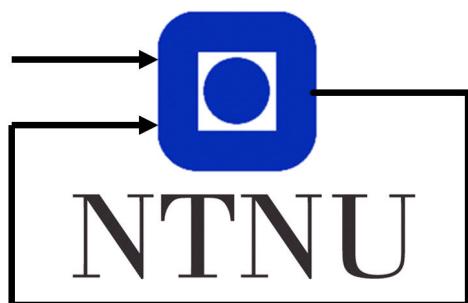


TTK4115 Linear System Theory Lab Report

Group 58
Anders Thallaug Fagerli: 758034
Bendik Åshaug Holm: 478445

November 18, 2019



Department of Engineering Cybernetics

Contents

1 Modeling, validation and manual control	1
1.1 Equations of motion	2
1.2 Parameter identification	3
1.3 Verification and manual control	4
2 Mono-variable control	8
2.1 PD Controller	8
2.2 Pole Placement	8
2.3 Harmonic Oscillator	11
3 Multi-variable control	13
3.1 State-space formulation	13
3.2 Linear Quadratic Regulators	13
3.3 Integral action	16
4 Intertial Measurement Unit	19
4.1 IMU characteristics	19
4.2 Gyroscope transform	21
4.3 Observability part I	22
4.4 Accelerometer	23
4.5 Observability part II	24
4.6 Noise	24
5 The Kalman filter	28
5.1 Discretization	29
5.2 State prediction	29
5.3 Error covariance prediction	30
5.4 State correction	31
5.5 Error covariance correction	31
5.6 Weighting matrix K	32
5.7 Tuning	32
5.8 The Kalman filter	35
6 Conclusion	37
Appendix	38
A Derivations	38
A.1 Modeling, validation and manual control	38
A.1.1 Equations of motion derivation	38
A.2 Mono-variable control	38
A.2.1 Transfer function derivation	38
A.3 Multi-variable control	38

A.3.1	Reference matrix derivation	38
A.3.2	Reference matrix integral action derivation	39
A.4	The Kalman filter	39
A.4.1	Error covariance prediction derivation	39
A.4.2	Error covariance correction derivation	40
A.4.3	Kalman gain derivation	41
B	MATLAB Code	41
B.1	acc_to_euler_angles.m	41
B.2	kalman_filter.m	41
C	MATLAB Figures	43
C.1	Equations of motion	43
C.2	Linear Quadratic Regulators	43
D	Simulink Block Diagrams	44
D.1	Equations of motion	44
D.2	Mono-variable control	45
D.3	Multi-variable control	46
D.4	Inertial Measurement Unit	47
D.5	The Kalman filter	48

1 Modeling, validation and manual control

This lab entails the modeling and control of the system illustrated in fig. 1¹. Two propellers, assumed to have equal masses of m_p , are connected by rods of length l_p to a rotating joint. The rotation angle of the propellers, the pitch, about this joint is denoted p . The joint is connected by a rod of length l_h to another joint capable of rotating so as to elevate the propeller craft off the ground. The other side of the elevation joint is connected by a rod of length l_c to a counterweight of mass m_c . The rotational angle of the craft about the elevation joint is denoted e . The elevation joint is also connected by a rod tethered to another vertical rotating joint perpendicular to the ground, allowing the entire structure to rotate about the vertical joint with a travel angle λ .

The propellers are capable of flight by applying the voltages V_f and V_b to the front and back propellers, respectively. This generates thrust with respective forces $F_f = K_f V_f$ and $F_b = K_f V_b$. The constant K_f , also known as the motor force constant, is assumed to be equal for both motors. The lift of the craft is provided by the sum of the thrust forces $F_f + F_b = K_f(V_f + V_b)$, and is counteracted by the gravitational forces acting on the two propellers, $F_{g,f} = m_p g$ and $F_{g,b} = m_p g$, respectively. The thrust is aided by gravity acting on the counterweight, $F_{g,c} = m_c g$. Rotation about the pitch axis is caused by the difference in the propeller thrust forces, given by $F_b - F_f = K_f(V_f - V_b)$.

Manipulating the sum of the propeller voltages to produce lift, as well as the difference of the propeller voltages to generate pitch, the system will be controlled to a desired pitch, elevation and travel.

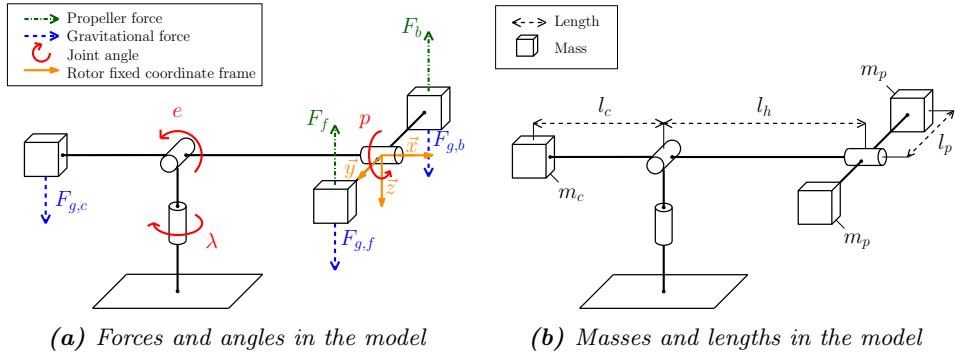


Figure 1: Depiction of relevant parameters in the model.

¹Figure 1 is sourced from material published on Blackboard.

1.1 Equations of motion

The system in fig. 1 can be described by the set of coupled ordinary differential equations for pitch p , elevation e and travel λ , by using Newtons 2nd law of motion for rotation,

$$J_\alpha \dot{\omega} = \sum \tau_\alpha = \sum F \cdot l_\alpha \quad (1)$$

where J_α is the moment of inertia about some angle α , $\dot{\omega}$ is the angular acceleration, τ_α is the momentum about the angle α and F is the force applied at distance l_α from the joint angle α .

Applying eq. (1) to the different angles of rotation in fig. 1a, the equations of motion can be derived for p , e and λ . Note the direction of positive rotation for the different angles from fig. 1a.

$$J_p \ddot{p} = l_p(K_f V_b - K_f V_f) \quad (2a)$$

$$J_e \ddot{e} = m_c g l_c \cos(e) - 2m_p g l_h \cos(e) + (K_f V_f + K_f V_b) l_h \cos(p) \quad (2b)$$

$$J_\lambda \ddot{\lambda} = (K_f V_f + K_f V_b) l_h \sin(p) \cos(e) \quad (2c)$$

As it is desired to model this as a linear system, the system is linearized around $[p \ e \ \lambda]^T = [0 \ 0 \ 0]^T$, giving the configuration in fig. 1. Using the small angle approximations, $\cos(x) \approx 1$ and $\sin(x) \approx x$, the equations of motion in eq. (2) may be linearized to,

$$J_p \ddot{p} = l_p(K_f V_b - K_f V_f) \quad (3a)$$

$$J_e \ddot{e} = m_c g l_c - 2m_p g l_h + (K_f V_f + K_f V_b) l_h \quad (3b)$$

$$J_\lambda \ddot{\lambda} = (K_f V_f + K_f V_b) l_h p \quad (3c)$$

Defining the sum of motor voltages as $V_s = V_f + V_b$ and the difference in motor voltage as $V_d = V_b - V_f$, eq. (3) may be re-written as,

$$J_p \ddot{p} = L_1 V_d \quad (4a)$$

$$J_e \ddot{e} = L_2 (V_s - V_{s,0}) \quad (4b)$$

$$J_\lambda \ddot{\lambda} = L_3 V_s p \quad (4c)$$

where $L_1 = K_f l_p$, $L_2 = K_f l_h$, $L_3 = K_f l_h$ and $V_{s,0} = \frac{2m_p g l_h - m_c g l_c}{K_f l_h}$ are constants. $V_{s,0}$ is here defined as the voltage needed to establish an equilibrium at the linearization point, and is found in eq. (3b) by setting $\ddot{e} = 0$ and solving for V_s , giving,

$$J_e \cdot 0 = m_c g l_c - 2m_p g l_h + K_f l_h V_{s,0}$$

$$V_{s,0} = \frac{2m_p g l_h - m_c g l_c}{K_f l_h}$$

Defining $\tilde{V}_s = V_s - V_{s,0}$ and solving eq. (4) for the angular acceleration about each angle, the final equations are given as,

$$\ddot{p} = K_1 V_d \quad (5a)$$

$$\ddot{e} = K_2 \tilde{V}_s \quad (5b)$$

$$\ddot{\lambda} = K_3 p \quad (5c)$$

where $K_1 = \frac{L_1}{J_p}$, $K_2 = \frac{L_2}{J_e}$ and $K_3 = \frac{V_{s,0} L_3}{J_\lambda}$ are constants. A linearization is here done for eq. (5c), which otherwise is expressed as,

$$\ddot{\lambda} = L_3 V_s p$$

$$\ddot{\lambda} = L_3 (\tilde{V}_s + V_{s,0}) p \quad (6)$$

Since $\ddot{\lambda}$ in eq. (6) is a function of \tilde{V}_s and p , which again are functions of time, eq. (6) is a higher order ODE. Linearization is done by omitting the higher order term \tilde{V}_s , and the resulting equation of motion for travel is eq. (5c).

1.2 Parameter identification

The encoder values for pitch, elevation and travel are reset every time Simulink is connected, normally with the helicopter-head resting on the table. This gives an offset from the depicted equilibrium configuration in fig. 1. As the pitch is at equilibrium while the helicopter-head is resting on the table, and the dynamics of the system is independent of where the travel angle is reset, the only motion to be configured is elevation. This can be configured by adding a constant to the encoder output for elevation. This constant was found experimentally to be approximately -31° when the helicopter-head is resting on the table.

The motor force constant K_f is needed to express the dynamics of the system, and can be found by solving $V_{s,0}$ for K_f , yielding eq. (7a) and eq. (7b):

$$V_{s,0} = \frac{2m_p g l_h - m_c g l_c}{K_f l_h} \quad (7a)$$

$$K_f = \frac{2m_p g l_h - m_c g l_c}{V_{s,0} l_h} \quad (7b)$$

Symbol	Parameter/Constant	Value	Unit
l_c	Distance from elevation axis to counterweight	0.46	m
l_h	Distance from elevation axis to helicopter head	0.66	m
l_p	Distance from pitch axis to motor	0.175	m
m_p	Motor mass	0.72	kg
m_c	Counterweight mass	1.92	kg
J_e	Moment of inertia for elevation	1.0335	kg m^2
J_λ	Moment of inertia for travel	1.0776	kg m^2
J_p	Moment of inertia for pitch	0.0441	kg m^2
e_{offset}	Elevation offset for equilibrium	-31	$^\circ$
K_1	Constant in equation of motion for pitch	0.5640	$1/\text{Vs}^2$
K_2	Constant in equation of motion for elevation	0.0908	$1/\text{Vs}^2$
K_3	Constant in equation of motion for travel	0.6117	$1/\text{s}^2$
K_f	Motor force constant	0.1421	N/V
$V_{s,0}$	Motor voltage offset for equilibrium	7.028	V

Table 1: Parameters and physical constants

$V_{s,0}$ is found by applying input through the joystick controller until the helicopter floats stationary in the horizontal position at equilibrium, while reading off the applied voltage. This was found to be approximately $V_{s,0} \approx 7\text{ V}$. Inserting this into eq. (7b), the motor force constant for the helicopter motors is calculated to be $K_f \approx 0.1421\text{ N/V}$. In total, the parameters and physical constants are summarized in table 1.

1.3 Verification and manual control

The proposed system in the previous sections can be verified by implementing the system in Simulink and comparing the outputs of the linearized model with the encoder outputs. Figure 22 in appendix D depicts the proposed system, where eq. (5) are solved for travel, pitch and elevation, and subsequently compared with the respective encoder values, here serving as ground truth. The joystick controller is here used as input.

Pitch

Figure 2 depicts how the linearized model compares to the encoder values for pitch. It is difficult to see how the model behaves around $p = 0$ from this plot, as the increase in angle from equilibrium results in a large deviation in the model, and will then scale the y-axis to extreme values. This illustrates how the model performs outside its linearization point, where large deviations result in unstable behavior that is difficult to counteract. This is a result of the integrator blocks in appendix D, fig. 22, which integrate up a large value

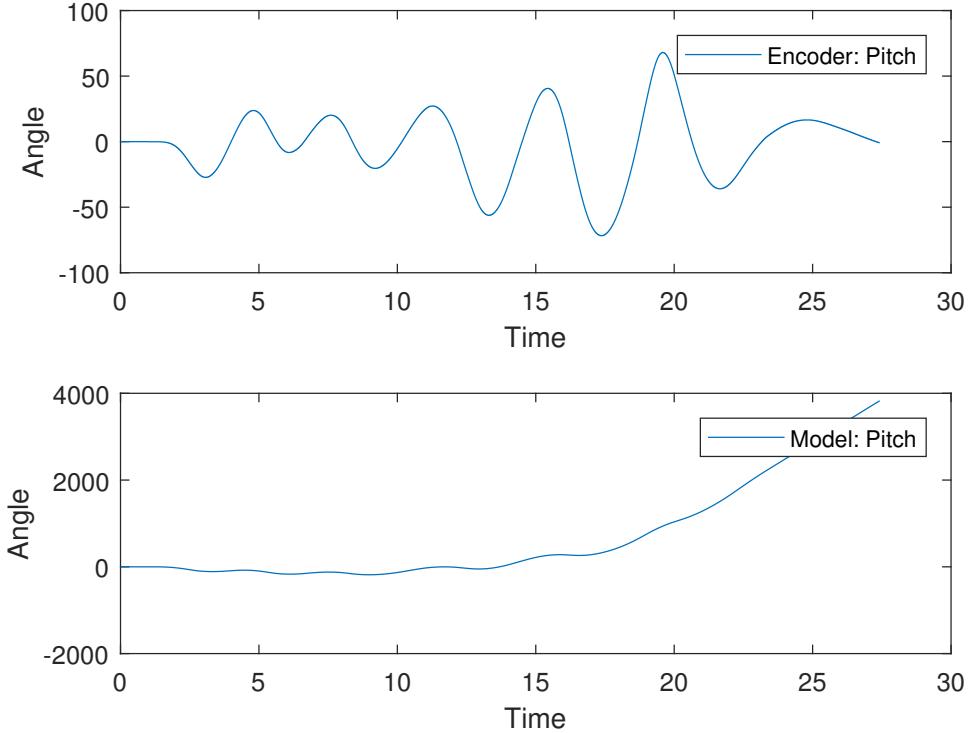


Figure 2: Pitch angle from encoder and model for increasing angles

over time.

To illustrate the behavior around $p = 0$, fig. 19 is included in appendix C. As expected, the model performs better at small angles and follows the encoder values to some degree, but there is still some deviation from the encoder values.

