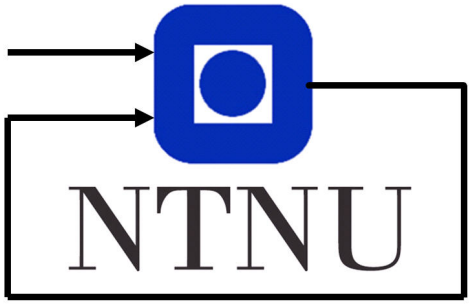
Helicopter Lab Report

Group 32
Student no. 118606
Student no. 477793
Student no. 122842

September 19, 2025



Department of Engineering Cybernetics

Abstract

This is the documentation for the helicopter lab work in TTK4115 Linear System Theory following the instructions given in [1] and [2]. It implements linear single-variable and multivariable control of a helicopter to improve stability. In the latter part, state estimators are added in the form of a Luenberger observer and a Kalman filter to reduce the error caused by measurement noise.

Contents

1 Introduction

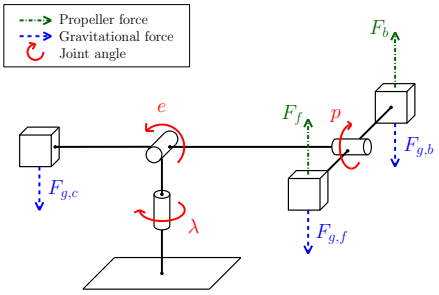


Figure 1: Illustration of helicopter setup. [1]

This is the documentation of the helicopter lab in TTK4115 Linear System Theory. It consists of four parts implementing different linear control methods of a helicopter attached with an arm to a table as seen in figure 1. The states required to control are pitch, pitch rate, elevation and elevation rate. The system reference is given through a Logitech joystick.

The first part is a monovariable PD-controller linearized with the helicopter parallel to the table.

The second part is a multivariable system with an LQR-controller, then extended to include integral action.

For the third part the system with the LQR controller is given a noisy output from an IMU, thus a Luenberger state estimator is added to the system.

Lastly, the Luenberger is replaced with a Kalman filter.

2 Part I – Monovariable Control

2.1 Introduction and Motivation

The goal of this part of the lab was to implement a PD controller for the pitch angle of the helicopter model and to investigate how different pole placements affect the closed-loop behaviour. The helicopter has three degrees of freedom (pitch p, elevation e, and travel λ), and the equations of motion were linearized around the equilibrium point $(p, e) = (0, 0)$.

After linearization, the pitch dynamics reduce to a second–order system with the transfer function

$$G(s) = \frac{K_1 K_{pp}}{s^2 + K_1 K_{pd} s + K_1 K_{pp}},$$

where K_1 is a constant determined from the physical parameters of the helicopter. This expression was the starting point for designing the PD controller used in the experiment.

To keep the elevation angle $e = 0$ in equilibrium, an offset voltage is required:

$$V_{s,0} = -\frac{L_2}{L_3}.$$

This offset was added to the elevation controller before the tests.

The pitch controller implemented in the lab was

$$V_d = K_{pp}(p_c - p) - K_{pd}\dot{p},$$

where p_c is the pitch reference from the joystick.

2.2 Hypothesis and Test Plan

To analyse how the controller gains influence the closed-loop behaviour, the transfer function was compared with the standard second–order form. By matching

$$s^2 + K_1 K_{pd} s + K_1 K_{pp} \quad \text{to} \quad s^2 + 2\zeta\omega_n s + \omega_n^2,$$

the relationships between the gains and the desired pole locations were obtained. These expressions were then evaluated using the experimentally based value $K_1 \approx 31.75$.

2.2.1 Pole Placement

The original idea was to choose different closed-loop poles (λ_1, λ_2) so that the pitch system would express clearly distinct behaviours. In theory, two real and negative poles yield an overdamped response, a repeated real pole produces a critically damped response, and a pair of complex poles with negative real part results in an underdamped, oscillatory response.

However, during the lab we did the opposite: we fixed K_{pd} and then experimented with different values for K_{pp}. Because of this, the gains we used did not actually correspond to the pole locations we had planned on paper. This means the "theoretical" pole placements (critically damped, underdamped, etc.) were not the same as the poles we actually implemented.

2.3 Experimental Results

Figure 2 shows the measured pitch responses for the different gain settings. For each test, an impulse-like change in the pitch reference was applied and the response was logged.

Even though the intention was to test one critically damped case, one overdamped case and one underdamped cases, the results showed that all configurations behaved mainly as overdamped systems. This happened because the gains we used produced two real negative poles in all cases (one slow pole and one very fast pole). In other words, none of the implemented gain sets actually resulted in complex poles.

Because of this, the responses mainly differed in how fast they converged, and only small oscillations appeared for the test where K_{pp} was set highest. This fits the pole analysis: the closed-loop poles always ended up on the real axis, so the system naturally behaved in a non-oscillatory way.

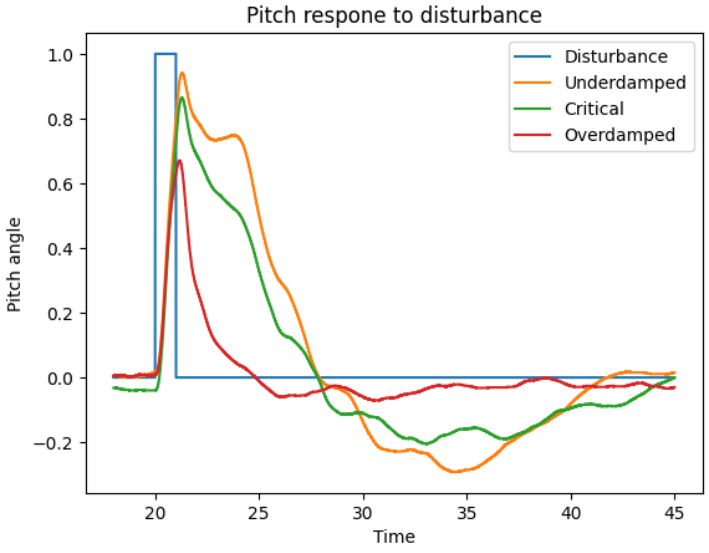


Figure 2: pitch responses for the set of tested values .

