

## Section I Introduction

### 1.1 Background

### 1.2 Modeling

## Section II Control System Design

### 2.1 Design specifications

### 2.2 Pole Placement

#### 2.2.1 Controller Design

#### 2.2.2 Simulate

### 2.3 LQR Controller

#### 2.3.1 Controller Design

#### 2.3.2 Simulate

### 2.4 State observer

#### 2.4.1 Observer Design

#### 2.4.2 Simulate

### 2.5 Decoupling Controller

#### 2.5.1 Controller Design

#### 2.5.2 Simulate

### 2.6 Servo Controller

#### 2.6.1 Controller Design

#### 2.6.2 Simulate

### 2.7 Minimize objective function

## Section III Conclusion

## Appendices

## Section I Introduction

### 1.1 Background

This is a real control problem in chemical engineering, where we are trying to control the reaction process that takes place in a tank shown in the following figure 1.1. The chemical reaction is described by  $A \rightarrow B$ .  $A$  and  $B$  stand for the reactant and product. The reaction is conducted in the large container located at the central of the tank, meanwhile there is water flow inside the surrounded container wall (we call it cooling jacket) to control the reaction temperature.

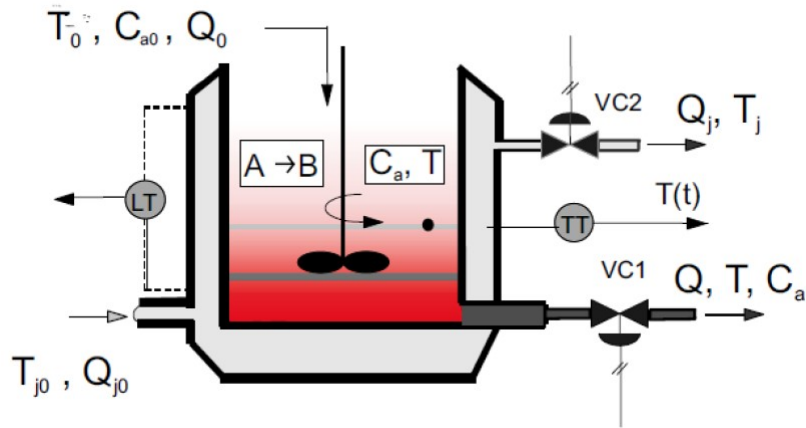


Figure 1.1 CSTR reactor

Our goal is to control the component of the reactant.

### 1.2 Modeling

For model-based control, the first step is to build an effective dynamic model for our target plant, the continuous-flow stirred tank reactor (CSTR) in this project. Define the state vector  $x = [C_a \ T \ T_j]^T$ , where  $C_a$  is the component concentration of the reactant,  $T$  is the reaction temperature in the tank, and  $T_j$  is the temperature of the outflow water of the cooling jacket. Define the control signal  $u = [F \ F_j]^T$ , where  $F$  is the outlet flow rate of the reaction;  $F_j$  is the flow rate of the water in the cooling jacket. Define the measurement vector  $y = [T \ T_j]^T$ , because we only have two temperature sensors. The system is described by

$$\begin{aligned} \dot{x} &= Ax + Bu + Bw \\ y &= Cy \end{aligned} \tag{1-1}$$

My matriculation number is A023935A, then  $a = 2$ ,  $b = 9$ ,  $c = 3$ ,  $d = 5$ , so:

$$A = \begin{bmatrix} -1.7 & -0.25 & 0 \\ 23 & -30 & 20 \\ 0 & -490 & -1140 \end{bmatrix}$$

$$B = \begin{bmatrix} 5 & 0 \\ -83 & 0 \\ 0 & -770 \end{bmatrix}$$

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

The initial condition of the system is assumed to be

$$x_0 = [1 \quad 100 \quad 200]^T$$

## Section II Control System Design

### 2.1 Design specifications

The transient response performance specifications for all the outputs  $y$  in state space model (1-1) are as follows:

- 1) The overshoot is less than 10%.
- 2) The 2% setting time is less than 30 seconds.

Consider a standard second-order system:

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2-1)$$

Its two poles determine performance completely. According to the transient response performance specifications, we can get:

$$\begin{cases} M_p = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} \leq 10\% \\ t_s = \frac{4.0}{\zeta\omega_n} \leq 30 \text{ s} \end{cases} \quad (2-2)$$

By solving (2-2), we can get  $\zeta > 0.59$ ,  $\zeta\omega_n \geq 0.133$ . We can choose:

$$\begin{cases} \zeta = 0.65 \\ \omega_n = 0.25 \end{cases}$$

The poles of the reference model are:

$$s = -0.1625 \pm 0.19i$$

### 2.2 Pole Placement

#### 2.2.1 Controller Design

Assume that we can measure all the three state variables. I need to design a state feedback controller via pole placement. Because state space (1-1) is a three-order system, we can choose a poles far from the dominant poles. The desired poles are:  $s_1 = -0.1625 + 0.19i$ ,  $s_2 = -0.1625 - 0.19i$ ,  $s_3 = -0.64$ . The determinate of the closed-loop matrix is

$$\det(sI - A_d) = (s - s_1)(s - s_2)(s - s_3) \quad (2-3)$$

We can get closed-loop matrix form (2-3)

$$A_d = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ s_1 s_2 s_3 & -(s_1 s_2 + s_1 s_3 + s_2 s_3) & s_0 + s_1 + s_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.04 & -0.2705 & -0.965 \end{bmatrix} \quad (2-4)$$