Much of the deviation is explained by how the helicopter is controlled by joystick. The model expects a continuous changing voltage applied to the helicopter motors when it is pitching back and forth, which is not always the case as a short excitation in pitch from the joystick may result in a larger change in pitch on the helicopter, a result of the pendulum motion the helicopter is exposed to about the pitch joint.

Travel

Figure 3 shows how the linearized model compares to the encoder values for travel and pitch. Pitch is here depicted for reference, as travel is a function of pitch. The model for travel deviates greatly from the encoder values, and is incomparable with any reasonable motion of travel. This is most likely a result of the four integration blocks in appendix D, fig. 22, integrating up deviations from equilibrium for both pitch and travel. In order to keep the system at equilibrium for both angles, the helicopter must be controlled very

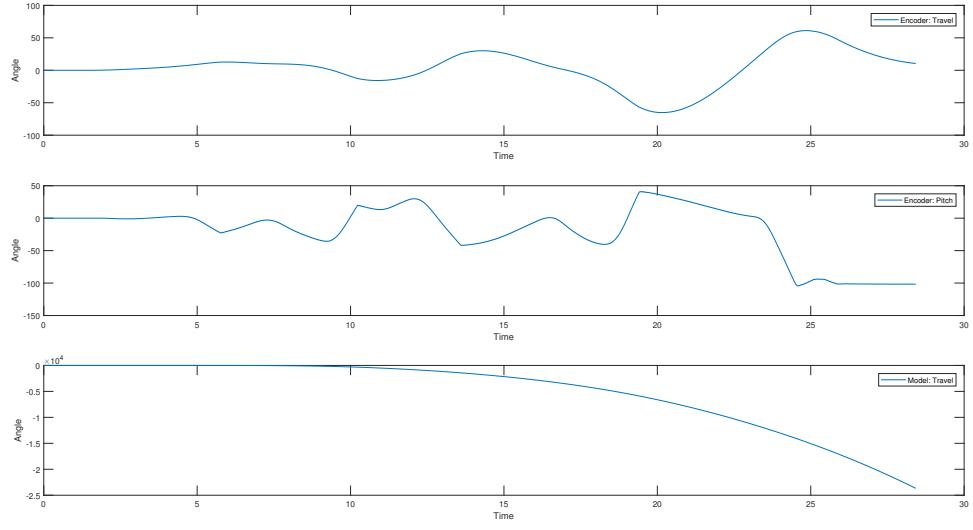


Figure 3: Travel and pitch angle from encoder and model for increasing angles

well, something that proved difficult with the joystick. It is therefore difficult to reason about the model correctness around equilibrium, but fig. 3 clearly shows how it deviates for angles outside equilibrium.

Elevation

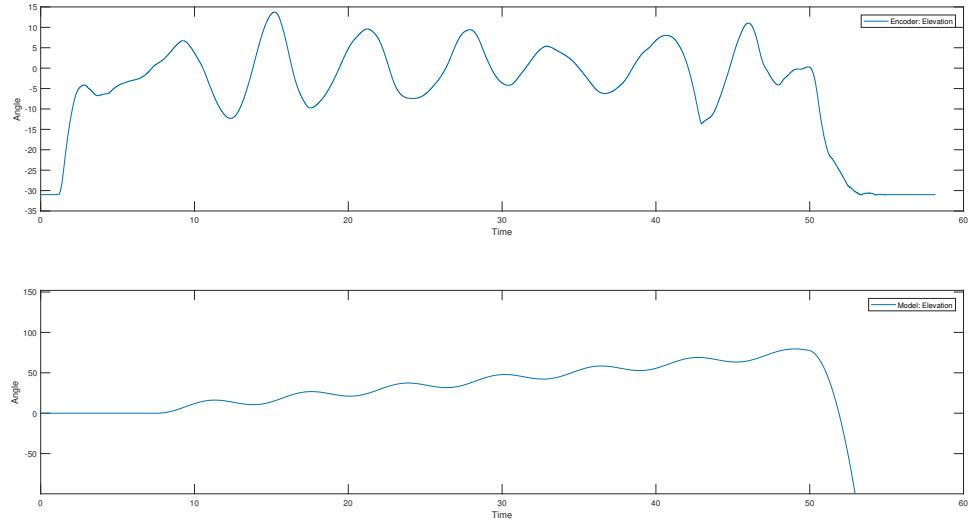


Figure 4: Elevation angle from encoder and model for a sine wave input

Figure 4 shows how the linearized model compares to the encoder values for elevation. As the initial elevation angle of the helicopter is far away from equilibrium, the original Simulink block diagram in fig. 22 cannot be used

to test correctness of elevation, as the subtraction of $V_{s,0}$ from an input of zero from the joystick will instantaneously be integrated to a large error. An option is to initialize the helicopter in the horizontal position, but then the encoder value for elevation will have an offset of 31° . It is therefore necessary to delay the signal to the system until the helicopter is around the equilibrium point. A solution is shown in fig. 23, appended in appendix D. As the helicopter proved difficult to control with joystick near the equilibrium, a sine wave with amplitude 1 V around $V_{s,0}$ was used as input.

From fig. 4 the model can be seen to approximate the sine wave at small angles around equilibrium, but fails at larger angles. Another issue is how the model gives a phase shifted angle compared to the real elevation of the helicopter. This may be a result of an inaccurate representation for the inertia in the helicopter, as the encoder values lag behind the estimated values, indicating that there is more inertia in the helicopter than modeled.

Summary

As seen from the previous sections, the linearized model for the equations of motion perform poorly outside the defined linearization point. Using an open-loop estimate for pitch, elevation and travel, as implemented in appendix D, fig. 22, is therefore not wise when regulating a highly dynamic system such as the helicopter. Open-loop regulators will in general require a highly precise model of the system in order to function properly, and as the model proposed in the previous sections greatly simplifies the dynamics of the helicopter, an open-loop estimate will function poorly. Some inaccuracies in the model include neglecting friction in joints, inaccurate physical parameters for the helicopter and neglecting to model the flow of air around the propellers in the protective grids. Perhaps the biggest inaccuracy when verifying the estimate for pitch, elevation and travel is that the model is only true when the propellers apply a force, which is not always the case when using the joystick.

2 Mono-variable control

The elevation controller provided in the Simulink file was connected to the system, setting the reference e_c to zero and adding $V_{s,0}$ to the controller output to extract V_s . The block diagram for the system is found in appendix D, fig. 25.

2.1 PD Controller

The pitch of the helicopter will now be controlled using a PD controller on the form,

$$V_d = K_{pp}(p_c - p) - K_{pd}\dot{p} \quad (8)$$

Inserting eq. (8) into eq. (5a) and applying the Laplace transform gives the transfer function between the pitch p and desired reference p_c ,

$$\frac{p}{p_c}(s) = \frac{K_1 K_{pp}}{s^2 + s K_1 K_{pd} + K_1 K_{pp}} \quad (9)$$

A detailed derivation of eq. (9) is found in appendix A from eq. (38). Note that the terms $p(0)$ and $\dot{p}(0)$ disappear during the Laplace transform as the pitch of the helicopter is stationary at initialization. A figure of the implemented pitch controller is appended in appendix D, fig. 26.

2.2 Pole Placement

To find the poles, eq. (9) can be written as,

$$\frac{p}{p_c}(s) = \frac{K_1 K_{pp}}{(s - \lambda_1)(s - \lambda_2)} \quad (10)$$

where λ_1 and λ_2 are the poles of the system. These are found by setting the numerator in eq. (9) to zero and solving for s , giving,

$$\lambda_1, \lambda_2 = \frac{-K_1 K_{pd} \pm \sqrt{(K_1 K_{pd})^2 - 4 K_1 K_{pp}}}{2} \quad (11)$$

The poles of the system can then be placed by K_{pp} and K_{pd} . To reason on the placement of these poles, notice that the time-domain solution to a state-space model includes the term,

$$e^{At} = Q e^{\Lambda t} Q^{-1},$$

where the system matrix A is diagonalized in terms of its eigenvalues Λ and eigenvectors Q . The eigenvalues of A are the poles in system, and as the time-domain solution includes an exponential raised to these poles, they must be placed in locations such that the output of the system converges with

time. Following this reasoning, negative real poles will give an exponentially decaying response, positive real poles will give an exponentially increasing response, and complex poles will give a sinusoidal response, as a complex exponential may be written in terms of cosines and sines,

$$e^{j\omega t} = \cos(\omega t) + j \sin(\omega t)$$

The values for K_{pp} and K_{pd} are chosen to check how the system behaves to these theoretically different scenarios. Both gains are normally assumed positive in the following sections.

Real negative overlapping poles

This configuration is achieved by removing the square root in eq. (11) completely, which occurs when,

$$K_{pp} = \frac{K_1 K_{pd}^2}{4}$$

Setting an arbitrary positive value for K_{pd} will now result in real negative overlapping poles.

Real negative non-overlapping poles

The constraint is now,

$$K_{pd}^2 > \frac{4K_{pp}}{K_1},$$

so the square root returns a real number. Setting K_{pd} higher than K_{pp} will result in this configuration.

Complex conjugate poles with negative real component

The constraint is now,

$$K_{pd}^2 < \frac{4K_{pp}}{K_1},$$

so the square root returns a complex number. Setting K_{pp} higher than K_{pd}^2 will result in this configuration.

Complex conjugate poles with positive real component

This configuration has the same constraint as the previous, but K_{pd} is now set to a negative value. This will move the poles over to the right half-plane in the s-plane, resulting in instability.

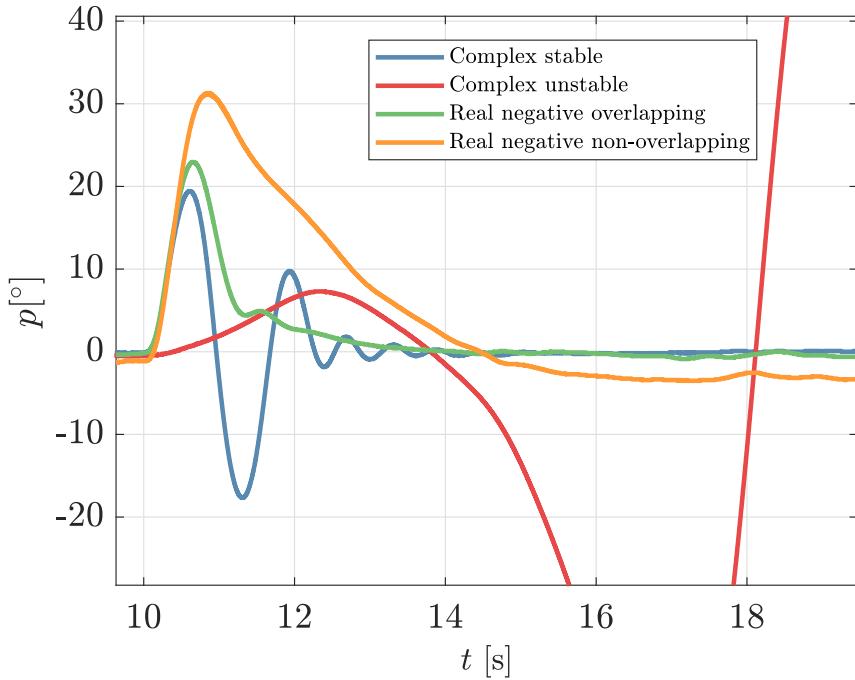


Figure 5: Experimental results of pitch angle for different pole configurations

Results

The experiments were conducted using the block diagram in fig. 25 in appendix D, simulating an impulse response at $t = 10$ s. This was done to simulate the same input for each experiment, resulting in responses that were comparable to each other. The result of the scenarios discussed above are depicted in fig. 5. The impulse responses of the respective configurations have different amplitudes due to the difference in distance between the poles and the origin in the s-plane, but the correspondence between the experiments and theory is clearly visible.

The real negative overlapping poles should in theory result in a critically damped system, but will due to inaccuracies in the model and model parameters result in a slightly overdamped system, as seen from the green graph. For the poles to be perfectly overlapping, the constant K_1 must be perfectly modeled, which is most likely not the case. The real poles of the system may therefore not necessarily be overlapping, and the system may thus not be critically damped.

The rest of the poles give expected responses:

- The real negative non-overlapping poles as seen in the yellow graph result in a slowly exponentially decaying response; an overdamped system.

- The complex conjugate poles with negative real component, the blue graph, gives an exponentially decaying sinusoidal response; an under-damped system.
- The complex conjugate poles with positive real component, illustrated by the red graph, results in an exponentially increasing sinusoidal response; an unstable system.

Another possible scenario is positive real poles, which give an exponentially increasing response. This system is clearly unstable, and was difficult to gather reasonable data for.

Choosing the negative overlapping poles as the best placement, K_{pp} and K_{pd} are found to be,

$$K_{pp} = 7.0925, \quad K_{pd} = 7.0925$$

for the poles $\lambda_1 = \lambda_2 = -2$. This may not give the best possible performance in practice, as the model isn't perfect, but is a theoretically sound choice.

2.3 Harmonic Oscillator

The system can be viewed as a harmonic oscillator of the form

$$G(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \quad (12)$$

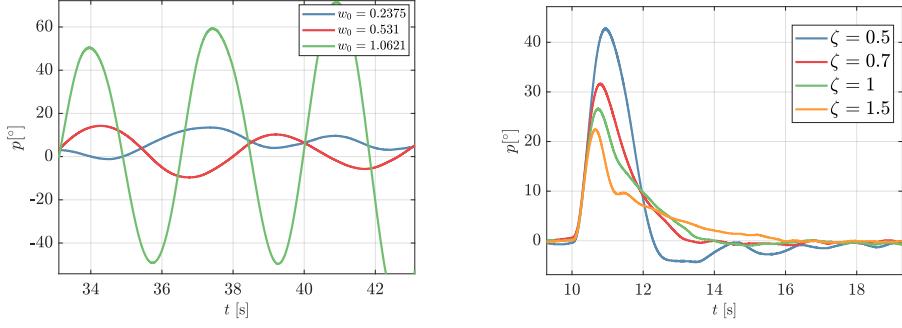
where ω_0 is the undamped frequency of the system and ζ is the damping ratio. If we have a system without damping, i.e $\zeta = 0$, the time domain response should in theory become a pure sine wave with frequency ω_0 , as eq. (12) results in the laplace transform of a sinusoid. If we then increase ω_0 , the sine wave should oscillate faster. On the other hand, keeping ω_0 constant and increasing ζ should make the sinusoid decay exponentially, until we are left with a pure exponential.