2.4 Discussion

From the measurements it was clear that the helicopter did not behave the way the theoretical pole placements suggested. Even though the test plan included cases that were meant to be underdamped, critically damped and overdamped, the actual responses were mostly overdamped. This matches the pole calculations based on the gains we used: since K_{pd} was kept fixed and only K_{pp} was changed, all resulting closed-loop poles ended up real and negative. Because of this, none of the implemented tests actually produced complex poles, and the system therefore showed little or no oscillation.

The real helicopter also introduces several effects that are not included in the linearized model, such as motor delay, friction, encoder noise, and coupling to the elevation and travel dynamics. All of these effects add extra damping, which further reduces the chance of observing oscillations even if the gains are close to producing complex poles.

Among the tested values, the largest K_{pp} gave the quickest response, and this test showed the only noticeable oscillatory behaviour. This makes sense, because increasing K_{pp} moves the poles further left in the complex plane and increases the speed of the response. For smaller values of K_{pp}, the system behaved more sluggishly and settled without overshoot, which is typical for two real poles where one pole dominates the behaviour.

Overall, the results showed that the model predicts the general trends correctly (larger gains give faster responses), but the real helicopter is more damped than the ideal second–order model. This explains why the system rarely showed the underdamped behaviour that was expected from the theoretical pole placements.

3 Part II - Multivariable control

3.1 Introduction and Motivation

Part II extends the single-input design to a multi-input, multi-output (MIMO) controller that regulates pitch and elevation using state-feedback. The helicopter is linearized around an operating point, and a Linear Quadratic Regulator (LQR) balances tracking performance against input effort via the weighting matrices $\mathbf{Q}$ and $\mathbf{R}$. Compared with direct pole placement, LQR provides a clearer tuning process and improved robustness through explicit trade-offs in $\mathbf{Q}$ and $\mathbf{R}$.

$$J = \int_0^\infty (\mathbf{x}^\top \mathbf{Q_{LQR}} \mathbf{x} + \mathbf{u}^\top \mathbf{R_{LQR}} \mathbf{u})\, dt \tag{1}$$

The work proceeds in two steps. First, the system is written in state-space form and controllability is verified:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{2}$$

$$\begin{bmatrix} \dot{p} \\ \ddot{p} \\ \ddot{e} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \end{bmatrix} \begin{bmatrix} \widetilde{V}_s \\ V_d \end{bmatrix} \tag{3}$$

We checked controllability by the rank of the controllability matrix 4. With rank three, the model is fully controllable.

$$\mathcal{C} = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A^2B} \end{bmatrix} \tag{4}$$

In the second part the system was augmented with another two states to include integral action as follows:

$$\begin{aligned} \dot{\gamma} &= p_c - p \\ \dot{\zeta} &= \dot{e_c} - \dot{e} \end{aligned} \tag{5}$$

$$\begin{bmatrix} \dot{p} \\ \ddot{p} \\ \ddot{e} \\ \dot{\gamma} \\ \dot{\zeta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \\ \dot{e} \\ \gamma \\ \zeta \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \widetilde{V}_s \\ V_d \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_c \\ \dot{e}_c \end{bmatrix} \tag{6}$$

State feedback with reference feedforward is applied. The joystick sets the reference $\mathbf{r}$.

$$\mathbf{r} = \begin{bmatrix} p_c \\ \dot{e}_c \end{bmatrix} \tag{7}$$

The reference-feed-forward has the form:

$$\mathbf{u} = \mathbf{Fr} - \mathbf{Kx}. \tag{8}$$

$$\mathbf{F} = (\mathbf{C}(\mathbf{BK} - \mathbf{A})^{-\mathbf{1}})^{-\mathbf{1}} \tag{9}$$

The optimal gain $\mathbf{K}$ is computed with the MATLAB function $lqr(\mathbf{A}, \mathbf{B}, \mathbf{Q}_{LQR}, \mathbf{R}_{LQR})$.

3.2 Hypothesis

Large state weighting relative to input weighting ($\mathbf{Q} \gg \mathbf{R}$) tends to keep states close to reference, but increases noise sensitivity and risk of jitter or instability. Strong state emphasis can also drive actuators into saturation; during saturation, the system behaves nonlinearly, and linear design assumptions break down.

Excessive input weighting ($\mathbf{R} \gg \mathbf{Q}$) limits motor voltages, yielding slow responses and, in extreme cases, failure to reach the reference.

With the augmented system we expect the system performance to be more reliable. A faster convergence as well as eliminating any possible stationary deviation.

With the augmented design, performance is expected to be more reliable: faster convergence and zero steady-state error. However, overly large $\mathbf{R}$ may still hinder reference tracking, and integrator windup can cause overshoot or oscillations.

3.3 Test Plan

Tuning of $\mathbf{Q}$ and $\mathbf{R}$ is performed step by step, varying one diagonal element at a time. Bryson's rule (10) sets reasonable ranges: each diagonal element equals the inverse square of the maximum allowed value for the corresponding state or input. The procedure begins with identity matrices; one diagonal element is increased to observe the effect. Changes that yield quicker, smoother responses (with acceptable overshoot/oscillation) are retained before moving to the next element. After $\mathbf{Q}$ tuning, the same process is applied to $\mathbf{R}$. While tuning, attention is paid to oscillations, jitters, overshoot, actuator saturation, settling time, and the ability to reach the reference.

$$\begin{aligned} \mathbf{Q} &= \begin{bmatrix} \frac{1}{p_{max}^2} & 0 & 0 \\ 0 & \frac{1}{\dot{p}_{max}^2} & 0 \\ 0 & 0 & \frac{1}{\dot{e}_{max}^2} \end{bmatrix} \\ \mathbf{R} &= \begin{bmatrix} \frac{1}{\widetilde{V}_{s,max}} & 0 \\ 0 & \frac{1}{V_{d,max}} \end{bmatrix} \end{aligned} \tag{10}$$

After adding the integral action, two additional parameters needed to be tuned by repeating the procedure. During testing, integrator windup was a new potential concern, as it can cause overshoot, oscillations, or, in the worst case, instability.

