Helicopter Lab report

TTK4115 - Linear System Theory

Group 31:
Øystein Brox - 741461
Morten Jansrud - 742003
Trym Nordal - 749720

Department of Engineering Cybernetics
NTNU
Norway
October 2016

Contents

Summary

The goal of the project is to control a mounted helicopter with a joystick using feed forward, monovariable control, LQR and state observers. The document will thoroughly derive mathematical models and discuss our choises for regulator optimizations.

1 Part 1 - Mathematical modeling

1.1 Problem 1 - Equations of motion

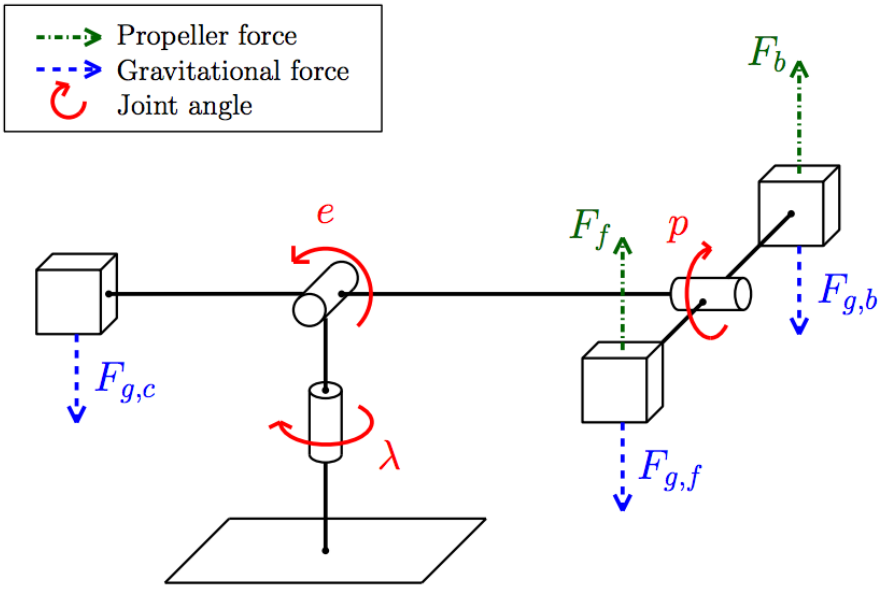

Figure 1: Helicopter model

Physics affecting the helicopter has to be defined in order to understand and control the system. Equations of motion are mathematically derived by using Newtons second law of rotation

$$\sum \tau = I\alpha$$

The helicopters ability to move forward and backward is determined by the pitch angle, which is formed by the difference in force applied on the front and the back motor. The motor forces are propotional with the applied voltage.

$$\sum \tau = F_f l_p - F_b l_p = K_f V_f l_p - K_f V_b l_p = K_f l_p V_d$$

$$J_p \ddot{p} = K_f l_p V_d = L_1 V_d \tag{1.1}$$

The elevation angle defines the lift of the helicopter. It is important to take into account gravity and that the elevation force from the motors are dependant on the pitch angle.

$$\begin{aligned}\sum \tau &= (F_{g,c} l_c - (F_{g,f} + F_{g,b}) l_h) \cos e + (F_f + F_b) l_h \cos p \\ &= g(m_c l_c - 2 m_p l_h) \cos e + K_f l_h V_s \cos p\end{aligned}$$

$$\begin{aligned}J_e \ddot{e} &= g(m_c l_c - 2 m_p l_h) \cos e + K_f l_h V_s \cos p \\ &= L_2 \cos e + L_3 V_s \cos p\end{aligned} \tag{1.2}$$

The travel angle is defined by the rotation around the Z-axis. The torque is dependant on the pitch and the elevation, since those angles affects the length of the arm.

$$\sum \tau = -(F_f + F_b) l_h \cos e \sin p = -K_f l_h V_s \cos e \sin p$$

$$J_\lambda \ddot{\lambda} = -K_f l_h V_s \cos e \sin p = L_4 V_s \cos e \sin p \tag{1.3}$$

1.2 Problem 2 - Linearization around a point

Linearization of the mathematical model is necessary to simplify the problem and become able to control the system with the tools provided. The linearization will be around the point $(p, e, \lambda)^T = (p^*, e^*, \lambda^*)^T$, with $p^* = e^* = \lambda^* = 0$, we have $(\dot{p}, \dot{e}, \dot{\lambda})^T = (0, 0, 0)^T$, $(\ddot{p}, \ddot{e}, \ddot{\lambda})^T = (0, 0, 0)^T$. Using 1.1 and 1.2 we find $(V_s^*, V_s^*)^T$:

$$1.1: \quad J_p \cdot 0 = L_1 V_d^* \quad \Rightarrow \quad V_d^* = 0$$

$$1.2: \quad J_e \cdot 0 = L_2 \cos 0 + L_3 V_s^* \cos 0 \quad \Rightarrow \quad V_s^* = -\frac{L_2}{L_3}$$

$$\begin{bmatrix} V_s^* \\ V_d^* \end{bmatrix} = \begin{bmatrix} -\frac{L_2}{L_3} \\ 0 \end{bmatrix} \tag{1.4}$$

Next, with the coordinate transform

$$\begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} = \begin{bmatrix} p \\ e \\ \lambda \end{bmatrix} - \begin{bmatrix} p^* \\ e^* \\ \lambda^* \end{bmatrix}$$

$$\begin{bmatrix} \tilde{V_s} \\ \tilde{V_d} \end{bmatrix} = \begin{bmatrix} V_s \\ V_d \end{bmatrix} - \begin{bmatrix} V_d^* \\ V_s^* \end{bmatrix}$$

the system can be rewritten as

$$\ddot{\tilde{p}} = \frac{L_1}{J_p}\tilde{V_d} \tag{1.5}$$

$$\ddot{\tilde{e}} = \frac{L_2}{J_e}\cos\tilde{e} + \frac{L_3}{J_e}\left(\tilde{V_s} - \frac{L_2}{L_3}\right)\cos\tilde{p} \tag{1.6}$$

$$\ddot{\tilde{\lambda}} = \frac{L_4}{J_\lambda}(\tilde{V_s} - \frac{L_2}{L_3})\cos\tilde{e}\sin\tilde{p} \tag{1.7}$$