Comparing eq. (12) with the transfer function in eq. (9), it can be seen that $\omega_0 = \sqrt{K_1 K_{pp}}$, which gives the following expression for the damping ratio,

$$\zeta = \frac{1}{2} \frac{K_1 K_{pd}}{\sqrt{K_1 K_{pp}}} \quad (13)$$

Results

Several experiments were conducted using different values for ω_0 and ζ . The first three trials removed the damping from the system, setting ζ to zero



(a) Impulse response for different ω_0 with $\zeta = 0$ (b) Impulse response for different ζ with $\omega_0 = 1.5$

Figure 6: Impulse response for different ω_0 and ζ for a harmonic oscillator

while ω_0 was adjusted, which in theory should result in a marginally stable system with angular frequency ω_0 . In terms of poles, the transfer function in eq. (12), with $\zeta = 0$, can be seen to have poles laying on the imaginary axis of the s-plane, known to give a marginally stable system. The response can be seen in fig. 6a, where three different values for ω_0 were tested. As expected, the system has no damping, and an increase in ω_0 gives a sinusoid with higher frequency. This also means that higher values for ω_0 gives faster responses, which is desirable as long as the system is stable. However, this will in practice result in the helicopter overshooting its reference pitch, as there is some delay in the system. The undamped frequency ω_0 is therefore a tuning parameter.

The impulse response for various ζ can be seen in fig. 6b, where $\omega_0 = 1.5$ for all trials. As expected, an increase in damping ratio results in increase of system damping. The optimal value for ζ should in theory be $\zeta = 1$, as this results in a critically damped system, as seen from eq. (12) which results in overlapping poles. In practice, however, it can be seen from fig. 6b that $\zeta = 0.7$ gives a faster response. This is most likely due to inaccuracies in the model, specifically K_1 as it is the only model parameter present in eq. (13). K_1 is again a function of the motor constant K_f and moment of inertia J_p , which are both likely to be somewhat inaccurate.

3 Multi-variable control

3.1 State-space formulation

The equations for pitch and elevation in eq. (5) can be put into a state-space formulation of the form,

$$\dot{x} = Ax + Bu$$

such that,

$$\dot{x} = \begin{bmatrix} \dot{p} \\ \ddot{p} \\ \ddot{e} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ V_d \end{bmatrix} \quad (14)$$

In general, a system is controllable if its controllability matrix \mathcal{C} has full rank, where,

$$\mathcal{C} = [B \ AB \ A^2B \ \dots \ A^{n-1}B]$$

with n states. We have that,

$$AB = \begin{bmatrix} 0 & K_1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad A^2B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

so the controllability matrix for the system in eq. (14) is,

$$\mathcal{C} = \begin{bmatrix} 0 & 0 & 0 & K_1 & 0 & 0 \\ 0 & K_1 & 0 & 0 & 0 & 0 \\ K_2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (15)$$

As the rows in eq. (15) are linearly independent, the rank of the matrix is $\text{rank}(\mathcal{C}) = 3$, so the matrix has full rank and the system is controllable. This means that any state can be reached by the system within a finite amount of time by some input $u(t)$.

3.2 Linear Quadratic Regulators

A linear quadratic regulator (LQR) minimizes the cost function,

$$J = \int_0^\infty x^T(t)Qx(t) + u^T(t)Ru(t)dt \quad (16)$$

where $x(t)$ is the system state, Q and R are weighting matrices for state and input respectively, and $u(t)$ is the input to the system that minimizes eq. (16). Determining Q and R is done by manual tuning, which is covered at a later point.

A controller with state-feedback and reference feed-forward is desired,

$$u = Fr_c - Kx \quad (17)$$

where F and K are gain matrices for the desired reference r_c and state x respectively. The LQR is designed to regulate pitch and elevation rate, so the reference r_c is chosen as $r_c = [p_c \dot{e}_c]^T$.

Determining K is a matter of finding the optimal solution to the optimization problem in eq. (16), which can be found by solving for P in the algebraic Riccati equation,

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (18)$$

and setting $K = R^{-1}B^T P$. This is done in MATLAB with the function `lqr(A,B,Q,R)`.

The reference gain F is chosen such that $\lim_{t \rightarrow \infty} p(t) = p_c$ and $\lim_{t \rightarrow \infty} \dot{e}(t) = \dot{e}_c$, meaning the states converge to the desired reference. As shown in appendix A, eq. (39), F is found to be,

$$F = (C(BK - A)^{-1}B)^{-1} \quad (19)$$

where C is the output matrix in $y = Cx$. Other values for F will make the system converge to other values than the reference, which is undesirable. An example is given in appendix C, fig. 20, where F is multiplied by a scalar value a and the step response for pitch angle is observed, with step size given as $p_c = 17.2^\circ$. As seen from fig. 20, doubling the value of F will double the reference set to the regulator.

A block diagram of the regulator is found in appendix D, where the system is depicted in fig. 27 and the LQR is depicted in fig. 28.

Tuning Q and R

The performance of the LQR is mainly determined by how well the weighting matrices Q and R are tuned to the desired performance of the system. In the cost function eq. (16), Q penalizes deviations between the current state x and the reference r_c . In the same manner, R penalizes the input vector u . Increasing Q will therefore lead to more aggressive control, as deviations from the reference are penalized more heavily. On the other hand, increasing R will lead to more conservative use of the actuators, as the application of input is penalized.

Defining Q and R by,

$$Q = \begin{bmatrix} q_1 & 0 & 0 \\ 0 & q_2 & 0 \\ 0 & 0 & q_3 \end{bmatrix}, \quad R = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix}$$

the performance of the helicopter can be tuned by adjusting the values in the diagonal of Q and R . The helicopter is desired to be fast and accurate,

and must be tuned accordingly. As both Q and R affect the performance similarly, R is kept constant at,

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and will thus not limit the input u . Q is then adjusted to achieve optimal behaviour.

As pitch and elevation are decoupled states, their respective weights can be tuned independently. Pitch is tuned first, then elevation.

The block diagram in appendix D, fig. 29, depicts the system for tuning pitch. A step function for pitch angle p is used to simulate the step response, so the system performance for each trial is comparable. To minimize the effect travel may have on pitch, the helicopter was physically restrained to a constant travel angle. The weights for pitch rate \dot{p} and elevation rate \dot{e} , q_2 and q_3 respectively, were held constant during the tuning of pitch angle with weight q_1 . The step response for three different values of q_1 can be seen in fig. 7.

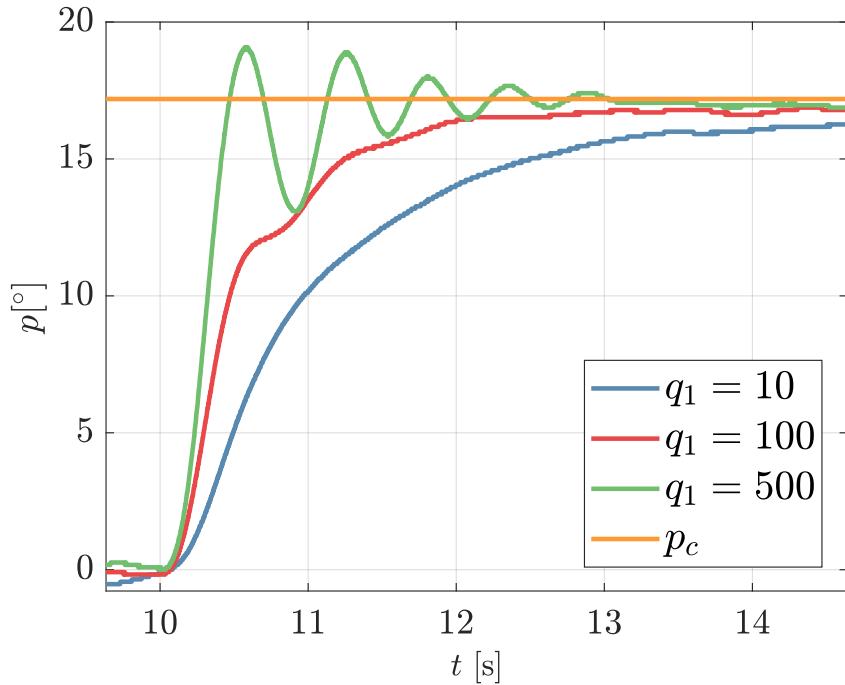


Figure 7: Step response when tuning q_1 for pitch angle with reference $p_c = 17.2^\circ$

The effect of the tuning parameter can clearly be seen from the behaviour of the helicopter in fig. 7. Too low values for q_1 will not penalize the deviation from the reference enough, and allows the controller more time to reach the reference. This results in an over-damped system, as illustrated by $q_1 = 10$.

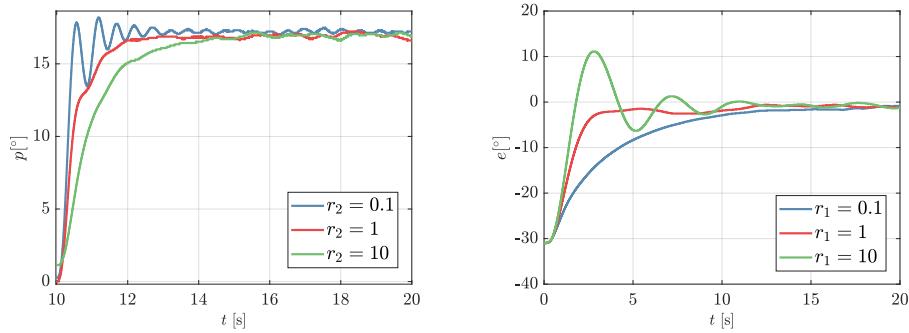
Too high values for q_1 will penalize the deviation too much, and the controller sets the input u too high, resulting in the underdamped system that occurs when $q_1 = 500$. By looking at the step response for each iteration of tuning, the value of q_1 for the critically damped system was found to be $q_1 = 100$. This value for q_1 is further used when tuning pitch rate and elevation rate. Worth noting from fig. 7 is the steady-state error from the reference, which will be improved upon later with the introduction of integral effect.

Pitch rate and elevation rate were tuned in the same manner; by looking at the step response and iterating until the critically damped response was achieved. This resulted in the configuration,

$$Q = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 40 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

giving the best performance in terms of accuracy and speed.

To illustrate the effect of R , which has been held constant previously, the configuration for Q is now kept constant while r_1 and r_2 vary. As r_1 penalizes \tilde{V}_s , which elevation is a function of, it is expected that high values will limit the elevation rate of the helicopter, giving an overdamped system. Similarly, low values will give an underdamped system. The same goes for r_2 , which affects the pitch. The result is displayed in fig. 8, where it can be seen that the behaviour is as expected from theory.



(a) Step response when tuning r_2 for pitch angle with reference $p_c = 17.2^\circ$ (b) Step response when tuning r_1 for elevation rate with reference $\dot{e}_c = 0 \text{ rad/s}$

Figure 8: Step response when tuning R

3.3 Integral action

As the previously described models for the helicopter are not perfect, integral effect needs to be introduced to remove steady-state error. This is done by integrating the deviation between the reference and measured output,

$$x_a = \int_0^t r(\tau) - Cx(\tau) d\tau$$

where x_a are augmented states in the state-space model. The differentiation of the augmented states are defined as $\dot{\gamma} = p - p_c$ and $\dot{\zeta} = \dot{e} - \dot{e}_c$, such that the state-space formulation in eq. (14) is redefined as,

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \\ \dot{e} \\ \gamma \\ \zeta \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ V_d \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} p_c \\ \dot{e}_c \end{bmatrix} \\ &= \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_a \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ -I \end{bmatrix} r \\ &= \bar{A}\bar{x} + \bar{B}u + \begin{bmatrix} 0 \\ -I \end{bmatrix} r \end{aligned} \quad (20)$$

where \bar{A} and \bar{B} are the augmentations of A and B , and $x_a = [\gamma \quad \zeta]^T$ are the additional states. Similarly, Q and R are augmented as,

$$\bar{Q} = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 40 & 0 & 0 \\ 0 & 0 & 0 & q_4 & 0 \\ 0 & 0 & 0 & 0 & q_5 \end{bmatrix}, \quad \bar{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

where q_4 and q_5 penalize the new states γ and ζ respectively. Notice that $\bar{R} = R$, as the augmentation of the state-space model does not affect the input u .

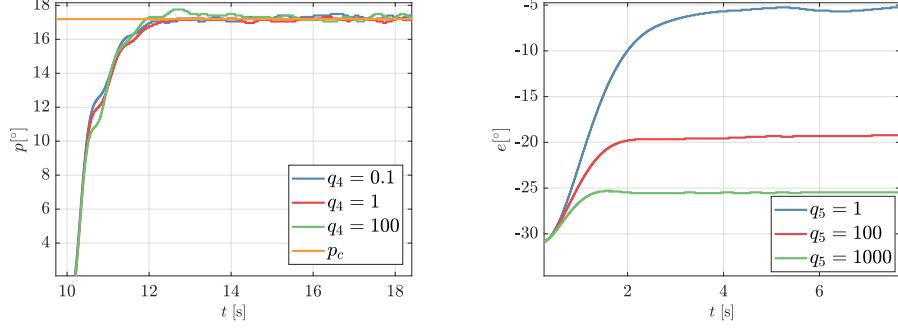
\bar{K} can now be found using the same approach as for K , but with the augmented matrices. Writing $\bar{K} = [K_1 \ K_2]$, the optimal augmented reference gain matrix \bar{F} is found to be,

$$\bar{F} = (C(BK_1 - A)^{-1}B)^{-1} = F$$

with a derivation appended in appendix A, eq. (40). From the derivation it can be seen that the output goes towards the reference ($y = Cx = r$) when $t \rightarrow \infty$ no matter what \bar{F} is, but may do so in an inordinate amount of time if \bar{F} is not chosen correctly. The effect of this can be seen in appendix C, fig. 21, were \bar{F} is multiplied by some scalar a . By e.g setting the reference-matrix to a high value, the controller will overshoot the actual reference and the integral effect will slowly remove the deviation from the reference.

The remaining part is finding q_4 and q_5 , which is done in a similar manner as when finding q_1 , q_2 and q_3 . Block diagrams for the new system are

appended in appendix D, fig. 30 and fig. 31. A step response was used for tuning the integral effect on pitch and elevation rate, which can be seen from fig. 9.



(a) Step response when tuning q_4 for pitch angle with reference $p_c = 17.2^\circ$ (b) Step response when tuning q_5 for elevation rate with reference $\dot{e}_c = 0 \text{ rad/s}$

Figure 9: Step response when tuning integral effect of Q

As seen from fig. 9a, the different weights on q_4 do not change the response significantly. Values over and around $q_4 = 100$ tend to give some overshoot, while values close to $q_4 = 1$ gives a near critically damped system. Notice that the steady-state error from fig. 7 is removed in fig. 9a, a result of the integral effect. In fig. 9b, q_5 was tuned by looking at the step response for elevation angle and checking the time until $\dot{e}_c = 0$ was achieved, which happens when the elevation angle stays constant. High values for q_5 tend to give a fast response, but overshoots the reference. In addition, the motors were observed to apply a frequently changing force, which may not be mechanically or electrically stable. Most noticeable is that the controller reaches its reference in elevation rate much faster, compared to fig. 8b. In fig. 8b, the reference is only reached at the linearization point, and has a steady-state error elsewhere. A possible reason for this is that the controller is optimized for linear behavior, and it is therefore not possible to maintain zero steady-state error outside this region with an imperfect model.