3.4 Results

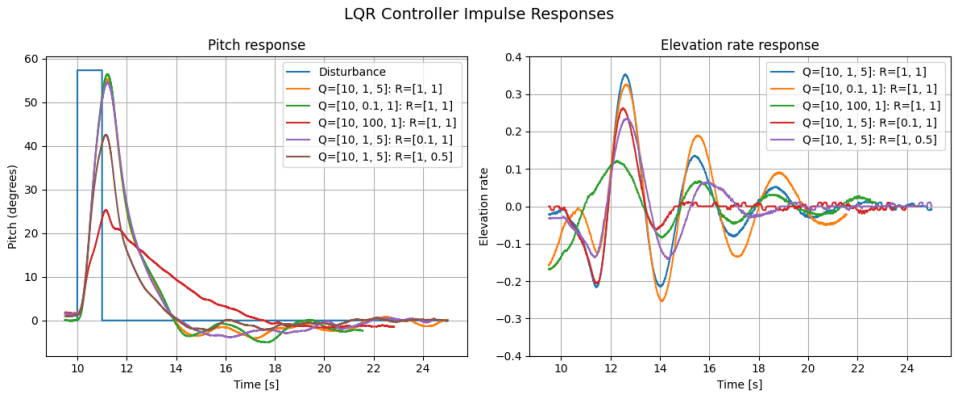


Figure 3: LQR results without integral action.

We consider the state vector $x = [p, \dot{p}, \dot{e}]^\top$ and inputs $u = [V_s, V_d]^\top$, with an LQR cost 1, where $Q = diag(q_p, q_{\dot{p}}, q_{\dot{e}})$ and $R = diag(r_{V_s}, r_{V_d})$. For simplicity, we refer to the diagonals of Q and R by their entries. Responses were recorded for a small impulse to keep the motion near the linearization point. Figure 3 shows representative pitch and elevation-rate responses for the choices of $\mathbf{Q}$ and $\mathbf{R}$

Starting with values inspired by Bryson's rule, we used $\mathbf{Q}' = [10, 1, 5]$, while using $R = \mathbb{I}$. This configuration produced a quick pitch response with some overshoot, but the elevation-rate was too lightly dampened, resulting in significant oscillations. The behavior is consistent with the coupling between pitch and elevation through the mechanics and the shared actuation.

We then decreased the penalties on the rate states. Pitch dynamics changed relatively little, but elevation-rate damping decreased.

To improve damping, we increased pitch-rate weight substantially and kept the others moderate. Pitch then moved a bit slower but settled cleanly with little or no overshoot. At the same time, the elevation rate improved slightly.

Lastly, we adjusted the motor voltage weights $\mathbf{R}$. The first element corresponds to the weight of the motor voltages V_s, and the second to the difference in voltage between the motors V_d. By following Bryson's rule we first reduced the motor voltage weighting. This resulted in a slightly better response in both pitch and a noticeable reduction in oscillations in elevation rate.

When the V_d weighting was decreased oscillations in pitch response improved slightly, but oscillations were re-introduced in the elevation rate.

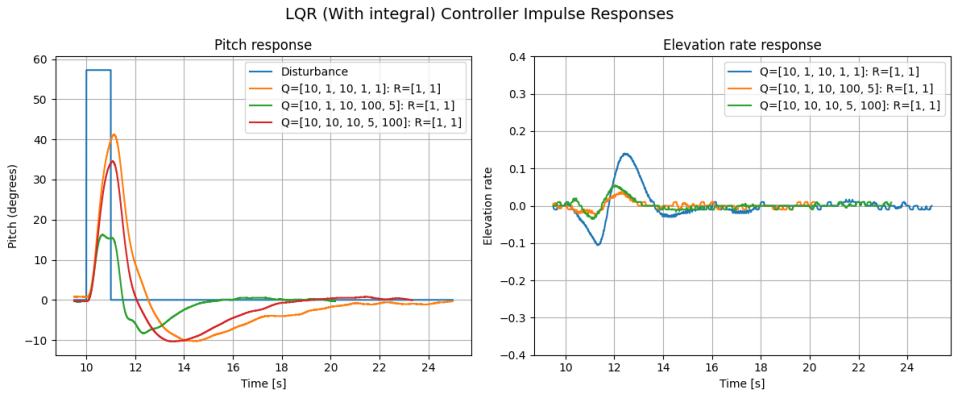


Figure 4: LQR results without integral action.

After the introduction of integral effect we decided to go with $\mathbf{R} = \mathbb{I}$ to reduce the chance of motor saturation aswell as for simplicity sake.

First we kept the integral action weighting at 1. This gave the response seen in Figure 4. It's a pretty fast response with no steady state deviations compared with previous results, though a pretty significany overshoot in both pitch and elevation rate response occurs. Compared with no integral action in Figure 5 the response is much faster.

The weight of the integral action of the pitch was then greatly increased to 100. This resulted in a very fast response aswell as reduced overshoot in both pitch and elevation rate.

Lastly the weight of the 2nd integral action was increased. This gave the response in Figure 4.

3.5 Conclusion

Although Bryson's rule provided a reasonable starting point, it proved insufficient on its own. The values of the weight did not correspond to the maximum of the corresponding state in our results. A better approach was to use it for initial values of the weight matrices, then tune them based on the system's performance.

It's also very clear that integral action improves control of the system. It leads to a quicker response with no steady state error. It is important to either avoid motor saturation by tuning weight matrices or add anti-windup measures such as clamping to avoid integrator windup.

In summary, the experiments show that rate penalties in Q are the primary lever for damping and overshoot control. Bryson's rule is useful for initialization, but effective performance requires tuning $\mathbf{Q}$ to realistic operating ranges and shaping $\mathbf{R}$ to reflect actuator limits.