Then, we need to obtain the controllable canonical form of this system. First, we need to compute the controllability matrix:

$$\begin{aligned} W_c &= [B \quad AB \quad A^2B] \\ &= [b_1 \quad b_2 \quad Ab_1 \quad Ab_2 \quad A^2b_1 \quad A^2b_2] \\ &= \begin{bmatrix} 5 & 0 & 12.25 & 0 & -672.08 & 3850 \\ -83 & 0 & 2605 & -15400 & 735531.7 & 18018000 \\ 0 & -770 & 40670 & 877800 & -47640250 & -993146000 \end{bmatrix} \end{aligned} \quad (2-5)$$

Because  $\text{rank}(W_c) = 3$ , the system is controllable, and the transformation matrix  $T$  exists. We can select 3 independent vectors from (2-4) in the order from left to right. And group them in a matrix  $E$  in the following form:

$$E = [b_1 \quad Ab_1 \quad b_2] = \begin{bmatrix} 5 & 12.25 & 0 \\ -83 & 2605 & 0 \\ 0 & 40670 & -770 \end{bmatrix} \quad (2-6)$$

Once we have the matrix  $E$ , we need to compute the inverse of  $E$ . Because there are two vectors are associated with  $u_1$ , one vector is associated with  $u_2$ , we can get  $d_1 = 2$ ,  $d_2 = 1$ . We need to chose the second and third rows of  $E^{-1}$ , then we can get transformation matrix  $T$

$$T = \begin{bmatrix} q_2^T \\ q_2^T A \\ q_3^T \end{bmatrix} = \begin{bmatrix} 0.0059 & 0.0004 & 0 \\ -0.0019 & -0.0122 & 0.0071 \\ 0.3122 & 0.0188 & -0.0013 \end{bmatrix} \quad (2-7)$$

The matrix  $\bar{A}$  and  $\bar{B}$  can be expressed as

$$\bar{A} = TAT^{-1}, \quad \bar{B} = TB \quad (2-8)$$

Then we need to design the feedback gain matrix for the controllable canonical form.

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

Next step, we need to compare  $\bar{A} - \bar{B}\bar{K}$  and  $A_d$ , and equalize them. By solving this equation, we can get  $\bar{K}$ , and the feedback matrix  $K$

$$K = \bar{K}T = \begin{bmatrix} 0.5236 & -3.0940 & 1.8245 \\ 0.2028 & 0.0090 & 1.8573 \end{bmatrix} \quad (2-9)$$

The controller can be expressed as:

$$u = -Kx + Ir$$

## 2.2.2 Simulate

The initial state is  $x_0$ , we can use Simulink to simulate the designed system. My simulate module is showed in the following figure 2.1.

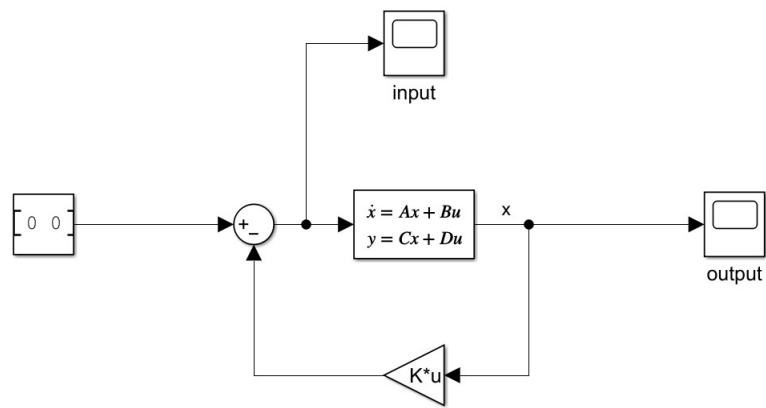


Figure 2.1 Pole placement module

All the three state responses to non-zero initial state  $x_0$  with zero external inputs are shown in figure 2.2. The control signal is shown in figure 2.3. The desired poles are:  $s_1 = -0.1625 + 0.19i$ ,  $s_2 = -0.1625 - 0.19i$ .