The resulting configuration after tuning is,

$$\bar{Q} = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 40 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 100 \end{bmatrix}, \quad \bar{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

4 Intertial Measurement Unit

So far, encoders have been used to measure the states used for feedback in the regulators. These give very accurate measurements, and will be regarded as ground truth for the states of the helicopter. On an actual helicopter, which is not mounted to fixed rotating arms, the use of encoders is not possible. An inertial measurement unit (IMU) is often used to measure states in these cases.

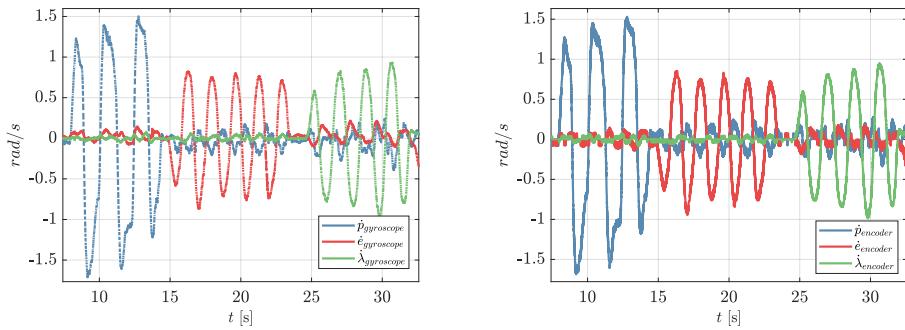
4.1 IMU characteristics

An IMU measures accelerations and rotational velocities in three dimensions, often modeled by,

$$\tilde{a}(t) = a(t) + b_a(t) + n_a(t) \quad (21a)$$

$$\tilde{\omega}(t) = \omega(t) + b_\omega(t) + n_\omega(t) \quad (21b)$$

where \tilde{a} and $\tilde{\omega}$ are the measured accelerations and rotational velocities, a and ω are the actual underlying accelerations and rotational velocities, b_a and b_ω are slowly varying sensor bias, and n_a and n_ω are additive white noise. Looking at this model, a big difference from using encoders is that the true state is hidden in noise, making use of direct measurements from the IMU unfavorable for state feedback. Estimation of the true states are therefore necessary, which will be discussed later.

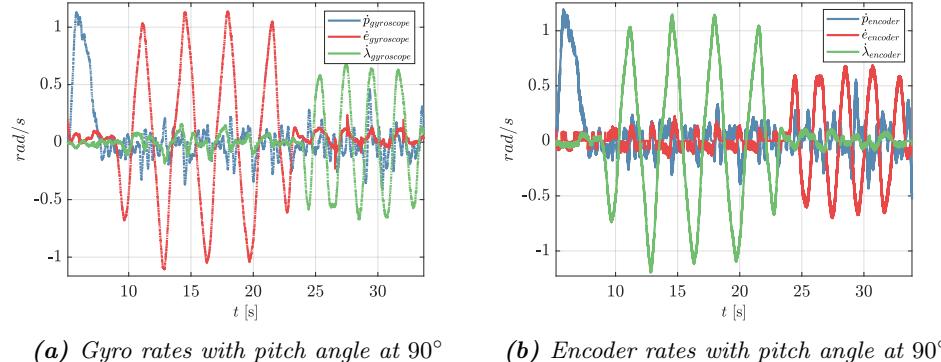


(a) Gyro rates when moving about one axis at a time (b) Encoder rates when moving about one axis at a time

Figure 10: Comparison of gyro rates from the IMU and encoder rates from the encoders when moving about one axis at a time

As seen from fig. 10, the encoder rates can apparently be easily replaced by the IMU's gyroscope. The IMU samples at a slightly slower rate than the encoders, as seen from the gaps in fig. 10a, but performs just as well otherwise. This may be a bit misleading, as the strength of the encoders are

in the angles, and not angle rates, which are obtained by differentiation on the angles. If the encoder angles were to be compared with the estimated angles from the gyroscope rates, the result would most likely show that the encoders perform better than the IMU.



(a) Gyro rates with pitch angle at 90° (b) Encoder rates with pitch angle at 90°

Figure 11: Comparison of gyro rates from the IMU and encoder rates from the encoders when moving about elevation and travel axis with pitch angle at 90°

A problem with using the IMU data directly can be seen in fig. 11, where the pitch of the helicopter is set to 90° before moving about the other axis. When moving the helicopter about the travel axis in a sinusoidal manner, seen from fig. 11b, the gyroscope measures angular velocity about the elevation axis instead, seen from fig. 11a. Likewise, when moving the helicopter about the elevation axis, the gyroscope measures angular velocity about the travel axis. The reason for this is that the IMU takes measurements in its local coordinate frame, seen from the orange coordinates in fig. 1a. When e.g. pitching the helicopter, the whole coordinate frame is rotated about the pitch axis, making a movement about an axis in the global frame translate to movement about other axes in the local frame. This must be corrected for in some way. An exception is of course the scenario in fig. 10, where the global frame lines up with the IMU's local frame, and the helicopter is moved about one axis at a time.

The accelerometer on the IMU measures accelerations in the three directions \vec{x} , \vec{y} and \vec{z} , defined by fig. 1a. The gravitational constant g is a reference for measuring accelerations in the different directions, and is measured positive in the upwards direction in the frame of the accelerometer. As with the gyroscope, the accelerations in the different directions are decomposed when the helicopter is at an angle with respect to equilibrium. This can be seen from fig. 12, where the helicopter starts at standstill on the table and is moved only about the elevation axis to equilibrium. When at standstill, with the gravitational constant as the only acceleration, one should expect the accelerometer to measure acceleration in \vec{z} only, but as the helicopter is tilted about 30° the gravitation is decomposed in both \vec{x}

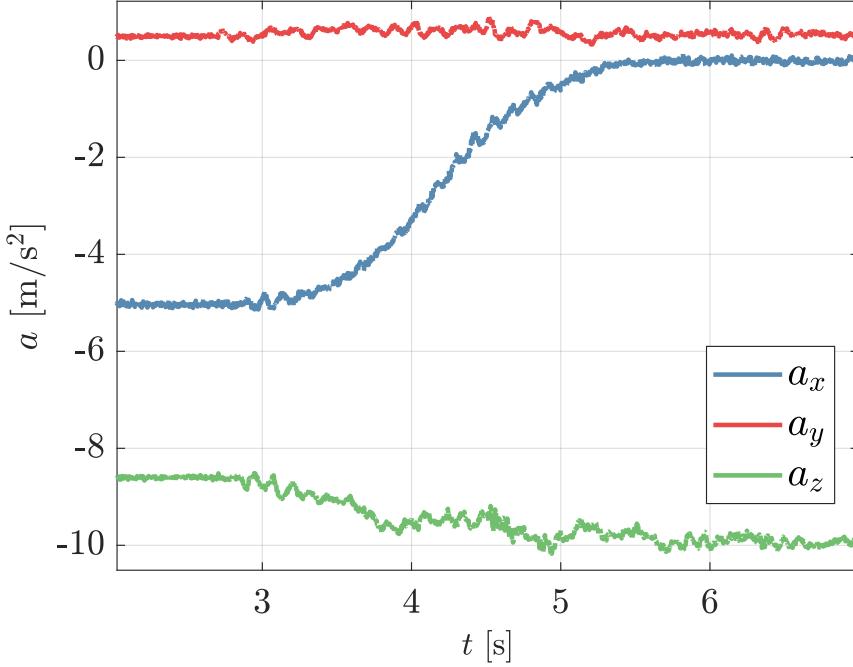


Figure 12: Accelerometer measurements when moving helicopter from table to equilibrium

and \vec{z} . When moved to equilibrium, a_x can be seen to decrease in absolute value towards zero, while a_z increases in absolute value towards $g \approx 9.81$. The negative sign on a_x and a_z is a consequence of the chosen coordinate frame in fig. 1a, as the accelerometer measures gravitation in the upwards direction. The slight off-set on the measurements at equilibrium may be due to the helicopter having an off-set from equilibrium, the IMU not being calibrated correctly or the effect of the bias terms in eq. (21).

4.2 Gyroscope transform

To correct for the gyroscope measurements being in the local coordinate frame of the IMU, a transform is applied to the measurements so the rotational velocities always are referenced to the global frame. This is done by multiplying the gyroscope vector with a rotation matrix,

$$R_{p,e,\lambda} = \begin{bmatrix} 1 & \sin(p)\tan(e) & \cos(p)\tan(e) \\ 0 & \cos(p) & -\sin(p) \\ 0 & \frac{\sin(p)}{\cos(e)} & \frac{\cos(p)}{\cos(e)} \end{bmatrix} \quad (22)$$

where p and e are the angles measured by the encoders. Note that this transform relies on the encoders, which in practice should be unavailable

when using an IMU. In the case where the use of encoders is not possible, these angles must be estimated by e.g values obtained with the IMU.

4.3 Observability part I

The transformed gyroscope measurements are to be inserted in the state-space model, now defined by,

$$\dot{x} = \begin{bmatrix} \dot{p} \\ \ddot{p} \\ \dot{e} \\ \ddot{e} \\ \dot{\lambda} \\ \ddot{\lambda} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ K_3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \\ e \\ \dot{e} \\ \lambda \\ \dot{\lambda} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ 0 & 0 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ V_d \end{bmatrix} \quad (23a)$$

$$y_{gyro} = \begin{bmatrix} \dot{p} \\ \dot{e} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \\ e \\ \dot{e} \\ \lambda \\ \dot{\lambda} \end{bmatrix} \quad (23b)$$

Note that the state vector x has double the dimension of the measurement vector y_{gyro} . Intuitively, this should be problematic for any controller that operates on the states in the state vector, as only three of the states are directly observable by measurements. Using these measurements as feedback may therefore not be sufficient for the chosen controller. The only way all states can be observed is if the measured states are functions of the other states, so the other states can be observed indirectly through the measured states. This is only the case for p , as it is a function of $\dot{\lambda}$. The observable subset of x is therefore p , \dot{p} , \dot{e} and $\dot{\lambda}$.

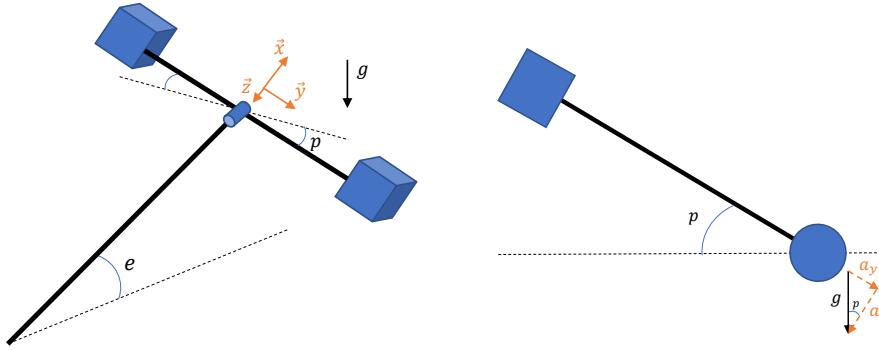
The same result can be found through the system observability matrix,

$$\mathcal{O} = \begin{bmatrix} p & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & K_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \\ \dot{p} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \\ e & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \\ \dot{e} & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \\ \lambda & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \\ \dot{\lambda} & \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \end{bmatrix}$$

where the rows in the transposed matrix corresponding to e and λ are zeros. This means the matrix has linearly dependent rows, and the observability matrix will therefore not have full rank, with e and λ as unobserved states.

4.4 Accelerometer

To increase the observability of the system, the accelerometer measurements are also included in the state space model. The accelerations are not used directly as measurement feedback, but provide information about the angle of rotation about the different axes. As mentioned previously, the acceleration measured at a stationary position is the gravitational constant decomposed in the directions \vec{x} , \vec{y} and \vec{z} , with magnitude depending on the rotation about the different axes. This can more easily be seen in fig. 13a.



(a) Rotation of the IMU's coordinate frame when rotating (b) Decomposition of g into a_y and a_z when pitching

Figure 13: IMU frame and accelerations when rotating helicopter

The acceleration in the different directions are from fig. 13a found to be,

$$a_x = g \sin(e) \quad (24a)$$

$$a_y = g \sin(p) \quad (24b)$$

$$a_z = g \cos(e) \cos(p) \quad (24c)$$

Solving for g , eq. (24) may be re-written as,

$$g = \frac{a_x}{\sin(e)} = \frac{a_y}{\sin(p)} = \frac{a_z}{\cos(e)\cos(p)}$$

The angles p and e may now be extracted from the accelerations. The pitch angle is found to be,

$$\begin{aligned} \frac{a_y}{\sin(p)} &= \frac{a_z}{\cos(e)\cos(p)}, \quad \cos(e) \approx 1 \\ p &= \arctan\left(\frac{a_y}{a_z}\right) \end{aligned} \quad (25)$$

using the small angle approximation for elevation angle. This is a reasonable approximation, as the helicopter is not expected to achieve large elevation angles with the setup in the lab. Additionally, all the controllers designed previously assume that this approximation holds. As these controllers are to be used with measurement feedback, this assumption will not add any more inaccuracy than is already present in the model, although the inaccuracy is now also present in the measurement model.

Solving for elevation angle is done in a similar manner, but can be done without assuming $\cos(p) \approx 0$. Using fig. 13b, the cosine of pitch can be seen to be $\cos(p) = a_z/g$, with $g = \sqrt{a_y^2 + a_z^2}$ by the Pythagorean theorem. The elevation angle may now be found as,

$$\begin{aligned} \frac{a_x}{\sin(e)} &= \frac{a_z}{\cos(e)\cos(p)} \\ e &= \arctan\left(\frac{a_x}{a_z} \cdot \cos(p)\right) \\ e &= \arctan\left(\frac{a_x}{a_z} \cdot \frac{a_z}{g}\right) \\ e &= \arctan\left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}\right) \end{aligned} \quad (26)$$

4.5 Observability part II

Using measurements from the accelerometer as well as gyro, the measurement vector may be written as,

$$y_{IMU} = \begin{bmatrix} \dot{p} \\ \dot{e} \\ \dot{\lambda} \\ p \\ e \end{bmatrix}$$

Following the same reasoning as before, the observable states are now p , \dot{p} , e , \dot{e} and $\dot{\lambda}$, with p and e observed through the accelerometer, using eq. (25) and eq. (26). Using the observability matrix, here omitted due to the size, the same conclusion can be found.