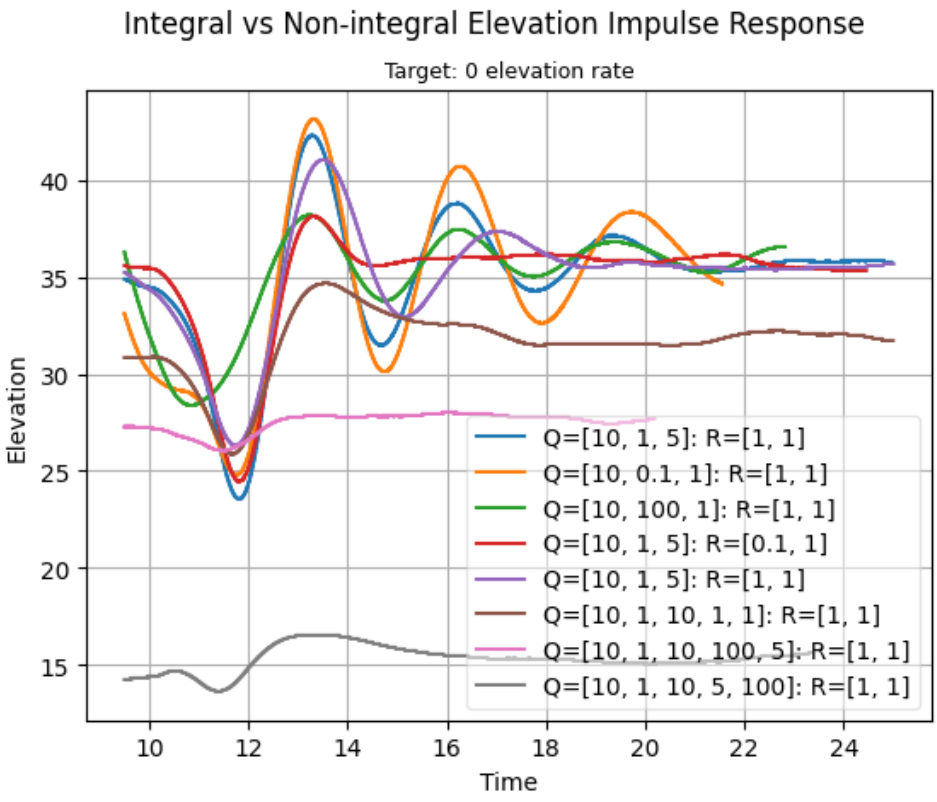


Figure 5: Integral vs no integral action.

4 Part III - Luenberger observer

4.1 Introduction and Motivation

In Part III, measurements were obtained using an inertial measurement unit (IMU). The IMU introduces considerable noise into measured signals. To mitigate noise effects, a state estimator in the form of a Luenberger observer is implemented. Observers estimate unmeasured or excessively noisy states, provided the system is observable. The resulting state-space formulation is:

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

$$\begin{bmatrix} \dot{p} \\ \ddot{p} \\ \dot{e} \\ \ddot{e} \\ \ddot{\lambda} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ K_3 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \\ e \\ \dot{e} \\ \dot{\lambda} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ 0 & 0 \\ K_2 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \widetilde{V}_s \\ V_d \end{bmatrix} \tag{11}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x}$$

$$\begin{bmatrix} p \\ e \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \\ e \\ \dot{e} \\ \dot{\lambda} \end{bmatrix} \tag{12}$$

As for observability the Matlab function $rank(obsv(\mathbf{A}, \mathbf{C}))$ was used to find the rank of the system. By trial and error the $\mathbf{C}$ matrix was adjusted until the system was observable (i.e. full rank). This resulted in the required states being e and $\dot{\lambda}$. The same conclusion can be made by looking at the state space formulation as the remaining states are only coupled with those two.

As the IMU only measures acceleration, some processing was required to measure p and e. The following equations were derived from the physical representation of the system described in 1:

$$\begin{aligned} p &= \arctan\left(\frac{a_y}{a_z}\right) \\ e &= \arctan\left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}\right) \end{aligned} \tag{13}$$

The linear observer in 14 was then implemented.

$$\dot{\hat{x}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{L}(\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}) \tag{14}$$

The observer gain $\mathbf{L}$ was selected to minimize deviation between measured outputs and estimated states and was computed by pole placement

using the MATLAB function $\texttt{place}(\mathbf{A}', \mathbf{C}', \mathbf{p})'$, where $\mathbf{p}$ contains the desired estimator poles.

4.2 Test plan

The test plan prioritizes estimator response speed. Poles near the origin in the left half-plane (LHP) yield slower estimator dynamics than poles placed farther into the LHP. To avoid instability capable of damaging the helicopter, initial poles are placed on the real axis close to the origin. Based on observed performance, poles are shifted leftward until convergence becomes sufficiently fast. Under careful monitoring, large negative poles are then tested. Finally, poles are placed in the right half-plane (RHP) to confirm the onset of instability.

4.3 Results

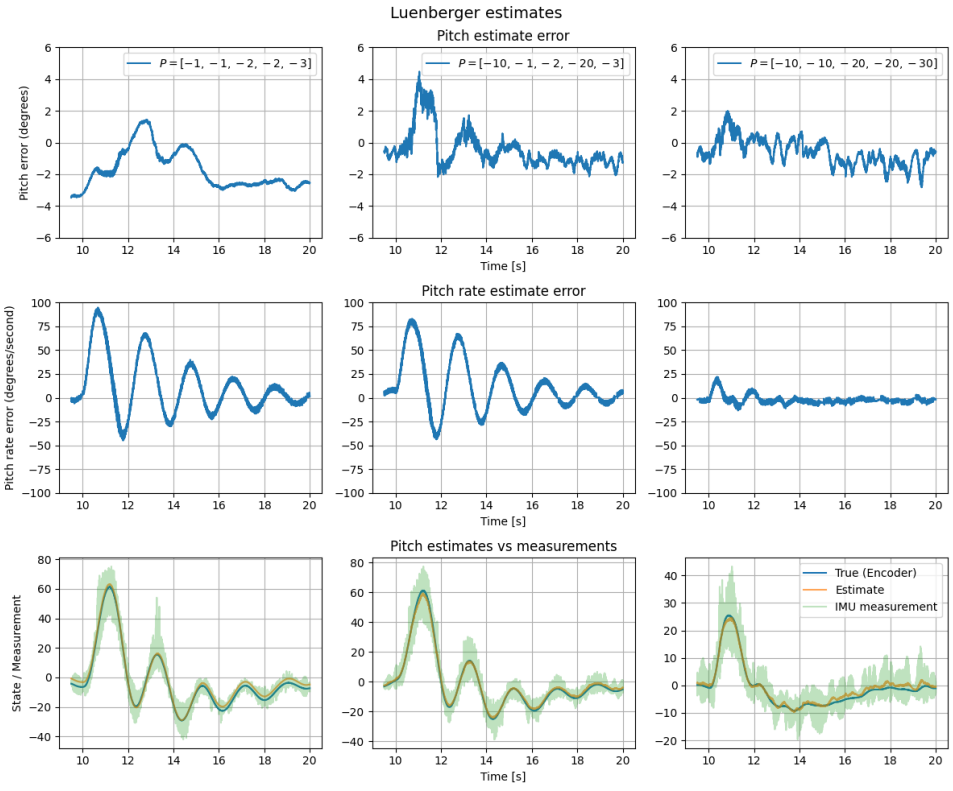


Figure 6: Plot of response for different poles used for Luenberger observer.

