ . It is given by the diagonal matrix with the following entries on the diagonal:  $0.01 * [0.50.50.11110.010.010.01]$ . The resulting responses are shown in figure 38. While most covariances were kept roughly around the initial value, the covariances of the bias estimates was significantly reduced. To understand why this was done, we have plotted the bias estimate for pitch for both the initial case and the optimal case beside each other in figure 37. Initially, it would seem like the bias estimate is too slow in the initial case, and would benefit from having its  $\mathbf{Q}$  entry increased. However, we opted for instead decreasing it and rather giving the pitch bias an initial value roughly around what it previously had been hard coded to. This is because we knew that the biases varied slowly, if they varied at all. Hence, once the biases have reached constant values, we don't want them to change rapidly based on the noisy measurement. The optimal bias response shows a smoother trajectory that slowly optimises the bias as opposed to starting at zero and approaching the bias in a more noisy fashion.

### 5.2.2 Comparison to Luenberger observer

In the design and implementation of the observers, the largest difference between the Kalman filter and the Luenberger observer relates to the way the tuning is performed. Tuning the covariance matrix of the system disturbance proved to be much more intuitive than tuning the poles of the Luenberger observer. This is because it is easy to see whether a state estimate is too noisy or too inaccurate, while it is difficult to conclusively see how one should change the poles of the estimation error dynamics.

Furthermore, the Kalman filter performance was much better than what we were able to achieve using the Luenberger observer. This is particularly evident when comparing our "optimal" Luenberger response in figure 28 to our optimal Kalman Filter response in figure 38.

One final important comparison is that, while the Luenberger observer can amplify measurement noise in the estimate, the Kalman filter can not produce estimates with more noise than the actual measurement noise.