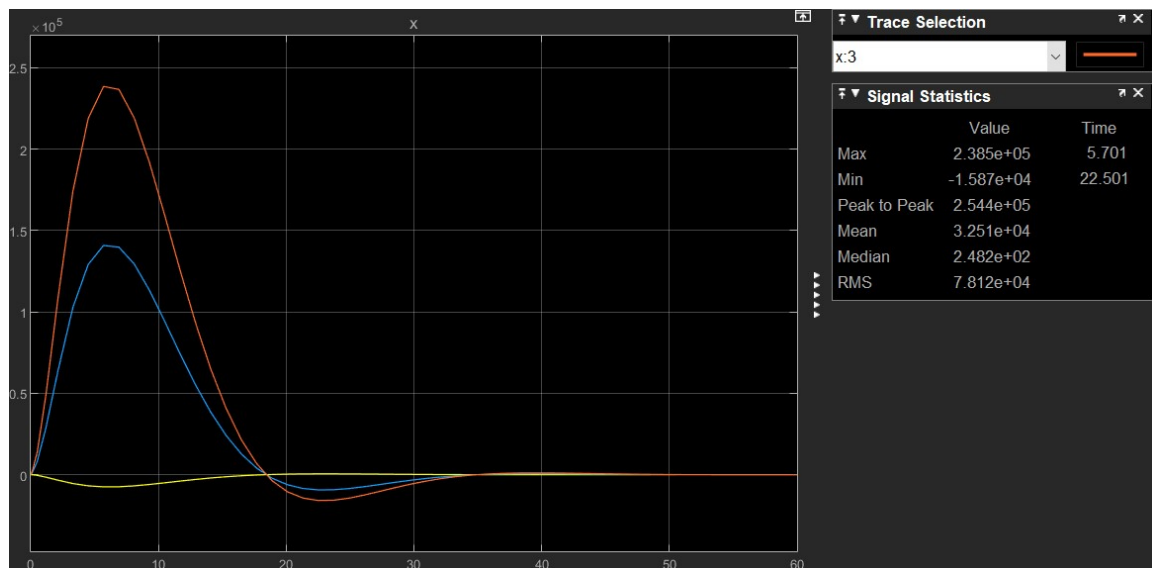


Figure 2.2 State response

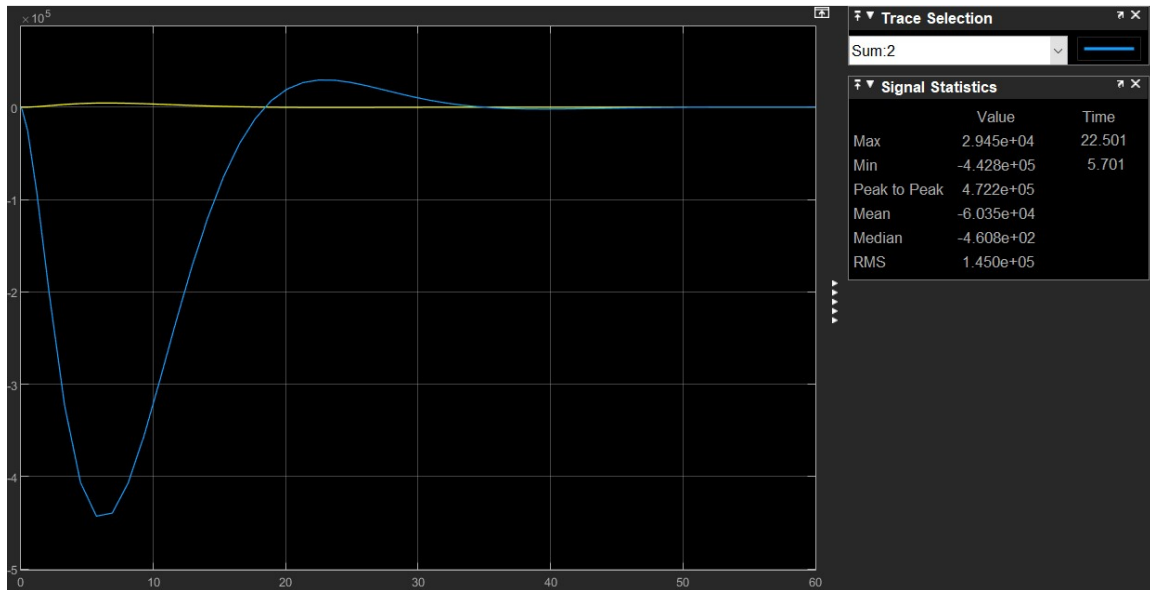


Figure 2.3 Control signal

If we increase  $\omega_n$  and  $\zeta$  at the same time, the real part part of the pole will goes away from original. The setting time of the system will be shorter and the input signal will be larger. If we decrease A and B, we get the opposite result. Simulation results are shown in the figure below.

