TTK4115 Linear System Theory
Helicopter Lab report

Group 62

Fan Gao 502425
Hongyu Zhou 502423
Shuyuan Shen 502427

NTNU

Department of Engineering Cybernetics
Norwegian University of Science and Technology

October 21,2018

Contents

1 Introduction

In this Helicopter Lab project, our task is to control a 3 DOF helicopter model. Based on simplification of the helicopter model and mathematical calculation, the equilibrium equations have been derived. In practice, PID control, feed-forward, state estimator and LQR method are involved in the system design and optimization of the helicopter's performance. Also, comparison between simulation results of different controllers is presented through graphs and comments.

2 Part I - Mathematical modeling

2.1 Problem 1

In order to analyze, control and optimize the helicopter, we should first mathematically model it. The model of the helicopter is depicted in Fig. 1. There are 3 point masses in the helicopter model: 2 masses (m_f and m_b) represent motors or propellers and 1 mass (m_c) is the counterweight to balance the force generated by motors. l_c, l_h and l_p denote distances between joints and masses. The 3 cubes represent 3 point masses, two of which in the same side are motors. The 3 cylinders are helicopter joints and the cylinder's axis equals to the axis of rotation respectively: p denotes pitch angle of heading, e denotes elevation angle and λ denotes travel angle. The pitch angle $p = 0$ represents horizontal heading and elevation angle $e = 0$ represents the 3 masses are of the same vertical position. Linear relationship between supplied voltage to the motors, and applied forces from the rotors are also assumed. Other forces, such as friction and centripetal forces are neglected.

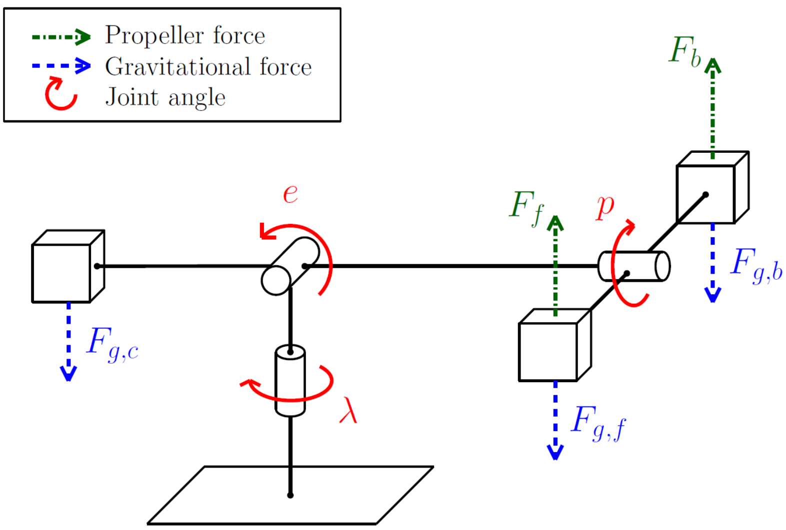

Figure 1: Helicopter model

The equations of motion (differential equations) for the pitch angle p, the elevation angle e, and the travel angle λ are shown below and the moments of inertia are respectively J_p, J_e and J_λ.

Pitch

Pitch motion can be modelled by the following equations

$$J_p\ddot{p} = F_f p - F_b l_p + F_{g,f} l_p cos(p) - F_{g,b} l_p cos(p) \tag{2.1}$$

with formulas are given by

$$F_f = K_f V_f \quad F_b = K_f V_b \quad F_{g,f} = F_{g,b} = m_p g \tag{2.2}$$

Thus

$$J_p\ddot{p} = K_f V_b l_p = L_1 V_d \tag{2.3}$$

Elevation

Elevation motion can be modelled by the following equations

$$J_e\ddot{e} = (F_f + F_b)cos(p)l_h - (F_{g,f} + F_{g,b})cos(e)l_h + F_{g,c} l_c cos(e) \tag{2.4}$$

with formula is given by

$$F_{g,c} = m_c g \tag{2.5}$$

Thus

$$J_e\ddot{e} = (m_c l_c - 2m_p l_h)gcos(e) + K_f V_s l_h cos(p) = L_2 cos(e) + L_3 V_s cos(p) \tag{2.6}$$

Travel

Travel motion can be modelled by the following equations

$$J_\lambda\ddot{\lambda} = -F_f sin(p) l_h cos(e) - F_b sin(p) l_h cos(e) = L_4 V_s cos(e) sin(p) \tag{2.7}$$

2.2 Problem 2

To linearize the equation of motion around$(p, e, \lambda)^T = (p^*, e^*, \lambda^*)^T$, where $p^* = e^* = \lambda^* = 0$, $(\dot{p}, \dot{e}, \dot{\lambda})^T = (0, 0, 0)^T$. We can use equations below and then get V_s and V_d

$$\begin{aligned} J_p\ddot{p}^* &= L_1 V_d^* = 0 \\ J_e\ddot{e}^* &= L_2 + L_3 V_s^* = 0 \\ J_\lambda\ddot{\lambda}^* &= L_4 V_s^* = 0 \end{aligned} \tag{2.8}$$

The solutions are

$$\begin{aligned} V_d^* &= 0 \\ V_s^* &= dfracm_c g l_c - 2m_p g l_h K_f l_h \end{aligned} \tag{2.9}$$

So that we can get the equation of motion by inserting V_s and V_d

$$\begin{aligned} \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \lambda \end{bmatrix} &= \begin{bmatrix} p \\ e \\ \lambda \end{bmatrix} - \begin{bmatrix} p^* \\ e^* \\ \lambda^* \end{bmatrix} \\ \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} &= \begin{bmatrix} V_s \\ V_d \end{bmatrix} + \begin{bmatrix} V_s^* \\ V_d^* \end{bmatrix} \end{aligned} \tag{2.10}$$

$$
\begin{aligned}
\begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \lambda \end{bmatrix} &= \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} \\
\begin{bmatrix} V_s \\ V_d \end{bmatrix} &= \begin{bmatrix} \tilde{V}_s - \dfrac{m_c g l_c - 2 m_p g l_h}{K_f l_h} \\ \tilde{V}_d \end{bmatrix}
\end{aligned}
\tag{2.11}
$$

$$
\begin{aligned}
J_p \ddot{p} &= K_f l_p (\tilde{V}_d + V_d^*) = K_f l_p \tilde{V}_d \\
J_e \ddot{e} &= K_f l_h cos(p)(\tilde{V}_s + V_s^*) + (m_c l_c - 2 m_p l_h) g cos(e) = K_f l_h \tilde{V}_s \\
J_\lambda \ddot{\lambda} &= -K_f V_s l_h sin(p) cos(e) = -(m_c l_c - 2 m_p l_h) g \tilde{p}
\end{aligned}
\tag{2.12}
$$

Using these values, the Jacobians can be calculated:

$$
\begin{bmatrix} J_p \ddot{\tilde{p}} \\ J_e \ddot{\tilde{e}} \\ J_\lambda \ddot{\tilde{\lambda}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -(m_c l_c - 2 m_p l_h) g & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} + \begin{bmatrix} 0 & K_f l_p \\ K_f l_h & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix}
\tag{2.13}
$$

$$
\begin{aligned}
\ddot{\tilde{p}} &= K_1 \tilde{V}_d \\
\ddot{\tilde{e}} &= K_2 \tilde{V}_s \\
\ddot{\tilde{\lambda}} &= K_3 \tilde{p}
\end{aligned}
\tag{2.14}
$$

$$
\begin{aligned}
K_1 &= K_f l_p \\
K_2 &= K_f l_h \\
K_3 &= -(m_c l_c - 2 m_p l_h) g
\end{aligned}
\tag{2.15}
$$

2.3 Problem 3

Let the signal from the x-axis of the joystick connect to the voltage difference V_d with a small gain and the signal from the y-axis of the joystick connect directly to the voltage sum V_s. We find it is really difficult to control the helicopter for many reasons. On the one hand, the unbalanced weight of m_b and m_f may require a small V_d but we cannot input it exactly; consequently, the deviation leads to bad behavior. On the other hand, we neglect the influence of friction which can also affect the motion of helicopter.