Linearization of a system on the form

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

around equilibrium $\mathbf{x}_0$ and $\mathbf{u}_0$ is done by following the formula

$$\dot{\mathbf{x}} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \end{bmatrix}\right|_{\mathbf{x}_0,\mathbf{u}_0} \mathbf{x} + \left.\begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \frac{\partial g_1}{\partial u_2} \\ \frac{\partial g_2}{\partial u_1} & \frac{\partial g_2}{\partial u_2} \\ \frac{\partial g_3}{\partial u_1} & \frac{\partial g_3}{\partial u_2} \end{bmatrix}\right|_{\mathbf{x}_0,\mathbf{u}_0} \mathbf{u}$$

After putting 1.5 - 1.7 into this formula we are left with

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{L_4}{J_\lambda}V_s^* & 0 & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 & \frac{L_1}{J_e} \\ \frac{L_3}{J_e} & 0 \\ 0 & 0 \end{bmatrix}\mathbf{u}$$

$$\Rightarrow \quad \ddot{\tilde{p}} = K_1\tilde{V_d} \tag{1.8}$$

$$\Rightarrow \quad \ddot{\tilde{e}} = K_2\tilde{V_s} \tag{1.9}$$

$$\Rightarrow \quad \ddot{\tilde{\lambda}} = K_3\tilde{p} \tag{1.10}$$

where $K_1 = \frac{L_1}{J_p}$, $K_2 = \frac{L_3}{J_e}$ and $K_3 = \frac{L_4}{J_\lambda}V_s^*$.

1.3 Problem 3 - Feed forward

The helicopter was not easy to steer when using feed forward control from the joystick to the voltage input without any manipulation. Multiplying the voltage sum by 8 made the helicopter respond faster and become easier to control, but it was still difficult.

A mathematical model will be a simplified version of reality. The helicopters physical behavior is, however, very much alike what we anticipated based on our theoretical model. Some discrepancies exists due to external physical parameters which was not included in our model, e.g. drag and friction.
The corresponding Simulink model is shown in figure 9 in Appendix B.

1.4 Problem 4 - Helicopter at equilibrium

V_s^* was found to be 7V by measurement. K_f can be solved by inserting the values for L_2, L_3 and V_s^* into 1.4

$$K_f = -\frac{g(m_c l_c - 2m_p l_p)}{l_h V_s^*} = 0,1427 \tag{1.11}$$

The expanded Simulink model is shown in figure 10 in Appendix B.

2 Part 2 – Monovariable control

2.1 Problem 1 - Pitch PD controller

A PD controller will be added in order to control the pitch angle

$$\tilde{V_d} = K_{pp}(\tilde{p_c} - \tilde{p}) - K_{pd}\dot{\tilde{p}} \tag{2.1}$$

The Simulink model of the pitch controller is showed in figure 12. This controller together with the elevation controller is implemented in the Simulink model as showed in figure 11. Both figures are found in Appendix B.

The transferfunction for $\frac{\tilde{p}(s)}{\tilde{p_c}(s)}$ is found by substituting 2.1 with $\tilde{V_d}$ in 1.8 and Laplace transforming:

$$\begin{aligned} & \ddot{\tilde{p}} = K_1(K_{pp}(\tilde{p_c} - \tilde{p}) - K_{pd}\dot{\tilde{p}}) \\ \Rightarrow \quad & \tilde{p}(s^2 + K_1 K_{pd} s + K_1 K_{pp}) = \tilde{p_c}(K_1 K_{pp}) \\ \Rightarrow \quad & \frac{\tilde{p}(s)}{\tilde{p_c}(s)} = \frac{K_1 K_{pp}}{s^2 + K_1 K_{pd} s + K_1 K_{pp}} \end{aligned} \tag{2.2}$$

When determining the poles of this transferfunction with regards to K_1 and K_{pd} it is preferred to have one pole with only a negative real part, making the system

critically damped. This is achieved by writing the transferfunction on the form $\frac{K\omega_0^2}{s^2+2\zeta\omega_0 s+\omega_0^2}$ and setting the damping ratio $\zeta = 1$. The regulator parameters can be written as $K_{pp} = \frac{\omega_0^2}{K_1}$ and $K_{pd} = 2\frac{\sqrt{K_1 K_{pp}}}{K_1}$. Increasing and tuning the natural frequency ω_0 to make the system faster gave us

$$\begin{aligned} K_{pp} &= 11.04 \\ K_{pd} &= 8.83 \end{aligned} \tag{2.3}$$

A plot of the step response of the system with these values are shown in figure 2 in Appendix A.

2.2 Problem 2 - Travel rate P controller

The travel rate $\dot{\bar{\lambda}}$ is to be controlled with a P controller

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

where $K_{rp} < 0$. Assuming $\tilde{p} = \tilde{p}_c$ and using 1.10, we get

$$\frac{\ddot{\bar{\lambda}}}{K_3} = K_{rp}(\dot{\bar{\lambda}}_c - \dot{\bar{\lambda}}) \quad \overset{\mathcal{L}}{\Rightarrow} \quad s\dot{\bar{\lambda}} = K_3 K_{rp}(\dot{\bar{\lambda}}_c - \dot{\bar{\lambda}}) \quad \Rightarrow \quad \frac{\dot{\bar{\lambda}}}{\dot{\bar{\lambda}}_c} = \frac{K_3 K_{rp}}{s + K_3 K_{rp}} \tag{2.4}$$

This can be written as

$$\frac{\dot{\bar{\lambda}}(s)}{\dot{\bar{\lambda}}_c(s)} = \frac{\rho}{s+\rho} \tag{2.5}$$

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

By including the P-controller the travel rate would reach a constant value based on the reference. A joystick gain of 0.8 and $K_{rp} = -1$ gave a fast and accurate response.

