# Control System Design for Continuous-Flow Stirred Tank Reactor

## EE5101/ME5401 Linear Systems Mini-Project

Praveen Kumar Krishnapur
Matriculation Number: A0304688U

Email: e1373776@u.nus.edu

November 17, 2024

# Abstract

This report investigates the design and simulation of control systems for a Continuous-Flow Stirred Tank Reactor (CSTR) using various methods, MATLAB and Simulink. Pole placement via the full-rank method achieved stable output decay, with faster responses from poles placed further from the origin. An LQR controller highlighted the trade-offs between control speed and energy cost through tuning $Q$ and $R$ matrices. A state observer was designed to estimate unmeasured states, quickly eliminating estimation errors with appropriate pole placement. Decoupling control enabled independent manipulation of outputs, ensuring stability and simplified operations. Integral control combined with a state observer provided effective reference tracking and disturbance rejection, using an augmented state-space model to minimize sensor requirements. While manipulating all three state variables directly was infeasible with two inputs, minimizing a weighted objective function produced results close to the desired steady state. These methods demonstrate practical insights into robust and efficient control system design.

# Contents

# 1 Introduction

The CSTR problem involves controlling a chemical reaction process in which a reactant is converted into a product within a tank. Due to the high cost and installation challenges of component concentration sensors, the system is limited to using two temperature sensors to measure the reaction and cooling jacket outflow temperatures. This project aims to control the reactant concentration indirectly using available temperature data, ensuring cost-efficiency in small-scale industrial setups.

My matriculation number is A0304688U; thus, using the last four digits, we set $a = 4$, $b = 6$, $c = 8$, and $d = 8$. This gives us the following matrices:

$$A = \begin{bmatrix} -1.7 & -0.25 & 0 \\ 23 & -30 & 20 \\ 0 & -660 & -860 \end{bmatrix}, \quad B = \begin{bmatrix} 7 & 0 \\ -118 & 0 \\ 0 & -1300 \end{bmatrix}$$

The measurement matrix $C$ and initial condition for the system are derived from the problem description:

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad x_0 = \begin{bmatrix} 1 \\ 100 \\ 200 \end{bmatrix}$$

The state-space representation for the CSTR system is given by:

$$\dot{x} = Ax + Bu + Bw, \quad y = Cx$$

where $x$ represents the state vector, $u$ the control input vector, and $w$ the load disturbance vector.

The time domain specifications for the system are as follows:

- The overshoot should be less than 10
- The 2% settling time should be less than 30 seconds.

To meet the time domain specifications, we calculate the damping ratio $\zeta$ and natural frequency $\omega_n$. The conditions are based on the overshoot and settling time requirements.

The overshoot $M_p$ is related to the damping ratio $\zeta$ by:

$$M_p = e^{\left(-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}\right)} < 0.1$$

Solving for $\zeta$, we find:

$$\zeta > 0.591$$

For my calculations I assume

$$\zeta = 0.8$$

. The 2% settling time $T_s$ is defined as:

$$T_s = \frac{4}{\zeta\omega_n} < 30 \text{ seconds}$$

Substituting $\zeta = 0.8$, we find:

$$\omega_n > \frac{4}{30 \times 0.8} \approx 0.167 \text{ rad/s}$$

Therefore, I choose the values as $\zeta = 0.8$ and $\omega_n = 0.2$.

Based on my $\zeta$ and $\omega_n$, the second-order transfer function is given by:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{0.04}{s^2 + 0.32s + 0.04}$$

From this transfer function, we calculate the poles by solving the characteristic equation:

$$s^2 + 0.32s + 0.04 = 0$$

The poles are:

$$s = -0.16 \pm 0.12j$$

Since it's a third-order system (3 state variables), we need to choose a third pole such that it is 2-5 times the real part of $\lambda_1$ or $\lambda_2$. Here, $\mathrm{Re}(\lambda_1 \text{ or } \lambda_2) = -0.16$, so the third pole can be chosen as $-1$.

Thus, the three poles are:

$$\lambda_1 = -0.16 + 0.12j, \quad \lambda_2 = -0.16 - 0.12j, \quad \lambda_3 = -1$$

The characteristic equation is:

$$(s+1)(s^2 + 0.32s + 0.04) = s^3 + 1.32s^2 + 0.36s + 0.04$$

# 2 Pole Placement (3.2.1)

In this section, we will design a state feedback controller for the system using the pole placement method. The goal is to place the system poles at desired locations to satisfy the time-domain specifications and achieve the required transient response. Given the three poles identified in the characteristic equation, the feedback gain matrix will be designed to place the closed-loop poles at these locations.

We want to determine the control law $u = -Kx + Fr$ such that the closed-loop system:

$$\dot{x} = (A - BK)x + BFr,$$

meets

$$\det(sI - (A - BK)) = s^n + \gamma_{n-1}s^{n-1} + \cdots + \gamma_0.$$

The desired pole locations are:

$$\lambda_1 = -0.16 + 0.12j, \quad \lambda_2 = -0.16 - 0.12j, \quad \lambda_3 = -1$$

We will calculate the state feedback gain matrix $K$ to achieve these pole locations.

## 2.1 Full Rank Method

We will use the Full Rank Method for Pole Placement in our analysis. This method involves determining whether the system is controllable and then selecting appropriate vectors for pole placement.

First, we check the controllability of the system by calculating the controllability matrix:

$$C = [B \ AB \ A^2B]$$

Calculating the controllability matrix, we find:

$$C == \begin{bmatrix} 7 & 0 & 17.6 & 0 & -955.17 & 6500 \\ -118 & 0 & 3701 & -26000 & 1.447 \times 10^6 & 2.314 \times 10^7 \\ 0 & -1300 & 77880 & 1.118 \times 10^6 & -6.9419 \times 10^7 & -9.4432 \times 10^8 \end{bmatrix}$$

The rank of this matrix is 3, which confirms that it is full rank, and hence, the system is controllable.

For a MIMO system, the next step is to select $n$ independent vectors from the $nm$ vectors in the controllability matrix in strict left-to-right order and group them with the same input in a square matrix $C$. In our case, this selection is $[b_1 \ Ab_1 \ b_2]$ as the first three columns are independent.

The index of input is $d_1 = 2$; $d_2 = 1$. Therefore, our controllability matrix $C$ is:

$$C = \begin{bmatrix} 7 & 17.6 & 0 \\ -118 & 3701 & 0 \\ 0 & 77880 & -1300 \end{bmatrix}$$

We then calculate $C^{-1}$ and use it to determine the transformation matrix $T$ as follows:

$$T = \begin{bmatrix} q_2^T \\ q_2^T A \\ q_3^T \end{bmatrix}$$

where $q_2$ and $q_3$ are the columns from $C^{-1}$ corresponding to the inputs.

Using $T$, we compute the transformed matrices $\bar{A}$ and $\bar{B}$ as:

$$\bar{A} = TAT^{-1}, \quad \bar{B} = TB$$

Assume the gain matrix $\bar{K}$ to be:

$$\bar{K} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \end{bmatrix}$$

We then compare the non-trivial rows from $\bar{A} - \bar{B}\bar{K}$ to a desired closed-loop matrix $A_d$. This results in six equations with six unknowns, which we solve to determine the gain matrix $\bar{K}$. The calculated $\bar{K}$ is:

$$\bar{K} = \begin{bmatrix} -1036.1 & 360.27 & 23.937 \\ 74842 & 0.36 & -1248.3 \end{bmatrix}$$

We then compute the original feedback gain as $K = \bar{K}T$. The resulting $K$ matrix is:

$$K = \begin{bmatrix} 1.1681 & -2.9838 & 1.784 \\ 0.24833 & 0.011681 & 0.96204 \end{bmatrix}$$

Finally, we check the eigenvalues of $A - BK$ and find them to be:

$$-0.16 + 0.12i, \quad -0.16 - 0.12i, \quad -1 + 0i$$

which match our desired pole locations. We have successfully placed our poles.

## 2.2 Experiments and Results

Using the designed controller, we calculate the step input response of the system. The Simulink model design used for this experiment is shown in Figure 1.

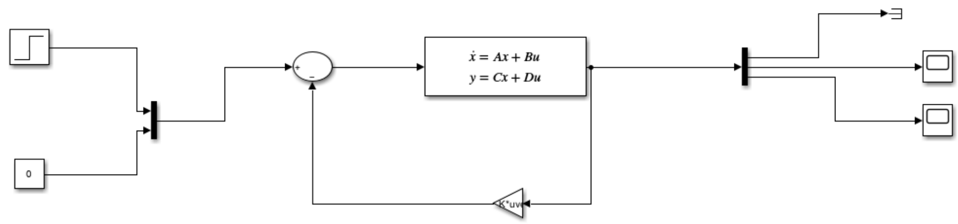



Figure 1: Simulink model design for step response experiment

The step input response of the system is shown in Figure 2.

From the images, we can observe the response of $x_2$ and $x_3$ when a step input is applied on input 1 and input 2. The settling time and overshoot values are:

$$\text{Settling time: } [23.22, 15.917, 25.162, 15.45] \text{ seconds}$$

$$\text{Overshoot: } [2.3448, 1.9492, 2.8431, 1.4201]\%$$

These values satisfy the time-domain specifications.

Below we can see the zero input response of the system.

From the figures, we can see that all three state variables $x_1$, $x_2$, and $x_3$ start from their initial conditions and reach a steady-state value within 30 seconds. This behavior demonstrates the stability of our control system.

We can change the poles and obtain different system behaviors while still adhering to the time domain specifications. Below are two such systems with different pole placements. The images are arranged in a 2x3 grid, where each column corresponds to the responses of $x_1$, $x_2$, and $x_3$, respectively, and each row corresponds to a specific pole design.