2.4 Problem 4

To adjust pitch motion and elevation motion to be zero, we change the input constant and then get $V_s^* = 6.6$V. So we calculate K_f by using the equation

$$
K_f = -\frac{m_c g l_c - 2 m_p g l_h}{V_s l_h} = 0.1513
\tag{2.16}
$$

Thus

$$
\begin{aligned}
K_1 &= 0.6006 \\
K_2 &= 0.0966 \\
K_3 &= -0.6117
\end{aligned}
\tag{2.17}
$$

3 Part II - Monovariable control

3.1 Problem 1

In this part, a PD controller is implemented by using the equation

$$\tilde{V}_d = K_{pp}(\tilde{p}_c - \tilde{p}) - K_{pd}\dot{\tilde{p}} \tag{3.1}$$

$$\ddot{\tilde{p}} = K_1\tilde{V}_d = K_1K_{pp}(\tilde{p}_c - \tilde{p}) - K_1K_{pd}\dot{\tilde{p}} \tag{3.2}$$

Using the laplace transformation, assuming $\tilde{p}(0) = 0$

$$s^2\tilde{p}(s) = K_1K_{pp}(\tilde{p}_c(s) - \tilde{p}(s)) - sK_1K_{pp}\tilde{p}_c(s) \tag{3.3}$$

Solving this gives us the transfer function

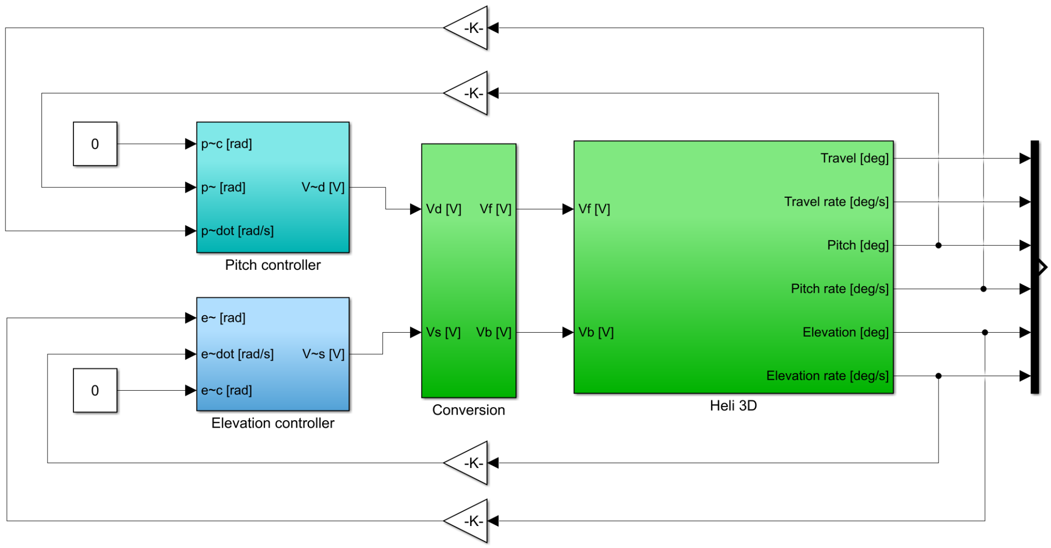

Figure 2: Part II, Problem 1

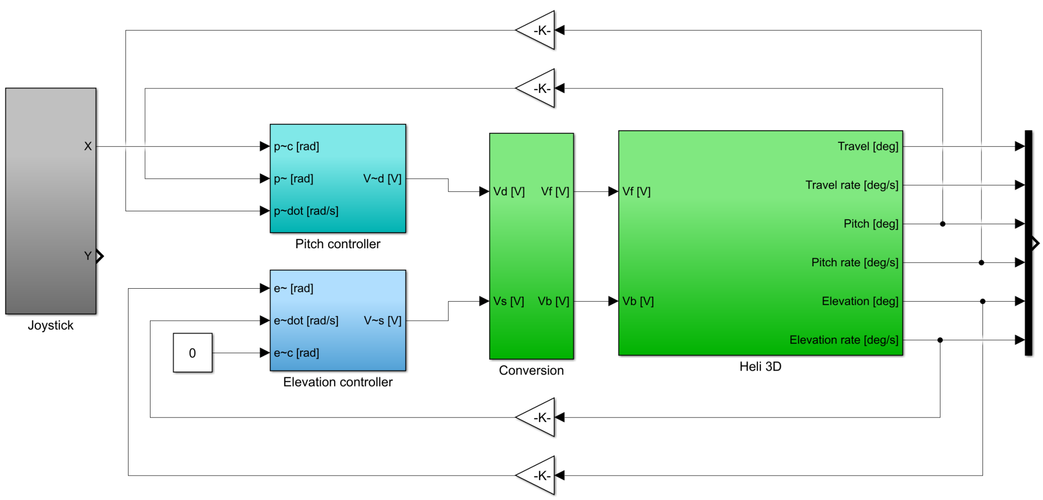

Figure 3: Part II, Problem 1 Connect the joystick

$$\frac{\tilde{p}(s)}{\tilde{p}_c(s)} = \frac{K_1 K_{pp}}{s^2 + sK_1 K_{pd} + K_1 K_{pp}} \tag{3.4}$$

The linearized pitch dynamics can be regarded as a second-order linearized system, given by the transfer function

$$\tilde{g}(s) = \frac{\omega_0^2}{s^2 + 2\omega_0 \xi s + \omega_0^2} \tag{3.5}$$

The pitch motion should react rapidly without much oscillations. The resonance frequency ω_0 decides how fast the pitch control will be. In order to get desired or optimal behavior of the helicopter motion, we need to choose proper ξ and ω_0

$$\begin{aligned} \omega_0^2 &= K_1 K_{pp} \\ 2\xi\omega_0 &= K_1 K_{pd} \end{aligned} \tag{3.6}$$

The response of system is influenced by the poles chosen, which is connected with ξ. On the one hand, if we chose $\xi < 1$, which means the poles had imaginary part, the helicopter would have an overshoot when stabilizing. On the other hand, if we chose $\xi > 1$, which gives two seperate poles only with negative real part, the response would be slower when reaching desired position. Therefore, we choose $\xi = 1$, which gives critical damping. With regard to ω, higher resonance frequency may damage the helicopter, or make it unstable. Hence we choose $\omega_0 = 2.5$ by testing.
Then we can get the K_{pp} and K_{pd} as follow

$$\begin{aligned} K_{pd} &= \frac{2\xi\omega_0}{K_1} = 8.3257 \\ K_{pp} &= \frac{\omega_0^2}{K_1} = 10.4071 \end{aligned} \tag{3.7}$$

3.2 Problem 2

In this problem a P-controller is implemented in the pitch control

$$\tilde{p}_c = K_{rp}(\dot{\lambda}_c - \dot{\lambda}) \tag{3.8}$$

When $\tilde{p} = \tilde{p}_c$

$$\ddot{\tilde{\lambda}} = K_3 \tilde{p} = K_3 \dot{\tilde{p}}_c = K_3 K_{rp}(\dot{\lambda}_c - \dot{\lambda}) \tag{3.9}$$

Using Laplace transformation

$$(s + K_3 K_{rp})\dot{\tilde{\lambda}}(s) = K_3 K_{rp} \dot{\tilde{\lambda}}(s) \tag{3.10}$$

We can calculate the transfer function

$$\frac{\dot{\tilde{\lambda}}(s)}{\dot{\tilde{\lambda}}_c(s)} = \frac{K_3 K_{rp}}{s + K_3 K_{rp}} = \frac{\rho}{s + \rho} \tag{3.11}$$

With

$$\rho = K_3 K_{rp} \tag{3.12}$$

The inner loop should be faster than the outer. The inner loop is given by $\omega_0 = 2.5$, K_{rp} is supposed to be chosen smaller. Choose $K_{rp} = -0.5$.