The new Simulink model is shown in figure 13 in Appendix B. A plot comparing the input reference form the joystick (x) and measured travel rate $\dot{\lambda}$ is shown in figure 3 in Appendix A.

2.3 Discussion

The regulator response is dependent on the poles chosen. In our case we wanted to create a critically damped system which has one pole with a negative real part. Had we chosen to include an imaginary part in our poles, we could expect an underdamped system where the regulator for the pitch did an overshoot when stabilizing. On the other hand, if we had two separate poles with only real parts, the system would have been overdamped reaching the desired value slower.

3 Part 3 - Multivariable control

3.1 Problem 1 - System of equations

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{3.1}$$

$$\boldsymbol{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \end{bmatrix} \qquad \boldsymbol{u} = \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{3.2}$$

With the state vector and the input vector in 3.2 in a state space formulation 3.1, **A** and **B** is

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{B} = \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \end{bmatrix} \tag{3.3}$$

Thus the state space is

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \ddot{\tilde{e}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \dot{\tilde{e}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ K_2 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix} \tag{3.4}$$

3.2 Problem 2 - Linear quadratic regulator

The controllability-matrix is obtained by

$$\boldsymbol{\mathcal{C}} = \begin{bmatrix} B & AB & A^2B & ... & A^{n-1}B \end{bmatrix}$$

Using **A** and **B** from 3.3 the resulting controllabiliy-matrix is

$$\boldsymbol{\mathcal{C}} = \begin{bmatrix} 0 & 0 & 0 & K_1 & 0 & 0 \\ 0 & K_1 & 0 & 0 & 0 & 0 \\ K_2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Since $rank(\mathcal{C}) = 3$ the system is controllable. Next, the controller on the form $\mathbf{u} = \mathbf{Pr} - \mathbf{Kx}$ is implemented. **r** is the reference, which will be input from the joystick, and **K** corresponds to the LQR. The control input $\mathbf{u} = -\mathbf{Kx}$ optimizes the cost function

$$J = \int_0^\infty (\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t))\, dt \tag{3.5}$$

Q and **R** are diagonal weighting matrices. **Q** and **R** were found with Brysons rule as a starting point. Brysons rule state the following

$$Q_{ii} = \frac{1}{maximum(x_i^2)} \quad and \quad R_{ii} = \frac{1}{maximum(u_i^2)} \tag{3.6}$$

That gave the following values og $\mathbf{Q}$ and $\mathbf{R}$

$$\boldsymbol{Q} = \begin{bmatrix} 91.2 & 0 & 0 \\ 0 & 10.1 & 0 \\ 0 & 0 & 91.2 \end{bmatrix} \qquad \boldsymbol{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.7}$$

However after some scaling and tuning of the practical system, these matrices gave the fastest and most accurate response

$$\boldsymbol{Q} = \begin{bmatrix} 91.2 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 100 \end{bmatrix} \qquad \boldsymbol{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.8}$$

$\mathbf{K}$ was obtained by the MATLAB command lqr(A,B,Q,R)

$$\boldsymbol{K} = \begin{bmatrix} 0 & 0 & 10 \\ 9.55 & 9.15 & 0 \end{bmatrix} \tag{3.9}$$

The $\mathbf{P}$ matrix is now chosen such that $\lim_{t\to\infty} \tilde{p}(t) = \tilde{p}_c$ and $\lim_{t\to\infty} \dot{\tilde{e}}(t) = \dot{\tilde{e}}_c$. With these conditions we get $\dot{x}_\infty = 0$ and $y_\infty = r_0$. $\mathbf{P}$ is found by solving the equations for the system as time approaches infinity

$$0 = \mathbf{A}\mathbf{x}_\infty + \mathbf{B}(\mathbf{P}\mathbf{r}_\infty - \mathbf{K}\mathbf{x}_\infty) \quad \Rightarrow \quad \mathbf{x}_\infty = (\mathbf{B}\mathbf{K} - \mathbf{A})^{-1}\mathbf{B}\mathbf{P}\mathbf{r}_\infty$$

$$\mathbf{r}_\infty = \mathbf{C}(\mathbf{B}\mathbf{K} - \mathbf{A})^{-1}\mathbf{B}\mathbf{P}\mathbf{r}_\infty \quad \Rightarrow \quad \mathbf{P} = (\mathbf{C}(\mathbf{B}\mathbf{K} - \mathbf{A})^{-1}\mathbf{B}\mathbf{P})^{-1} \tag{3.10}$$

After solving 3.10 in MATLAB, the values of $\mathbf{P}$ are found to be

$$\boldsymbol{P} = \begin{bmatrix} 0 & 10 \\ 9.55 & 0 \end{bmatrix}$$

In this controller design Brysons rule was used as a starting point. After physically steering the helicopter we found that some of the states needed tuning. The pitch rate and the elevation rate was increased to get a faster response in the two states. Pitch rate and elevation rate were the hardest states to use with Brysons rule, as you must know the maximum acceptable value of the states. These maximum values were guesstimated. After tuning, the values of 3.8 worked well. The Simulink model of this system is showed in figure 14 in Appendix B.

3.3 Problem 3 - The integral effect

Two extra states are needed by adding an integral effect

$$\begin{aligned} \dot{\gamma} &= \tilde{p} - \tilde{p_c} \\ \dot{\zeta} &= \dot{\tilde{e}} - \dot{\tilde{e}_c} \end{aligned} \tag{3.11}$$

$$\boldsymbol{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \gamma \\ \zeta \end{bmatrix} \quad and \quad \boldsymbol{u} = \begin{bmatrix} \tilde{V_s} \\ \tilde{V_d} \end{bmatrix} \tag{3.12}$$

The new matrixes and model was implemented in Simulink as shown in appendix B and figure 15.