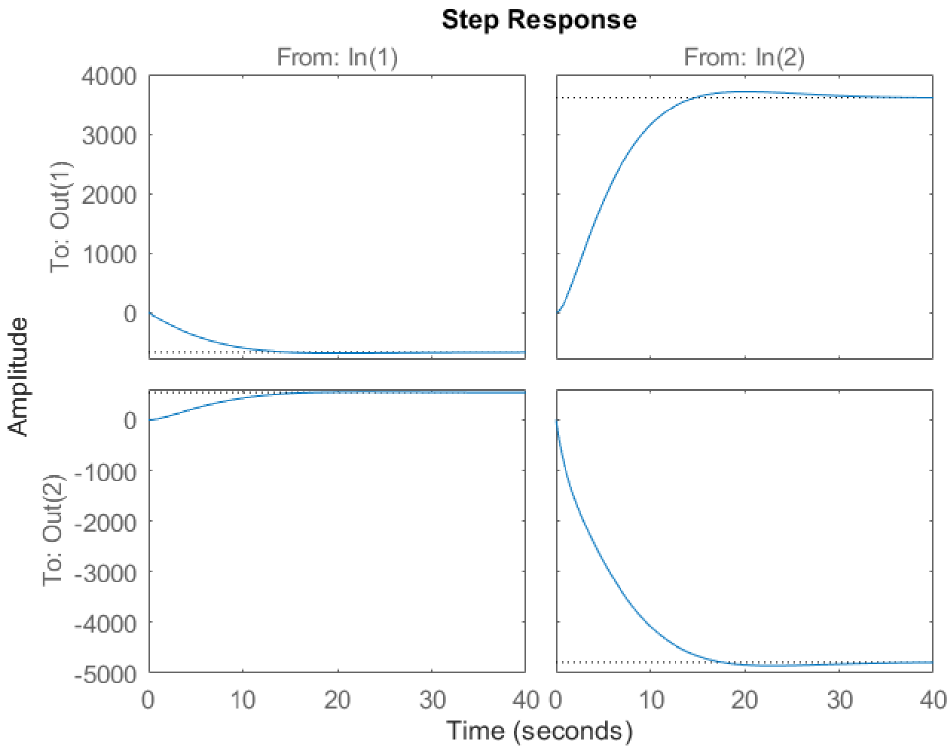
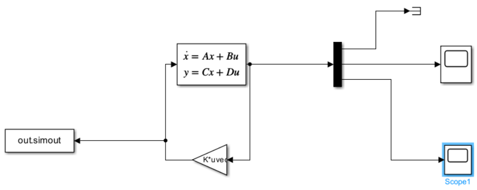
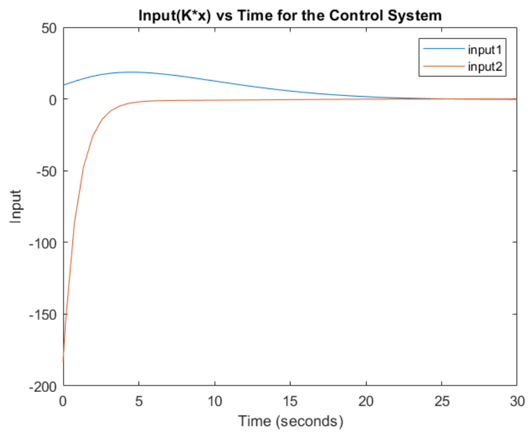


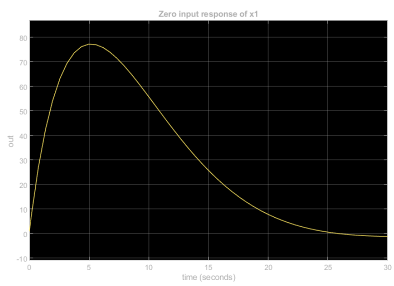

Figure 2: Step input response of the system



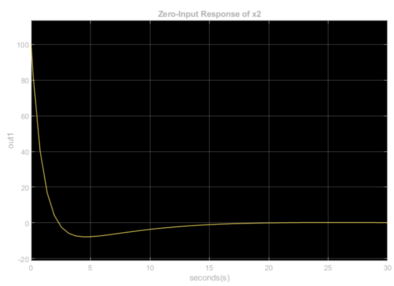
(a) Simulink design for zero input response



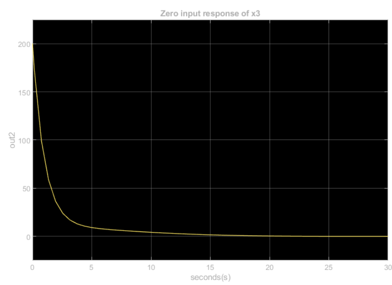

(b) Input $(K \cdot x)$ vs. time for zero input case



(c) $x_1$ response for zero input case

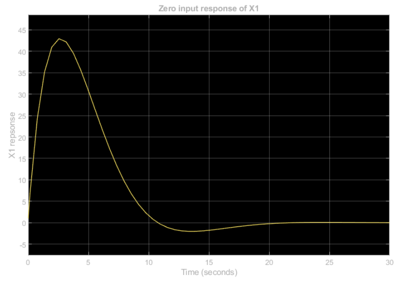


(d) $x_2$ response for zero input case

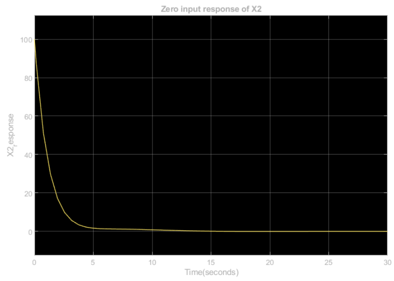


(e) $x_3$ response for zero input case

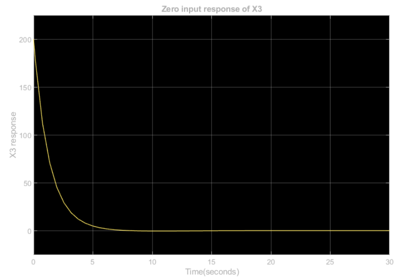
Figure 3: Zero input response of the system. (a) Simulink design, (b) Input $(K \cdot x)$ vs. time, (c) $x_1$ response, (d) $x_2$ response, (e) $x_3$ response



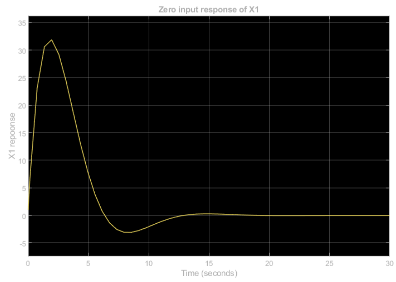

(a) $x_1$ response (Row 1)



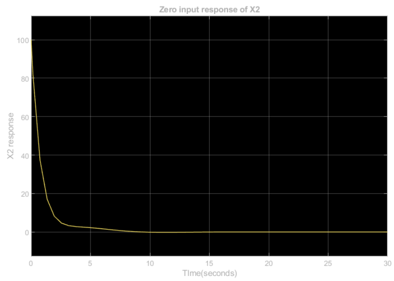

(b) $x_2$ response (Row 1)



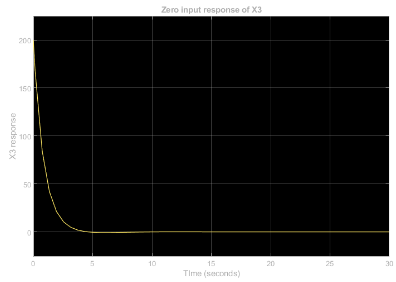

(c) $x_3$ response (Row 1)




(d) $x_1$ response (Row 2)




(e) $x_2$ response (Row 2)




(f) $x_3$ response (Row 2)

Figure 4: System responses for different pole placements. The first row corresponds to $\zeta = 0.7$, $\omega = 0.4$ with poles at $-0.28 \pm 0.286i$, $-0.8$. The second row corresponds to $\zeta = 0.6$, $\omega = 0.6$ with poles at $-0.36 \pm 0.48i$, $-1.2$.

When poles are placed further from the real axis, the system generally becomes faster, resulting in reduced settling times, which improves the response speed. However, increasing the distance can also introduce higher oscillations if the damping ratio is low. Properly balancing the pole positions helps achieve stability while satisfying time domain specifications.

# 3 LQR Control (3.2.2)

## 3.1 Method

The LQR optimal control is to find the control law that minimizes

$$J = \frac{1}{2} \int_0^\infty \left( x^T Q x + u^T R u \right) dt,$$

and it turns out to be in the form of linear state feedback: $u = r - Kx$.

The first step is to find the positive definite solution $P$ of the Riccati equation:

$$PA + A^T P - PBR^{-1}B^T P + Q = 0$$

where $A$ and $B$ are plant parameters, and $Q$ and $R$ are parameters provided at the beginning of the LQR method.

The matrices $Q$ and $R$ need to be chosen carefully to achieve a balance between response speed and energy efficiency. Generally, a larger $Q$ results in a faster response, while a larger $R$ reduces energy consumption. Typically, $Q$ and $R$ are selected as diagonal matrices to allow for individual weighting of specific state and control variables, especially if certain variables require higher penalties due to undesirable responses.