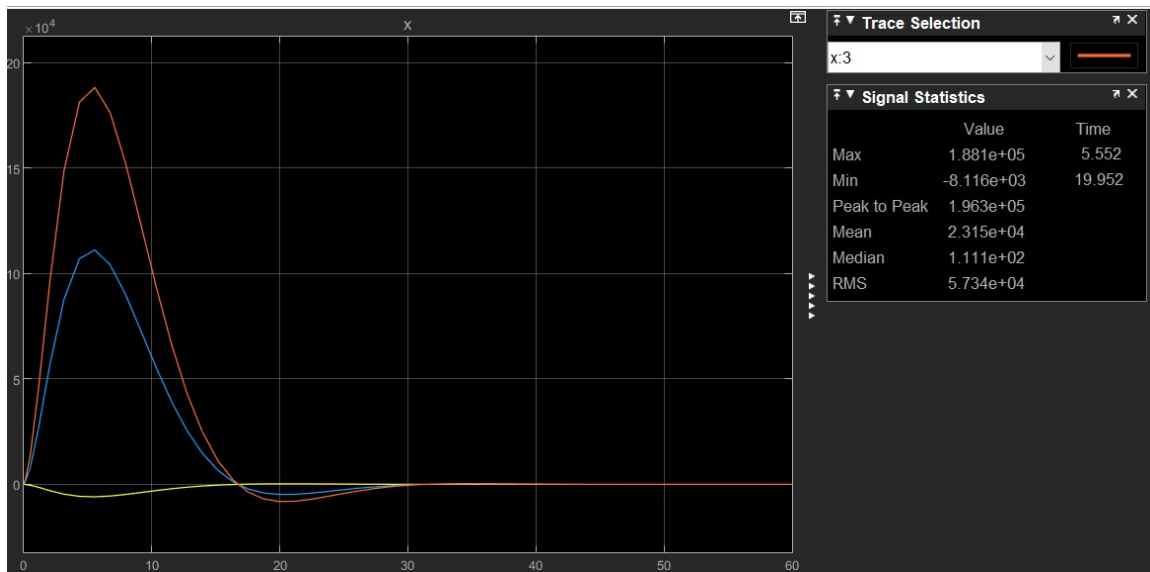


Figure 2.4 Increase  $\zeta$  from 0.65 to 0.7

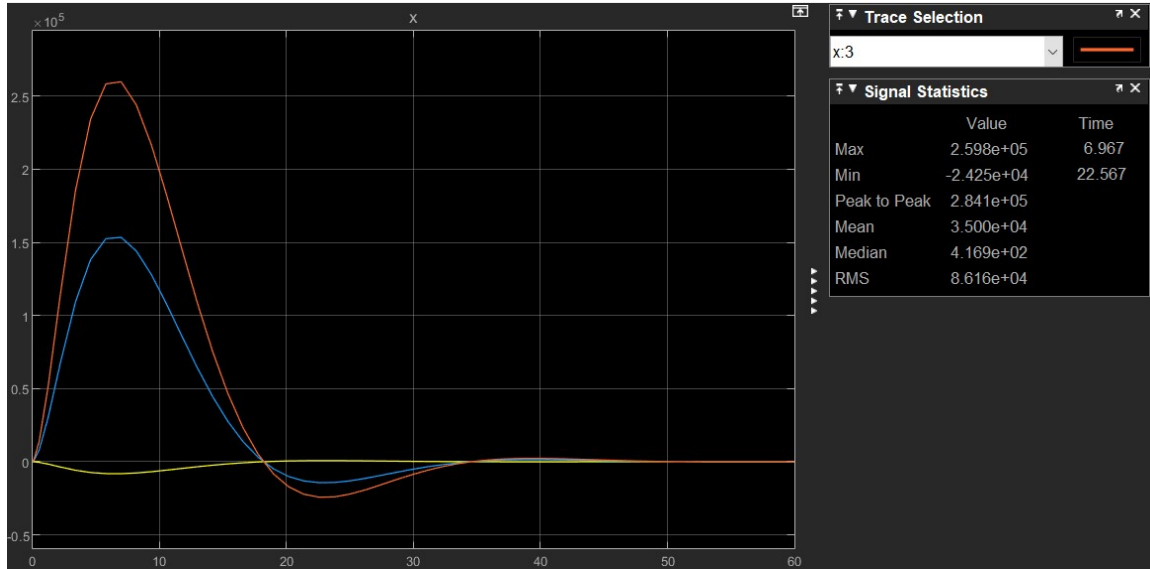


Figure 2.5 Decrease  $\zeta$  form 0.65 to 0.6

## 2.3 LQR Controller

### 2.3.1 Controller Design

Assume that we can measure all the three state variables, we need to design a state feedback controller using the LQR method. Assume that the matrix  $Q$  and  $R$  is:

$$Q = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 500 & 0 \\ 0 & 0 & 1000 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.1 & 0 \\ 0 & 100 \end{bmatrix}$$

Then we need minimize the following quadratic cost function.

$$J = \frac{1}{2} \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (2-10)$$

The optimal control law turns out to be in the form of linear state feedback:

$$u = -Kx \quad (2-11)$$

Let's try the Lyapunov method this time. Let's define Lyapunov function

$$V(x) = x^T P x \quad (2-12)$$

where  $P$  is a positive definite matrix.  $P > 0$ . And we will find out an equation for  $P$  such that

$$\frac{dV(x)}{dt} \leq 0 \quad (2-13)$$

$$\frac{dV(x)}{dt} = x^T (A^T P + P A + Q - P B R^{-1} B^T P) x + (B^T P x + R u)^T R^{-1} (B^T P x + R u) - x^T Q x - u^T R u$$

In order to make the first term on the RHS to be zero, we choose  $P$  such that



$$A^T P + PA + Q = PBR^{-1}BP \quad (2-14)$$

We can use systematic way of eigenvalue-eigenvector based algorithm to solve the *ARE* equation. First, let define a  $2n \times 2n$  matrix:

$$\Gamma = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} = \begin{bmatrix} -1.7 & -0.25 & 0 & -250 & 4150 & 0 \\ 23 & -30 & 20 & 4150 & -68890 & 0 \\ 0 & -490 & -1140 & 0 & 0 & -5929 \\ -10 & 0 & 0 & 1.7 & -23 & 0 \\ 0 & -500 & 0 & 0.25 & 30 & 490 \\ 0 & 0 & -1000 & 0 & -20 & 140 \end{bmatrix} \quad (2-15)$$

Then, let the eigenvector of  $\Gamma$  corresponding to stable  $\lambda_i$ ,  $i = 1, 2, 3$ .

$$\begin{bmatrix} v \\ \mu \end{bmatrix} = \begin{bmatrix} -0.0591 & 0.0755 & -0.0182 \\ 0.9865 & 0.0041 & 0.2997 \\ 0.1269 & -0.0013 & 0.9245 \\ 0 & 0.9953 & 0 \\ 0.0829 & 0.06 & 0.0123 \\ 0.0185 & 0 & 0.2345 \end{bmatrix} \quad (2-16)$$

$$P = \mu v^{-1} = \begin{bmatrix} 13.1426 & 0.7872 & 0.0041 \\ 0.7872 & 0.1330 & -0.0143 \\ 0.0041 & -0.0143 & 0.2584 \end{bmatrix}$$

We can get the feedback matrix  $K$ :

$$K = -R^{-1}B^T P = \begin{bmatrix} 3.7776 & -71.0203 & 12.0312 \\ -0.0313 & 0.1097 & -1.9896 \end{bmatrix} \quad (2-17)$$

The controller can be expressed as:

$$u = -Kx + Ir$$

### 2.3.2 Simulate

The initial state is  $x_0$ , we can use Simulink to simulate the designed system. My simulate module is showed in the following figure 2.6.