3.4 Discussion

The linear quadratic regulator (LQR) makes it easy to tune each state individually to get the desired response from the system. The regulator gain K are proportional with the ratio between the weighting matrices Q and R, so multiplying both matrixes will not affect the system.

By adding the integrating states in problem 3 the steady state error of the pitch is removed and the elevation reach the desired value.

4 Part 4 - State estimation

4.1 Problem 1 - State-space formulation

In this section the states were estimated instead of measured as in the previous parts. The systems from 1.5 - 1.7 can be derived by a state-space formulation of the form

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \end{aligned} \tag{4.1}$$

$$\boldsymbol{x} = \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} \tilde{V_s} \\ \tilde{V_d} \end{bmatrix}, \qquad \boldsymbol{y} = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} \tag{4.2}$$

4.2 as state vector, input vector and output vector gives

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\tilde{p}} \\ \ddot{\tilde{p}} \\ \dot{\tilde{e}} \\ \ddot{\tilde{e}} \\ \dot{\tilde{\lambda}} \\ \ddot{\tilde{\lambda}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ K_3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & K_1 \\ 0 & 0 \\ K_2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_s \\ \tilde{V}_d \end{bmatrix}$$

$$\boldsymbol{y} = \begin{bmatrix} \tilde{p} \\ \tilde{e} \\ \tilde{\lambda} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \tilde{p} \\ \dot{\tilde{p}} \\ \tilde{e} \\ \dot{\tilde{e}} \\ \tilde{\lambda} \\ \dot{\tilde{\lambda}} \end{bmatrix}$$

4.2 Problem 2 - Linear observer

The observability of the system were obtained thorugh the MATLAB function

```
O = obsv(A,B)
```

This matrix has *rank = 6* which means that the system is observable A linear observer is created on the form

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{L}(\boldsymbol{y} - \boldsymbol{C}\hat{\boldsymbol{x}}) \tag{4.3}$$

The closed loop observer poles (eigenvalues of (**A-LC**)) can be chosen as preferred because the system is observable. The placement of the poles does influence the systems behavior, which is important to keep in mind when choosing the poles. The observer should be faster than the system (**A-BK**), and a general rule of thumb is to have the poles of the observer to converge around ten times faster than the system. The way to achieve this is to let the poles of the observer have large negative real values. The **L** matrix was found by using the MATLAB-function

```
L = place(A',C',poles)
```

See figure 4 for plot with the first controller from part III and figure 5 for the controller with integral effect. The Simulink model of the observer is showed in 17 and its implemented in the rest of the Simulink model in figure 16.

4.3 Problem 3 - Observer without pitch

Observability of a system depends on which states that are measured. Again using the MATLAB function as in problem 2 the observability matrix when measuring $\tilde{e}$ and $\tilde{\lambda}$ is found to have $rank = 6$, proving that it is observable. However, when measuring $\tilde{p}$ and $\tilde{e}$ the observability matrix only have $rank =$

4, thus it is not observable. A state observer based on the measurements in equation 4.4 is being made

$$\boldsymbol{y} = \begin{bmatrix} \tilde{e} \\ \tilde{\lambda} \end{bmatrix} \tag{4.4}$$

Plot of measurement vs estimation is given in figure 8.

4.4 Discussion

4.4.1 Problem 4.2

After some testing it was found that our observer worked well when the poles were spread in a fan shape with a radius of almost 80 (figure 6). This is 20 times the value of the largest eigenvalue of (**A-BK**). We also tried with only poles on the real axis (firgure 7). There was no noticable difference in the behavior between these two pole placements.

4.4.2 Problem 4.3

Even though the system is observable when $\tilde{e}$ and $\tilde{\lambda}$ are measured, it does not imply that the helicopter can be controlled well using an observer. The problem occurs when you try to control the travel which is dependant on the pitch, an unmeasured state.

A smooth travel rate is predictable and makes it possible to control the helicopter. In the case of rapid changes in the travel angle or sudden stop, the pitch estimation goes way off the actual measurement, the ability to control the helicopter breaks, it self-accelerates and in most cases crashes.

The pitch estimations appears to be more inaccurate the faster you try to to sample a state, and limitations to the observer has to be implemented to make the system stable. After testing we conclude that the observer made the most accurate estimations by placing poles on the real axis, thus making the oberserver over damped.

The L matrix in the observer scales the deviation between the measured and estimated states.

Based on the plots, the elevation estimation seems to be working correctly as it has very few deviations from the actual measurement. The reason is because we use the elevation measurement itself to create estimations.