For this design, we assume:

$$Q = \begin{bmatrix} 15 & 0 & 0 \\ 0 & 900 & 0 \\ 0 & 0 & 150 \end{bmatrix}, \quad R = \begin{bmatrix} 200 & 0 \\ 0 & 160 \end{bmatrix}$$

To solve the Riccati equation, we need to form a $2n \times 2n$ (6x6) matrix, defined as:

$$\Gamma = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix}$$

We then select $n$ stable eigenvectors of this matrix to form the $\nu$ and $\mu$ vectors. In our case, the $\Gamma$, $\nu$, and $\mu$ matrices are as follows:

$$\Gamma = 10^4 \times \begin{bmatrix} -0.00017 & -0.000025 & 0 & -0.0000245 & 0.000413 & 0 \\ 0.0023 & -0.003 & 0.002 & 0.000413 & -0.006962 & 0 \\ 0 & -0.066 & -0.086 & 0 & 0 & -1.05625 \\ -0.0015 & 0 & 0 & 0.00017 & -0.0023 & 0 \\ 0 & -0.09 & 0 & 0.000025 & 0.003 & 0.066 \\ 0 & 0 & -0.015 & 0 & -0.002 & 0.086 \end{bmatrix}$$

$$\nu = \begin{bmatrix} 0.00009465 & -0.01447 & -0.04668 \\ -0.01511 & 0.29652 & -0.00299 \\ 0.99728 & -0.18473 & 0.00720 \end{bmatrix}$$

$$\mu = \begin{bmatrix} -0.00053935 & 0.08071 & -0.99634 \\ -0.03561 & 0.93336 & -0.07120 \\ 0.06271 & -0.00806 & -0.000399 \end{bmatrix}$$

Once we have $\nu$ and $\mu$, the solution to the Riccati equation, $P$, is given by:

$$P = \mu\nu^{-1}$$

The resulting matrix $P$ is:

$$P = \begin{bmatrix} 21.2602 & 1.3206 & 0.0174 \\ 1.3206 & 3.2202 & 0.0130 \\ 0.0174 & 0.0130 & 0.0631 \end{bmatrix}$$

From this $P$, we can calculate our gain matrix $K$ using the formula:

$$K = R^{-1}B^T P$$

Thus, the gain matrix $K$ is:

$$K = \begin{bmatrix} -0.0350 & -1.8537 & -0.0070 \\ -0.1418 & -0.1053 & -0.5125 \end{bmatrix}$$

## 3.2 Experiments and Results

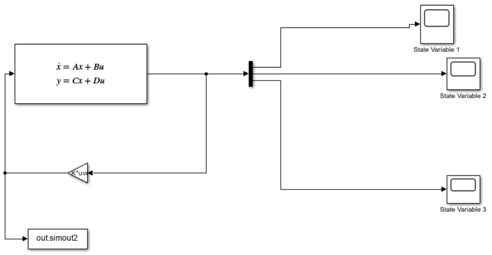
Using the gain matrix calculated above, the design is implemented in Simulink. The control signal size and the state responses for a zero input case with a non-zero initial state are shown in the figure below.

The step input response for the same system is shown in Figure 6. The settling time and overshoot values are as follows:

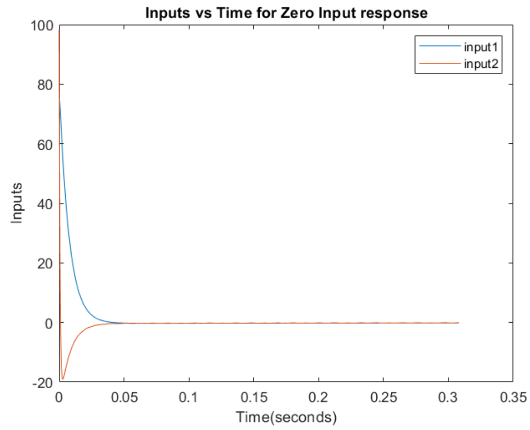
$$\text{Settling Time} = \begin{bmatrix} 4.4305 & 8.7667 \\ 5.1873 & 4.3162 \end{bmatrix}$$

$$\text{Overshoot} = \begin{bmatrix} 34.3343 & 0 \\ 0 & 33.5878 \end{bmatrix}$$

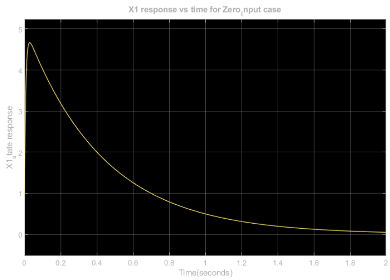
Clearly, the overshoot values are not acceptable as per our specifications. Hence, we need to adjust our $Q$ and $R$ values to ensure that the settling time is below 30 seconds and the overshoot is less than 10%.



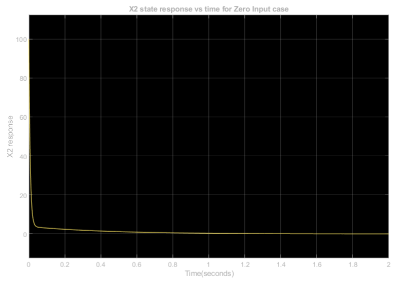

(a) Simulink design for zero input case



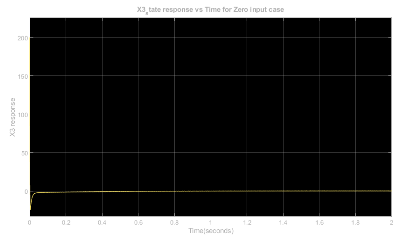

(b) Control signal size $Kx$ vs. time



(c) $x_1$ state response



(d) $x_2$ state response



(e) $x_3$ state response

Figure 5: Results for zero input case with non-zero initial state: (a) Simulink design, (b) Control signal size $Kx$ vs. time, (c) $x_1$ state response, (d) $x_2$ state response, (e) $x_3$ state response.

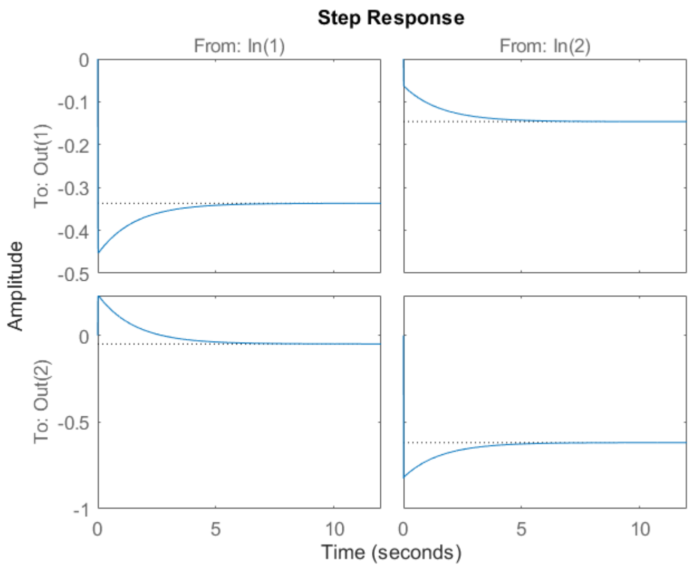



Figure 6: Step input response of LQR system

Here, our state variables are deviating significantly from the steady-state value at the moment the step input is applied. To mitigate this, we can assign higher $Q$ values to the elements corresponding to the 2nd and 3rd state variables, which will penalize large deviations more heavily. Additionally, increasing the $R$ values helps to minimize the control inputs, ensuring no drastic inputs are applied to the system.

Taking:

$$Q = \begin{bmatrix} 15 & 0 & 0 \\ 0 & 10000 & 0 \\ 0 & 0 & 2800 \end{bmatrix}, \quad R = \begin{bmatrix} 800 & 0 \\ 0 & 520 \end{bmatrix}$$

we find that the settling time and overshoot values are as follows:

$$\text{Settling Time} = \begin{bmatrix} 3.9407 & 14.3518 \\ 7.7959 & 3.8314 \end{bmatrix}$$

$$\text{Overshoot} = \begin{bmatrix} 9.8367 & 6.8060 \\ 0 & 9.9561 \end{bmatrix}$$

These values are within the specified time domain requirements, so these $Q$ and $R$ matrices are suitable for the system. We observe that the increased $R$ values result in a longer settling time compared to the previous design with smaller $R$ values. Additionally, the higher $Q$ values effectively reduce the overshoot by minimizing abrupt changes in the state variables.

# 4 LQR Controller with Observer (3.2.3)

## 4.1 Method

To design a state observer and simulate the resultant observer-based LQR control system, a full-order observer/-controller combination is used. In this method, the pair $(A, C)$ must be observable to ensure that the pair $(A^T, C^T)$ is controllable by duality. To check the observability of $(A, C)$, we construct the following observability matrix:

$$O = \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix}$$

Here, we have:

$$O = 10^5 \times \begin{bmatrix} 0 & 0.00001 & 0 \\ 0 & 0 & 0.00001 \\ 0.00023 & -0.0003 & 0.0002 \\ 0 & -0.0066 & -0.0086 \\ -0.07291 & -1.230575 & -1.78 \\ -1.518 & 5.874 & 7.264 \end{bmatrix}$$

The rank of this matrix is 3, indicating that the pair $(A, C)$ is observable. By duality, $(A^T, C^T)$ is controllable. We can use the controllability matrix from Section **3.2.3** (LQR section).

To form the observer, we need to stabilize the matrix $A - LC$ as follows:

$$\dot{\tilde{x}} = (A - LC)\tilde{x}$$

This stabilization ensures that the state estimation error $\tilde{x}$ converges to zero as $t \to \infty$.

To find $L$ that stabilizes $A - LC$, we compare the characteristic equation to the desired poles. The characteristic equation is:

$$\det[sI - (A - LC)] = \det[sI - (\tilde{A} - \tilde{B}\tilde{K})]$$

where $\tilde{A} = A^T$, $\tilde{B} = C^T$, and $\tilde{K} = L^T$.