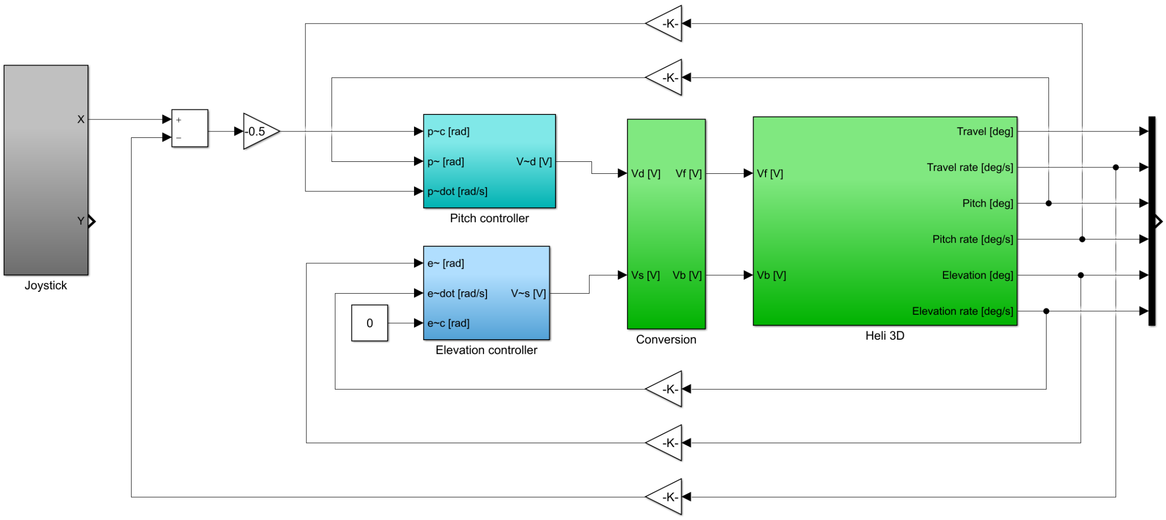

Figure 4: Part II, Problem 2

4 Part III - Multivariable control

4.1 Problem 1

In this part, we need to control the pitch angle $\tilde{p}$ and the elevation rate $\dot{\tilde{e}}$ with a multivariable controller.
First, establish the state space equation as follow

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{4.1}$$

The state vector $\mathbf{x}$ and input vector $\mathbf{u}$ are given as

$$\mathbf{x} = \begin{bmatrix} \tilde{p} & \dot{\tilde{p}} & \dot{\tilde{e}} \end{bmatrix}^T \quad \mathbf{u} = \begin{bmatrix} \tilde{V}_s & \tilde{V}_d \end{bmatrix}^T \tag{4.2}$$

Matrices $\mathbf{A}$ and $\mathbf{B}$ are stated as

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 1 \\ 0 & K_1 \\ K_2 & 0 \end{bmatrix} \tag{4.3}$$

4.2 Problem 2

We first examine the controllability

$$\mathcal{C} = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & K_1 & 0 & 0 \\ 0 & K_1 & 0 & 0 & 0 & 0 \\ K_2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.4}$$

rank($\mathcal{C}$)=3, the controllability matrix has full rank so that the system is controllable.
Then we make a controller

$$\mathbf{u} = \mathbf{P}\mathbf{r} - \mathbf{K}\mathbf{x} \quad \mathbf{r} = \begin{bmatrix} \tilde{p}_c \\ \dot{\tilde{e}}_c \end{bmatrix} \tag{4.5}$$

Where the matrix $\mathbf{K}$ corresponds to LQR for which the control input $\mathbf{u} = -\mathbf{K}\mathbf{x}$.
Optimizes the cost function

$$\mathbf{J} = \int_0^\infty \mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)dt \tag{4.6}$$

The K matrix can be calculated by MATLAB command lqr(A,B,Q,R). When it comes to the calculation of $\mathbf{P}$ matrix, it should fulfill that $\mathbf{y}(t) \to r$ as $t \to \infty$

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{BK})\mathbf{x} + \mathbf{BPr} = 0 \tag{4.7}$$

$\mathbf{r}$ is given as

$$\mathbf{r} = \mathbf{C}\mathbf{x}_\infty = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{p}_c \\ \dot{\tilde{p}}_c \\ \dot{\tilde{e}}_c \end{bmatrix} \tag{4.8}$$

Thus we have

$$\mathbf{r}_\infty = \mathbf{C}\mathbf{x}_\infty = [\mathbf{C}(\mathbf{BK} - \mathbf{A})^{-1}\mathbf{BP}]\mathbf{r}_\infty \tag{4.9}$$

$$\mathbf{P} = [\mathbf{C}(\mathbf{BK} - \mathbf{A})^{-1}\mathbf{B}]^{-1} \tag{4.10}$$

Usually, $\mathbf{Q}$ and $\mathbf{R}$ are diagonal. So we set the $\mathbf{Q}$ and $\mathbf{R}$ matrix to be identity matrix and then change the diagonal elements respectively to optimize the performance of the helicopter. Our final $\mathbf{Q}$ and $\mathbf{R}$ matrix are set below

$$\mathbf{Q} = \begin{bmatrix} 90 & 0 & 0 \\ 0 & 25 & 0 \\ 0 & 0 & 90 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.11}$$

Therefore the $\mathbf{K}$ and $\mathbf{P}$ matrices are

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 9.4868 \\ 9.4868 & 7.5229 & 0 \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} 0 & 9.4868 \\ 9.4868 & 0 \end{bmatrix} \tag{4.12}$$

It is evident that we give $\mathbf{Q}$ elements higher value than $\mathbf{R}$ elements. That is because we focus more on the behavior of the helicopter and should definitely make the cost for mistakes in the states higher, regardless of the cost of input. But in real engineering, we need also take the input cost into consideration.
With this tuning, the control system behaved desirable. The pitch adjusts fast and accurately without excessive oscillations, and the elevation adjusts well too. However, one problem is that when we use joystick to make the helicopter further away from its linearized area and then let go of the joystick, the helicopter would go back to the equilibrium point instead of just keeping stable. We contend that it is not possible to maintain zero error for a closed-loop system without integral effect if there are disturbances in the system. Later in next problem the comparison of step response between with and without integral effect will be present.

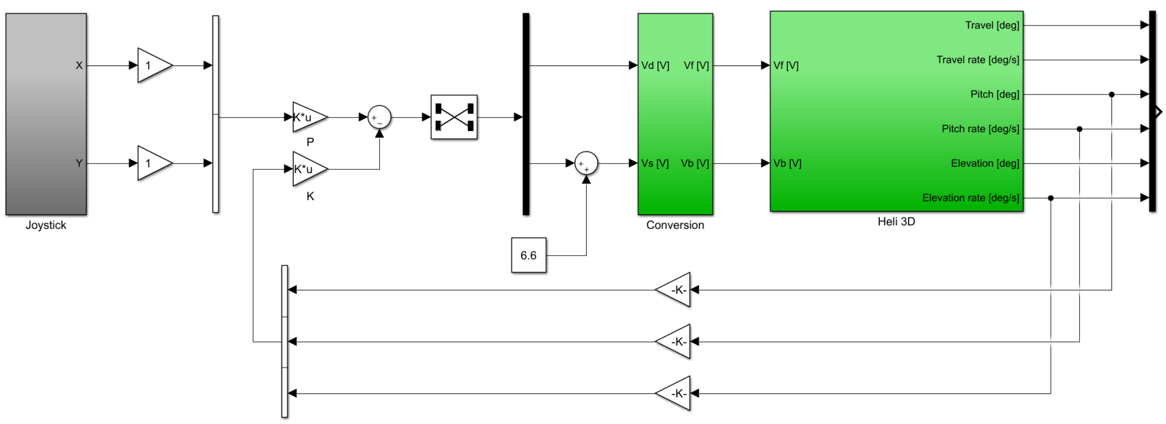

Figure 5: Part III, Problem 2

4.3 Problem 3

In this part, we need to include an integral effect for the elevation rate and the pitch angle.

$$\begin{aligned} \dot{\gamma} &= \tilde{p} - \tilde{p}_c \\ \dot{\zeta} &= \dot{\tilde{e}} - \dot{\tilde{e}}_c \end{aligned} \tag{4.13}$$