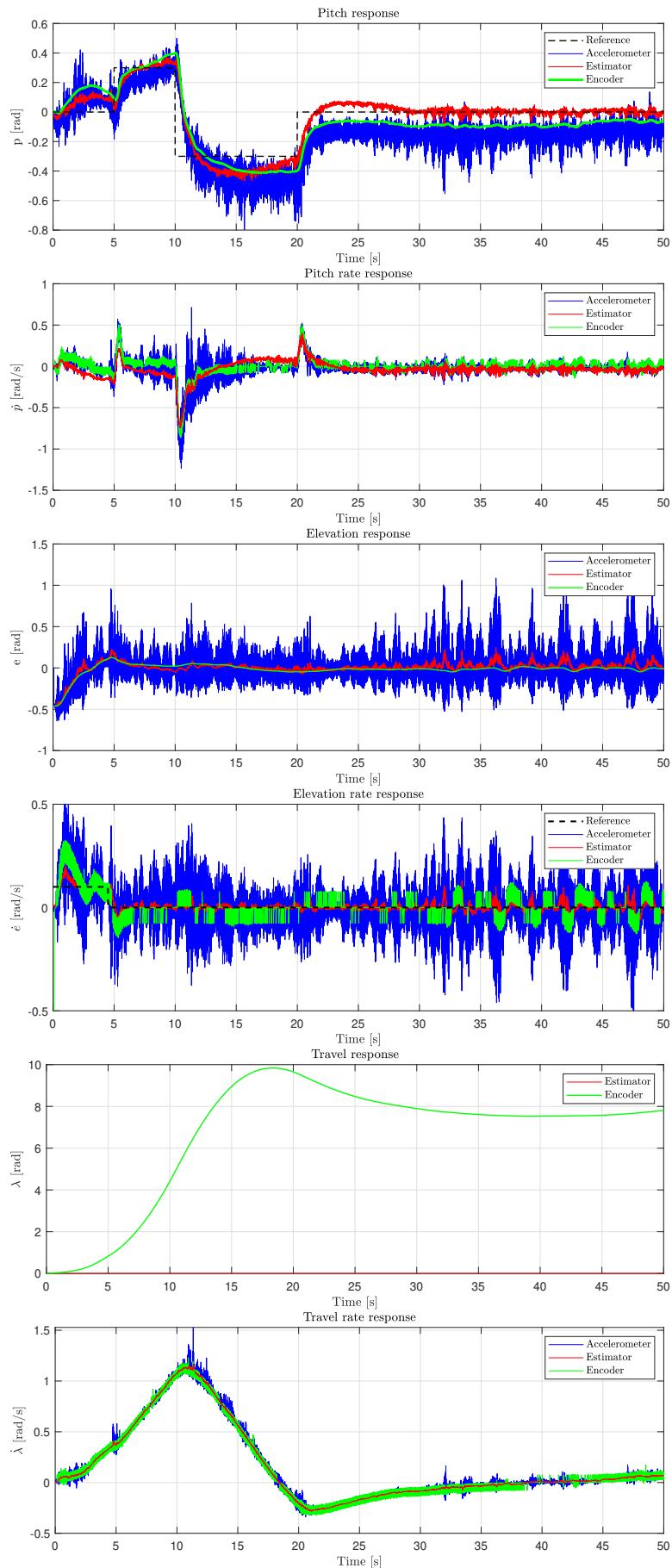


Figure 31: Encoder, accelerometer and estimator response with  $\mathbf{Q} = 0.01\mathbf{I}$