This process is similar to pole placement in Section **2**. The desired poles are chosen to be 3-5 times the dominant pole of the LQR system. Finding $L$ is analogous to finding $K$ in the pole placement method.

### 4.2 Experiments and Results

Consider that the initial states for the observer, $\hat{x}$, are set to $\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$, while the initial states for the plant are given by $x_0 = \begin{bmatrix} 1 \\ 100 \\ 200 \end{bmatrix}$. The $Q$, $R$, and $K$ matrices from Section **3** are used in this design.

The poles for the observer are placed at $[-0.5, -0.5, 1]$. By following the steps similar to Section **2** (Pole Placement), we arrive at the observer gain matrix $L$ as follows:

$$L = \begin{bmatrix} -0.21348 & 0 \\ -30.2 & 20 \\ -660 & -859.5 \end{bmatrix}$$

The observer control system has the plant:

$$\dot{x} = Ax + Bu, \quad y = Cx,$$

the observer:

$$\dot{\hat{x}} = A\hat{x} + Bu + L[y - C\hat{x}]$$

and the control law:

$$u = -K\hat{x} + r$$

To simulate the above, the following Simulink model is used:

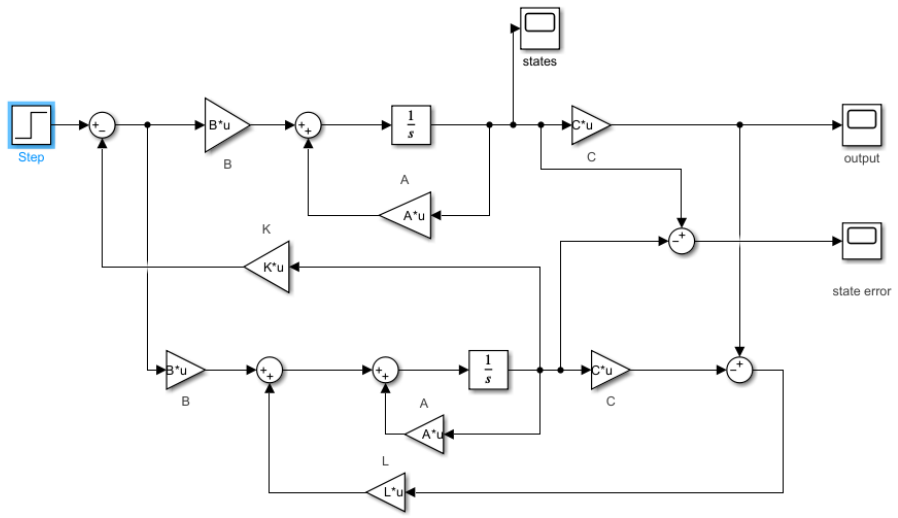
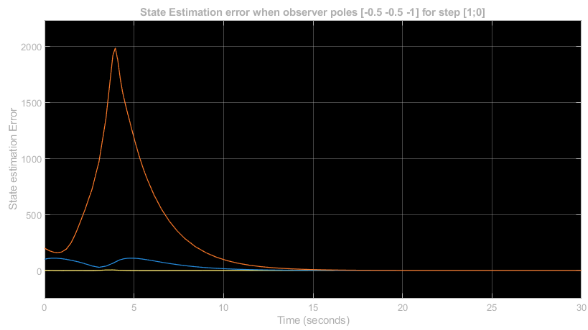


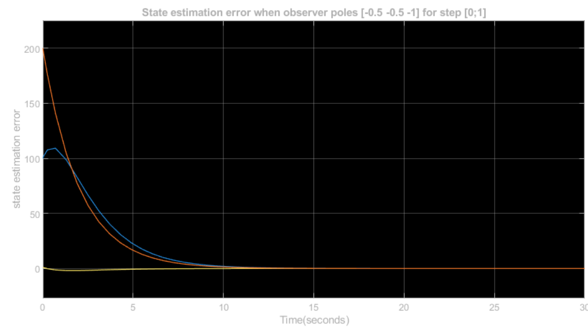

Figure 7: Simulink model for observer-based LQR control system

With the given initial conditions and observer poles, the state estimation errors and outputs for step inputs $[1; 0]$ and $[0; 1]$ are shown in Figure 8. Keeping everything else the same but changing the observer poles to $[-1, -1, -2]$, we get the state estimation errors and outputs for step inputs $[1; 0]$ and $[0; 1]$, as shown in Figure 9.

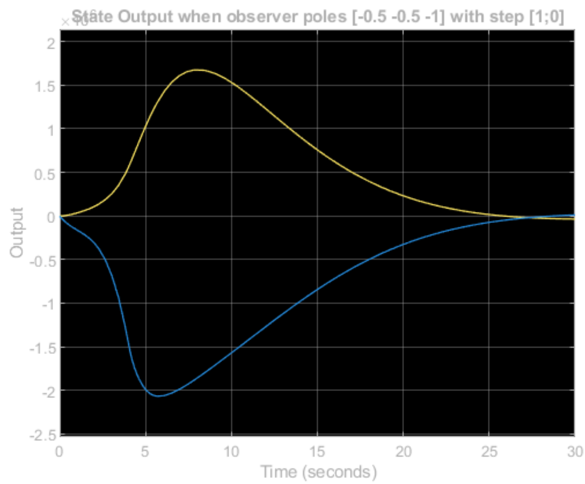
It is evident that initially there is an error between the true states and the estimated states. However, with the closed-loop control of the observer, this error converges to zero rapidly, resulting in no difference between the states and outputs over time. As we move the observer poles further from the imaginary axis, the state estimation error decays faster, indicating that the more stable the poles, the quicker the convergence between the estimated states and the actual states.



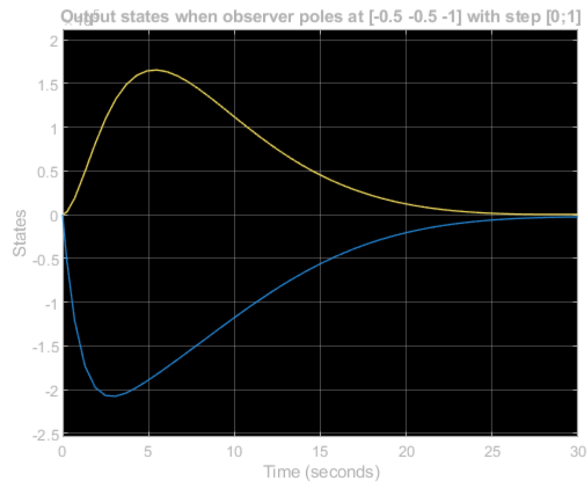

(a) Estimation error for input $[1; 0]$




(b) Estimation error for input $[0; 1]$

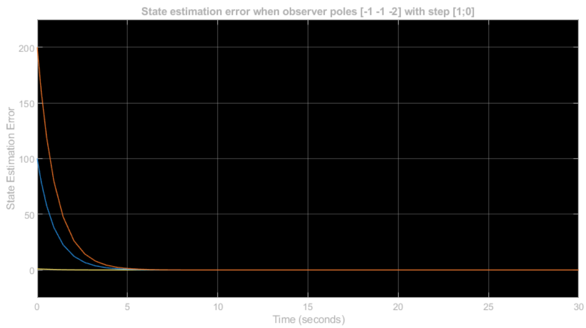



(c) Output for input $[1; 0]$

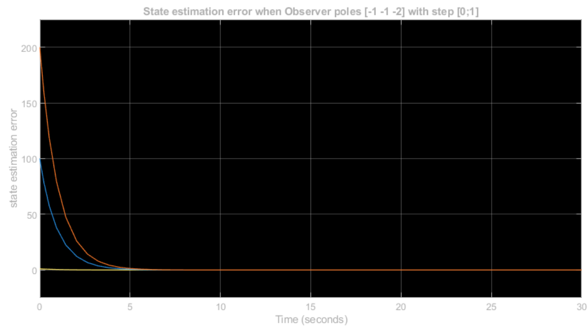



(d) Output for input $[0; 1]$

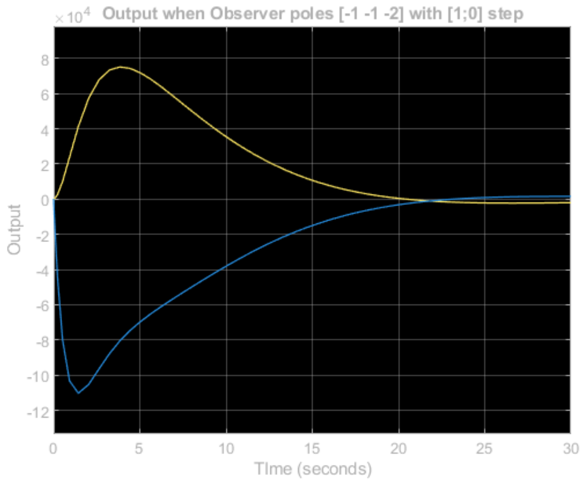
Figure 8: Observer controller system with poles at $[-0.5, -0.5, -1]$



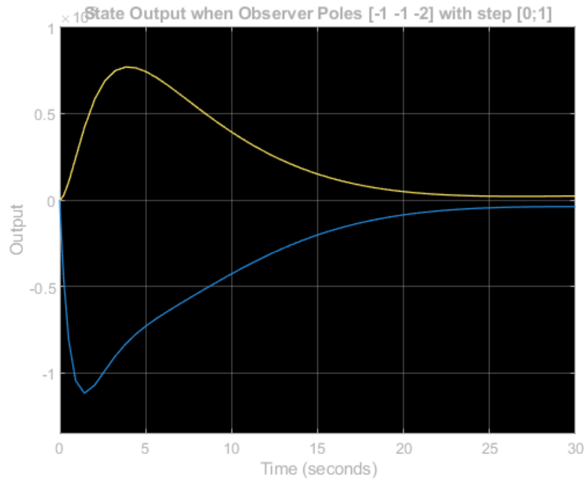

(a) Estimation error for input $[1;0]$




(b) Estimation error for input $[0;1]$




(c) Output for input $[1;0]$




(d) Output for input $[0;1]$

Figure 9: Observer controller system with poles at $[-1, -1, -2]$

# 5 Decoupling Control (3.2.4)

## 5.1 Method

Decoupling control aims to decouple a coupled plant, which is often required in practice to simplify operations. There are two methods to decouple a plant: state feedback with the state-space model and output feedback with the transfer function matrix. Since this is a linear state-space model, the first method (state feedback) is more suitable.