With integration, the helicopter can track the reference much better because the integral effect removes stationary error. Also, Use LQR to minimize γ and ζ.
After tuning in the same manner as without integral effect, the following $\mathbf{Q}$ and $\mathbf{R}$ matrices were chosen

$$\mathbf{Q} = \begin{bmatrix} 90 & 0 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 & 0 \\ 0 & 0 & 90 & 0 & 0 \\ 0 & 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 0 & 40 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.14}$$

Then we get the $\mathbf{P}$ and $\mathbf{K}$ matrices

$$\begin{aligned} \mathbf{K} &= \begin{bmatrix} 0 & 0 & 14.8622 & 0 & 6.3246 \\ 14.5312 & 8.5670 & 0 & 7.0711 & 0 \end{bmatrix} \\ \mathbf{P} &= \begin{bmatrix} 0 & 14.8622 \\ 14.8622 & 0 \end{bmatrix} \end{aligned} \tag{4.15}$$

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \ddot{\tilde{e}} \\ \dot{\gamma} \\ \dot{\zeta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \\ \gamma \\ \zeta \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\tilde{p}_c \\ -\dot{\tilde{e}}_c \end{bmatrix} \tag{4.16}$$

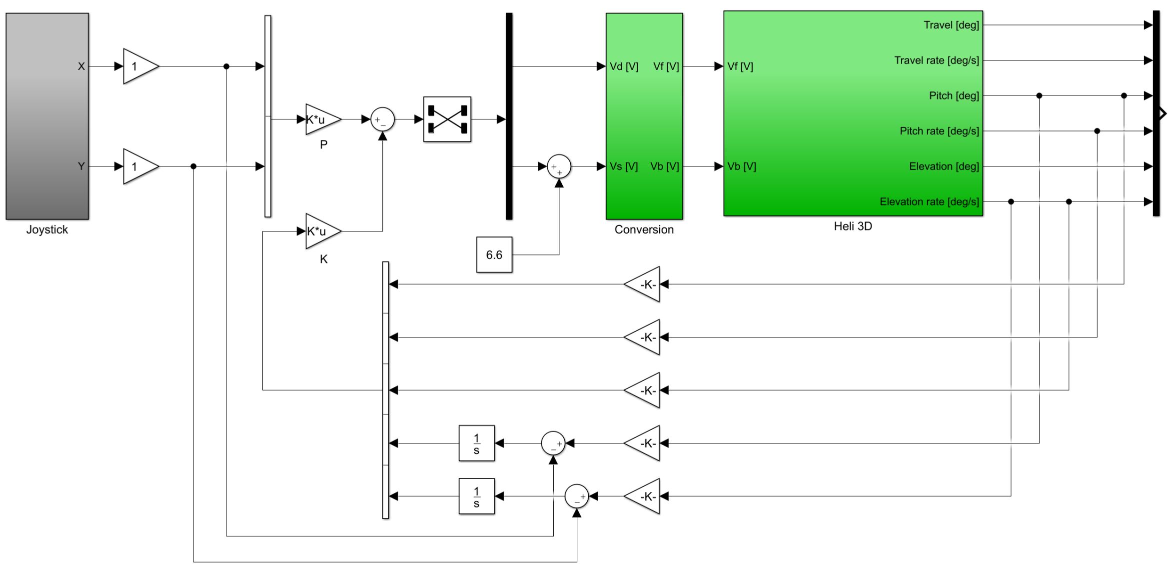

Figure 6: Part III, Problem 2

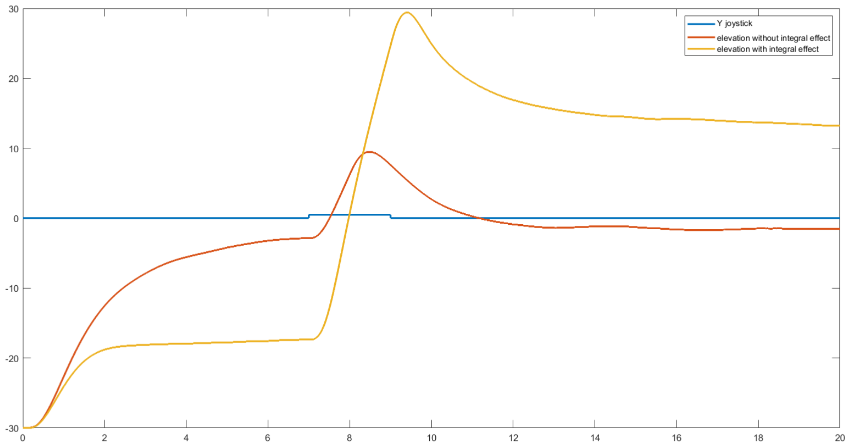

Figure 7: Step responce without and with Integral effect

Shown in the figure 7, we combined two step blocks in Simulink to generate the step input from $t = 7s$ to $t = 9s$ with the magnitude of 0.5, so that the difference of elevation between the helicopter with and without integral effect becomes obvious.
As discussed in last question, the helicopter would go back to the equilibrium point when $\dot{\tilde{e}} = 0$, which coincides with the result of step responce. On the contrary, the helicopter with integral effect tracks the given reference $\dot{\tilde{e}}$ better and is able to remain stable even if it is out of the linearized area. This is because of disturbance compensation provided by integral action. Also, one of states results from integral action, i.e. $\zeta = \tilde{e} - \tilde{e}_c$, will be minimized by LQR controller, which tries to make $\tilde{e}$ equals $\tilde{e}_c$.

5 Part IV - State estimation

Based on the previous control, we can only detect the helicopter's motion regardless of relevant motion rate. In this section, we involve state estimator to measure the motion rate.

5.1 Problem 1

We derive the state-space equation From previous mathematical calculation and then change the state vector $\mathbf{x}$ and input vector $\mathbf{u}$.

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \end{aligned} \tag{5.1}$$

$$\mathbf{x} = \begin{bmatrix} \tilde{p} & \dot{\tilde{p}} & \tilde{e} & \dot{\tilde{e}} & \tilde{\lambda} & \dot{\tilde{\lambda}} \end{bmatrix}^T \quad \mathbf{u} = \begin{bmatrix} \tilde{V}_s & \tilde{V}_d \end{bmatrix}^T \quad \mathbf{y} = \begin{bmatrix} \tilde{p} & \tilde{e} & \tilde{\lambda} \end{bmatrix}^T \tag{5.2}$$

Respectively, we can get the $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ matrix.

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ K_3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ 0 & 0 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.3}$$

The observability matrix is:

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} & \mathbf{CA} & \mathbf{CA}^2 & \mathbf{CA}^3 & \mathbf{CA}^4 & \mathbf{CA}^5 \end{bmatrix}^T \tag{5.4}$$

rank($\mathcal{O}$)=6, which means the system is observable. So we can continue our observer design.

5.2 Problem 2

Using state estimation, the state-space equation is:

$$\begin{aligned} \dot{\hat{\mathbf{x}}} &= \mathbf{A}\hat{\mathbf{x}} + \mathbf{L}(\mathbf{y} - \hat{\mathbf{y}}) \\ \hat{\mathbf{y}} &= \mathbf{C}\hat{\mathbf{x}} \end{aligned} \tag{5.5}$$

The error is defined as:

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}} \tag{5.6}$$

Given that our measured y has noise $\mathbf{n}$

$$\mathbf{y}_m = \mathbf{y} + \mathbf{n} \tag{5.7}$$

Then the error rate equals to:

$$\dot{\mathbf{e}} = (\mathbf{A} - \mathbf{LC})\mathbf{e} - \mathbf{Ln} \tag{5.8}$$

Through changing the pole placement of $(\mathbf{A} - \mathbf{LC})$, we can design the estimator to have a good performance. We should give the pole a larger negative real number for