5 Appendix A - Plots

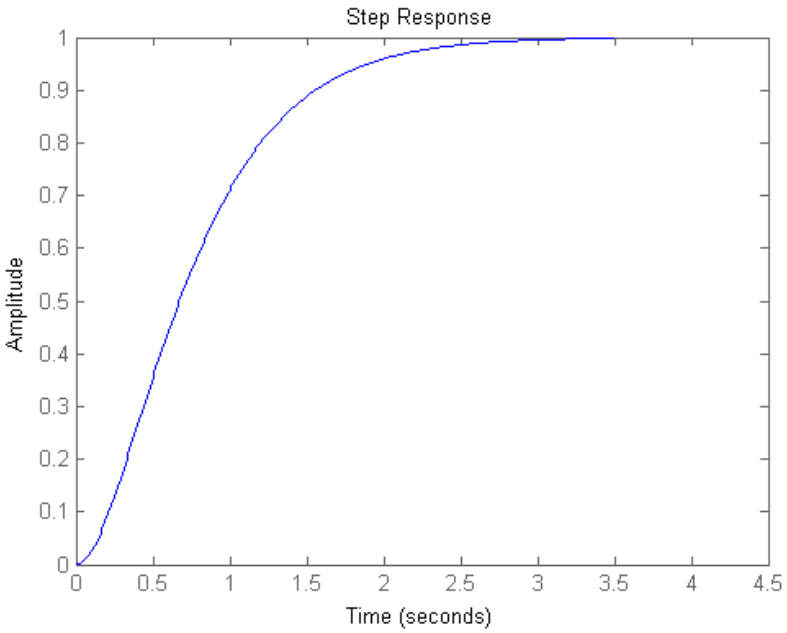

Figure 2: Step Response of the system in Part 1, problem 2

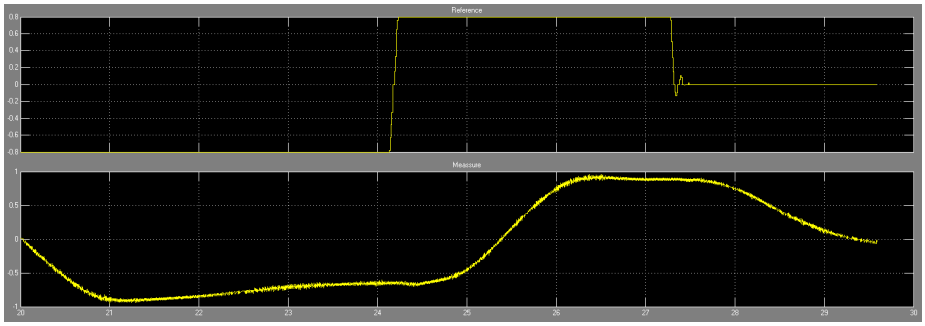

Figure 3: Comparison of the input reference x and the measured travel rate $\dot{\lambda}$. The reference is on top.

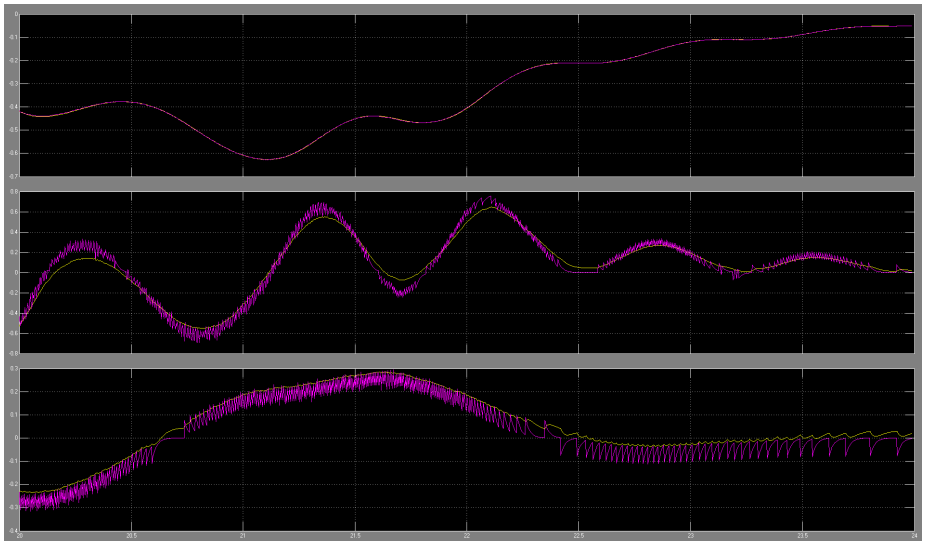

Figure 4: This shows the controller values in Part 4, problem 2 with the first controller. From the top: p, $\dot{p}$ and $\dot{e}$, the estimated value is yellow and measured is purple.

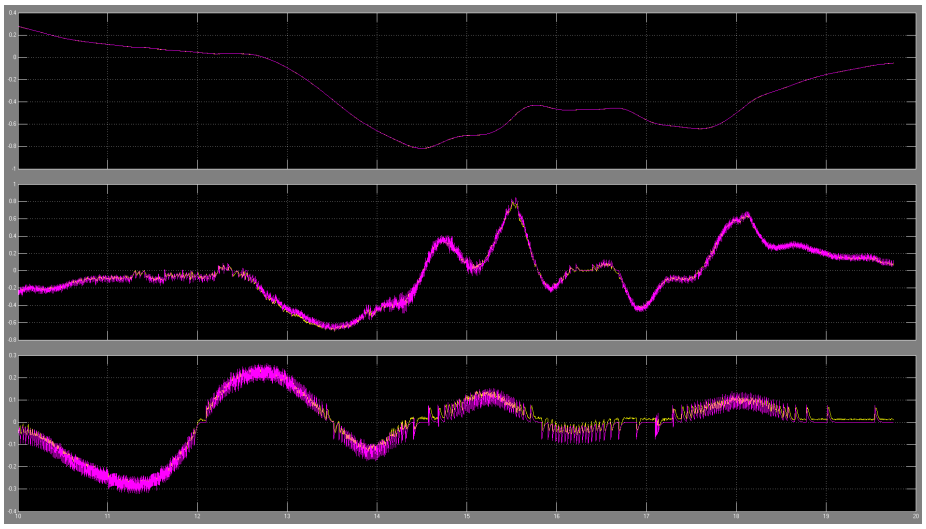

Figure 5: This shows the controller values in Part 4, problem 2 with the second controller. From the top: p, $\dot{p}$ and $\dot{e}$, the estimated value is yellow and measured is purple.