4.6 Noise

The most notable difference between the measurements from the IMU and encoders is the additive noise. This can be seen from the previous plots, but is also shown in fig. 14 for two different cases, using the gyroscope as an example.

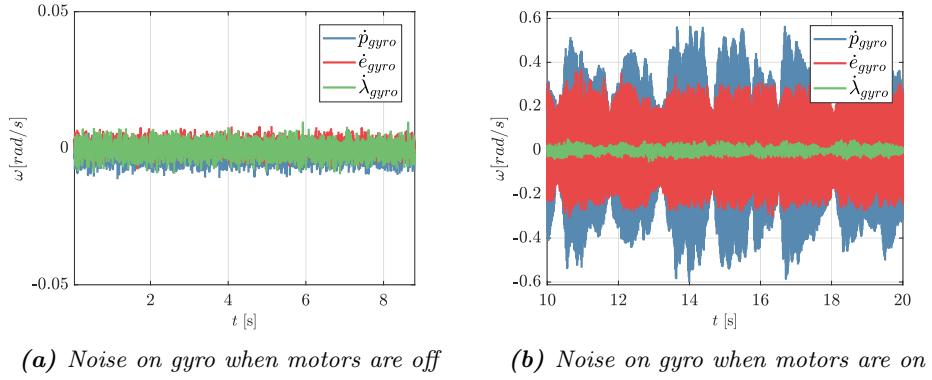


Figure 14: Comparison of noise on gyroscope measurements at equilibrium when motors are off/on

In fig. 14a, the helicopter is kept still on the table with motors turned off while measurements are taken from the IMU. The angular velocity in each direction should obviously be zero in this case, but the IMU detects non-zero angular velocities due to the noise.

In fig. 14b, the helicopter is kept in equilibrium by the LQR from Lab 3 while measurements are taken from the IMU. The angular velocity in each direction should in this case also be zero, but is heavily affected by the noise. Notice the difference in magnitude on the axis between fig. 14a and fig. 14b. The reason for the increased noise while the helicopter is flying is likely due to vibrations on the helicopter body and increased electromagnetic interference from the motors and cables when drawing higher currents.

The pitch rate in fig. 14b differs from the rest, which is a result of the helicopter varying its pitch in small, quick movements to regulate towards a reference of zero pitch angle. An adding factor is also that the LQR is tuned aggressively. The travel rate has lower noise than the rest due to the helicopter being held physically at a constant travel angle during the measurements, and is therefore not prone to the same vibrations as elevation and pitch.

The correctness of the model of the IMU in eq. (21) depends on the noise being white, where white noise is characterized by having zero mean and being uncorrelated between time-steps. The zero-mean property can be seen from fig. 14, but the noise being uncorrelated is harder to justify from the plots, although the amplitude of the noise is fairly uniform, indicating white noise. Another property of white noise is a constant flat power spectral density, which is examined in fig. 15 for the gyroscope noise while the helicopter is kept at the table. As seen from the figure, the power spectral density is relatively constant over all frequencies, indicating that white noise may be a reasonable model for the noise on the IMU.

The signal covariance matrix for each case, R_{ground} and R_{flying} for heli-

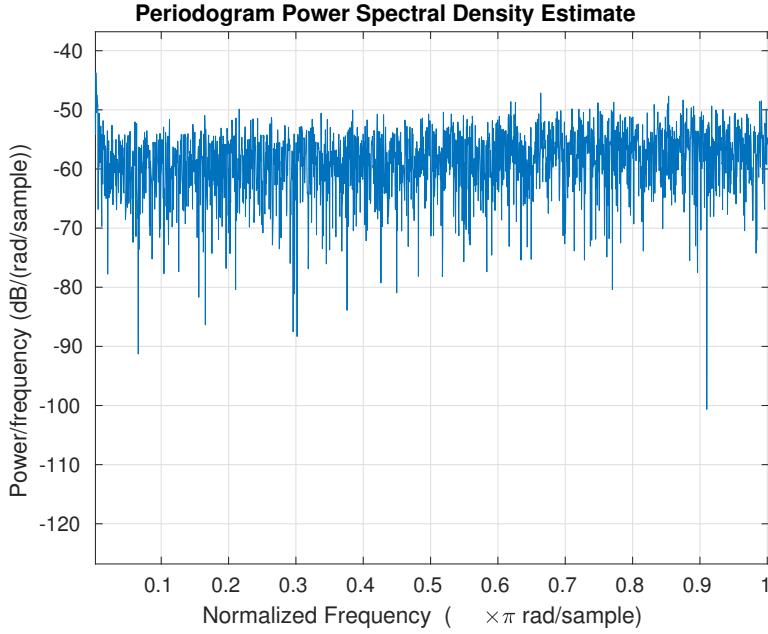


Figure 15: Power spectral density of gyroscope measurements

copter on ground and flying respectively, gives the variance and covariance of the states. As seen in eq. (27), the measurements from the IMU have a larger spread from the mean when the helicopter is flying than when laying still. This corresponds well with fig. 14, as larger noise is expected to give higher variance. Especially pitch has a high variance, which may be due to the same reason as for the varying pitch rate in fig. 14b.

$$R_{ground} = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & -0.0001 & 0.0000 \\ 0.0000 & 0.0000 & -0.0001 & 0.0029 & -0.0005 \\ 0.0000 & 0.0000 & 0.0000 & -0.0005 & 0.0001 \end{bmatrix} \quad (27a)$$

$$R_{flying} = \begin{bmatrix} 0.0028 & -0.0033 & 0.0001 & 0.0036 & -0.0057 \\ -0.0033 & 0.0112 & -0.0001 & -0.0092 & -0.0091 \\ 0.0001 & -0.0001 & 0.0001 & 0.0002 & -0.0002 \\ 0.0036 & -0.0092 & 0.0002 & 0.0106 & 0.0068 \\ -0.0057 & -0.0091 & -0.0002 & 0.0068 & 0.0708 \end{bmatrix} \quad (27b)$$

As previously mentioned, the biggest difference between the measurements from the IMU and encoders is the added noise. This noise will heavily affect any attempt to control the helicopter using direct measurements with e.g an LQR, as the input to the controller is a noisy signal that fluctuates rapidly. To illustrate this, LQR was used with IMU measurements as state

feedback, depicted in appendix D, fig. 32. The conversion of IMU measurements to state measurements is depicted in appendix D, fig. 33, with Matlab functions implementing eq. (25), eq. (26), given in appendix B.1, and transformation eq. (22), supplied in the task. As expected, the helicopter is near impossible to control when using measurements directly, so a method for filtering the measurements to retrieve estimates of the true states is desired.

5 The Kalman filter

As observed in the previous section, using the IMU measurements directly as feedback leads to insufficient control. This is made apparent by the model of the IMU in eq. (21), where the true state is hidden in noise, giving a noisy input to the controller. An estimate of the true state is therefore desired, and can be achieved by using a fitting observer.

Approaching the filtering problem from a Bayesian perspective, the posterior density $p(x_k|x_{1:k-1}, y_{1:k}, u_{1:k})$ is desired for each timestep k , modeled as a probability density function to account for the uncertainty in the model. Applying the Markov assumption on this model, the posterior density can be written as $p(x_k|x_{k-1}, u_k)$, giving the dependency structure in fig. 16.

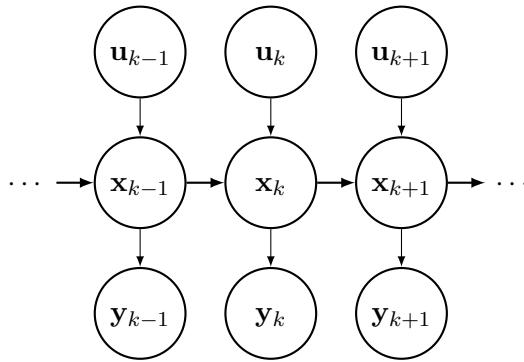


Figure 16: Markov chain with states hidden in measurements

This may seem like a drastic assumption to make, but notice that the discretized state-space model is on the form,

$$x_{k+1} = A_d x_k + B_d u_k + w_{d,k} \quad (28a)$$

$$y_k = C_d x_k + v_{d,k} \quad (28b)$$

$$w_d \sim \mathcal{N}(0, Q_d), \quad v_d \sim \mathcal{N}(0, R_d) \quad (28c)$$

giving the same recursive formulation as in fig. 16, with added process and measurement noise. Assuming that the noise is Gaussian and the model is linear, the optimal filter for estimating the states in a model described by fig. 16 can be shown to be the Kalman filter. The noise from the IMU is white Gaussian, as shown in the previous section, but the linear model for the helicopter is an approximation of a nonlinear model, and the Kalman filter will thus not be an optimal filter, though it should perform sufficiently. The Kalman filter consists of two steps: prediction and correction, where prediction will be handled first.

5.1 Discretization

As a Kalman filter implemented on a computer uses discrete time-steps, the state-space model needs to be discretized to eq. (28), given by,

$$A_d = e^{AT}, \quad B_d = \left(\int_{\tau=0}^T e^{A\tau} d\tau \right) B, \quad C_d = C$$

with $T = 0.002$ s. The discrete-time measurement noise covariance R_d is given by the measured covariance while flying from the previous task, R_{flying} , and the process noise covariance Q_d remains a tuning parameter, describing the uncertainty in the model.

$$A_d = \begin{bmatrix} 1 & 0.002 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.002 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.002 \\ 0.0012 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_d = \begin{bmatrix} 0 & 0 \\ 0 & 0.0011 \\ 0 & 0 \\ 0.00018 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$C_d = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

5.2 State prediction

The predicted state in the next time-step amounts to the expected state, so taking the expectation of eq. (28a) gives,

$$\bar{x}_{k+1} = A_d \bar{x}_k + B_d u_k \quad (29)$$

as the predicted state. If the model were perfect, this prediction would be sufficient for controlling the helicopter. Not only is the model imperfect, it is also a linear approximation, and control by predictions only will therefore not suffice. This prediction is in a sense equivalent to the model's estimates in section 1.3, and has similar results. A comparison is given in fig. 17, where a sine wave is given as reference on pitch angle. The helicopter is kept around equilibrium to ensure the linearized model is valid, but it does not follow the encoder values particularly well. The signal shape is seen to be somewhat similar, but the values are off by multiple degrees, and the predictions have a clear downwards trend. This means the Kalman filter most likely will have to rely on measurements for correcting the off-set in the predictions. The comparison in fig. 17 may be somewhat misleading, as there is some suspicion to the spike in fig. 17a at $t = 5$ s. The reason for this

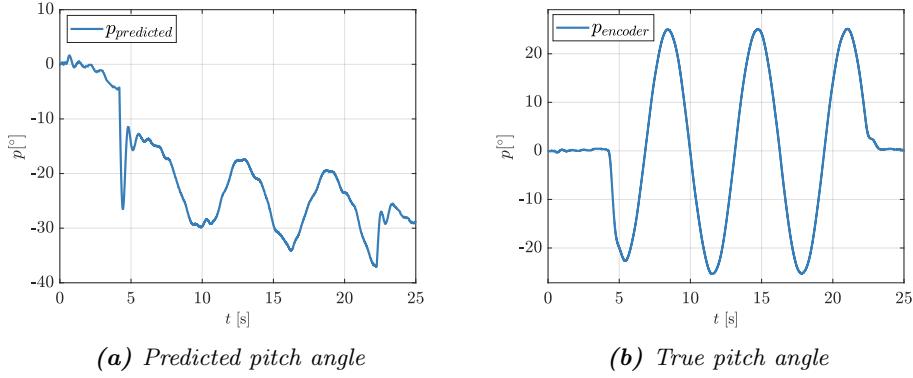


Figure 17: Comparison of predicted and true pitch angle with a sine wave input

is unknown, but as it shifts the prediction more than it should it affects the rest of the predictions. Another uncertainty is whether the linearized model is valid for the amplitude applied by the sine wave.

Using only the prediction step in the Kalman filter to control the helicopter will, as discussed, not work well. Around the linearization point it proved possible for some time, until the predictions drifted too much. Due to drift, a larger control input from the joystick was required over time to keep the helicopter stable, until it could not be controlled anymore. As mentioned, the model is imperfect and an approximation, and the helicopter proved impossible to control outside the linearization point. This open-loop estimator will therefore not work outside the linearization point, and within it will work poorly for some time.

5.3 Error covariance prediction

The error in the state prediction is given by,

$$\bar{\varepsilon}_k = x_k - \bar{x}_k$$

with covariance denoted by \bar{P}_k . As for the state prediction, a prediction of the covariance into the next time-step is desired, given by,

$$\begin{aligned} \bar{P}_{k+1} &= \text{Cov}(\bar{\varepsilon}_{k+1}) \\ &= \mathbb{E}[\bar{\varepsilon}_{k+1}\bar{\varepsilon}_{k+1}^T] \\ &= A_d\bar{P}_kA_d^T + Q_d \end{aligned} \quad (30)$$

with a derivation appended in appendix A, eq. (41).

It is now clear how the process noise Q_d affects the uncertainty in the predictions. Setting Q_d to higher values will set a higher uncertainty in the

predictions from the model, and will in the opposite case set lower uncertainty. The effect of Q_d on the estimates is not apparent at this point though, but Q_d will at a later point indirectly weight the model's predictions against new measurements by the Kalman gain when the estimates are updated, in effect telling how much the predictions should be trusted.

The diagonal in Q_d represents the variance of each state, which cannot be negative. \bar{P}_k will thus always increase for each time-step when only doing predictions according to eq. (29) and eq. (30). Tuning Q_d is therefore postponed until measurements are included in the correction step.

5.4 State correction

As seen previously, the uncertainty in the predictions will increase unbounded if they are not corrected for by measurements. When doing correction to obtain the state estimate, the filter must weigh the model's prediction against the new measurement based on the uncertainty of both. This weighting is incorporated in a weighting matrix K_k , which will be discussed later. The state estimate is then denoted,

$$\hat{x}_k = \bar{x}_k + K_k(y_k - \bar{y}_k) \quad (31)$$

where y_k is the measurement and $\bar{y}_k = E[y_k] = C_d\bar{x}_k$ is the prediction.