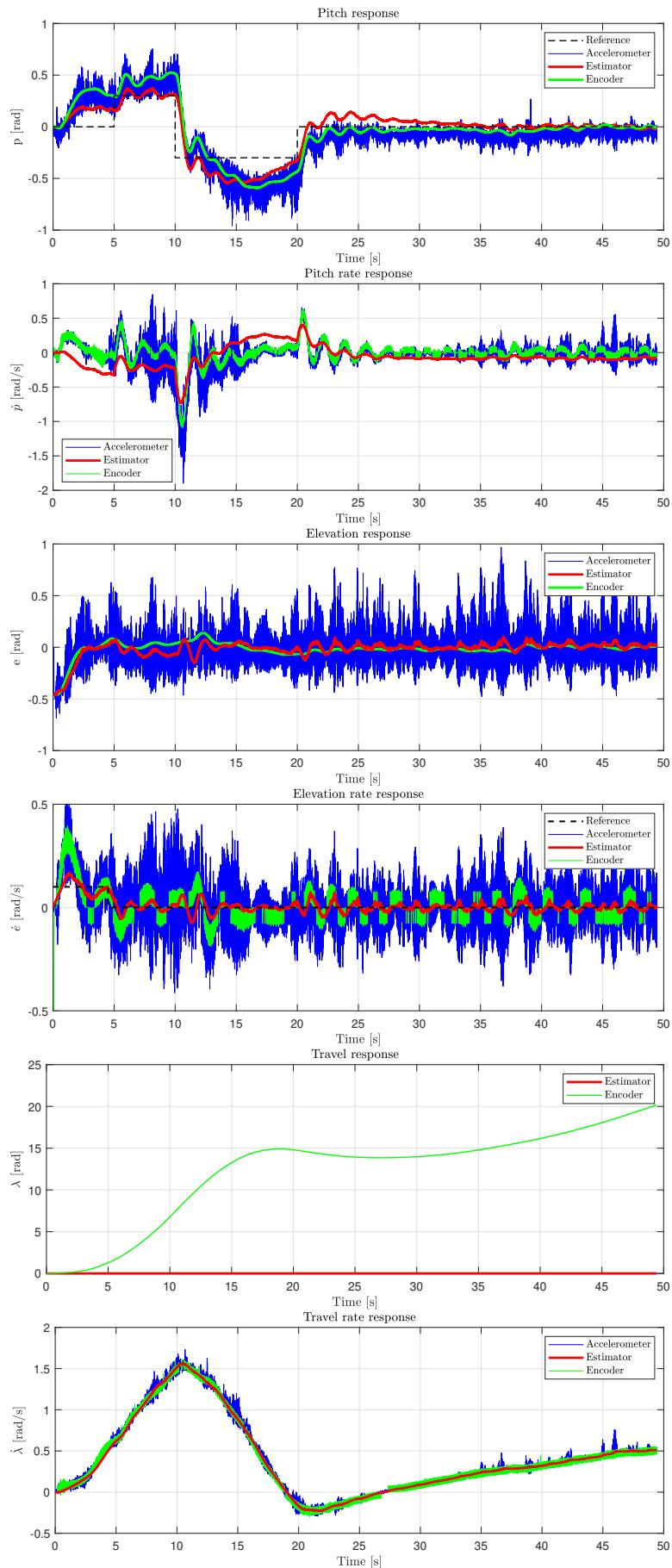


Figure 32: Encoder, accelerometer and estimator response with  $\mathbf{Q} = 0.0001\mathbf{I}$