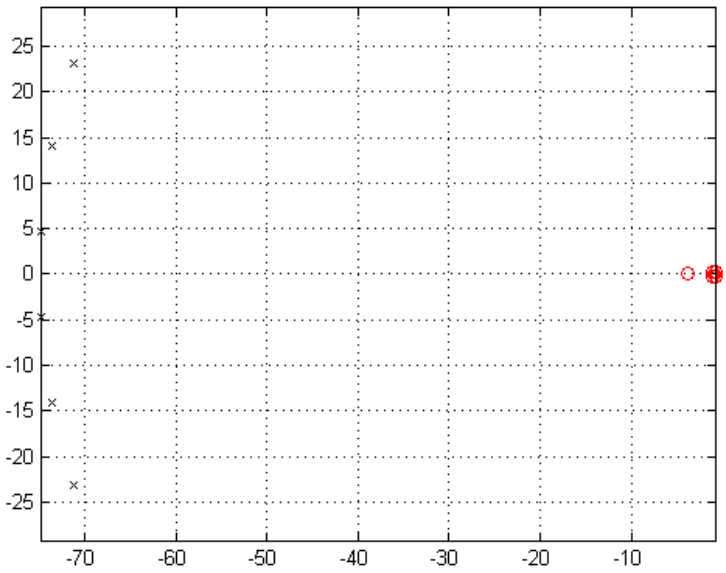

Figure 6: Poles spread with r=8

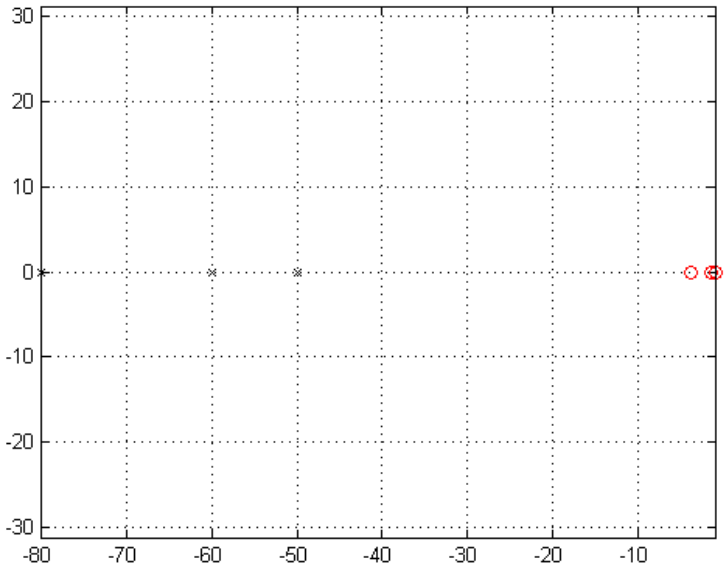

Figure 7: Poles with only real parts

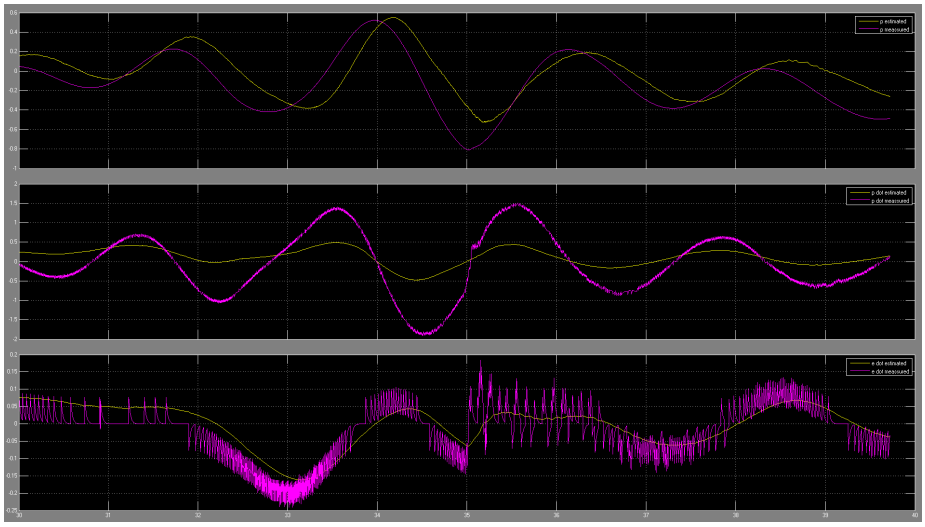

Figure 8: This shows the controller values in Part 4, problem 3. From the top: p, $\dot{p}$ and $\dot{e}$, the estimated value is yellow and measured is purple.