The results of pole placement for the Luenberger observer are shown in Figure 3, and all experiments used real poles. With the initial choice $p = [-1, -1, -2, -2, -3]$, the estimator performance was poor: convergence

was slow, the pitch estimate had a clear steady-state bias, and the pitch rate error oscillated with a large time constant. The bias caused the helicopter to drift away from the linearization point.

To improve the estimator, some poles were moved further into the LPH $p = [-10, -1, -2, -20, -3]$. This improved the estimator slightly. More noise is introduced in the pitch estimate error, but the bias has been significantly decreased. Pitch rate error still has a lot of oscillations with a large time constant.

Lastly the rest of the poles were moved, resulting in $p = [-10, -10, -20, -20, -30]$. This change yielded a markedly better estimator: the response became faster, the estimation error stayed close to zero, and oscillations in the pitch-rate estimate were largely suppressed. A small residual bias in pitch remained, consistent with IMU-based angle computation and unmodeled effects.

A more aggressive placement $p = [-100, -100, -200, -200, -300]$ was also tested. This led to an unstable implementation and unusable data.

Overall, the moderate placement $p = [-10, -10, -20, -20, -30]$ provided the best trade-off between speed and noise sensitivity: it was sufficiently fast to track the plant without noticeably amplifying measurement noise.

4.4 Conclusion

Pole-placement tuning of the Luenberger observer shows that moving the estimator poles moderately into the left half-plane improves estimation quality, while overly aggressive placements amplify noise and can destabilize the system. The configuration $p = [-10, -10, -20, -20, -30]$ achieved fast convergence with minimal oscillations and only a small bias, achieving both a responsive and robust system.

5 Part IV - Kalman filter

5.1 Introduction and Motivation

To obtain an optimal state estimate under sensor noise and model uncertainty, a discrete-time Kalman filter is implemented. The Kalman filter minimizes the mean-square estimation error. Compared with a Luenberger observer tuned by pole placement, the Kalman filter explicitly incorporates the measurement-noise covariance and process-noise covariance. The correction (update) step of the discrete filter is

The Kalman gain $\mathbf{L_k}$ is given by:

$$\mathbf{L_k} \;=\; \mathbf{P_k}^{-}\,\mathbf{C}^{\top}\left(\mathbf{C}\,\mathbf{P_k}^{-}\,\mathbf{C}^{\top}+\mathbf{R_d}\right)^{-1} \tag{15}$$

Update the estimate with the measurement:

$$\hat{\mathbf{x}}_\mathbf{k} \;=\; \hat{\mathbf{x}}_\mathbf{k}^{-}+\mathbf{L_k}\big(\mathbf{y_k}-\mathbf{C}\,\hat{\mathbf{x}}_\mathbf{k}^{-}\big) \tag{16}$$

Compute error covariance for updated estimate:

$$\mathbf{P_k} \;=\; (\mathbf{I}-\mathbf{L_k}\mathbf{C})\,\mathbf{P_k}^{-}\,(\mathbf{I}-\mathbf{L_k}\mathbf{C})^{\top}+\mathbf{L_k}\mathbf{R_d}\mathbf{L_k}^{\top}. \tag{17}$$

The prediction step which estimates the next time step:

$$\begin{aligned}\hat{\mathbf{x}}_{\mathbf{k+1}}^{-} &= \mathbf{A}\hat{\mathbf{x}}_\mathbf{k}+\mathbf{B}\mathbf{u_k}+\mathbf{w_k}\\ \mathbf{y_k} &= \mathbf{C}\mathbf{x_k}+\mathbf{v_k}.\end{aligned} \tag{18}$$

$$\hat{\mathbf{P}}_{\mathbf{k+1}}^{-} = \mathbf{A}\hat{\mathbf{P}}_\mathbf{k}\mathbf{A}^{\top}+\mathbf{Q} \tag{19}$$

Process noise $\mathbf{w_k}$ and measurement noise $\mathbf{v_k}$ are zero-mean white Gaussian, described by 20:

$$\begin{aligned}\mathbf{w} &\sim \mathbf{N}(0,\mathbf{Q}_d)\\ \mathbf{v} &\sim \mathbf{N}(0,\mathbf{R}_d)\end{aligned} \tag{20}$$

The state vector is $\mathbf{x} = \begin{bmatrix} p & \dot{p} & e & \dot{e} & \lambda & \dot{\lambda}\end{bmatrix}^{\top}$. This choice yields the following continuous-time model in 21:

$$\mathbf{A}=\begin{bmatrix}0&1&0&0&0&0\\0&0&0&0&0&0\\0&0&0&1&0&0\\0&0&0&0&0&0\\0&0&0&0&0&1\\K_3&0&0&0&0&0\end{bmatrix},\mathbf{B}=\begin{bmatrix}0&0\\0&K_1\\0&0\\K_2&0\\0&0\\0&0\end{bmatrix},\mathbf{C}=\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\end{bmatrix}. \tag{21}$$

The discrete-time matrices differ from the continuous-time matrices. Discrete-time versions $\mathbf{A_d}, \mathbf{B_d}, \mathbf{C_d}$ were obtained using the *c2d* function in MATLAB with a sampling period of 0.002 s.

The process-noise covariance $\mathbf{Q_d}$ serves as a tuning parameter. The measurement-noise covariance $\mathbf{R_d}$ was estimated by measuring the helicopter at the linearization point while running the controller and estimator from the previous parts.

5.2 Hypothesis and Test Plan

The covariance $\mathbf{Q_d}$ controls the assumed process noise and thereby the reliance on measurements versus the model. Increasing $\mathbf{Q_d}$ increases the Kalman gain and places more weight on measurements; decreasing $\mathbf{Q_d}$ places more weight on the model. Similar to Luenberger observer design, choosing $\mathbf{Q_d}$ too small relative to $\mathbf{R_d}$ yields a slow, overconfident filter, whereas choosing it too large makes the filter chase measurement noise, producing jittery or oscillatory estimates.