In this method, we first find the relative degree using $\{A, B, C\}$ to ensure that $G(s)$ is non-singular and decoupling is possible. To find the relative degree:

$$\sigma_i = \begin{cases} \min\left(j \mid c_i^T A^{j-1} B \neq 0^T,\ j = 1, 2, \ldots, n\right); \\ n, \quad \text{if } c_i^T A^{j-1} B = 0^T,\ j = 1, 2, \ldots, n \end{cases}$$

Here, we calculate $C(1,:) \times B$ as follows:

$$C(1,:) \cdot B = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 7 & 0 \\ -118 & 0 \\ 0 & -1300 \end{bmatrix} = \begin{bmatrix} -118 & 0 \end{bmatrix}$$

Similarly,

$$C(2,:) \cdot B = \begin{bmatrix} 0 & -1300 \end{bmatrix}$$

Since for $j = 1$ in both cases the product is not equal to zero, this implies that the relative degrees are:

$$\sigma_1 = 1, \quad \sigma_2 = 1$$

Now that we have calculated the relative degrees, we proceed to build the $B^*$ matrix.

$$B^* = \begin{pmatrix} c_1^T A^{\sigma_1 - 1} B \\ c_2^T A^{\sigma_2 - 1} B \\ \vdots \\ c_m^T A^{\sigma_m - 1} B \end{pmatrix}$$

Since $\sigma_1 = \sigma_2 = 1$, we have:

$$B^* = \begin{pmatrix} c_1^T B \\ c_2^T B \end{pmatrix}$$

Calculating each term, we get:

$$B^* = \begin{pmatrix} -118 & 0 \\ 0 & -1300 \end{pmatrix}$$

Clearly, this matrix $B^*$ is non-singular. Therefore, there exists an $F$ and $K$ to decouple the system.

$$F = (B^*)^{-1}$$

Calculating the inverse, we get:

$$F = \begin{pmatrix} -0.0085 & 0 \\ 0 & -0.0008 \end{pmatrix}$$

In order to form a stable decoupled system, we use the following theorem:

$$F = (B^*)^{-1}, \quad K = (B^*)^{-1} \begin{pmatrix} c_1^T \varphi_{f_1}(A) \\ c_2^T \varphi_{f_2}(A) \\ \vdots \\ c_m^T \varphi_{f_m}(A) \end{pmatrix}$$

where

$$\varphi_{f_i}(A) = A^{\sigma_i} + \gamma_{i1} A^{\sigma_i - 1} + \cdots + \gamma_{i\sigma_i} I,$$

in which

$$\varphi_{f_i}(s) = s^{\sigma_i} + \gamma_{i1} s^{\sigma_i - 1} + \cdots + \gamma_{i\sigma_i}$$

corresponds to the stable characteristic polynomial of the $i$-th input-output pair.

Here, we assume $H(s) = \begin{bmatrix} \frac{1}{s+1} & 0 \\ 0 & \frac{1}{s+1} \end{bmatrix}$, which provides two stable poles for the system. Therefore, we have:

$$\varphi_{f_1}(A) = \varphi_{f_2}(A) = A + I$$

Calculating $A + I$, we get:

$$A + I = \begin{bmatrix} -0.7000 & -0.2500 & 0 \\ 23.0000 & -29.0000 & 20.0000 \\ 0 & -660.0000 & -859.0000 \end{bmatrix}$$

To calculate $C^{**}$, we use the following formula:

$$C^{**} = \begin{pmatrix} c_1^T \varphi_{f_1}(A) \\ c_2^T \varphi_{f_2}(A) \end{pmatrix} = \begin{pmatrix} c_1^T (A + I) \\ c_2^T (A + I) \end{pmatrix}$$

Therefore,

$$C^{**} = \begin{bmatrix} 23 & -29 & 20 \\ 0 & -660 & -859 \end{bmatrix}$$

Therefore, we calculate $K$ as follows:

$$K = (B^*)^{-1} C^{**}$$

Calculating this product, we get:

$$K = \begin{bmatrix} -0.1949 & 0.2458 & -0.1695 \\ 0 & 0.5077 & 0.6608 \end{bmatrix}$$

We have therefore decoupled and stabilized the system with the matrices $K$ and $F$ as stated above.

## 5.2 Experiments and Results

A step input is given as the reference signal to the system. Two kinds of outputs are observed:

1. When the initial conditions are zero.
2. When the initial conditions are set to $x_0 = \begin{bmatrix} 1 \\ 100 \\ 200 \end{bmatrix}$.

Below are the images of the Simulink model used to simulate the system, the step input response of System 1 and System 2 with inputs $[1; 0]$ and $[0; 1]$, respectively.

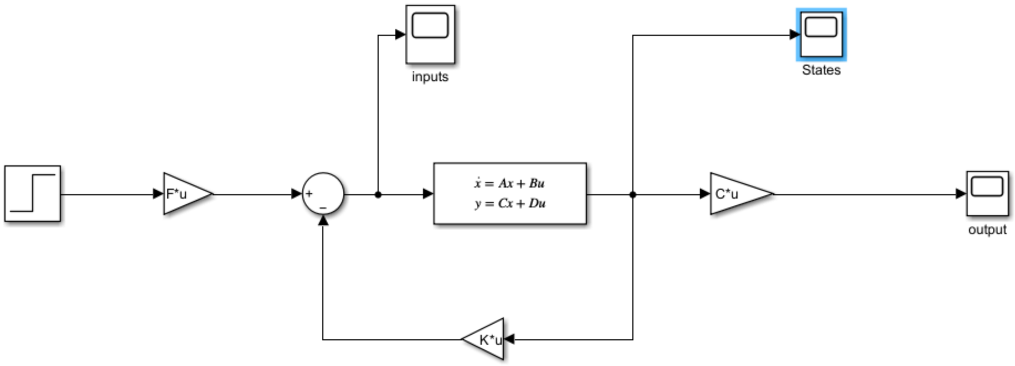
As expected after decoupling, the second and third state responses are decoupled from input 1 and input 2. When a $[1; 0]$ step input is applied:

1. In System 1, we observe that the second state variable converges to 1, while the third state variable remains at zero.
2. In System 2, the second state variable goes to 1 from an initial value of 100, and the third state variable decays to zero from an initial value of 200.

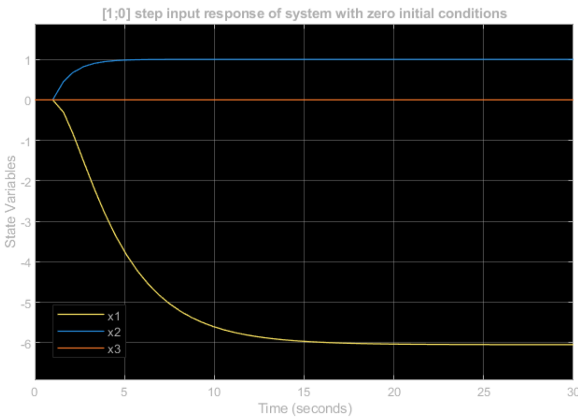
When a $[0; 1]$ step input is applied:

1. In System 1, the third state variable converges to 1, while the second state variable remains at zero.
2. In System 2, the third state variable goes to 1 from an initial value of 200, and the second state variable decays to zero from an initial value of 100.

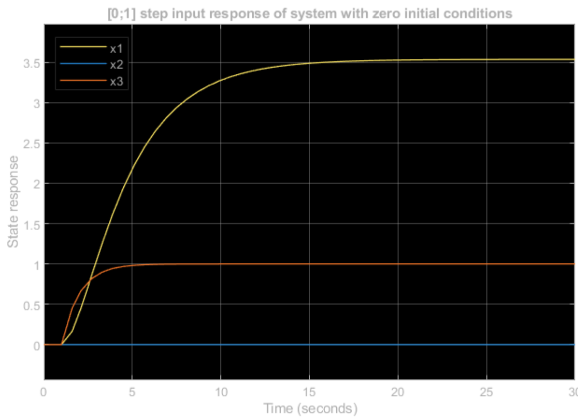
Both systems exhibit BIBO stability and internal stability, as evidenced by the fact that the output variables converge to the step input when their respective input is applied. Additionally, the first state variable remains stable across all four cases.