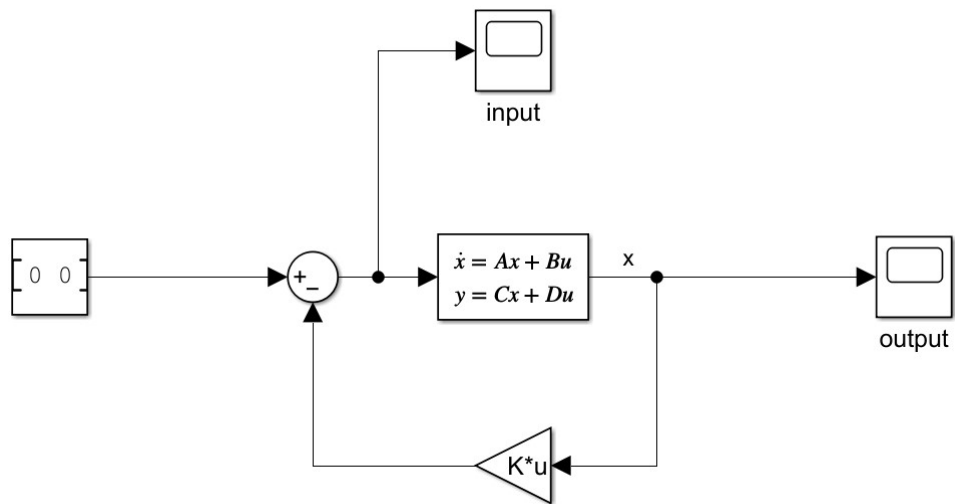


Figure 2.6 LQR module

All the three state responses to non-zero initial state  $x_0$  with zero external inputs are shown in figure 2.7. The control signal is shown in figure 2.8. The  $Q$  and  $R$  matrix are:

$$Q = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 500 & 0 \\ 0 & 0 & 1000 \end{bmatrix}, R = \begin{bmatrix} 0.1 & 0 \\ 0 & 100 \end{bmatrix}.$$