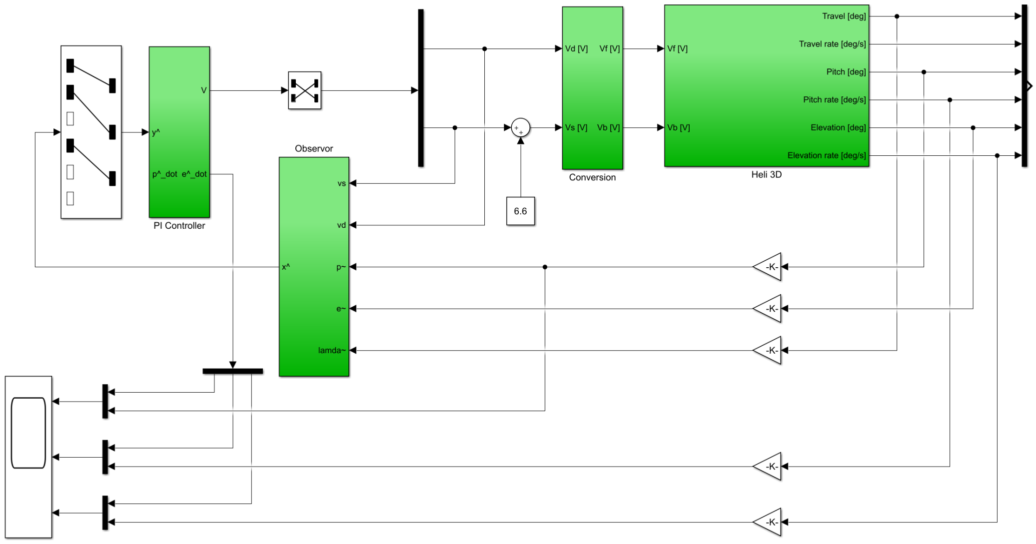

Figure 8: Part IV, Problem 2

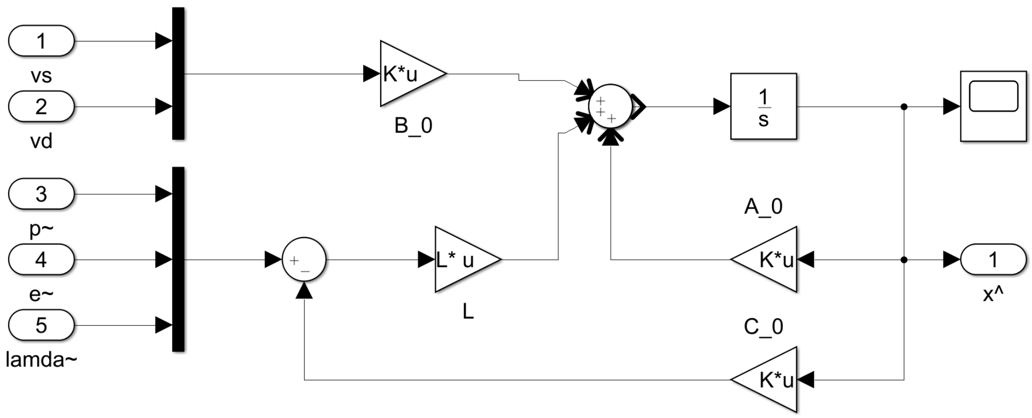

Figure 9: Simulink diagram of observer

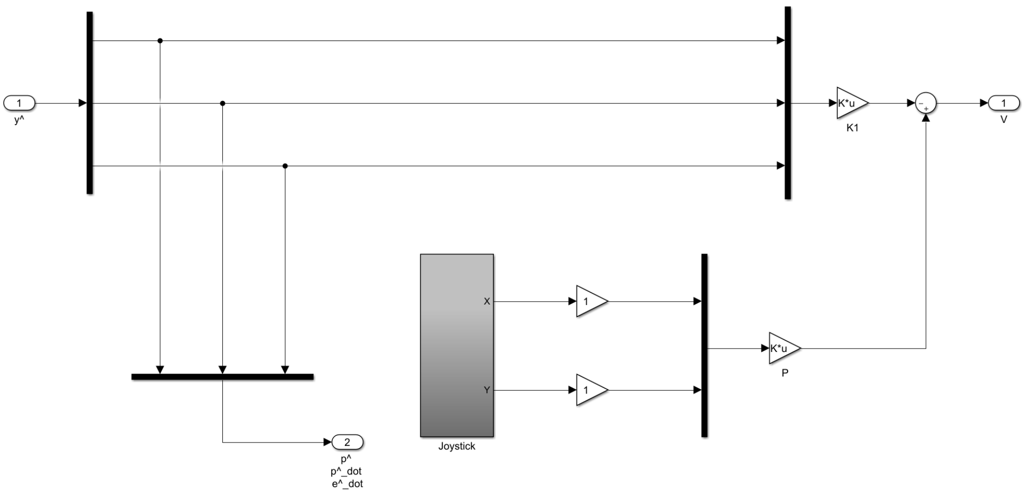

Figure 10: Simulink diagram of PI controller

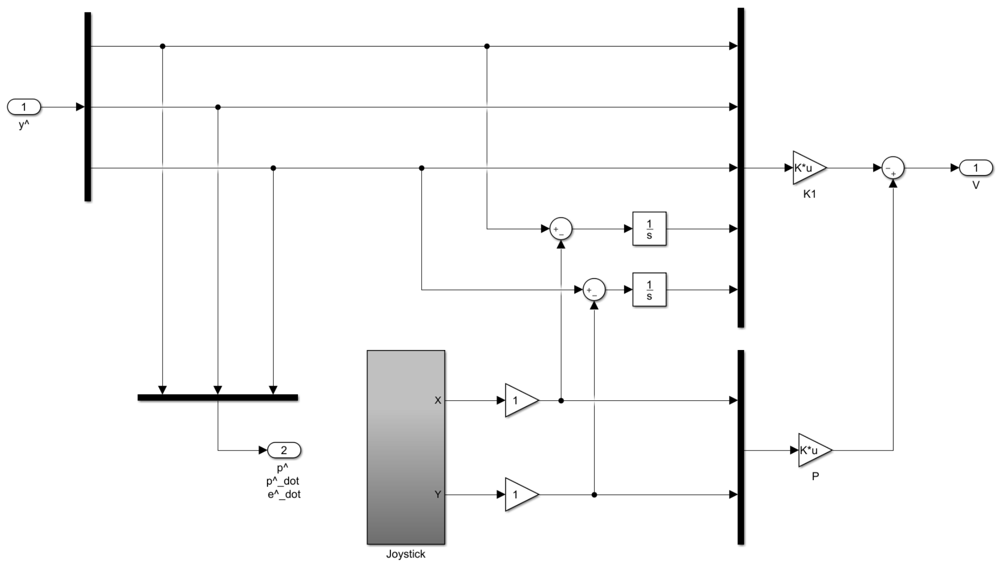

Figure 11: Simulink diagram of integrator PI controller

the error dynamics should be faster than the system dynamics. At the same time, higher values $\mathbf{L}$ leads to more noise. Lower values of L will also result in slower estimators. In MATLAB, after setting our desired poles, we can use the function '$place(\mathbf{A}, \mathbf{C}, poles)$' to get the $\mathbf{L}$ matrix. The estimator poles are distributed in the left half plane, manually set at a circular arc, the radius of which is set to a multiple times the distance to the most negative pole of the system itself. From figure 13, we can observe that the estimator underestimates pitch rate. If we want estimator has more accurate estimation, we could simply increase the radius that decides the poles of estimator. However, we also discovered that this solution would result in noise of estimated elevation rate. Therefore, the accuracy of estimated pitch rate is slightly sacrificed in order to prevent noise. Even though, the estimator is able to follow the real states with acceptable error.

$$\mathbf{L} = \begin{bmatrix} 43.0436 & 1.6473 & -4.5657 \\ 495.5321 & 30.0391 & -104.5870 \\ -2.3810 & 45.4193 & -3.5928 \\ -68.5365 & 535.7391 & -85.5120 \\ 6.1042 & 3.4747 & 44.9383 \\ 143.1120 & 80.4775 & 517.6262 \end{bmatrix} \tag{5.9}$$

The Simulink diagram is shown in figure 8. The observer and PI controllor are shown in figure 9 and 10 respectively.