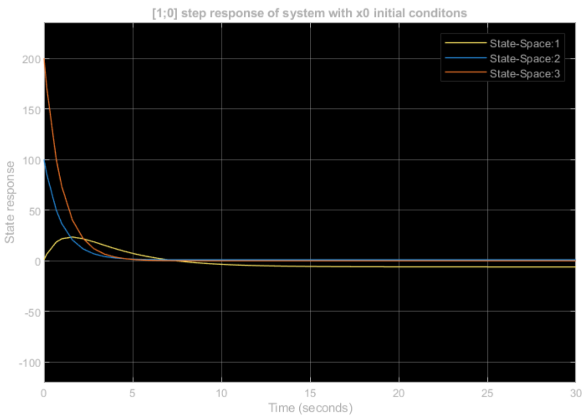

(a) Simulink model used for simulation



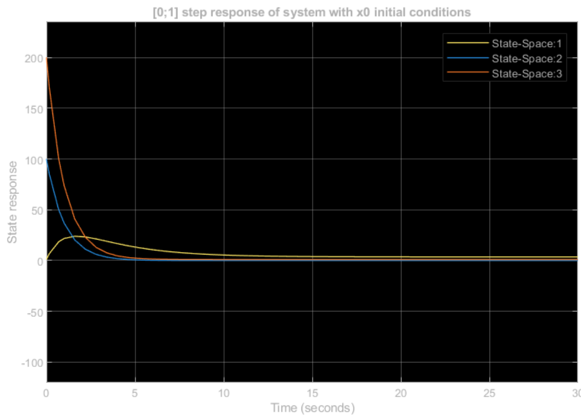

(b) Step response of System 1 with input [1; 0]




(c) Step response of System 1 with input [0; 1]




(d) Step response of System 2 with input [1; 0]




(e) Step response of System 2 with input [0; 1]

Figure 10: Simulink model and step input responses for System 1 and System 2. (a) Simulink model, (b)-(c) Step response of System 1, (d)-(e) Step response of System 2.

# 6 Integral Control + State Observer (3.2.5)

## 6.1 Method

In this task, the outputs are required to settle to the desired settling point $y_{sp} = \begin{bmatrix} 100 \\ 150 \end{bmatrix}$. Additionally, a disturbance $w = \begin{bmatrix} -2 \\ 5 \end{bmatrix}$ is applied at $t = 10$ seconds. Integral control is used to track the settling point, and because only two sensors are available to track the states, designing a state observer to estimate the states is necessary.

For integral control, we start by forming an augmented system and checking its controllability.

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} B_w \\ 0 \end{bmatrix} w + \begin{bmatrix} 0 \\ I \end{bmatrix} r$$

Check whether the augmented system is controllable or not:

$$\text{rank} \begin{pmatrix} A & B \\ -C & 0 \end{pmatrix} = 5 = n + m$$

The augmented matrices $\bar{A}, \bar{B}, \bar{B}_w, \bar{B}_r, \bar{C}$ are as follows:

$$\bar{A} = 10^2 \times \begin{bmatrix} -0.017 & -0.0025 & 0 & 0 & 0 \\ 0.23 & -0.3 & 0.2 & 0 & 0 \\ 0 & -6.6 & -8.6 & 0 & 0 \\ 0 & -0.01 & 0 & 0 & 0 \\ 0 & 0 & -0.01 & 0 & 0 \end{bmatrix}$$

$$\bar{B} = \bar{B}_w = \begin{bmatrix} 7 & 0 \\ -118 & 0 \\ 0 & -1300 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\bar{B}_r = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\bar{C} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

The rank of the augmented system is 5, which makes the system controllable.

An integral controller can be designed for this augmented system using the LQR method from Section **3**. Here, the $Q$ and $R$ matrices are assumed to be:

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 10 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

Using these $Q$ and $R$ matrices, we form the $\Gamma$ matrix and then solve the Riccati equation. The solution to the Riccati equation, $P$, is:

$$P = \begin{bmatrix} 9.7953 & 0.5719 & 0.0004 & -15.1287 & 8.7733 \\ 0.5719 & 0.0514 & -0.0001 & -0.9164 & 0.5217 \\ 0.0004 & -0.0001 & 0.0015 & -0.0001 & -0.0017 \\ -15.1287 & -0.9164 & -0.0001 & 40.5406 & -17.7254 \\ 8.7733 & 0.5217 & -0.0017 & -17.7254 & 20.5246 \end{bmatrix}$$

The control matrix $K$ is:

$$K = \begin{bmatrix} 2.1561 & -4.1323 & 0.0232 & 4.4628 & -0.2893 \\ -1.1671 & 0.1869 & -3.8593 & 0.2893 & 4.4628 \end{bmatrix}$$

Since the state is augmented, the $K$ matrix can be divided into $K_1$ and $K_2$, corresponding to the proportional and integral control components, respectively.

$$K_1 = \begin{bmatrix} 2.1561 & -4.1323 & 0.0232 \\ -1.1671 & 0.1869 & -3.8593 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 4.4628 & -0.2893 \\ 0.2893 & 4.4628 \end{bmatrix}$$

To design the observer, we follow a similar approach as outlined in Section **4**. The observer poles are placed 3-5 times further from the imaginary axis than the controller poles. Here, the controller poles are:

$$\text{controller\_poles} = \begin{bmatrix} -20000 & -5 & -2000 \end{bmatrix}$$

Placing the observer poles at these values and using pole placement, we obtain:

$$L = 10^4 \times \begin{bmatrix} 0.0286 & 0 \\ 0.1973 & 0.0020 \\ -0.0660 & 1.9140 \end{bmatrix}$$

With this setup, the required integral controller and state observer are designed using which the output follows the reference input and can eliminate the disturbance.

## 6.2 Experiments and Results

To simulate the designed Integral Control + State Observer system, the following Simulink model is used:

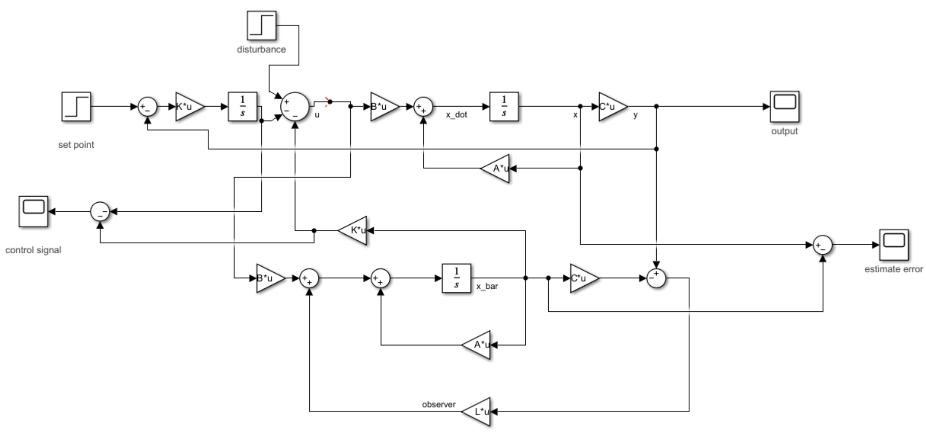
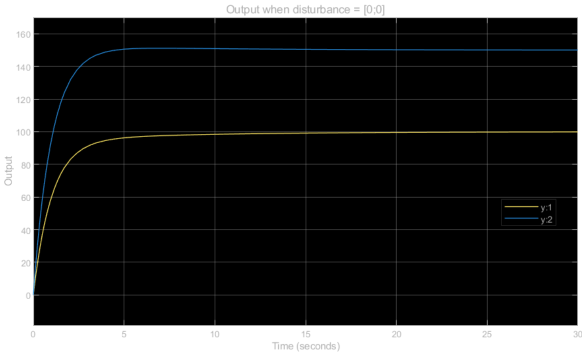


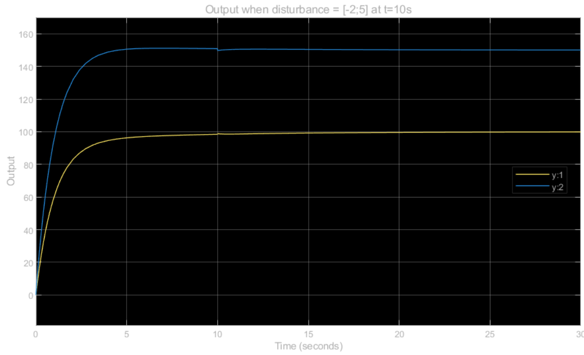

Figure 11: Simulink model for Integral Control + State Observer system

The following figures show the outputs, control signals, and state estimation errors for the cases without disturbance and with disturbance applied at $t = 10$ seconds, respectively.

When the disturbance is applied, slight deviation in the outputs can be observed but it quickly settles back to the settling point. This is also reflected as the input controls change slightly at $t = 10$ seconds when the disturbance is applied. Using integral control (servo control), we can achieve reference tracking for the output and eliminate steady-state disturbances. The state observer estimates unmeasured states, allowing for control where sensors may be costly or impractical to implement.

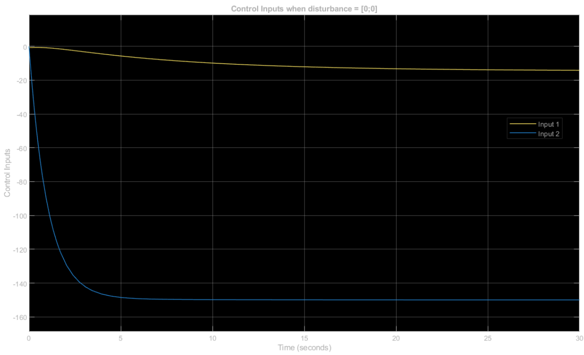



(a) Outputs without disturbance

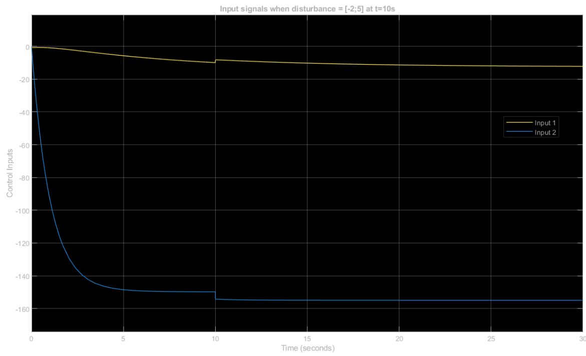



(b) Outputs with disturbance

Figure 12: Outputs for the system with and without disturbance




(a) Control signals without disturbance




(b) Control signals with disturbance

Figure 13: Control signals for the system with and without disturbance


(a) State estimation errors without disturbance


(b) State estimation errors with disturbance

Figure 14: State estimation errors for the system with and without disturbance

# 7 Manipulate Three State Variables Directly? (3.2.6)

## 7.1 Reasoning

Given that our target is to maintain the states $x$ around a specified set point $x_{sp} = \begin{bmatrix} 5 \\ 250 \\ 300 \end{bmatrix}$ at steady state, starting from the initial state $x_0 = \begin{bmatrix} 1 \\ 100 \\ 200 \end{bmatrix}$, the aim to manipulate all three state variables directly instead of focusing solely on the outputs. Assuming that the state-space representation remains the same with matrices $A$, $B$, and $C$, a similar approach to the integral control can be applied.