We begin with a small value relative to $\mathbf{R_d}$ and increase it incrementally until undesirable behavior appears; then we decrease it to the smallest value that achieves rapid convergence without noticeable jitter.

5.3 Results

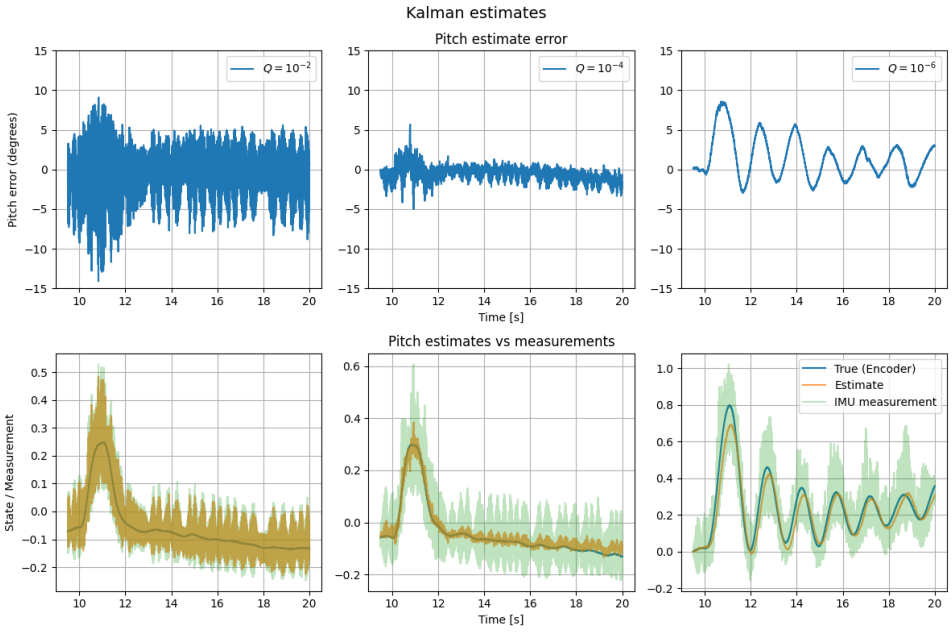


Figure 7: Plot of impulse response for different $\mathbf{Q}$ matrices for the Kalman filter.

First the measurement covariance matrix was found:

$$\mathbf{R_d} = \begin{bmatrix} 0.0036 & 0.0026 & 0.0010 & -0.0049 & 0.0916 & 0.0032 \\ 0.0026 & 0.0311 & 0.0289 & -0.0058 & -0.0908 & 0.0029 \\ 0.0010 & 0.0289 & 0.0390 & -0.0039 & -0.5889 & -0.0066 \\ -0.0049 & -0.0058 & -0.0039 & 0.0131 & 0.0576 & 0.0010 \\ 0.0916 & -0.0908 & -0.5889 & 0.0576 & 201.4733 & 3.4459 \\ 0.0032 & 0.0029 & -0.0066 & 0.0010 & 3.4459 & 0.0939 \end{bmatrix}. \tag{22}$$

As most of the values in the $\mathbf{R_d}$ matrix is in the order of 10^{-2} the first measurement was done using this value. This resulted in a desired response, but with significant noise in the states as seen in Figure 7.

$\mathbf{Q_d}$ was then decreased further to 10^{-4}, putting more trust in the plant. This resulted in a lot less noise in the system, while still having a good response.

Lastly $\mathbf{Q}$ was set to 10^{-4}. This resulted in a poor estimator with oscillations in the output. The system was still stable, but oscillations reduced the performance.

5.4 Conclusion

With the measured $\mathbf{R_d}$ in place, tuning the process noise $\mathbf{Q_d}$ was fairly straightforward. Too large and the filter became jittery, too small and it deviated from the physical system. A middle range value of around 10^{-4} achieved a good balance, giving fast, stable estimates that stayed close to the encoder and filtered out IMU noise. Compared with the Luenberger observer, the Kalman filter handled sensor noise a lot better. It removed any bias in the estimation error, achieved a good balance between noise and performance and was a lot easier to tune.

References

[1] EJ. *'Helicopter lab assignment'*. Department of Engineering Cybernetics NTNU. 2025.

[2] EJ. *'Helicopter lab preparation'*. Department of Engineering Cybernetics NTNU. 2025.

«I arbeidet med denne oppgaven har vi benyttet kunstig intelligens (KI) som verktøy på følgende måte: Chat-GPT ble brukt til å foreslå formuleringer og gi språklige forbedringer, men alt innhold er gjennomgått, redigert og faglig vurdert av oss. Chat-GPT er ikke brukt til å generere hele avsnitt eller til å besvare oppgaven i sin helhet. Vi har heller ikke lastet opp personopplysninger eller sensitiv informasjon i KI-verktøyet. Vi står selv ansvarlig for det faglige innholdet i oppgaven.».

A Simulink diagrams

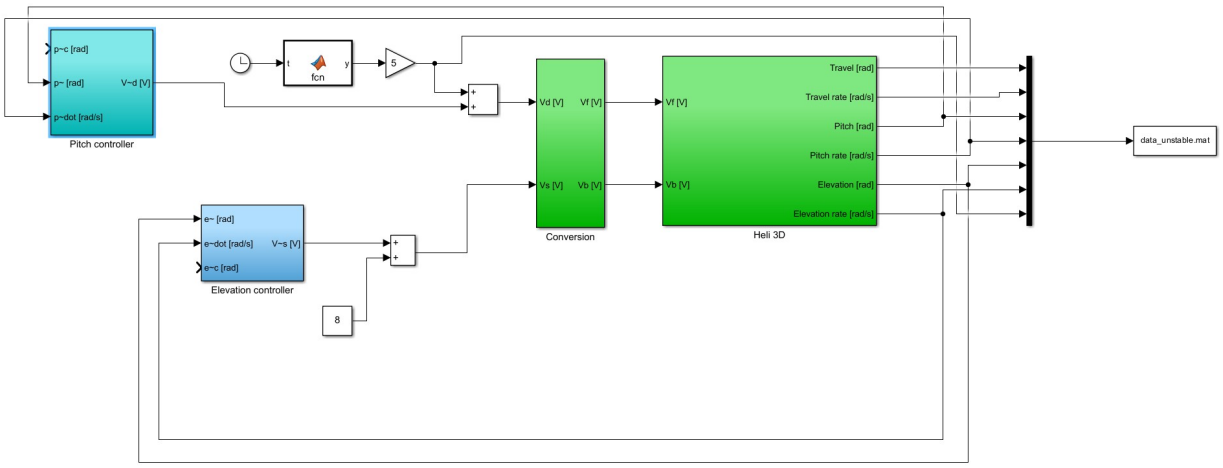


Figure 8: Simulink diagram for Part I.