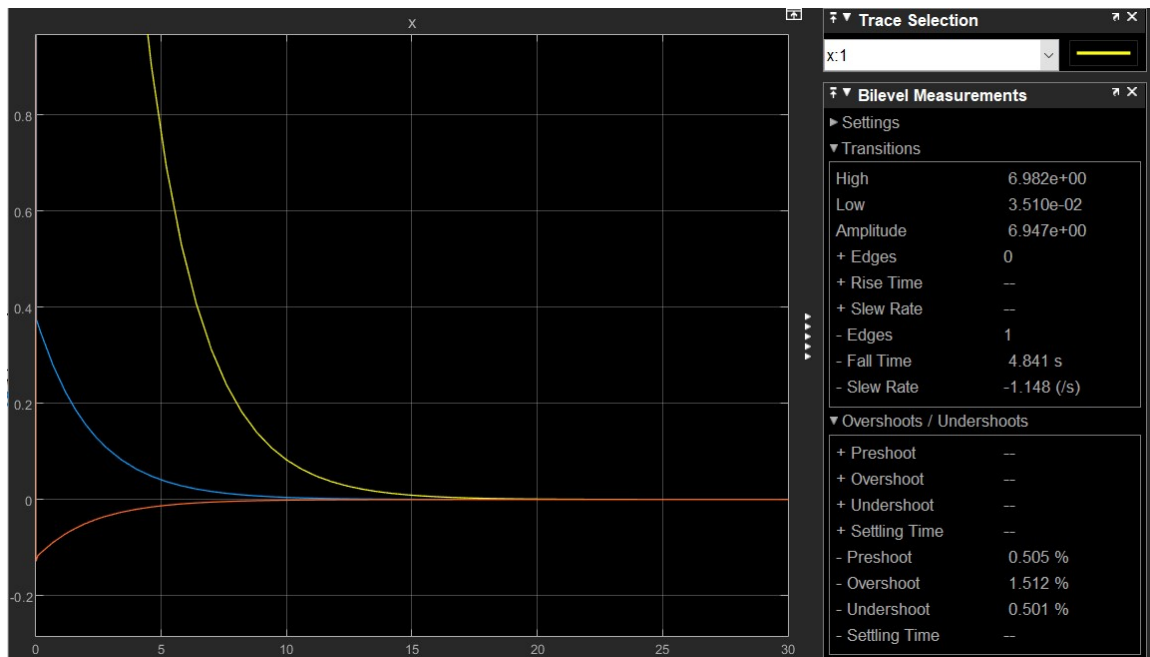


Figure 2.7 State response

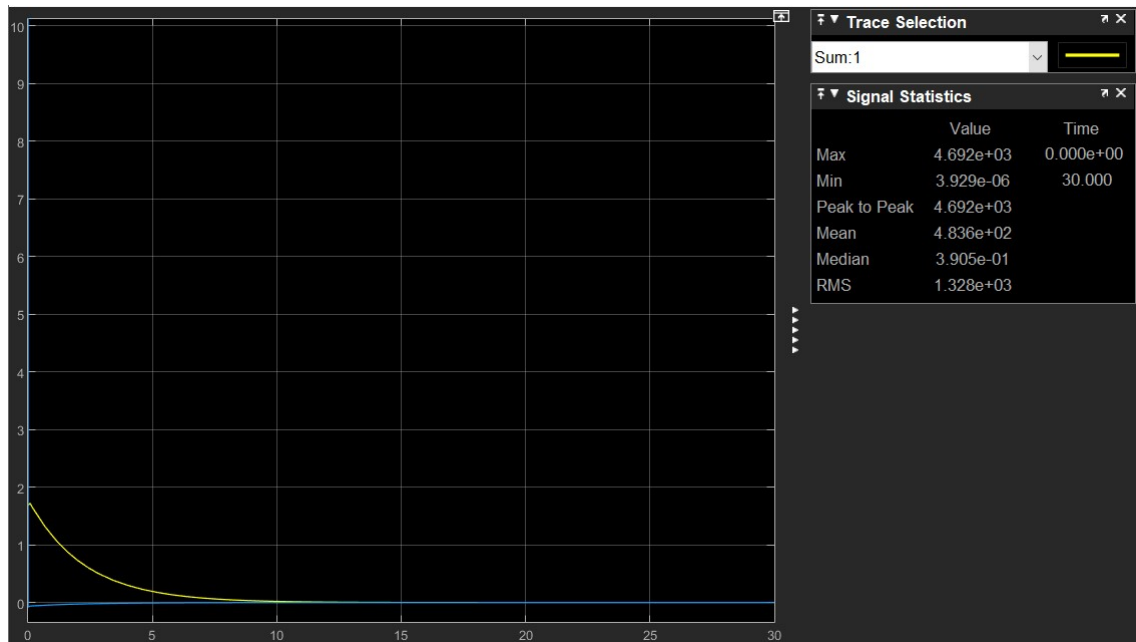


Figure 2.8 Control signal

If we increase the number in  $Q$ , The setting time of the system will be shorter and the input signal will be larger. If we increase the number in  $R$ , The setting time of the system will be longer and the input signal will be smaller. Three number in matrix  $Q$  corresponds to three state separately. Similarly, two number in matrix  $R$  corresponds to two input signal separately. Simulation results are shown in the figure below.