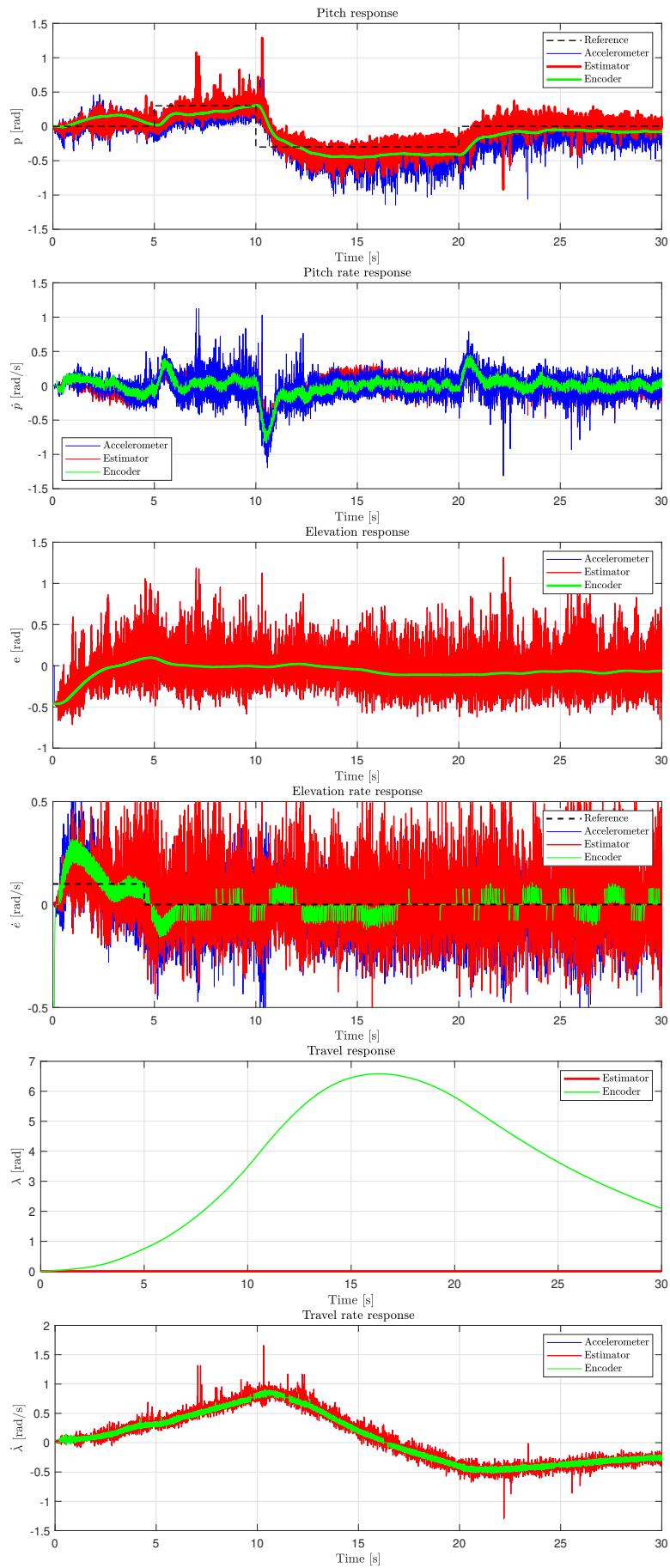


Figure 33: Encoder, accelerometer and estimator response with  $\mathbf{Q} = 1000\mathbf{I}$

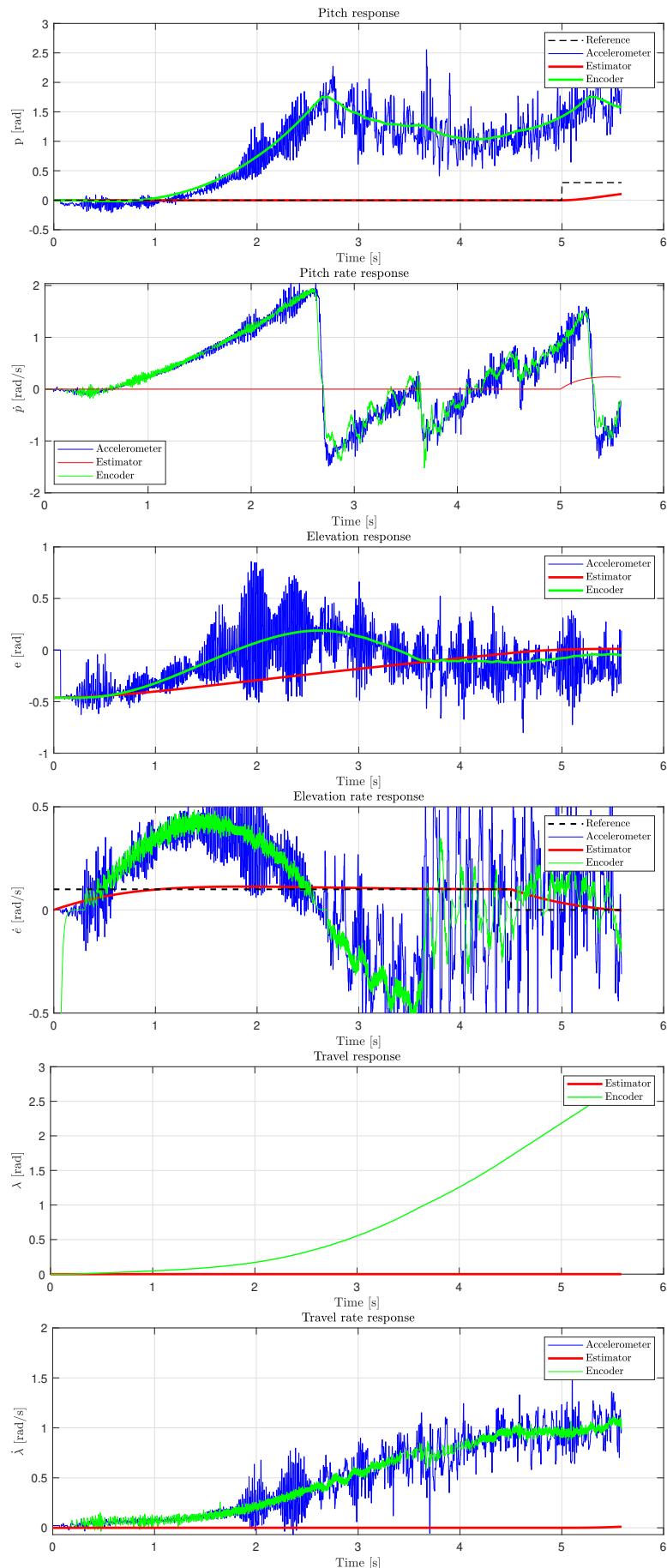


Figure 34: Encoder, accelerometer and estimator response with  $\mathbf{Q} = \mathbf{0}$