6 Appendix B - Simulink Models

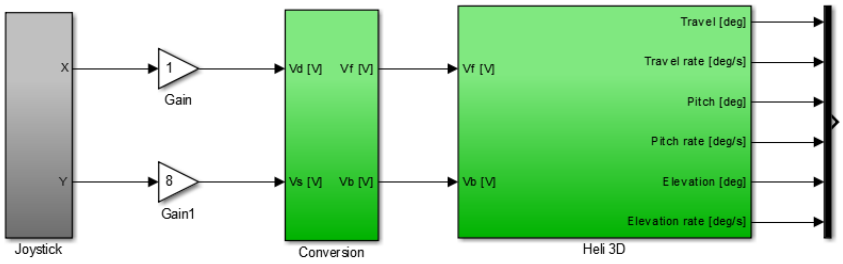

Figure 9: Part 1 - Problem 3

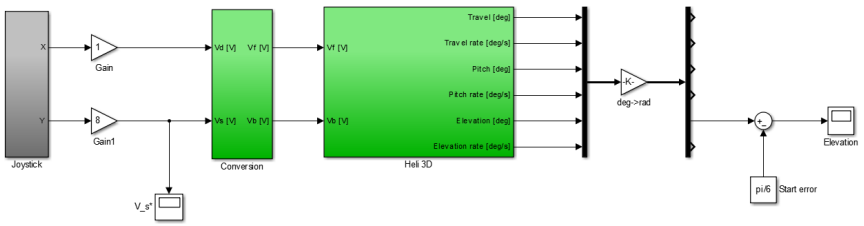

Figure 10: Part 1 - Problem 4

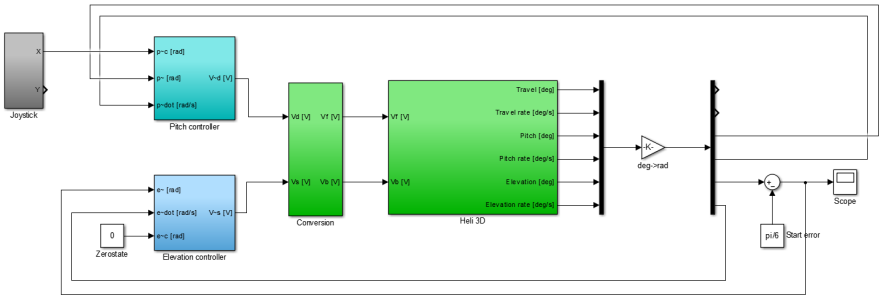

Figure 11: Part 2 - Problem 1

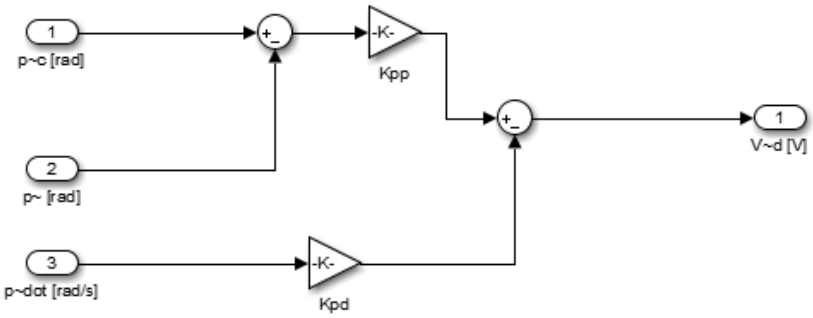

Figure 12: The pitch controller. First used in Part 2 - Problem 1

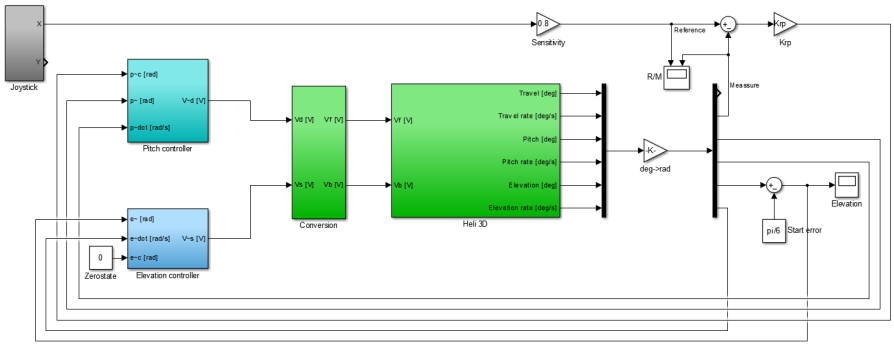

Figure 13: Part 2 - Problem 2

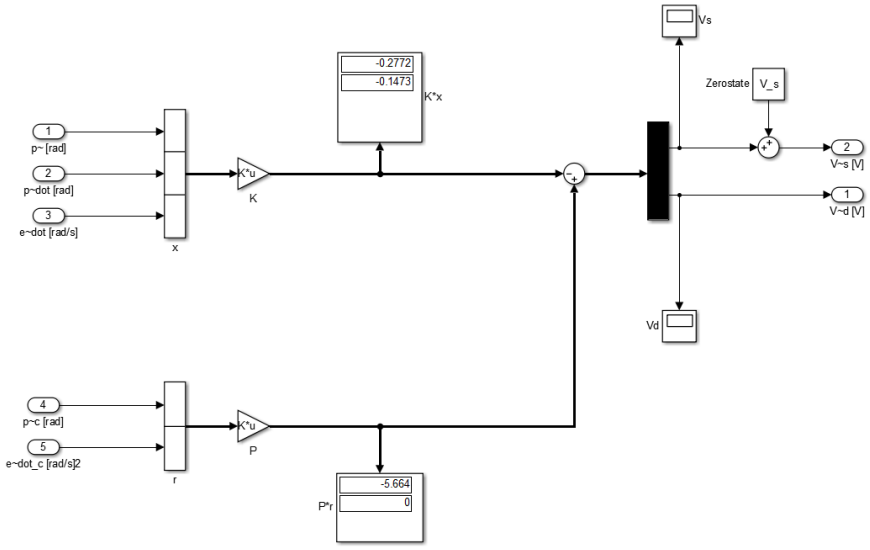

Figure 14: The LQR in Part 3 - Problem 2

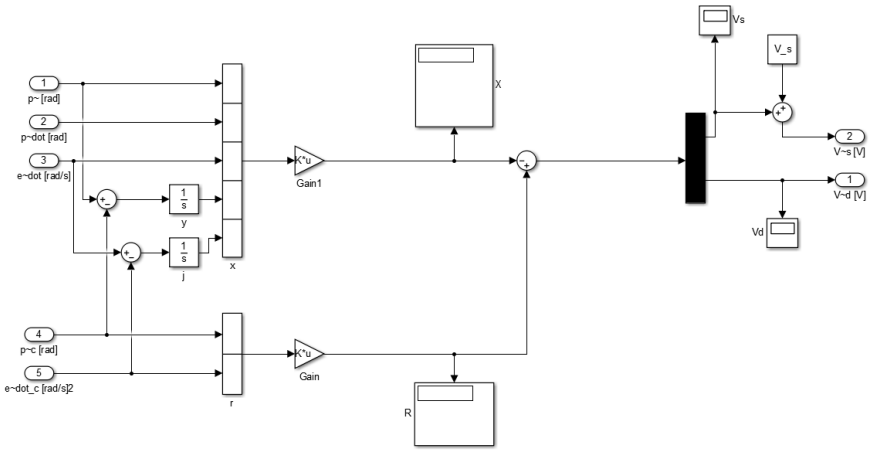

Figure 15: The LQR in Part 3 - Problem 3

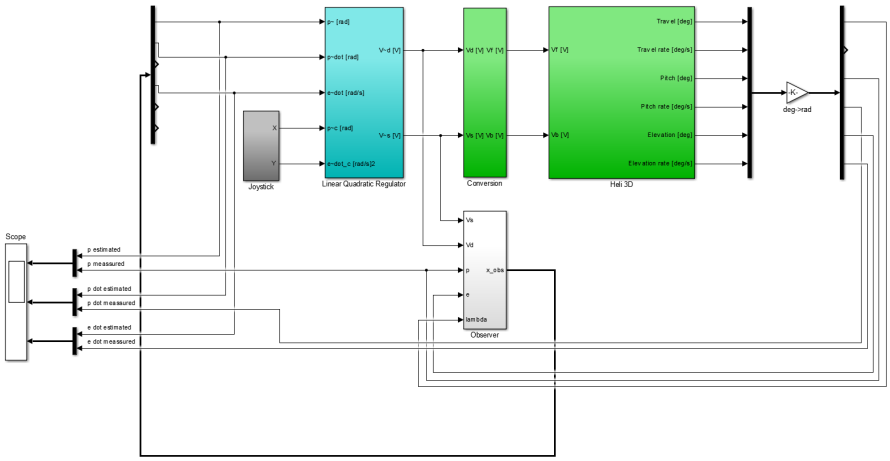

Figure 16: Part 4 - Problem 2

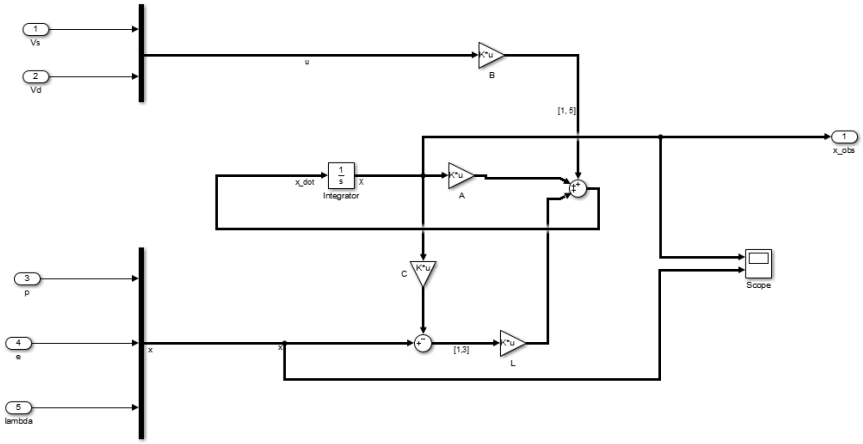

Figure 17: The observer block, first used in Problem 4 - Part 2

7 Appendix C - MATLAB script

```
%%%%%%%%%%% Calibration of the encoder and the hardware for the specific
%%%%%%%%%%% helicopter
Joystick_gain_x = 1;
Joystick_gain_y = -1;
```

PART I

```
%%%%%%%%%%% 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]

V_s = 7;
K_f = -(g*(m_c*l_c-2*m_p*l_h))/(V_s*l_h);
```

PART II

Problem 1

```
w_0 = 2.5;

L_1 = K_f*l_p;
L_2 = g*(m_c*l_c-2*m_p*l_h);
L_3 = K_f*l_h;
L_4 = -L_3;

J_p = 2*m_p*(l_p)^2;
J_e = m_c*(l_c)^2 + 2*m_p*(l_h)^2;
J_y = m_c*(l_c)^2 + 2*m_p*((l_h)^2+(l_p)^2);

K_1 = L_1/J_p;
K_2 = L_3/J_e;
K_3 = L_4*V_s/J_y;

%%%%%%%%%%% Regulator paramaters - Method 1 - w0 og critical damping
Kpp = (w_0)^2/K_1;
Kpd = 2*sqrt(K_1*Kpp)/K_1;