We derive the controller from previous part and then tune the PI control system by changing $\mathbf{Q}$ and $\mathbf{R}$

$$\mathbf{Q} = \begin{bmatrix} 20 & 0 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 & 0 \\ 0 & 0 & 90 & 0 & 0 \\ 0 & 0 & 0 & 30 & 0 \\ 0 & 0 & 0 & 0 & 40 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.10}$$

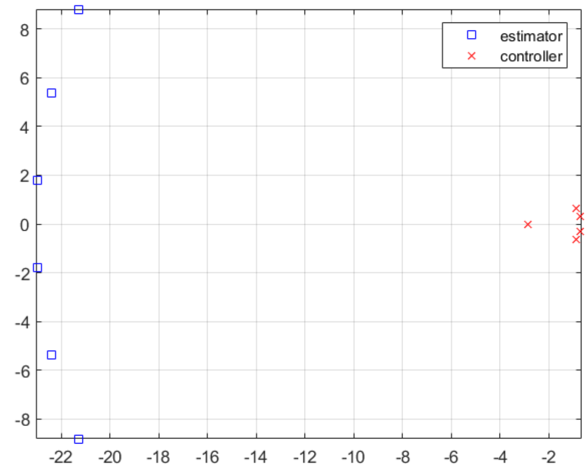

Figure 12: Poles of controller and estimator

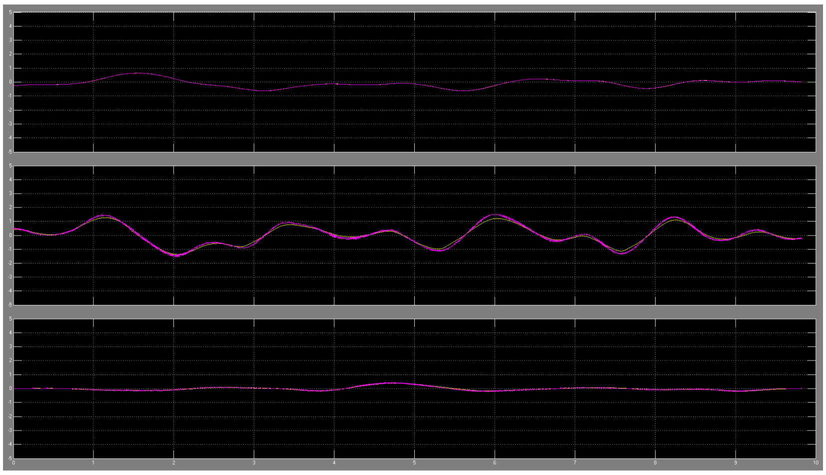

Figure 13: p, $\dot{p}$ and $\dot{e}$ comparison between measurements and estimations

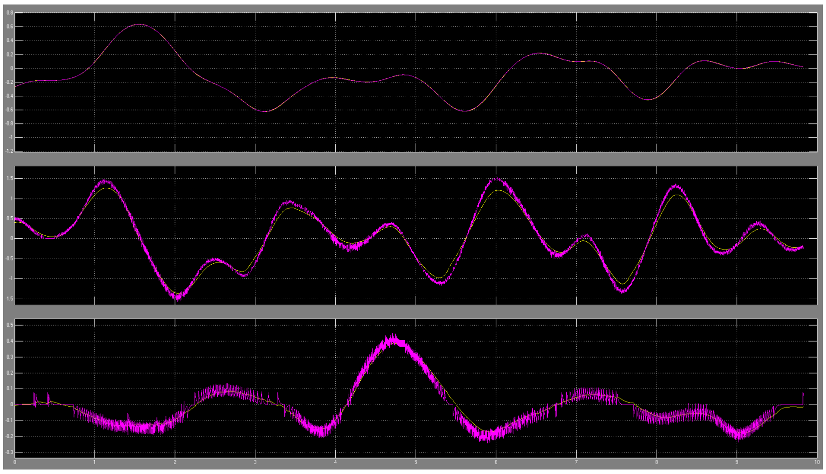

Figure 14: Zoom in view

5.3 Problem 3

In the last problem, we first calculate the Observability matrix for $\mathbf{y} = [\tilde{e}, \tilde{\lambda}]^T$ and $\mathbf{y} = [\tilde{p}, \tilde{e}]^T$.
For $\mathbf{y} = [\tilde{e}, \tilde{\lambda}]^T$, the $\mathbf{C}$ matrix is

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.11}$$

The observability matrix is:

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} & \mathbf{CA} & \mathbf{CA}^2 & \mathbf{CA}^3 & \mathbf{CA}^4 & \mathbf{CA}^5 \end{bmatrix}^T \tag{5.12}$$

The rank of the observability matrix $\mathcal{O}$ is 6, so the system is observable. We can use this measurement vector to construct our observer. For $\mathbf{y} = [\tilde{p}, \tilde{e}]^T$, the $\mathbf{C}$ matrix is

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{5.13}$$

The observability matrix is:

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} & \mathbf{CA} & \mathbf{CA}^2 & \mathbf{CA}^3 & \mathbf{CA}^4 & \mathbf{CA}^5 \end{bmatrix}^T \tag{5.14}$$

The rank of the observability matrix $\mathcal{O}$ is 4, so the system is not observable and thus $\mathbf{y} = [\tilde{p}, \tilde{e}]^T$ cannot be a measurement vector for observer. The reason is that we linearised the mathematical equation by approximating $\sin(p) = p$, $\cos(p) = 1$ and $\cos(e) = 1$ around $p = e = \lambda = 0$, which results in 2.14. This formula gives us a way of calculating $\tilde{p}$, i.e. $\ddot{\tilde{\lambda}} = K_3\tilde{p}$. While if we try to integrate $\tilde{\lambda}$, the lost information when differentiating can not be obtained. Therefore, it is not observable if one only measures $\tilde{p}$ and $\tilde{e}$.
We try to tune the control system by changing $\mathbf{Q}$ and $\mathbf{R}$ matrices . Here is our tuned parameters of $\mathbf{Q}$ and $\mathbf{R}$ matrices.

$$\mathbf{Q} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 90 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0.01 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} 250 & 0 \\ 0 & 250 \end{bmatrix} \tag{5.15}$$

Through the measurement vector $\mathbf{y} = [\tilde{e}, \tilde{\lambda}]^T$, we designed the observer and here is our Simulink diagram.

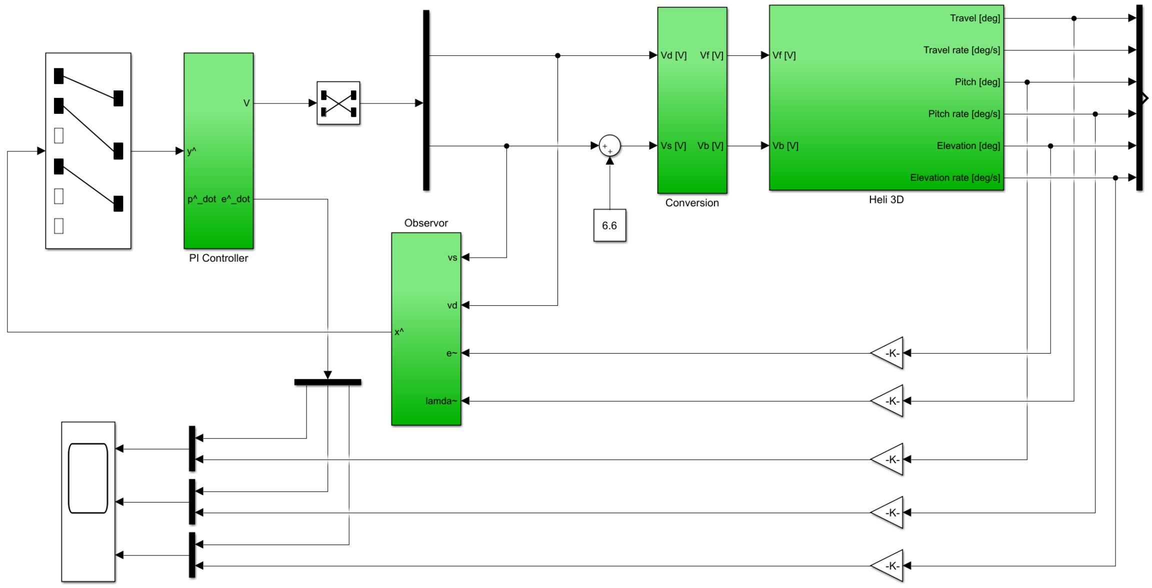

Figure 15: Closed loop simulation in Part IV, Problem 3

Figure 16 is our open-loop Simulink diagram. Through it, we can compare the deviation from the estimated and measured states and thus tune the controller more efficiently.