The state-space equations are given by:

$$\dot{x} = Ax + Bu + B_w w$$
$$y = Cx$$

where $x$ represents the state vector, $u$ is the control input, $w$ is the disturbance, and $y$ is the output.

To manipulate all three state variables, we introduce three integrators, one for each channel:

$$v(t) = \int_0^t e(\tau)\, d\tau$$

where $e(t) = r - x(t)$ is the error signal between the reference $r$ and the state $x$.

The derivative of $v(t)$ is given by:

$$\dot{v}(t) = r - x(t)$$

Form the augmented system by including the integrator in the state-space model. This results in the following system equations:

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -I & 0 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} B \\ 0 \end{bmatrix} w + \begin{bmatrix} 0 \\ I \end{bmatrix} r$$
$$y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix}$$

In this augmented system, the matrices are defined as follows:

$$\bar{A} = \begin{bmatrix} A & 0 \\ -I & 0 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad \bar{C} = \begin{bmatrix} C & 0 \end{bmatrix}$$

To check the controllability of this augmented system, we calculate the controllability matrix:

$$\text{Controllability matrix (Columns 1-6)} =$$
$$\begin{bmatrix} 7 & 0 & 17.6 & 0 & -955.17 & 6500 \\ -118 & 0 & 3701 & -26000 & 1.447 \times 10^6 & 2.314 \times 10^7 \\ 0 & -1300 & 77880 & 1.118 \times 10^6 & -6.9419 \times 10^7 & -9.4432 \times 10^8 \\ 0 & 0 & 7 & 0 & 17.6 & 0 \\ 0 & 0 & -118 & 0 & 3701 & -26000 \\ 0 & 0 & 0 & -1300 & 77880 & 1.118 \times 10^6 \end{bmatrix}$$

$$\text{Controllability matrix (Columns 7-12)} =$$

$$\begin{bmatrix} -3.60 \times 10^5 & -5.80 \times 10^6 & 3.59 \times 10^8 & 4.91 \times 10^9 & -3.05 \times 10^{11} & -4.14 \times 10^{12} \\ -1.43 \times 10^9 & -1.96 \times 10^{10} & 1.22 \times 10^{12} & 1.65 \times 10^{13} & -1.03 \times 10^{15} & -1.39 \times 10^{16} \\ 5.87 \times 10^{10} & 7.97 \times 10^{11} & -4.96 \times 10^{13} & -6.72 \times 10^{14} & 4.18 \times 10^{16} & 5.67 \times 10^{17} \\ -955.17 & 6500 & -3.60 \times 10^5 & -5.80 \times 10^6 & 3.59 \times 10^8 & 4.91 \times 10^9 \\ 1.45 \times 10^6 & 2.31 \times 10^7 & -1.43 \times 10^9 & -1.96 \times 10^{10} & 1.22 \times 10^{12} & 1.65 \times 10^{13} \\ -6.94 \times 10^7 & -9.44 \times 10^8 & 5.87 \times 10^{10} & 7.97 \times 10^{11} & -4.96 \times 10^{13} & -6.72 \times 10^{14} \end{bmatrix}$$

The rank of this controllability matrix is 2, which makes the augmented system uncontrollable. Therefore, it is not possible to maintain the states $x$ around an arbitrary reference input with the current state-space system.

Alternatively, there are two inputs and three outputs for this system. When the system's states converge to the settling point by the closed-loop LQR Controller, we have:

$$\dot{x} = 0,$$

which implies:

$$0 = Ax + Bu.$$

To meet the desired reference values, we set:

$$x = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}.$$

Substituting this into the steady-state equation gives:

$$0 = A \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} + B \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}.$$

This equation can be rewritten as:

$$B \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = -A \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}.$$

Expanding this equation, we have:

$$\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = - \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}.$$

Since $B \in \mathbb{R}^{3\times 2}$ and $u \in \mathbb{R}^2$, we are trying to solve a system of 3 equations with only 2 unknowns (the two inputs $u_1$ and $u_2$):

This system is **underdetermined** for generic values of $r_1, r_2, r_3$, meaning that, in general, there is no solution for arbitrary values of these references. Hence, this again proves that it is not possible to maintain the states $x$ around an arbitrary reference input with the current state-space configuration.

In this case, we only want to keep the state variables at steady state close enough to the set point $x_{sp}$. To place different emphasis on the exactness of the three state variables, we aim to minimize the following objective function:

$$J(x_s) = \frac{1}{2}(x_s - x_{sp})^T W (x_s - x_{sp}),$$

where $W = \mathrm{diag}(11, 12, 13)$ is a weighting matrix, and $x_s$ is the state vector at steady state.
Let $u_1$ and $u_2$ be the inputs at steady state. The steady-state value for the state-space system is given by:

$$x_s = \lim_{s\to 0}(sI - A)^{-1}BU,$$

where $U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$.
Substituting this expression into the objective function and minimizing with respect to $u_1$ and $u_2$ using the Jacobian, we obtain $u_1$ and $u_2$ as:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 5.6822 \\ -316.93 \end{bmatrix}$$

## 7.2 Experiments and Results

The following Simulink model was used to simulate the state space system with the calculated steady-state inputs:

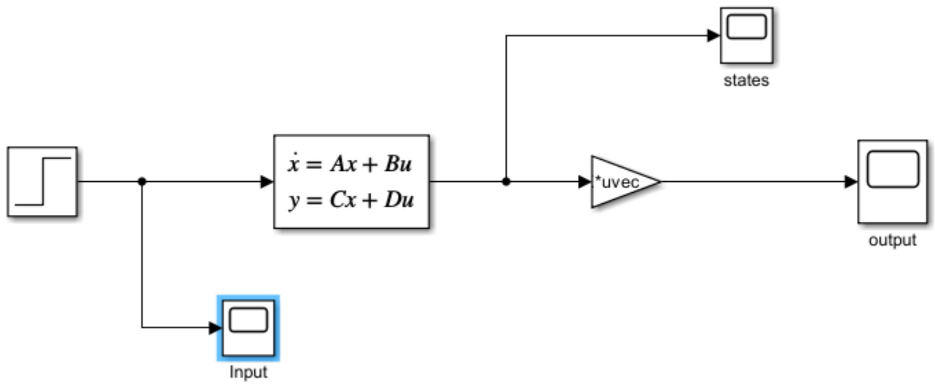
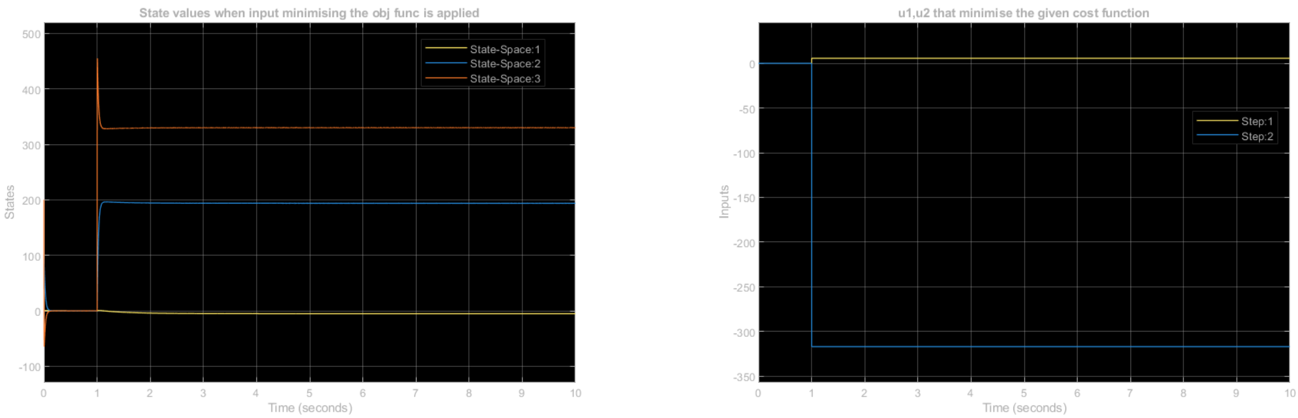



Figure 15: Simulink model for state space system simulation with calculated inputs

Below are the plots of the steady-state values of the state variables and the corresponding step inputs $u_1$ and $u_2$:




(a) Steady-state values of state variables

(b) Step inputs $u_1$ and $u_2$

Figure 16: Steady-state values and step inputs

From the figures, it can be observed that the steady-state values are reasonably close to $x_{sp}$. Adjusting the $W$ matrix allows us to place greater emphasis on the accuracy of specific states. The current $W$ matrix has nearly equal values along its diagonal, but these values can be modified to achieve improved accuracy for selected state variables.