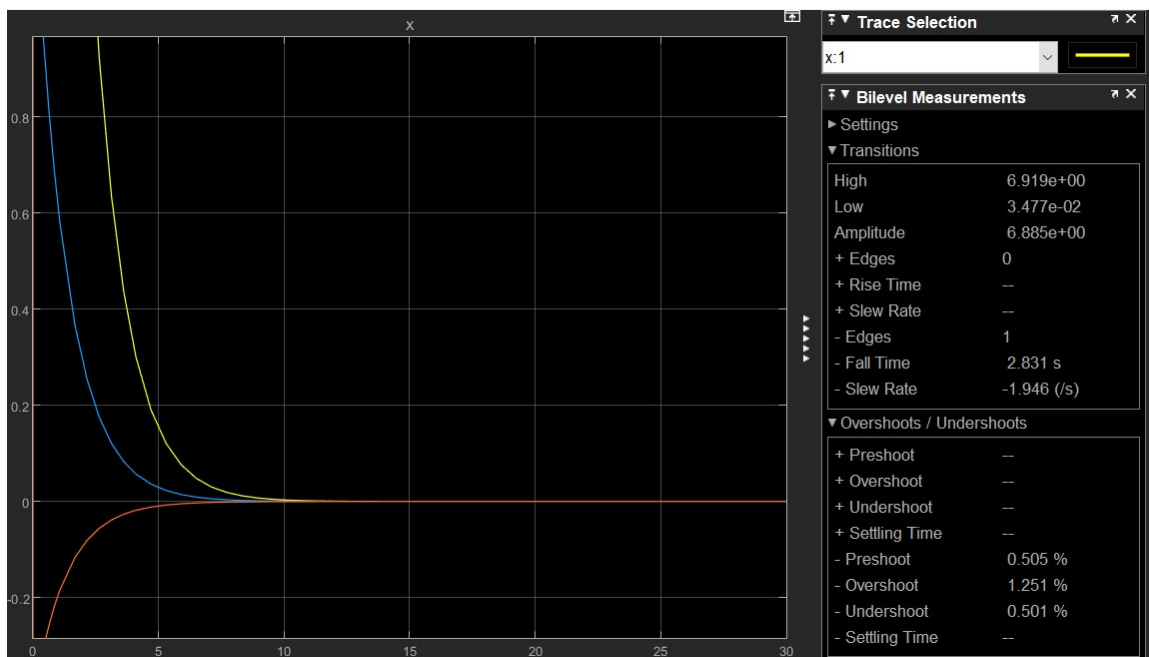


Figure 2.9 increace  $Q_{11}$  from 10 to 50

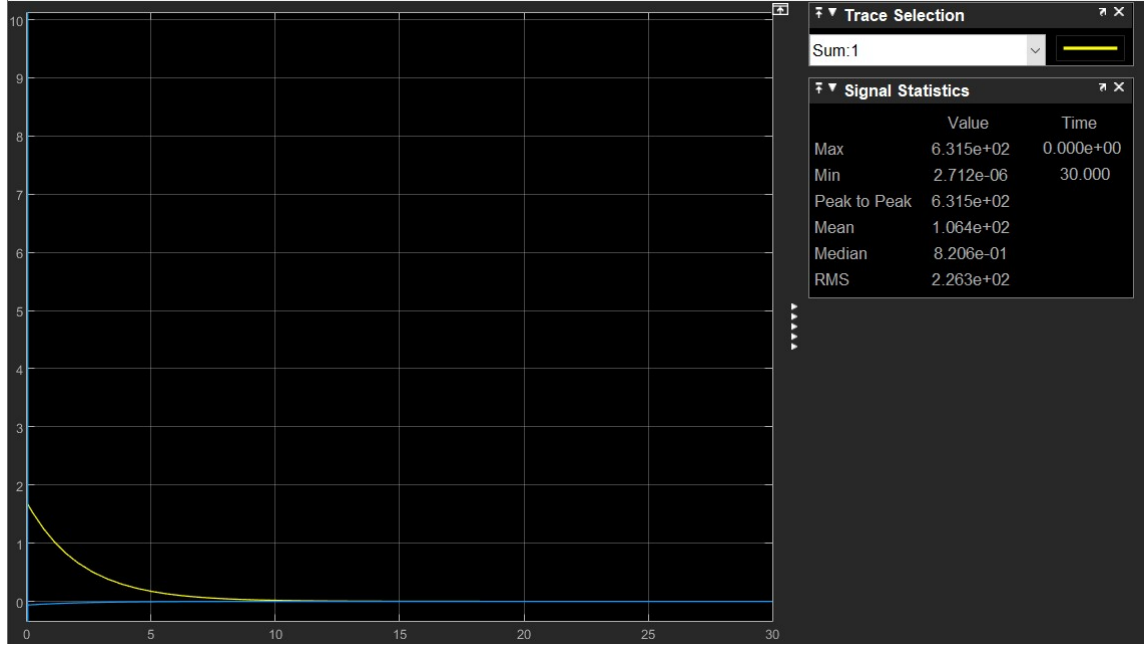


Figure 2.10 increase  $R_{11}$  from 0.1 to 10

## 2.4 State observer

### 2.4.1 Observer Design

In this task we still use LQR controller. Because LQR need state feedback, but we can only measure two of them. We should design an observer to estimate the other state. Here, I choose reduce-order observer to complete this task. Let  $\xi = Tx$ , so that  $\hat{x} = \begin{bmatrix} C \\ T \end{bmatrix}^{-1} \begin{bmatrix} y \\ \xi \end{bmatrix}$ . The estimation error  $e$  and  $\xi$  can be expressed as:

$$\begin{aligned} \dot{\xi} &= D\xi + Eu + Gy \\ e &= \xi - Tx \end{aligned} \quad (2-18)$$

In this system,  $d$  is a scalar. Let  $d = -3$ ,  $T = [t_1 \ t_2 \ t_3]$ ,  $G = [g_1 \ g_2]$ , we need to design  $T$  and  $G$ , so that

$$\begin{aligned} dT - TA + GC &= 0 \\ T &= (A - dI)^{-1}GC \end{aligned} \quad (2-19)$$

Let  $G = [10 \ 20]$ , then we can get  $T = [0.7831 \ -0.0443 \ -0.0184]$ . The matrix  $\begin{bmatrix} C \\ T \end{bmatrix}$  is nonsingular.  $E = TB = [7.5894 \ 14.1439]$ . The observer is thus determined as

$$\dot{\xi} = -3\xi + [7.5894 \ 14.1439]u + [10 \ 20]y \quad (2-20)$$

Then reconstruct the state variables from the plant's output and the observer's output as:

$$\hat{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.7831 & -0.0443 & -0.184 \end{bmatrix}^{-1} \begin{bmatrix} y(t) \\ \xi(t) \end{bmatrix} = \begin{bmatrix} 0.0565 & 0.0235 & 1.2770 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y(t) \\ \xi(t) \end{bmatrix} \quad (2-21)$$

## 2.4.2 Simulate

The initial state is  $x_0$ , we can use Simulink to simulate the designed system. My simulate module is showed in the following figure 2.11.