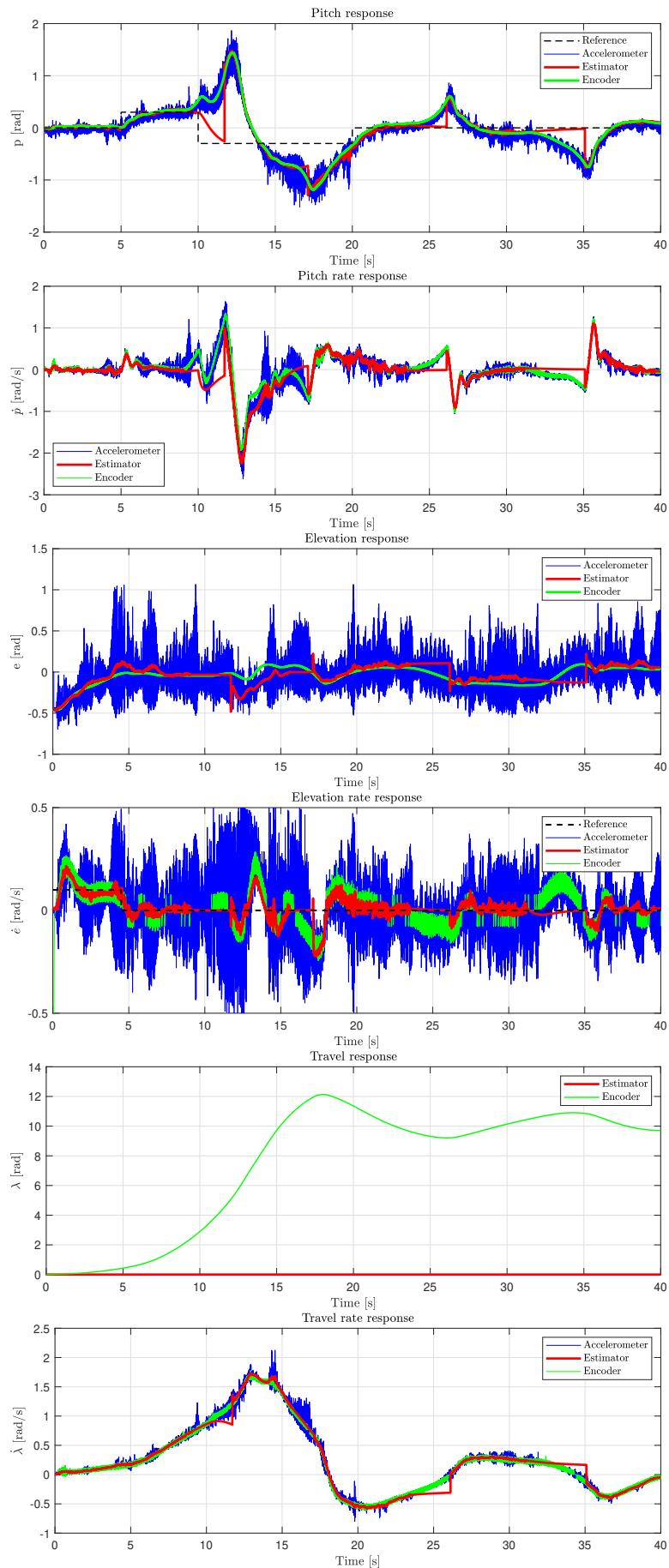


Figure 35: Encoder, accelerometer and estimator response with manual switching of the output signal. 54