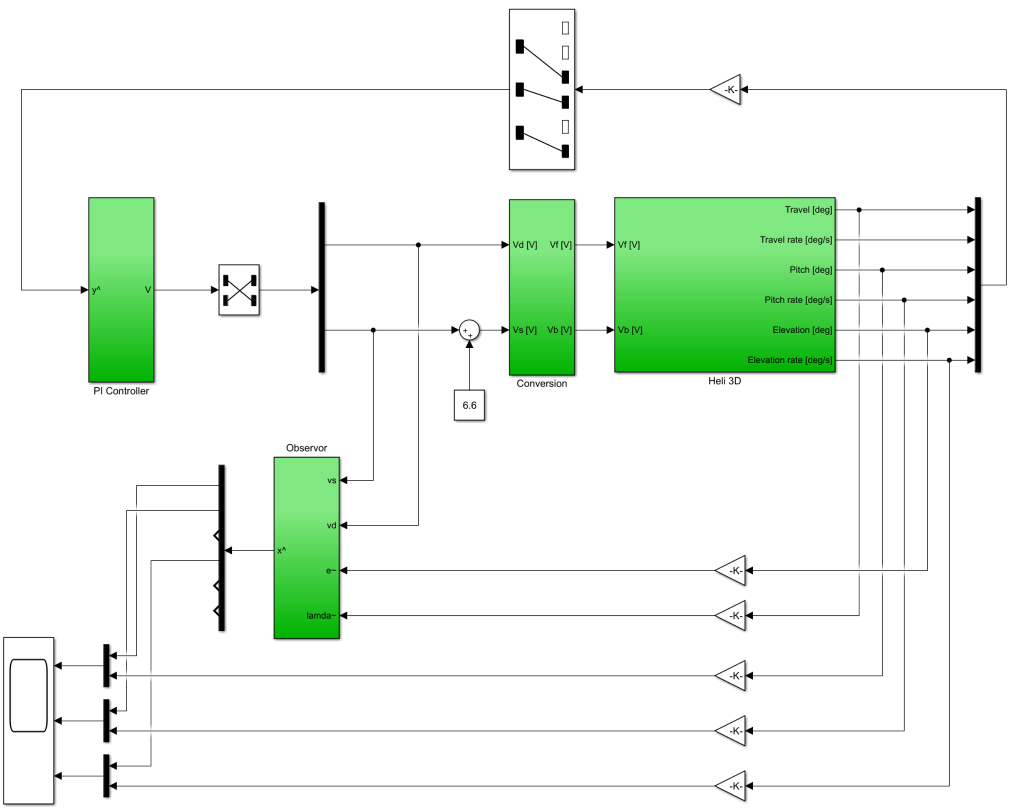

Figure 16: Open-loop simulation in Part IV, Problem 3

The result of open-loop simulation are shown below.

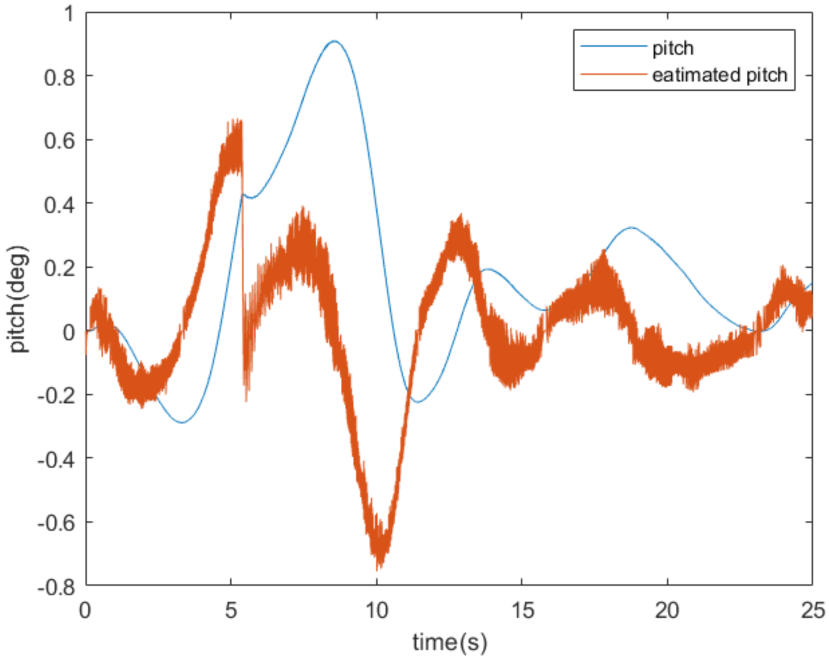

Figure 17: Open-loop pitch

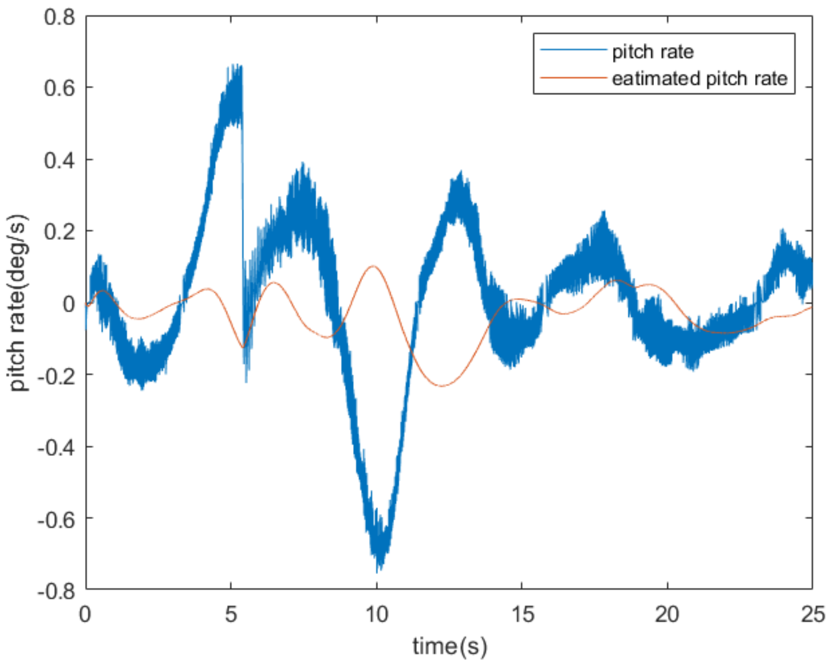

Figure 18: Open-loop pitch rate

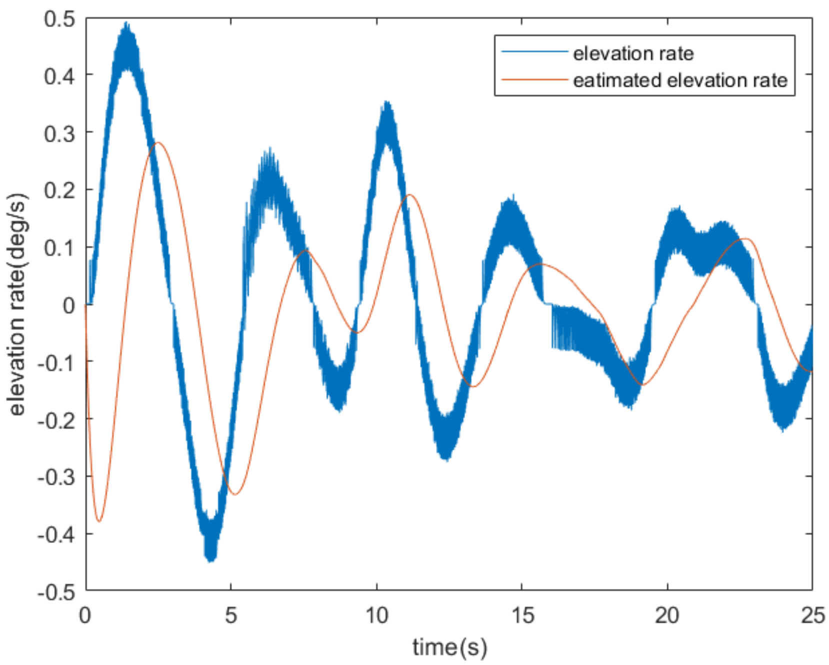

Figure 19: Open-loop elevation rate

Even though in theory it should be possible to control the helicopter by measuring only $\tilde{p}$ and $\tilde{e}$, it turned out to be very difficult. When we used the closed loop system, the controller was not able to regulate the pitch angle well because this state was not measured and in the end crashed.