In the time-steps where measurements are not received, the best estimate is the prediction from the model,

$$\hat{x}_k = \bar{x}_k \quad (32)$$

5.5 Error covariance correction

The error in the state estimate is given by,

$$\hat{\varepsilon}_k = x_k - \hat{x}_k$$

with covariance denoted by \hat{P}_k . The covariance in the error determines the uncertainty in the state estimate, and can be written as,

$$\begin{aligned} \hat{P}_k &= \text{Cov}(\hat{\varepsilon}_k) \\ &= E[\hat{\varepsilon}_k \hat{\varepsilon}_k^T] \\ &= (I - K_k C_d) \bar{P}_k (I - K_k C_d)^T + K_k R_d K_k^T \end{aligned} \quad (33)$$

the derivation for which is included in appendix A, eq. (42). Again, in the time-steps where measurements are not received, the best estimate is a prediction,

$$\hat{P}_k = \bar{P}_k \quad (34)$$

5.6 Weighting matrix K

The last piece of the Kalman filter is finding the optimal weighting matrix, K_k , known as the Kalman gain. To obtain accurate state estimates, the Kalman gain should be chosen such that it minimizes the expected estimation error, $E[(x - \hat{x}_k)^T(x_k - \hat{x}_k)]$. This amounts to minimizing the trace of the estimated error covariance, $\text{tr}(\hat{P}_k)$, which will be done by differentiating with respect to K_k , using the following properties of the trace,

$$\frac{\partial \text{tr}(AB)}{\partial A} = B^T, \quad \frac{\partial \text{tr}(ACA^T)}{\partial A} = 2AC \quad (35a)$$

$$\text{tr}(D) = \text{tr}(D^T), \quad \text{tr}(A + B) = \text{tr}(A) + \text{tr}(B) \quad (35b)$$

The minimization results in a Kalman gain of,

$$K_k = \bar{P}_k C_d^T (C_d \bar{P}_k C_d^T + R_d)^{-1} \quad (36)$$

with a derivation appended in appendix A, eq. (43).

The Kalman filter can now be implemented using eq. (29), eq. (30), eq. (31), eq. (33) and eq. (36), using eq. (32) and eq. (34) as estimates in time-steps where measurements are not received. The final system is depicted in appendix D, fig. 34, with the Kalman filter implemented in fig. 35 and the LQR implemented in fig. 36. The Kalman filter is implemented as a MATLAB function, given in appendix B.2.

5.7 Tuning

As previously mentioned, the process noise Q_d is a tuning parameter that describes the uncertainty in the model's predictions, which with the uncertainty in the measurements, R_d , determine how the Kalman gain K_k weighs predictions against measurements for each time-step. R_d could just as well have been a tuning parameter if there was no reasonable way to retrieve representative data for the measurement noise, but R_{flying} should be representative.

To ease the initial tuning of Q_d , the input to the LQR in appendix D, fig. 34, is provided by the encoders, while the state estimates provided by the Kalman filter are compared to the true states provided by the encoders. The process noise can then be tuned until the estimates are relatively close to the true states. For the results from each iteration of Q_d to be comparable, the same data must be collected for each run. This means the helicopter must perform roughly the same maneuvers each time, which can be done by e.g. providing deterministic signals to the reference, otherwise provided by the

joystick. The provided reference signal should be representative for the expected maneuvers by joystick when tuning Q_d , so Q_d does not underestimate or overestimate the uncertainty in the maneuvers. For the initial tuning, a sinusoid with amplitude 20° was applied to the pitch reference, while keeping the elevation rate zero. This should account for the expected maneuvers by joystick in pitch, but may not be sufficient for elevation.

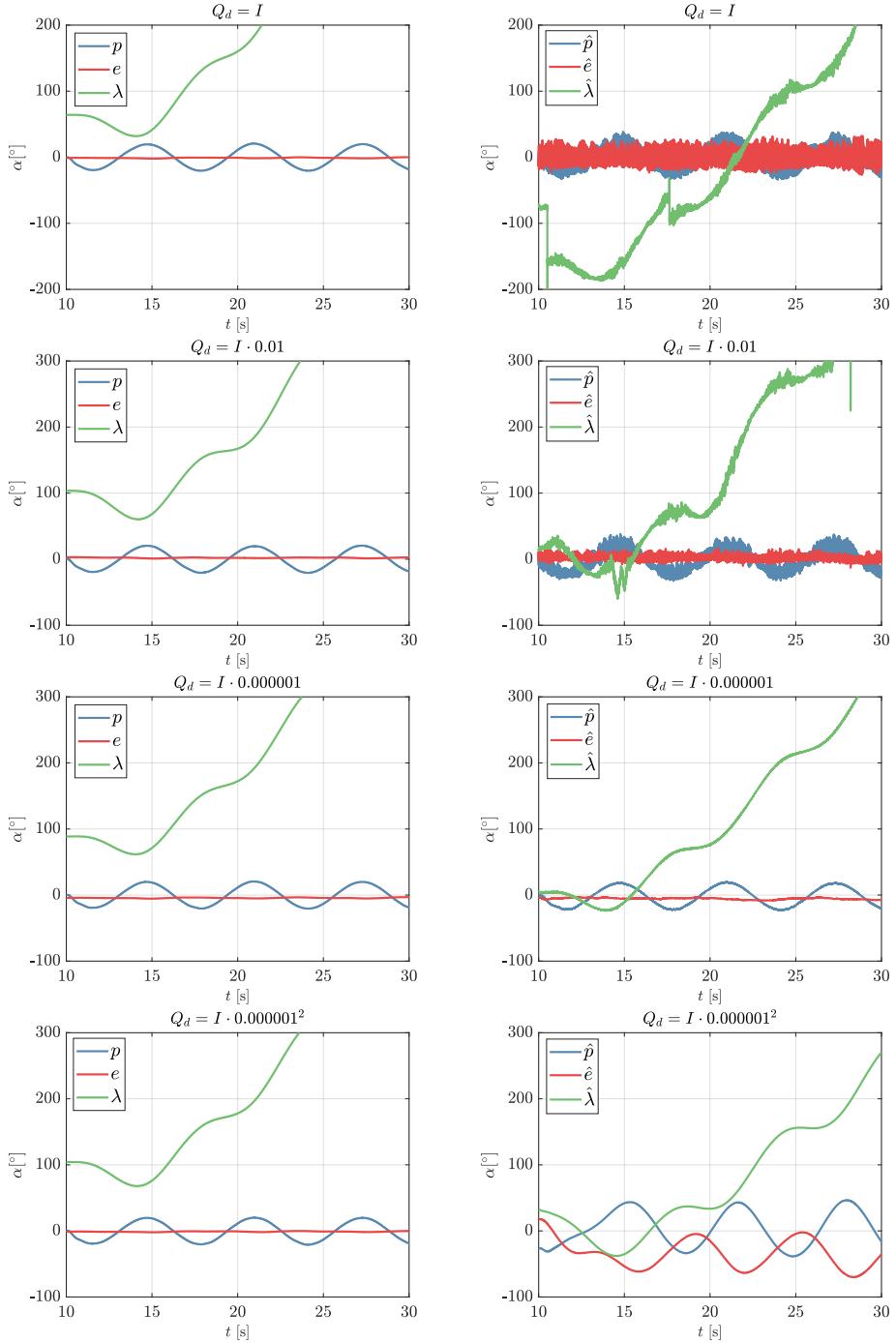
To evaluate the performance of the Kalman filter, some metric should be used to compare the true and estimated states. The root mean squared error (RMSE) could be such a metric, which needs to be minimized. Solely using the RMSE will however overfit Q_d to the specific maneuvers provided by the sinusoid, and may give a filter that performs poorly to other maneuvers. An experimental approach based on observing the behavior of the helicopter may be an equally sufficient metric, and gives more intuition towards how Q_d affects the filter. An example of four different values of Q_d is shown in fig. 18 for the angles, comparing encoder values to estimated values obtained with the IMU.

From fig. 18, it can be seen how higher values of Q_d let through more noise, while lower values smooth out the estimates. This corresponds well with the theoretical understanding of Q_d , as high uncertainty in the process model should make the filter rely more on measurements, while lower uncertainty should make the filter rely more on the model. High and low process uncertainty is of course relative to the measurement uncertainty, R_d . As seen, $Q_d = I \cdot 0.000001$ provides reasonable estimates compared to the true values. Higher values rely too much on measurements and give noisy estimates, while lower values rely too much on the model and give wrong estimates. The travel angle λ has an off-set of about 100° from the true angle, which may be a result of poor initialization before $t = 10$ s. As the travel angle is not incorporated into the LQR, this was not investigated further.

The value found for Q_d is suspiciously low compared to R_d . At a very late point in the lab, a mistake was found in the implementation of the Kalman filter in appendix B.2, where the system matrix A_d had not been transposed in the covariance prediction in line 5. Q_d was therefore tuned to a wrong implementation of the Kalman filter, which oddly enough gave a decent, but not satisfying, performing helicopter. The process above for tuning Q_d was therefore repeated with a correct implementation, but collecting data was not prioritized. Although fig. 18 depicts the tuning process for a wrong implementation, the approach for tuning Q_d is still correctly illustrated.

A similar approach may be used for the angular velocities, although the encoder values suffer from large spikes due to differentiation and are therefore more difficult to compare to.

As mentioned previously, the strategy for tuning Q_d above may not give a sufficient observer for all expected maneuvers by joystick. In particular, the elevation rate undergoes mild changes compared to what can be expected, and may therefore not be tuned sufficiently. A reasonable strategy at this



(a) True angles obtained by encoders

(b) Estimated angles obtained by IMU

Figure 18: Comparison of true angles and estimated angles for different Q_d

point, given that the initial tuning is decent, may be to provide the controller with state estimates from the filter instead of encoder values and observe the response of the helicopter to joystick input. This is not a very quantitative metric for performance, but it is after all the feel and look of the helicopter's response that is the end goal. This will thus be the strategy for the final tuning.

5.8 The Kalman filter

The Kalman filter is now to be used as state feedback to the LQR controller, depicted in appendix D, fig. 34. The initial tuning of Q_d may not give the best performance, but will now be tuned further by experimentation to give a good response.

The response of the pitch was initially seen to behave in a slowly decaying sinusoidal manner, indicating that the filter may weigh measurements too heavily. The first element of Q_d was therefore lowered to allow the filter to rely more on the model, until the response was satisfactory. The other states were tuned in a similar manner, with focus on pitch, pitch rate and elevation rate as these are used in the controller. The final value for Q_d after tuning was

$$Q_d = \begin{bmatrix} 0.0003 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

In retrospect, Q_d should probably have been evaluated in a more systematic manner, e.g. as the tuning for the LQR was in section 3.2. It is however difficult to tune to every possible maneuver the helicopter can make, so Q_d was after the initial tuning evaluated purely based on the look and feel of the response, ultimately giving a good response.

A noticeable off-set in the equilibrium of pitch was seen when using the Kalman filter, where the LQR would maintain a small angle in pitch. The reason was found to be an off-set in the accelerometer's measurement along the y-axis, measuring -0.4 m/s^2 in equilibrium, where it should be zero. This was accounted for in appendix B.1. The reason for this off-set is unknown, and as previously mentioned, it may be due to the IMU not being calibrated correctly, sensor bias or perhaps degradation with the harsh use of the helicopter on the lab.

There are now four possible ways to fly the helicopter: using encoders, raw measurements, predictions or the Kalman filter for state-feedback. Using raw measurements from the IMU or predictions from the model proved as poor observers, where measurements from the IMU are too noisy and will

give rapidly fluctuating states to the controller, while flying open-loop with predictions lacks accuracy of the model, giving an unstable controller especially outside the linearization point. The encoders where used in section 3, giving good performance. Unexpectedly, the response is better with state feedback from the Kalman filter than with the encoders. This is most likely due to the angular velocities, where the IMU gives better measurements than the differentiation of encoder values. This is clearly seen from the control vector u_k , which has a lot of noise when using encoders, due to the spikes in pitch rate and elevation rate. Using the gyroscope instead, u_k is a lot smoother, depending on how Q_d is tuned.

6 Conclusion

Several control schemes have been investigated for the control of a helicopter, using both encoders and an IMU for state observation. The helicopter was first modeled as a non-linear system, and subsequently linearized to allow the use of linear methods for control and estimation. A PD controller was implemented for controlling the pitch of the helicopter, where the behaviour was compared to theoretical observations on pole placement and harmonic oscillators, ultimately providing satisfactory compliance with theory. A linear quadratic regulator (LQR) was later implemented for controlling pitch and elevation rate, resulting in good performance after tuning.

The use of an IMU for state feedback was then investigated, and afterwards how noise affects the performance of the regulator. A Kalman filter was developed for retrieving estimates of true states hidden in noise, which was used in combination with the LQR for control of the helicopter.

After implementing LQR, state estimation and feedback, the performance of the helicopter appears satisfactory. It maneuvers well within the confines of linearization, and it adjusts to new references with a rapid and stable response. In fact, the helicopter has a slightly better performance after tuning the Kalman filter than it did when relying on the encoders.

A Derivations

A.1 Modeling, validation and manual control

A.1.1 Equations of motion derivation

$$\begin{aligned} J_p \ddot{p} &= F_{g,f} l_p \cos(p) - F_{g,b} l_p \cos(p) + F_b l_p - F_f l_p \\ J_p \ddot{p} &= m_p g l_p \cos(p) - m_p g l_p \cos(p) + K_f V_b l_p - K_f V_f l_p \\ J_p \ddot{p} &= l_p (K_f V_b - K_f V_f) \end{aligned} \quad (37a)$$

$$\begin{aligned} J_e \ddot{e} &= F_{g,c} l_c \cos(e) - F_{g,b} l_h \cos(e) - F_{g,f} l_h + (F_f + F_b) l_h \cos(p) \\ J_e \ddot{e} &= m_c g l_c \cos(e) - 2m_p g l_h \cos(e) + (K_f V_f + K_f V_b) l_h \cos(p) \end{aligned} \quad (37b)$$

$$\begin{aligned} J_\lambda \ddot{\lambda} &= (F_b + F_f) l_h \sin(p) \cos(e) \\ J_\lambda \ddot{\lambda} &= (K_f V_f + K_f V_b) l_h \sin(p) \cos(e) \end{aligned} \quad (37c)$$

A.2 Mono-variable control

A.2.1 Transfer function derivation

$$\begin{aligned} \ddot{p} &= K_1 K_{pp} (p_c - p) - K_1 K_{pd} \dot{p} \\ \mathcal{L}(\ddot{p}) &= \mathcal{L}(K_1 K_{pp} (p_c - p) - K_1 K_{pd} \dot{p}) \\ s^2 p(s) - sp(0) - \dot{p}(0) &= K_1 (K_{pp} (p_c(s) - p(s)) - s K_{pd} p(s)) - p(0) \\ s^2 p(s) + K_1 K_{pp} p(s) + s K_1 K_{pd} p(s) &= K_1 K_{pp} p_c(s) \\ p(s)(s^2 + s K_1 K_{pd} + K_1 K_{pp}) &= K_1 K_{pp} p_c(s) \\ \frac{p}{p_c}(s) &= \frac{K_1 K_{pp}}{s^2 + s K_1 K_{pd} + K_1 K_{pp}} \end{aligned} \quad (38)$$