% Problem 2

Krp = -1;
```

PART III

Problem 2

```
A = [0 1 0; 0 0 0; 0 0 0];
B = [0 0; 0 K_1; K_2 0];
Q = [91.2 0 0; 0 50 0; 0 0 100];
R = [1 0; 0 1];
C = [1 0 0; 0 0 1];
K = lqr(A,B,Q,R);
P = inv(C*inv(-A+B*K)*B);

% Problem 3

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];
Q =  [91.2 0 0 0 0; 0 50 0 0 0; 0 0 100 0 0; 0 0 0 50 0; 0 0 0 0 50];
R =  [1 0; 0 1];
C =  [1 0 0 0 0; 0 0 1 0 0];
K =  lqr(A,B,Q,R);
P =  [0 9.999999999999998;9.549869109050656 0];
```

PART IV

Problem 2

```
%%%%%%%%%%%%%% Using regulator design from part III problem 2
A_o =  [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_o =  [0 0; 0 K_1; 0 0; K_2 0; 0 0; 0 0];
C_o =  [1 0 0 0 0 0; 0 0 1 0 0 0; 0 0 0 0 1 0];

%Found by testing
%poles = [-80,-80,-70+38i,-70-38i,-60+50i,-60-50i];

%Pole spreading
es = eig(A-B*K);
r0 = max(abs(es));

fr = 20;
phi = pi/10;
r = r0*fr;

spread = -phi:(phi/(2.5)):phi;
poles = -r*exp(1i*spread);
```

```
plot(real(es),imag(es),'or',real(poles),imag(poles),'kx');grid on;axis equal
L_o = place(transpose(A_o), transpose(C_o),poles).';

% Problem 3
%%%%%%%%%%%%%% Using regulator design from part III problem 2
poles = [-8,-8,-5,-20,-1,-11];
L_o = place(transpose(A_o), transpose(C_o),poles).';
```

References

[1] *Helicopter lab assignment.* Itslearning, 2015.

[2] J. P. Hespanha. *Linear Systems Theory.* Princeton University Press, 2009.

[3] Wikipedia. State observer. URL `https://en.wikipedia.org/wiki/State_observer`.