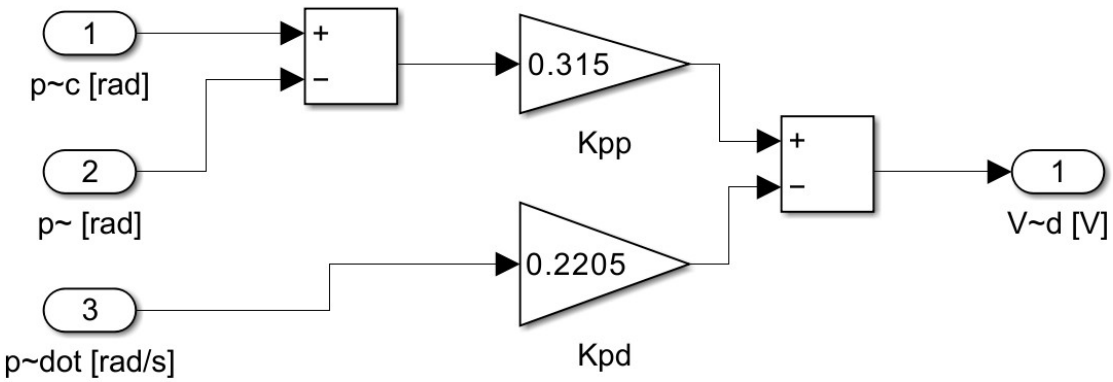


Figure 9: Pitch controller.