A.3 Multi-variable control

A.3.1 Reference matrix derivation

$$\begin{aligned} \dot{x} &= Ax + Bu, \quad u = Fr_c - Kx \\ &= Ax + B(Fr_c - Kx), \quad t \rightarrow \infty \\ &= 0 \end{aligned}$$

\Rightarrow

$$\begin{aligned} Ax_\infty + BF_r c - BKx_\infty &= 0 \\ (A - BK)x_\infty &= -BF_r c \\ x_\infty &= (BK - A)^{-1}BF_r c \end{aligned}$$

\Rightarrow

$$\begin{aligned}
y &= Cx, \quad x = x_\infty \\
y_\infty &= C(BK - A)^{-1}BFr_c, \quad y_\infty = r_c \\
I &= C(BK - A)^{-1}BF \\
F &= (C(BK - A)^{-1}B)^{-1}
\end{aligned} \tag{39}$$

A.3.2 Reference matrix integral action derivation

$$\begin{aligned}
\dot{x} &= \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_a \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ -I \end{bmatrix} r_c, \quad u = \bar{F}r_c - [K_1 \ K_2] \begin{bmatrix} x \\ x_a \end{bmatrix} \\
&= \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_a \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} (\bar{F}r_c - [K_1 \ K_2] \begin{bmatrix} x \\ x_a \end{bmatrix}) + \begin{bmatrix} 0 \\ -I \end{bmatrix} r \\
&= \begin{bmatrix} (A - BK_1)x - BK_2x_a + B\bar{F}r_c \\ Cx - r_c \end{bmatrix}, \quad t \rightarrow \infty \\
&= \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\end{aligned}$$

$x_a = [0 \ 0]^T$ when final states are reached

\Rightarrow

$$\begin{aligned}
(A - BK_1)x_\infty + B\bar{F}r_c &= 0 \\
x_\infty &= (BK_1 - A)^{-1}B\bar{F}r_c
\end{aligned}$$

\Rightarrow

$$\begin{aligned}
y_\infty &= Cx_\infty \\
&= C(BK_1 - A)^{-1}B\bar{F}r_c, \quad y_\infty = r_c \\
\bar{F} &= (C(BK_1 - A)^{-1}B)^{-1}
\end{aligned} \tag{40}$$

A.4 The Kalman filter

A.4.1 Error covariance prediction derivation

$$\begin{aligned}
\bar{\varepsilon}_{k+1} &= x_{k+1} - \bar{x}_{k+1} \\
&= A_d x_k + B_d u_k + w_k - (A_d \bar{x}_k + B_d u_k) \\
&= A(x_k - \bar{x}_k) + w_k \\
&= A\bar{\varepsilon}_k + w_k
\end{aligned}$$

\Rightarrow

$$\begin{aligned}
\bar{P}_{k+1} &= \text{Cov}(\bar{\varepsilon}_{k+1}) \\
&= \text{Cov}(A_d \bar{\varepsilon}_k + w_k), \quad \bar{\varepsilon}_k \text{ and } w_k \text{ independent} \\
&= \text{Cov}(A_d \bar{\varepsilon}_k) + \text{Cov}(w_k) \\
&= A_d \text{Cov}(\bar{\varepsilon}_k) A_d^T + \text{Cov}(w_k) \\
&= A_d \bar{P}_k A_d^T + Q_d
\end{aligned} \tag{41}$$

A.4.2 Error covariance correction derivation

$$\begin{aligned}
\hat{P}_k &= \mathbb{E}[\hat{\varepsilon}_k \hat{\varepsilon}_k^T] \\
&= \mathbb{E}[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]
\end{aligned}$$

$$\begin{aligned}
x_k - \hat{x}_k &= x_k - (\bar{x}_k - K_k(y_k - \bar{y}_k)) \\
&= (I - K_k C_d)(x_k - \bar{x}_k) - K_k v_k
\end{aligned}$$

\Rightarrow

$$\begin{aligned}
\hat{P}_k &= \mathbb{E}[(x_k - \bar{x}_k - K_k v_k)((x_k - \bar{x}_k) - K_k v_k)^T] \\
&= \mathbb{E}[(x_k - \bar{x}_k)((x_k - \bar{x}_k)^T)] \\
&\quad - \mathbb{E}[(x_k - \bar{x}_k)(K_k v_k)^T] \\
&\quad - \mathbb{E}[(K_k v_k)((x_k - \bar{x}_k)^T)] \\
&\quad + \mathbb{E}[(K_k v_k)(K_k v_k)^T]
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}[(x_k - \bar{x}_k)((x_k - \bar{x}_k)^T)] &= \mathbb{E}[(x_k - \bar{x}_k)v_k^T K_k^T] \\
&= (I - K_k C_d) \mathbb{E}[(x_k - \bar{x}_k)v_k^T] K_k^T \\
&= (I - K_k C_d) \cdot 0 \cdot K_k^T \\
&= 0 \\
&= \mathbb{E}[(K_k v_k)((x_k - \bar{x}_k)^T)]
\end{aligned}$$

\Rightarrow

$$\begin{aligned}
\hat{P}_k &= \mathbb{E}[(x_k - \bar{x}_k)((x_k - \bar{x}_k)^T)] + \mathbb{E}[(K_k v_k)(K_k v_k)^T] \\
&= (I - K_k C_d) \mathbb{E}[(x_k - \bar{x}_k)(x_k - \bar{x}_k)^T] (I - K_k C_d)^T + K_k \mathbb{E}[v_k v_k^T] K_k^T \\
&= (I - K_k C_d) \bar{P}_k (I - K_k C_d)^T + K_k R_d K_k^T
\end{aligned} \tag{42}$$

A.4.3 Kalman gain derivation

$$\begin{aligned}\hat{P}_k &= (I - K_k C_d) \bar{P}_k (I - K_k C_d)^T + K_k R_d K_k^T \\ \text{tr}(\hat{P}_k) &= \text{tr}((I - K_k C_d) \bar{P}_k (I - K_k C_d)^T) + \text{tr}(K_k R_d K_k^T)\end{aligned}$$

$$\begin{aligned}(I - K_k C_d) \bar{P}_k (I - K_k C_d)^T &= (\bar{P}_k - K_k C_d \bar{P}_k)(I - K_k C_d)^T \\ &= \bar{P}_k - (K_k C_d \bar{P}_k)^T - K_k (C_d \bar{P}_k) + K_k (C_d \bar{P}_k C_d^T) K_k^T\end{aligned}$$

Using eq. (35),

$$\begin{aligned}\frac{\partial}{\partial K_k} \text{tr}((I - K_k C_d) \bar{P}_k (I - K_k C_d)^T) &= -\frac{\partial \text{tr}(K_k C_d \bar{P}_k)}{\partial K_k} - \frac{\partial \text{tr}(K_k C_d \bar{P}_k)}{\partial K_k} + \frac{\partial \text{tr}(K_k C_d \bar{P}_k C_d K_k^T)}{\partial K_k} \\ &= -2(C_d \bar{P}_k)^T + 2K_k C_d \bar{P}_k C_d^T \\ \frac{\partial \text{tr}(K_k R_d K_k^T)}{\partial K_k} &= 2K_k R_d\end{aligned}$$

\Rightarrow

$$\begin{aligned}\frac{\text{tr}(\hat{P}_k)}{\partial K_k} &= -2(C_d \bar{P}_k)^T + 2K_k C_d \bar{P}_k C_d^T + 2K_k R_d \\ &= -2(C_d \bar{P}_k)^T + 2K_k (C_d \bar{P}_k C_d^T + R_d)^T, \quad \text{minimize} \\ &= 0\end{aligned}$$

\Rightarrow

$$K_k = \bar{P}_k C_d^T (C_d \bar{P}_k C_d^T + R_d)^{-1} \quad (43)$$

B MATLAB Code

B.1 acc_to_euler_angles.m

```

1 function [pitch, elevation] = acc_to_euler_angles(acc_vec)
2     a_x = acc_vec(1);
3     a_y = acc_vec(2) + 0.4;
4     a_z = acc_vec(3);
5
6     pitch = atan2(a_y, a_z);
7     elevation = atan2(a_x, sqrt(a_y^2 + a_z^2));
8 end

```

B.2 kalman_filter.m

```

1  function [x_est, P_est] = kalman_filter(x_prev, P_prev, u_k, ...
2                                              A, B, C, Q, R, y, isNewData)
3      %% Kalman predict
4      x_pred = A*x_prev + B*u_k;
5      P_pred = A*P_prev*A' + Q;
6      y_pred = C*x_pred;
7
8      %% Kalman gain
9      K = P_pred*C' / (C*P_pred*C' + R);
10
11     %% Kalman update
12     if (isNewData)
13         x_est = x_pred + K*(y - y_pred);
14         I = eye(6);
15         P_est = (I - K*C)*P_pred*(I - K*C)' + K*R*K';
16     else
17         x_est = x_pred;
18         P_est = P_pred;
19     end
20 end

```

C MATLAB Figures

C.1 Equations of motion

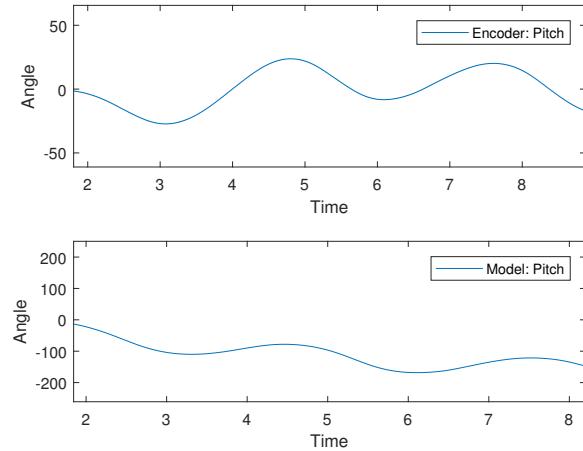


Figure 19: Pitch angle from encoder and model around linearization point

C.2 Linear Quadratic Regulators

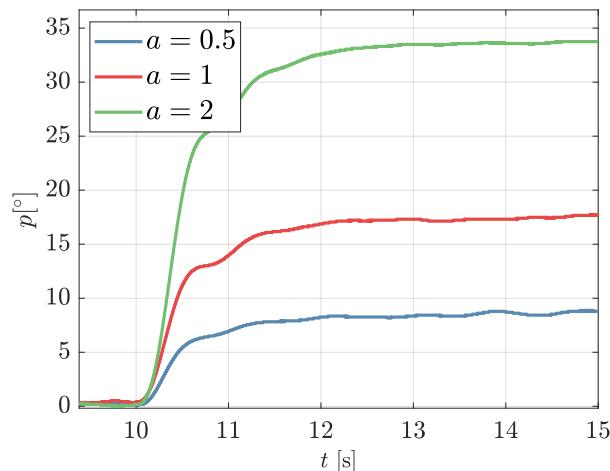


Figure 20: Step response for pitch angle with scalar multiplications of reference-matrix F , by some scalar a

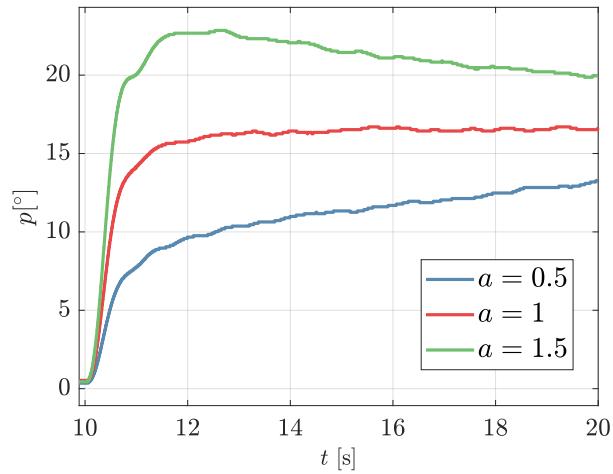


Figure 21: Step response for pitch angle with scalar multiplications of reference-matrix \bar{F} , by some scalar a , with integral effect