When experimenting with the radius of poles, we found that large radius(20-30 times the distance to the most negative pole of the system itself) resulted in very large estimation of pitch angle and pitch rate. We believe this is because the travel, $\tilde{\lambda}$, which is polluted by noise, is differentiated two and three times to get the pitch angle and pitch rate. To differentiate noise is to amplify it, which gives very large magnitude of estimated pitch and pitch rate. Therefore we need to decrease the radius. Besides, we lowered the first two diagonal values to decrease this impact.

We also choose the last two diagonal value quite small to minimize the effect integral action. The reason is that the signal from observer is noisy while that from joystick is not. This makes integral effect largely influenced by noise. Hence, we decided to use very small values, in order to weaken integral effect.

To get better results, we tried to move one or two poles at a time (in the complex conjugated case), for example, to move the poles near to the real axis or to move poles at smaller circle arc. However, these tries do not make an obvious difference and the observer still has bad performance.

6 Conclusion

The control system for a 3-DOF helicopter has been developed using linear system theory. PID controllers, LQR and state estimator are applied in the control system with good performance. However, the control system is not perfect as in the last problem the helicopter is not controlled well because of the difficulty in finding suitable poles of the estimator.
From this project, group members have learned a lot both about using MATLAB and Simulink and understanding abstract concepts of linear system theory. It is interesting to tune different parameters and to observe how the helicopter behaves, from which group members is able to have an insight in these sophisticated theories and understand how they work.
Finally, we gratefully acknowledge assistance from lecturer and teaching assistants. It might be hard to obtain these results without their help since all of group members major in marine technology without cybernetics background.

7 MATLAB script

```
%%%%%%%%%%% Physical constants
g = 9.81; % gravitational constant [m/s^2]
l_c = 0.46; % distance elevation axis to counterweight [m]
l_h = 0.66; % distance elevation axis to helicopter head [m
   ]
l_p = 0.175; % distance pitch axis to motor [m]
m_c = 1.92; % Counterweight mass [kg]
m_p = 0.72; % Motor mass [kg]


%%%%%%%%%%% Moment of inertia
J_p = 2*m_p*l_p*l_p;  %pitch
J_e = m_c*l_c*l_c + 2*m_p*l_h*l_h; %elevation
J_t = m_c*l_c*l_c + 2*m_p*(l_h^2+l_p^2); %travel


%%%%%%%%%%% Sum of the motor voltage V_s0 at euqilibrium
   point, obtained
%%%%%%%%%%% by measurement
V_s0 = 6.6;


%%%%%%%%%%% Pamameters of linearized equations of motion
K_f = (2*m_p*l_h-m_c*l_c)/(V_s0*l_h)*g;
K_1 = K_f*l_p/J_p;
K_2 = K_f*l_h/J_e;
K_3 = (m_c*l_c-2*m_p*l_h)*g/J_t;

omega=2.5;
damping=1;
K_pd=2*omega*damping/K_1;
K_pp=omega^2/K_1;

%%%%%%%%%%% part 3.2
Q=[90,0,0;0,25 0;0,0,90];
R=[1 0;0 1];
A=[0 1 0;0 0 0;0 0 0];
B=[0 0;0 K_1; K_2 0];
K=lqr(A,B,Q,R);
C=[1 0 0;0 0 1];
P=inv(C*inv(B*K-A)*B);

%%%%%%%%%%% Another way of finding P, also applicable in
   part 3.3
% Paramaters
syms K_1 K_2 K_11 K_12 K_13 K_21 K_22 K_23;
```

```
% Matrix
B=[0 0;0 K_1;K_2 0];
K=[K_11 K_12 K_13;K_21 K_22 K_23];
A=[0 1 0;0 0 0;0 0 0];
% Computation
P_0=inv(B*K-A)*B; % Dimension of P_0: 3*2. With second row
   as zeros
P_0(2,:)=[]; % Delete second row
% Expression of P using elements of K
P=inv(P_0);


%%%%%%%%%%% part 3.3
Q=[90 0 0 0 0;0 25 0 0 0;0 0 90 0 0;0 0 0 50 0;0 0 0 0 40];
R=[1 0;0 1];
A=[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];
B=[0 0;0 K_1; K_2 0;0 0;0 0];
K=lqr(A,B,Q,R);
P=[K(1,1) K(1,3);K(2,1) K(2,3)]; % Expression of P using
   elements of K


%%%%%%%%%%% part 4.2
% Without integral effect
Q=[20 0 0 ;0 25 0 ;0 0 90 ];
R=[1 0;0 1];
A=[0 1 0;0 0 0;0 0 0];
B=[0 0;0 K_1; K_2 0];
K=lqr(A,B,Q,R);
P=[K(1,1) K(1,3);K(2,1) K(2,3)];
% With integral effect
Q=[20 0 0 0 0;0 25 0 0 0;0 0 90 0 0;0 0 0 30 0;0 0 0 0 40];
R=[1 0;0 1];
A=[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];
B=[0 0;0 K_1; K_2 0;0 0;0 0];
K=lqr(A,B,Q,R);
P=[K(1,1) K(1,3);K(2,1) K(2,3)]; % Expression of P using
   elements of K
A_0=[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;];
B_0=[0 0;0 K_1;0 0;K_2 0;0 0;0 0];
C_0=[1 0 0 0 0 0;0 0 1 0 0 0;0 0 0 0 1 0;];
D=eig(A);
O=[B A*B A*A*B A*A*A*B A*A*A*A*B A*A*A*A*A*B];
N=rank(O);

% Pole placement
es=eig(A-B*K);
```

```
r0=max(abs(es));
% Radial
fr=8;
% fr=100; % estimated states are poluted by noise
% fr=0.1; % observor is slower than controller (helicopter
   is not able to keep stable)
% Sector
phi=pi/8;
r=r0*fr;
spread=-phi:(phi/2.5):phi;
poles=-r*exp(1i*spread);
plot(real(poles),imag(poles),'sb',real(es),imag(es),'rx');
   grid on;axis equal;

L=place(transpose(A_0),transpose(C_0),poles).';


%%%%%%%%%%% part 4.3
Q=[3 0 0 0 0;0 3 0 0 0;0 0 90 0 0;0 0 0 0.01 0;0 0 0 0
   0.01];
R=[250 0;0 250];
A=[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];
B=[0 0;0 K_1; K_2 0;0 0;0 0];
K=lqr(A,B,Q,R);
P=[K(1,1) K(1,3);K(2,1) K(2,3)]; % Expression of P using
   elements of K
A_0=[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;];
B_0=[0 0;0 K_1;0 0;K_2 0;0 0;0 0];
C_0=[0 0 1 0 0 0;0 0 0 0 1 0;];
D=eig(A);
O=[B A*B A*A*B A*A*A*B A*A*A*A*B A*A*A*A*A*B];
N=rank(O);

% Pole placement
es=eig(A-B*K);
r0=max(abs(es));
% Radial
fr=8;
phi=pi/8;
r=r0*fr;
spread=-phi:(phi/2.5):phi;
poles=-r*exp(1i*spread);

%  To move one or two poles to get better results
poles1=poles;
poles1(1)=-1.5+0.25*i;poles1(6)=-1.5-0.25*i;
plot(real(poles1),imag(poles1),'rx');
```

```
L=place(transpose(A_0),transpose(C_0),poles1).';
```

References

[1] Chi-Tsong Chen, *Linear System Theory and Design*. Oxford University Press, international fourth edition, 2013.

[2] Kristoffer Gryte, *TTK4115 Helicopter lab assignment, volume 4.5*. Department of Engineering Cybernetics NTNU, August 2015.

[3] Morten D.Pedersen, *Lecture 5-State feedback*. Department of Engineering Cybernetics NTNU, August 2018.