# 8 Conclusions

In the full rank method for pole placement, a control system was designed where the outputs decay to zero when zero input is applied. The selected poles were chosen to ensure stability, and it was observed that by placing the poles further from the origin, the output decayed to zero more quickly. This demonstrated the trade-off between response speed and stability margin. This method successfully achieved the desired system behavior at zero input.

In the LQR controller design, similar behavior was observed where the output decays to zero under zero input. The elements of the $Q$ matrix were adjusted to control the trade-off between response speed and energy cost. Larger values in $Q$ resulted in faster convergence of state variables to zero, albeit with a larger control signal and higher energy consumption. Conversely, increasing the elements of $R$ led to slower responses but required smaller control signals and reduced energy costs. This balance allowed for optimal control based on system requirements.

In the system designed with a state observer, an initial error existed between the true states and the estimated states. However, with closed-loop control, the observer quickly drove the error to zero. By placing the observer poles further from the origin, the estimation error decayed faster. State observers are necessary in cases where certain state variables cannot be directly measured, allowing the system to estimate unobservable states accurately and maintain control stability.

In the decoupling control design, the outputs were successfully decoupled from the inputs, achieving independent control of each output. The system exhibited both internal and external stability, with the outputs and state values converging to steady values. Decoupling control is crucial in systems where independent manipulation of outputs is needed, facilitating easier and more effective system control. The decoupling also improved the overall system response, enabling it to meet stability requirements for practical operation.

For the integral control combined with a state observer, the system's outputs successfully tracked the reference signals due to the integrator's effect. The integral control allowed reference tracking and eliminated steady-state error. When a disturbance was applied at $t = 10$ seconds, its effect on the system was minimized, seen as minor spikes in the output and input values. The augmented system, using integral control, enabled the use of fewer sensors, reducing system costs while achieving effective disturbance rejection.

In exploring the manipulation of all three state variables directly, it was concluded that three inputs are required to independently control all three states. The current system, with only two inputs, could not follow an arbitrary reference signal for all state variables. However, minimizing a specific objective function with steady-state values allowed for approximate results. The steady-state values were close to the desired set point, with minor errors that could potentially be minimized by adjusting the weighting matrix $W$ to emphasize specific states.

Overall, the analysis demonstrated that each control method—pole placement, LQR, state observer, decoupling, and integral control—provides unique advantages depending on the system's goals, whether they involve reference tracking, disturbance rejection, or independent state control. By fine-tuning design parameters like the $Q$, $R$, and $W$ matrices, further improvements in system performance can be achieved based on practical requirements.

# Appendices

# A MATLAB Code

## A.1 Main Script

```
clear all;
clc;

%% Given Information
% Given matrices
A = [-1.7, -0.25, 0;
     23, -30, 20;
     0, -660, -860];
B = [7, 0;
     -118, 0;
     0, -1300];
C = [0, 1, 0;
     0, 0, 1];
D = [0 0;
     0 0];
% initial conditions
x0 = [1 100 200]';

% calculated zeta and omega based on time domain specifications
zeta = 0.8;
omega = 0.2;

%% pole placement
[p1, p2] = pole_estimator(zeta, omega); % 3rd pole is assumed to be -1

P = [p1 p2 -1];
% K = pole_placement(A,B,P);

% K is used to simulate the system using zero_input_pole_placement.slx

%% LQR
% define Q and R matrices

Q = [15 0 0;
     0 900 0;
     0 0 150];
R = [200 0;
     0 160];

[K, V, U, P, M] = LQR_own(A,B,Q,R);

% K is used to simulate the system using step_input_lqr.slx and
% zero_input_lqr.slx

%% LQR controller with observer

% check for observability of pair(A,C)
obsr = [C; C*A; C*A*A];
if rank(obsr) < 3
        error('it is unobservable!');
end

% We use pole placement to find observer L
% observer poles:
P = [-0.5 -0.5 -1];
% L = pole_placement(A,B,P);

% L and K from LQR are used to simulate using observer_controller.slx

%% Decoupling control

[F,K] = decoupling(A,B,C);

% these F,K values combined with K from LQR are used to simulate using
% decoupler_ss.slx
```

```
%% Intergal control + state observer

y_sp=[100;150]; % settling point
disturbance=[-2;5]; % disturbance load applied at t=10s

Contr = rank([A B;C zeros(2,2)]);
if Contr ~= size(A,1) + size(C,1)
    error("The matrix is un-controllable");
end

% defining augmented matrices
A_bar=[A zeros(3,2);-C zeros(2,2)];
B_bar=[B;zeros(2,2)];
B_w_bar=[B;zeros(2,2)];
B_r_bar=[zeros(3,2);eye(2)];
C_bar=[C zeros(2,2)];

% defining Q and R to solve the augmented system
Q=[1 0 0 0 0;
   0 1 0 0 0;
   0 0 1 0 0;
   0 0 0 1 0;
   0 0 0 0 1]*10;

R=[1 0
   0 1]*0.5;

% solving using LQR
gamma=[A_bar -B_bar/R*B_bar'
    -Q -A_bar'];
[vector,value]=eig(gamma);

value=sum(value);
v=vector(:,find(real(value)<0));

P=v(6:10,:)/v(1:5,:);
K=real(inv(R)*B_bar'*P);

K1 = K(:,1:3);
K2 = K(:,4:5);

% to design the observer
% controller poles
desired_poles = [-20000 -5 -2000];
L = place(A',C',desired_poles)';

% using K1, K2, L the system is simulated using servos.slx

%% Question 6

% calculating given weight matrix
a=4;
b=6;
c=8;
d=8;
w=[a+b+1 0 0
    0 c+4 0
    0 0 d+5];

syms s u1 u2
U=[u1;u2];
xsp=[5 250 300]'; % settling point
xs=inv(s*eye(3)-A)*B*U;
xs=subs(xs,s,0); % steady state value of the system

% solving the jacobian to find the u1,u2 that minimize the loss function.
J=1/2*(xs-xsp)'*w*(xs-xsp);
jacob = jacobian(J, [u1 u2]);
ans = vpasolve(jacob==0);
u1=double(ans.u1);
u2=double(ans.u2);
u=[u1;u2];
```

```
% This u is used to simulate the system using state_control_last_model.slx
```

Listing 1: Main MATLAB Script

### A.2 Pole Placement Function

```
function K = pole_placement(A,B,P)
    % this function is used to place poles by pole placement
    % use Full rank method .i.e. controllable canonical form to calculate
    % poles

    % required characteristic polynomial
    lambda1 = P(1);
    lambda2 = P(2);
    lambda3 = P(3);

    syms s
    polynomial=(s-lambda1)*(s-lambda2)*(s-lambda3);
    Ad_cof=double(coeffs(polynomial));

    % controlability check:
    ctrb = [B A*B A*A*B];
    if rank(ctrb) < 3
        error('it is uncontrollable!');
    end

    for i=1:6
        WR(i) = rank(ctrb(:,1:i));
    end

    rho = [2 1]; % The controllability Index is determined by inspection of
                 % WR
    WA = [ctrb(:,1)'; ctrb(:,3)'; ctrb(:,2)']';

    M = inv(WA);
    M1 = M(rho(2),:);
    M2 = M(rho(1)+rho(2),:);

    % to find T - transformation matrix
    T = [M1; M1*A; M2];
    T_inv = inv(T);

    A_bar = T*A*T_inv;
    B_bar = T*B;


    % rounding off the low values to zero
    A_bar(abs(A_bar)<10^(-10))=0;
    B_bar(abs(B_bar)<10^(-10))=0;

    A_bar = round(A_bar, 2);
    B_bar = round(B_bar, 2);

    % solving for K
    K_bar = sym('k', [2 3]);
    closed_loop_matrix= A_bar - B_bar*K_bar

    % let the desired closed loop matrix be:
    Ad = [0 1 0;
        0 0 1;
        -Ad_cof(1:3)];

    % solving the equation
    equation = Ad == closed_loop_matrix;

    K_num=solve(equation)
    K_ans=struct2array(K_num);
    K_ans=double(K_ans);
    Kbar=[K_ans(1:3);
         K_ans(4:6)];
```

```

    % tranforming K back to original
    K = Kbar*T;

end
```

Listing 2: Pole Placement Function 1

### A.3 Pole Estimator Function

```
function [pole1, pole2] = pole_estimator(zeta, omega)
pole1 = -zeta * omega + omega * sqrt(zeta^2 - 1);
pole2 = -zeta * omega - omega * sqrt(zeta^2 - 1);
end
```

Listing 3: Pole Estimator Function

### A.4 LQR Function

```
function [K, V, U, P, M] = LQR_own(A,B,Q,R)
M = [ A    -B/R*B';
     -Q    -A'];

[evec, eval] = eig(M);
eval = sum(eval);
evec_stable = evec(:,find(real(eval)<0));
% P = evec_stable(,:)
V = evec_stable(1:3,:);
U = evec_stable(4:6,:);
P = U/V;

K = real(R \ B' * P);

end
```

Listing 4: LQR Function

### A.5 Decoupling Function

```
function [F,K] = decoupling(A,B,C)
    % the relative degree calculated
    sigma = [1 1];

    Bstar=[C(1,:)*A^(sigma(1)-1)*B;
       C(2,:)*A^(sigma(2)-1)*B];

    if det(Bstar) == 0
        error('Bstar matrix is singular')
    end

    F = inv(Bstar);

    % calculating C**
    % assuming H(s) = diag(1/(s+1))
    phi = A + eye(3);

    C_star = [C(1,:)*phi;
              C(2,:)*phi];
    K= Bstar\C_star;
end
```

Listing 5: Decoupling Function