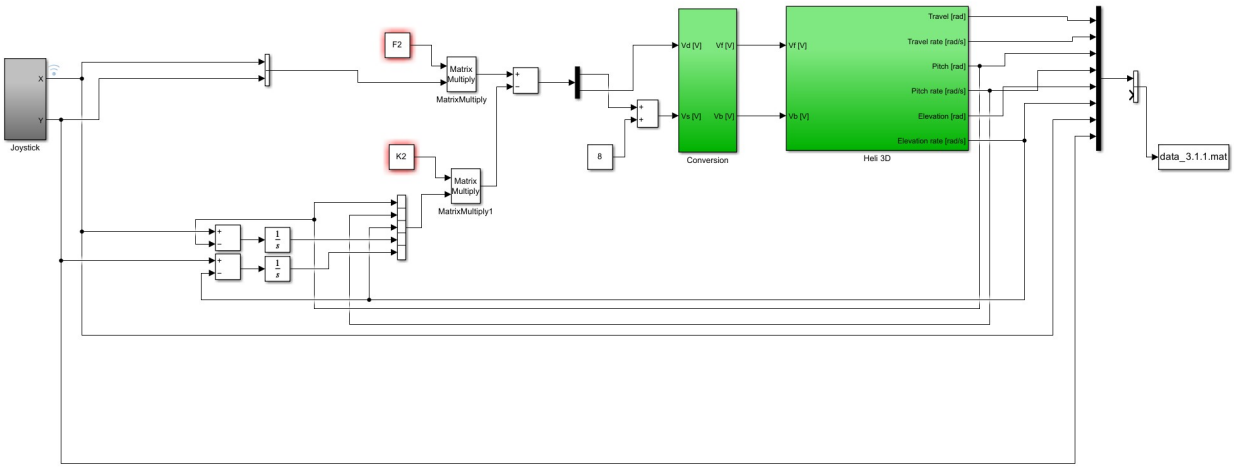


Figure 10: Simulink diagram for Part II

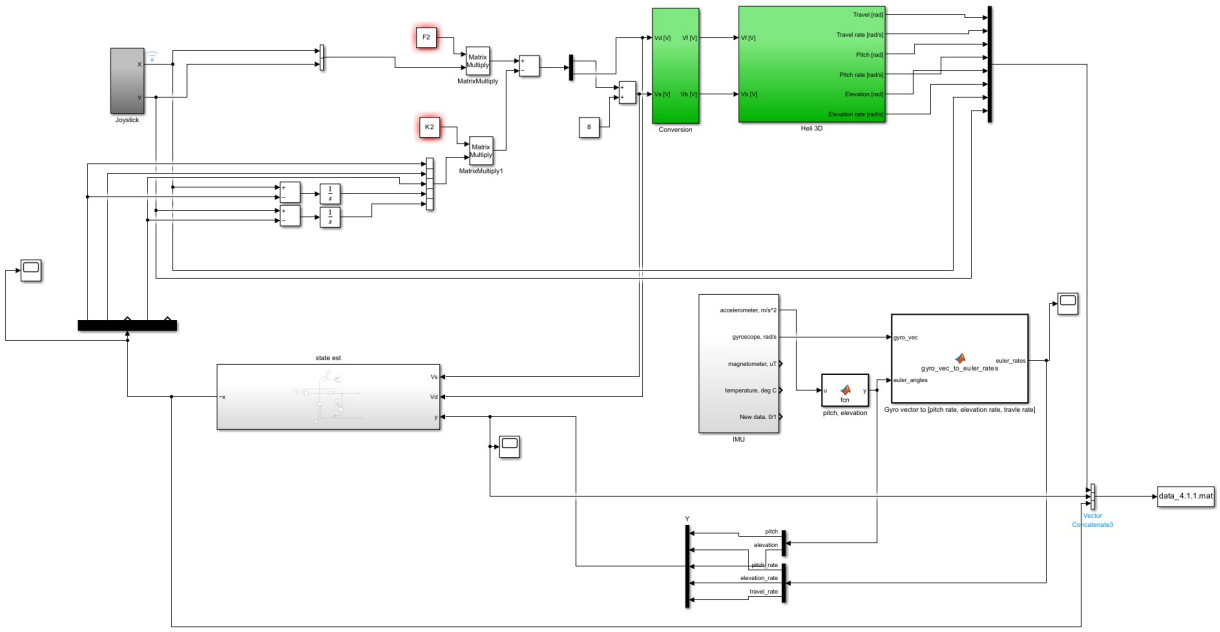


Figure 11: Simulink diagram for Part III

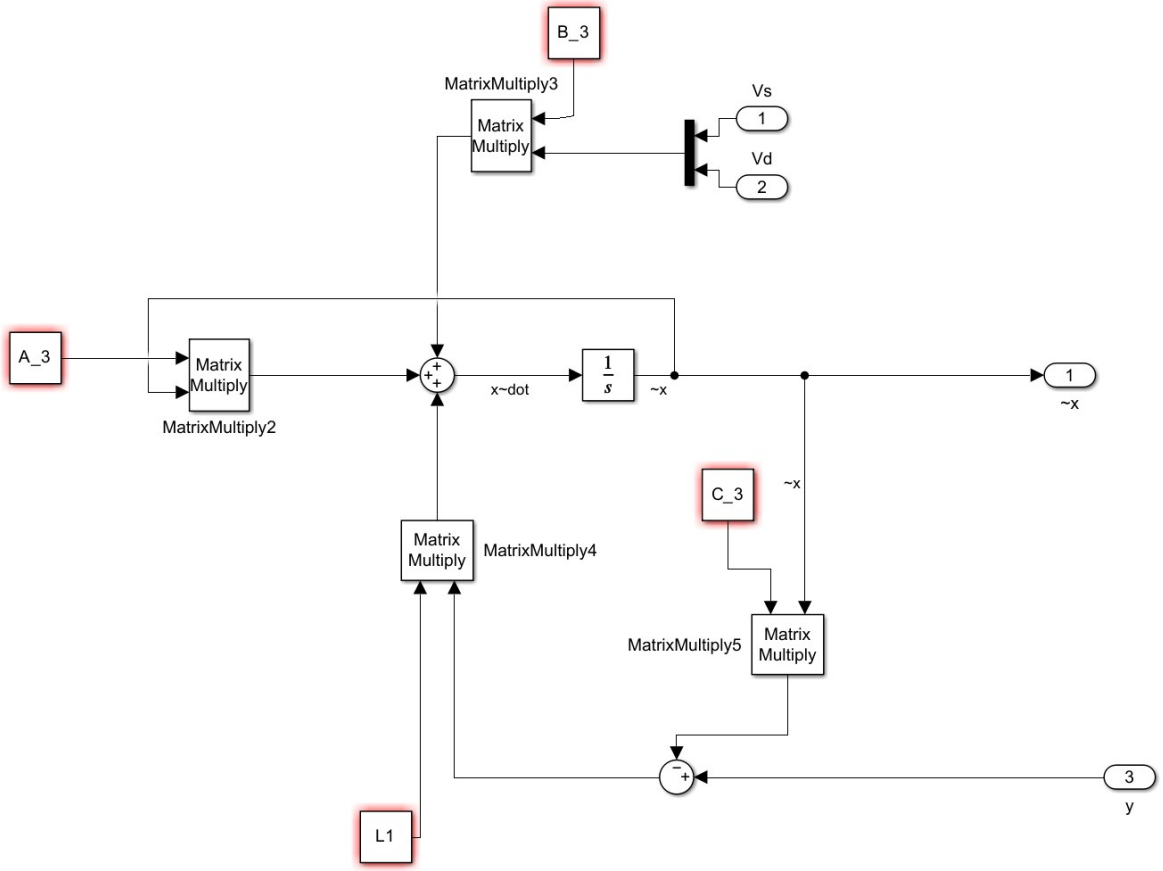


Figure 12: Simulink diagram for Luenberger estimator.

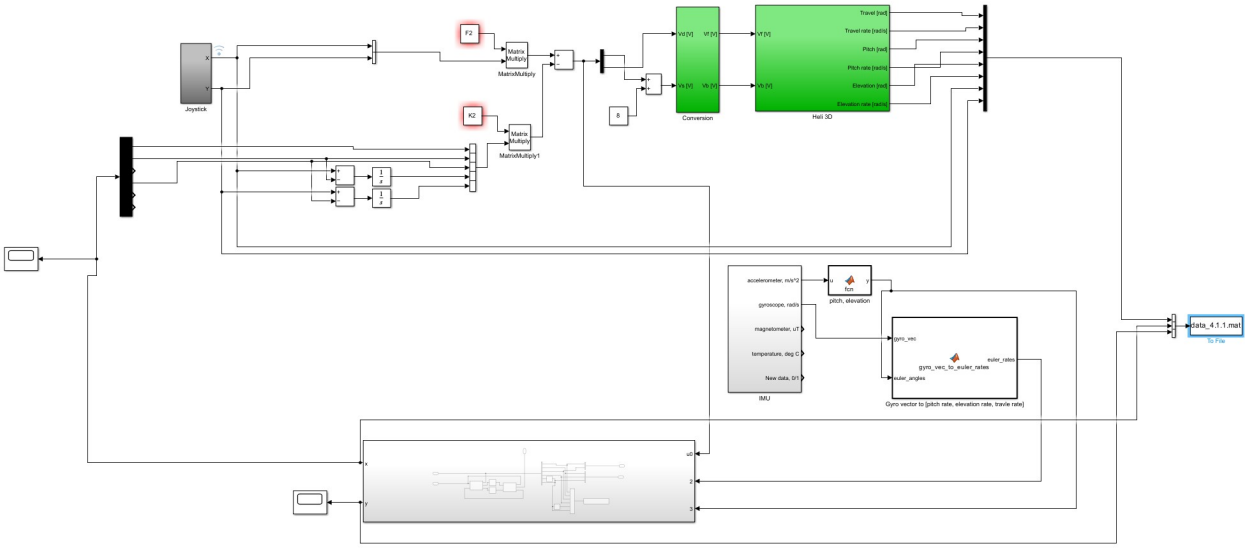


Figure 13: Simulink diagram for Part IV.

Figure 14: Simulink diagram for Kalman filter.