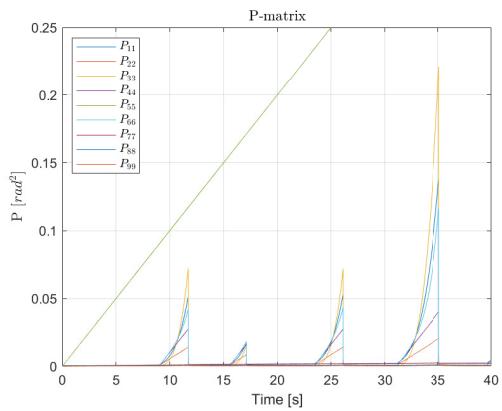


Figure 36: Estimation error variances with manual switching of the output signal.

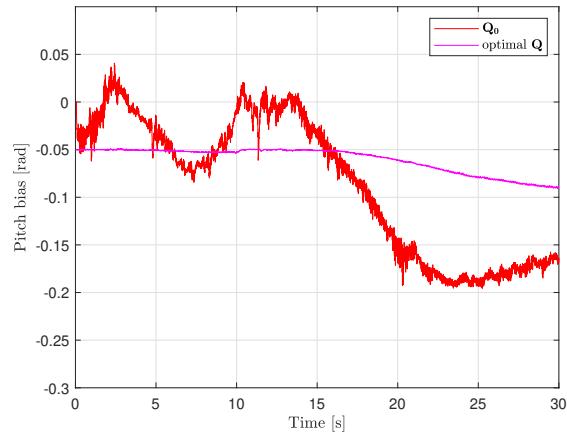


Figure 37: Pitch bias estimates for initial  $\mathbf{Q}_0 = 0.01\mathbf{I}$  and optimal  $\mathbf{Q}$