D Simulink Block Diagrams

D.1 Equations of motion

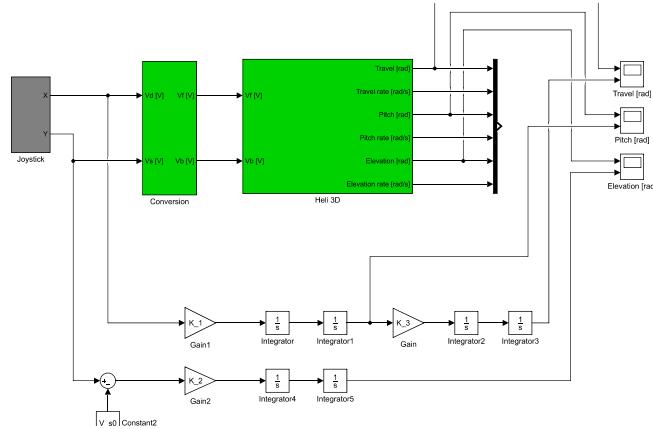


Figure 22: Simulink model in Part I with joystick input

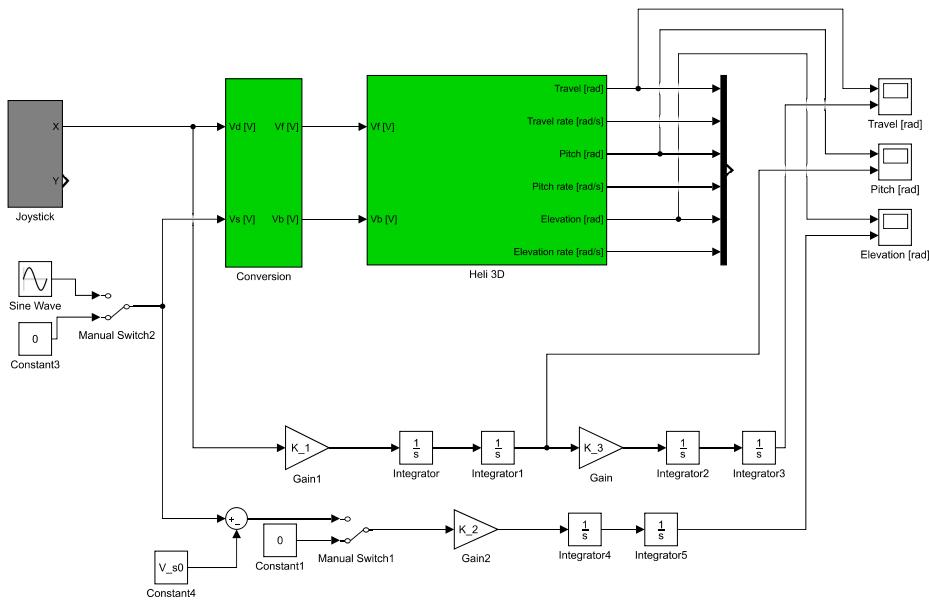


Figure 23: Simulink model in Part I with a sine wave input

D.2 Mono-variable control

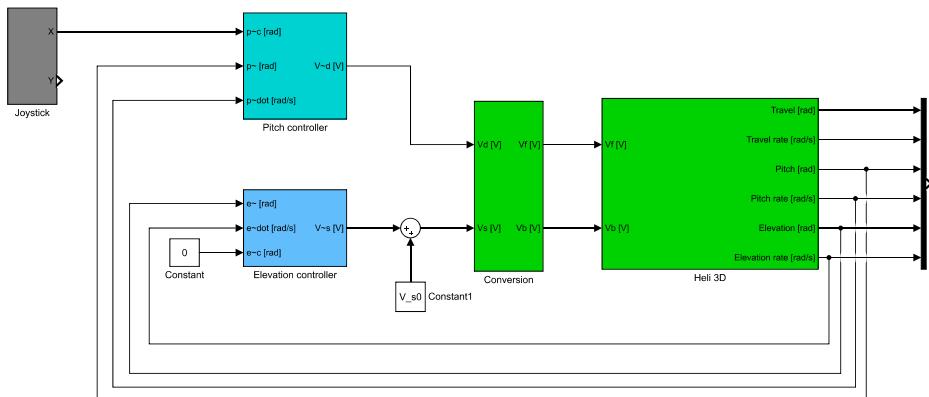


Figure 24: Simulink model in Part II with joystick input

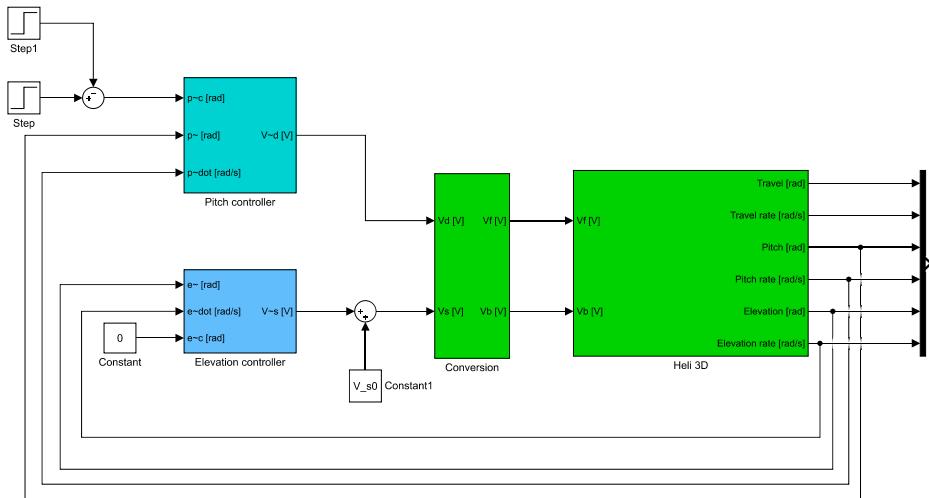


Figure 25: Simulink model in Part II with impulse response

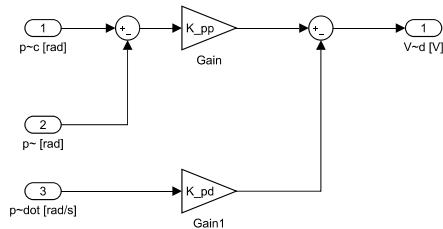


Figure 26: Simulink model for pitch controller in Part II

D.3 Multi-variable control

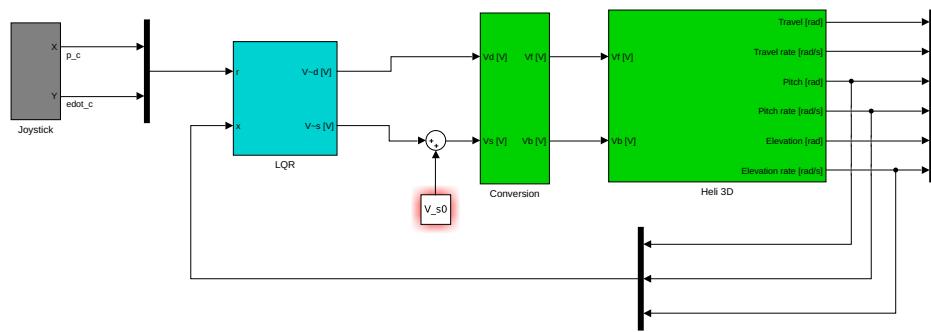


Figure 27: Simulink model for multi-variable control in Part III

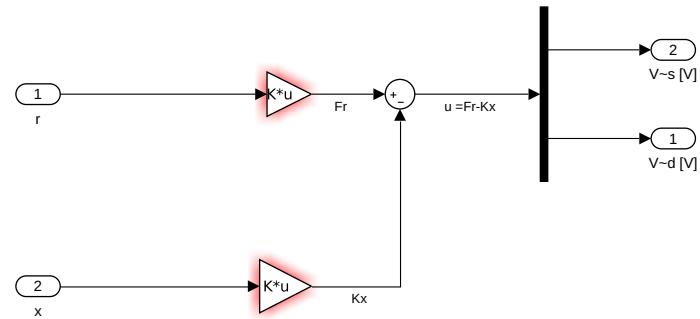


Figure 28: Simulink model for LQR in Part III

D.4 Inertial Measurement Unit

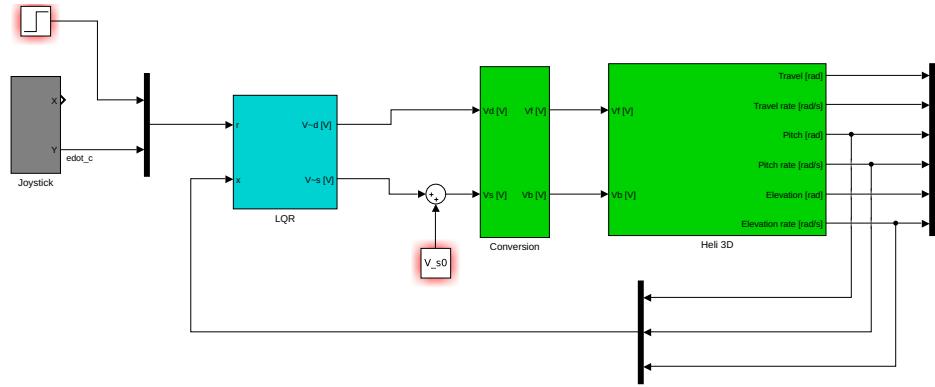


Figure 29: Simulink model with step input for pitch in Part III

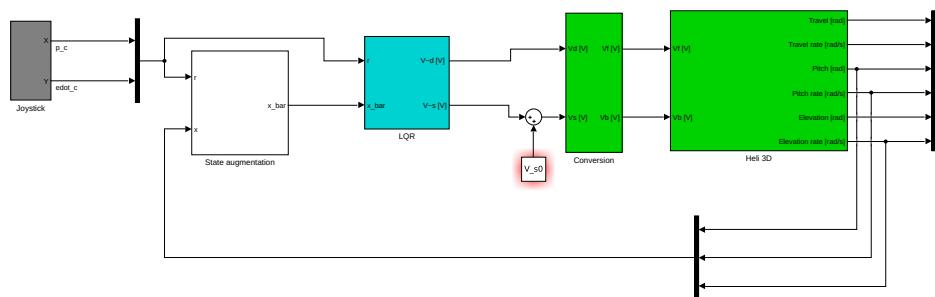


Figure 30: Simulink model with integral effect in Part III

D.5 The Kalman filter

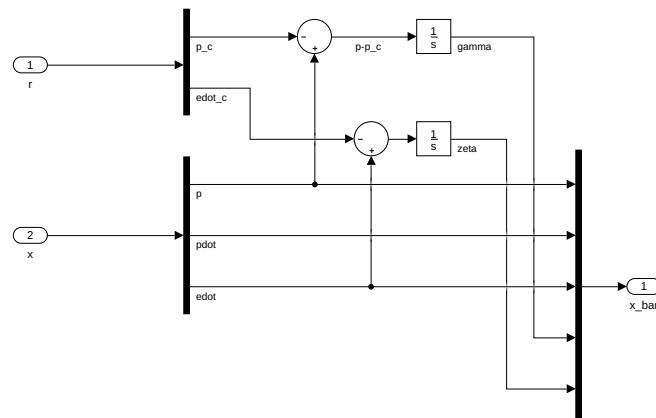


Figure 31: Simulink model for state augmentation in Part III

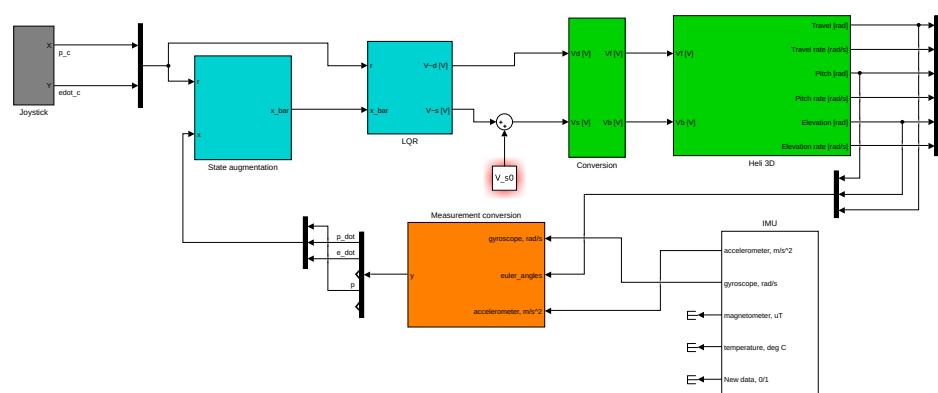


Figure 32: Simulink model for IMU feedback in Part IV

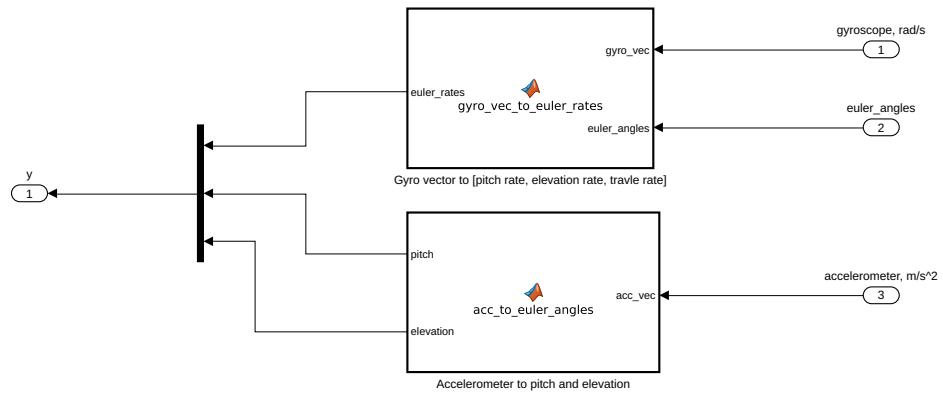


Figure 33: Simulink model for conversion of measurement from IMU in Part IV

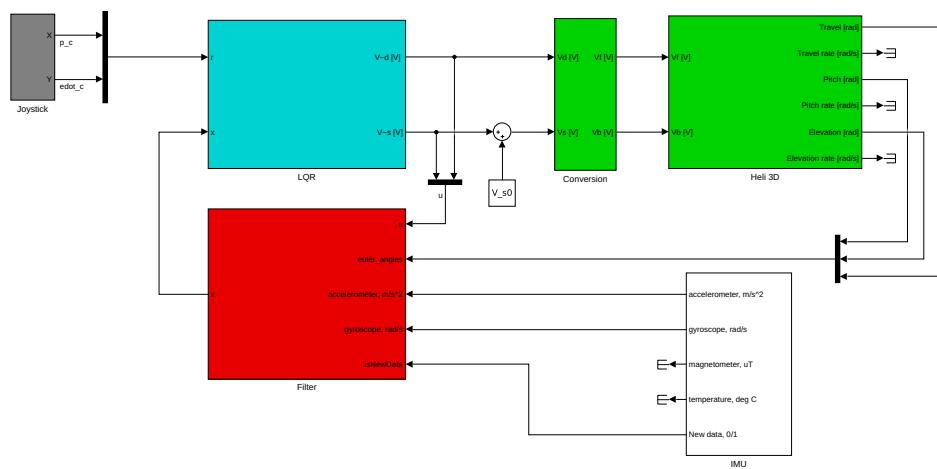


Figure 34: Simulink model for LQR with Kalman filtering in Part VI

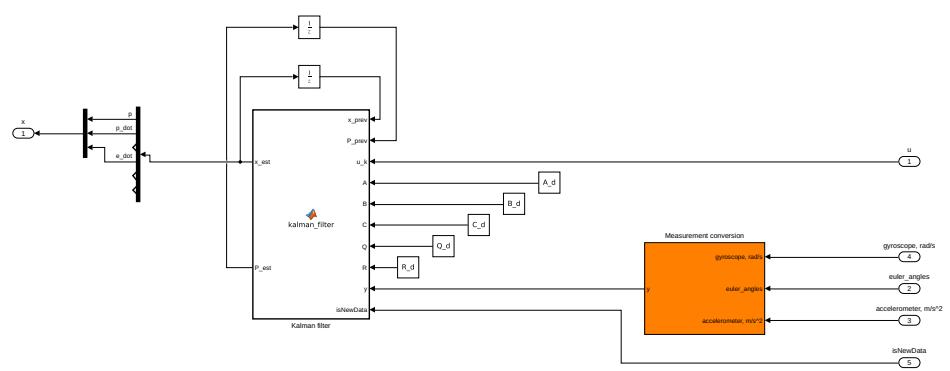


Figure 35: Simulink model for the Kalman filter in Part VI

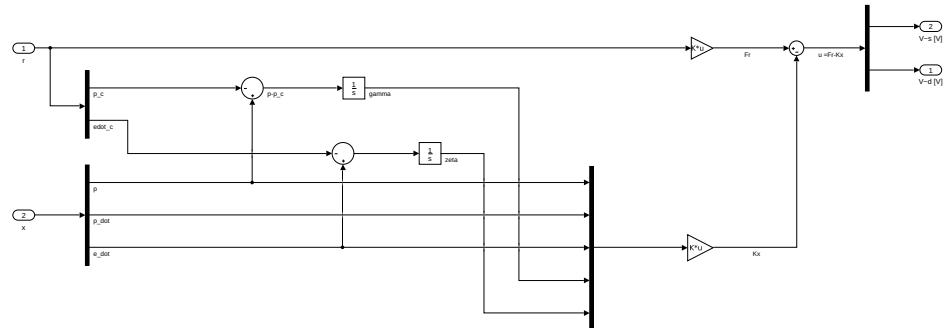


Figure 36: Simulink model for LQR in Part VI