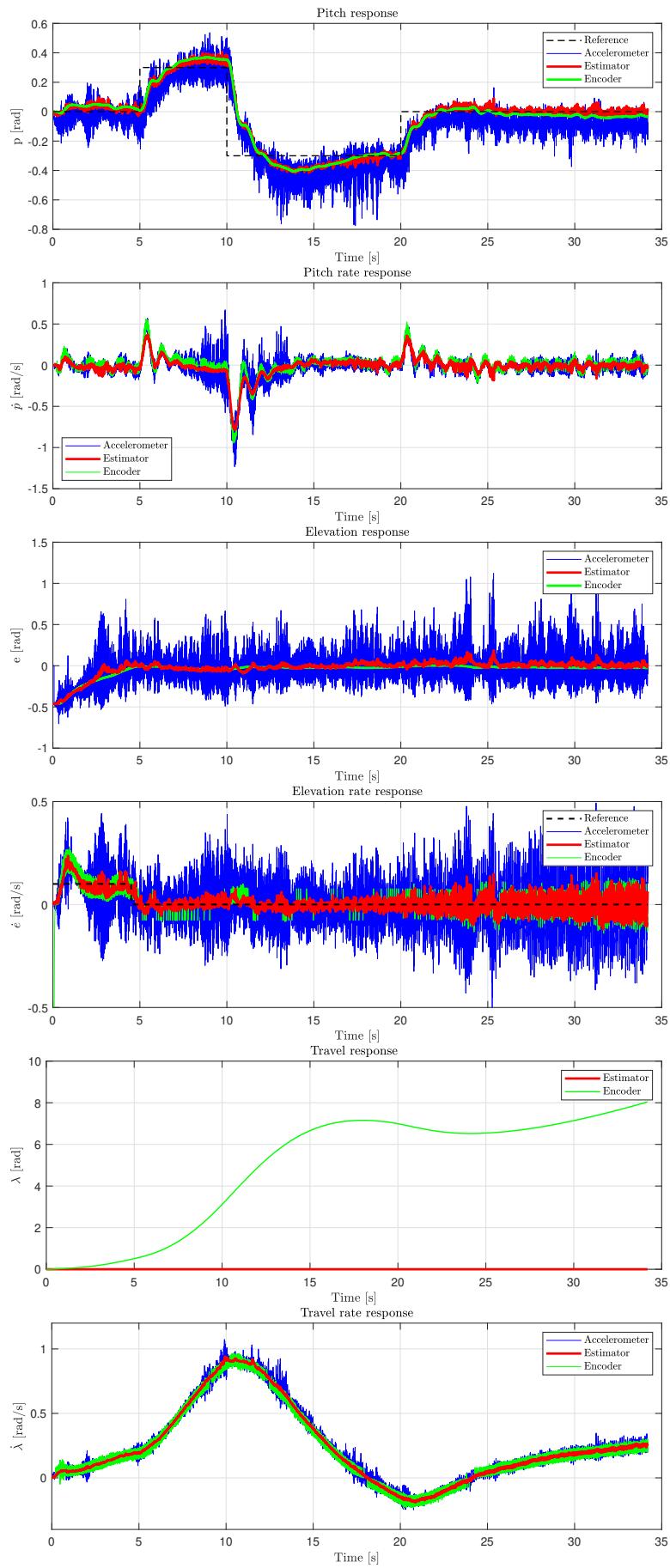


Figure 38: Encoder, accelerometer and estimator response with  $\mathbf{Q} = \text{Optimal}$

### 5.3 Conclusion

The Kalman filter has provided an intuitive way of tuning the state estimates via the variance of the model disturbance in the covariance matrix  $\mathbf{Q}$ . Large variances were shown to induce noisy estimates because the model is given less importance by the Kalman filter algorithm, such that the noisy measurement is passed more through. Conversely, small variances lead to smooth estimates that follow the encoder values less precisely. Bias estimation proved to be very easy and effective, and a significant improvement to hard coding these.

It is worth noting that it is not always the case that trusting the measurement more than the model produces more precise control. In cases where the model is very good, and the measurement is severely noisy, one may be better off going for a small  $\mathbf{Q}_d$  in order to trust the model and filter out much of the measurement noise.

## A Parameters and constants

Table 6: Parameters and values.

Symbol	Parameter	Value	Unit
$l_h$	Distance from elevation axis to helicopter head	0.66	m
$l_c$	Distance from elevation axis to counterweight	0.40	m
$l_p$	Distance from pitch axis to motor	0.175	m
$K_f$	Force constant motor	0.234	N/V
$J_p$	Moment of inertia for pitch	0.04	kg m <sup>2</sup>
$J_e$	Moment of inertia for elevation	0.87	kg m <sup>2</sup>
$J_\lambda$	Moment of inertia for travel	0.91	kg m <sup>2</sup>
$m_p$	Motor mass	0.65	kg
$m_c$	Counterweight mass	1.92	kg
$g$	Gravitational constant	9.81	m/s <sup>2</sup>
$V_{s0}$	Necessary summed voltage for hovering	5.7	V

Table 7: Constants and corresponding formulas.

Symbol	Formula
$K_1$	$\frac{K_f}{2m_p l_p}$
$K_2$	$\frac{K_f l_h}{m_c l_c^2 + 2m_p l_h^2}$
$K_3$	$\frac{l_h K_f V_{s0}}{m_c l_c^2 + 2m_p (l_h^2 + l_c^2)}$
$K_f$	$\frac{2g m_p l_h - l_c m_c g}{l_h V_{s,0}}$
$L_1$	$K_f l_p$
$L_2$	$l_c g m_c - 2g m_p l_h$
$L_3$	$K_f l_h$
$L_4$	$K_f l_h$

Notes:

- $K_1$  was wrong during the tests presented in section 2.  $l_p$  was mistakenly replaced with  $l_h$ .
- $K_f$  was wrong with approximately a factor of 2 due to a parentheses misplacement in the implementation.

## References

- [1] *Helicopter Lab Assignment.* PDF on Blackboard for TTK4115. Accessed: 2021-11-20.