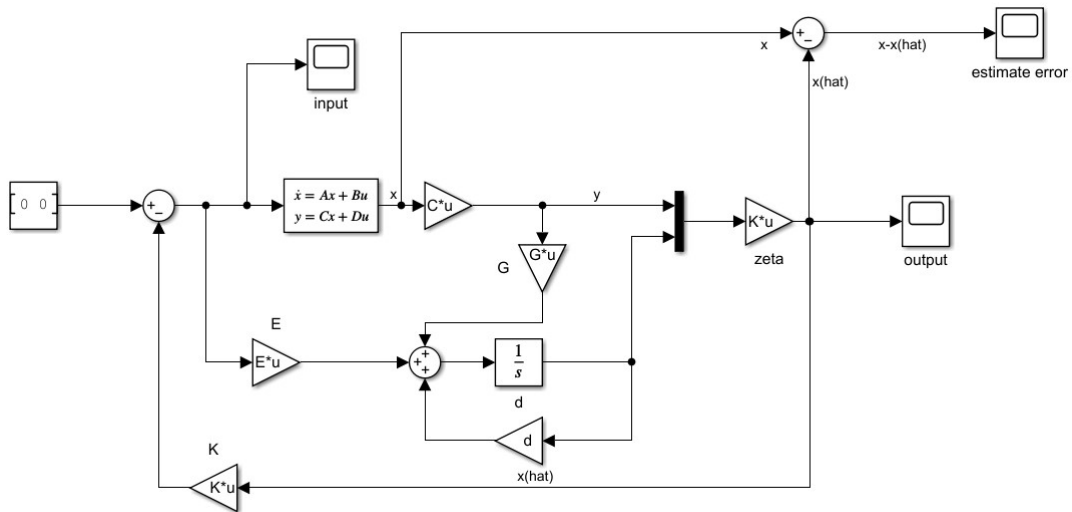


Figure 2.11 Reduce-order observer

The estimation error is shown in figure 2.12

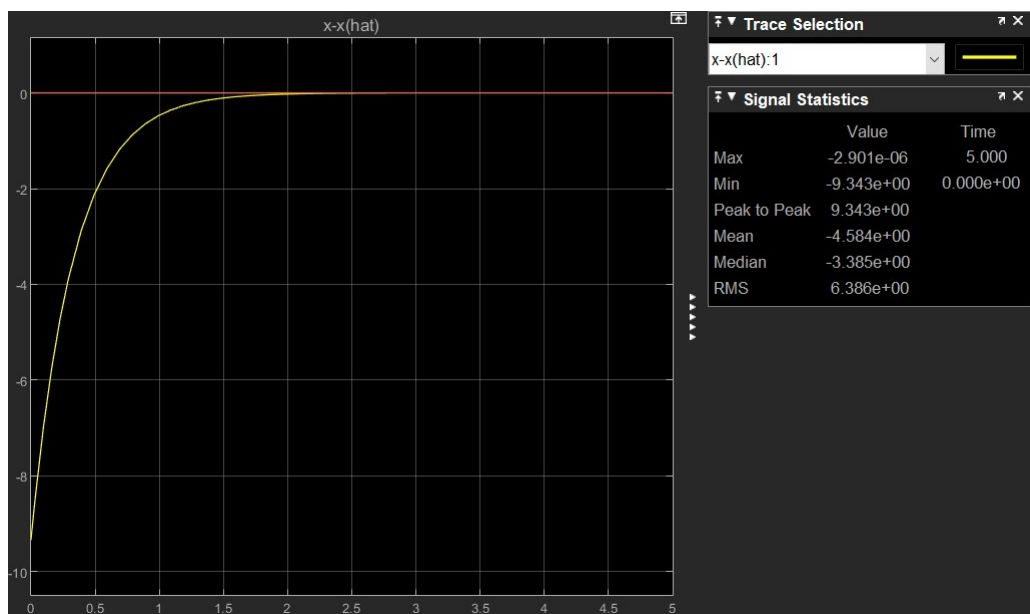


Figure 2.12 Estimation error

If we increase  $D$ , the estimation error will go to zero much faster. Otherwise, it will slower

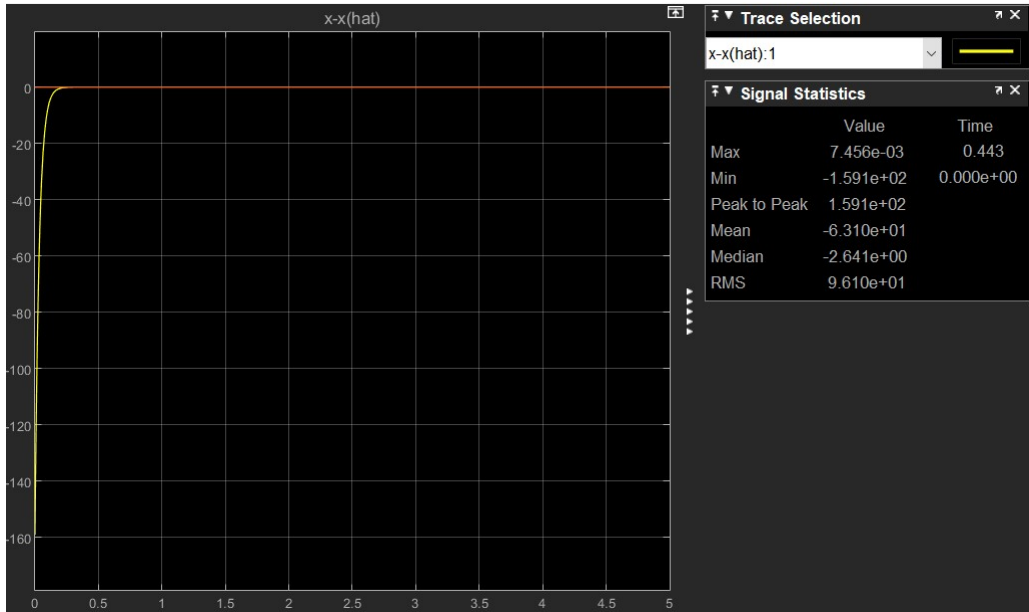


Figure 2.13 d=-30

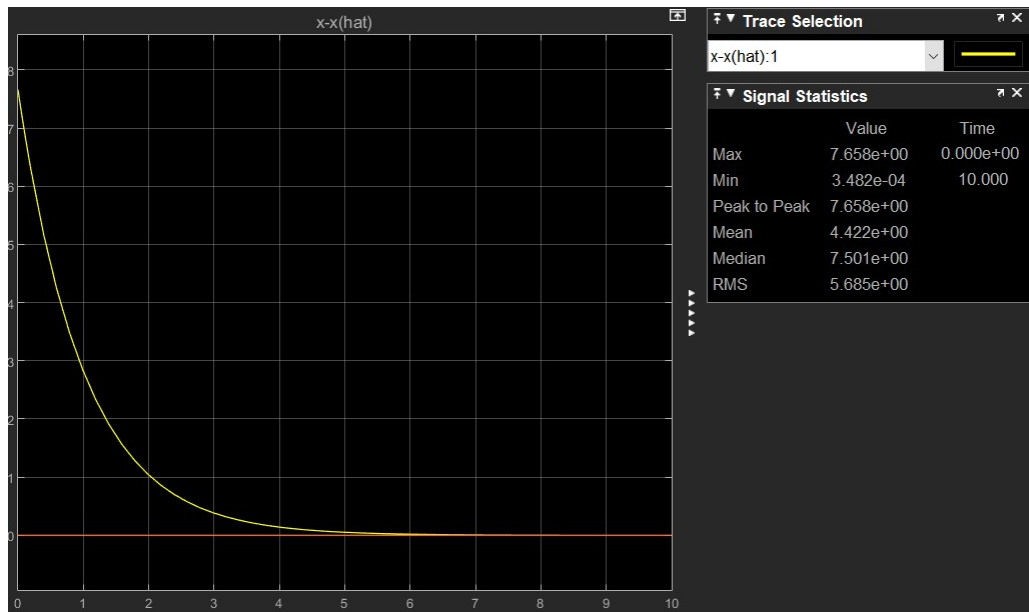


Figure 2.14 d=-1

## 2.5 Decoupling Controller

### 2.5.1 Controller Design

In this task, we need to design a decoupling controller with closed-loop stability.  $c_i^T$  represents the  $i$  rows of the matrix  $C$ . Because  $c_1^T B = [-80 \ 0]$ ,  $c_2^T B = [0 \ -770]$ ,  $\sigma_1 = 1$ ,  $\sigma_2 = 1$ . The transfer function can be written as

$$G(s) = \begin{bmatrix} s^{-\sigma_1} & 0 \\ 0 & s^{-\sigma_2} \end{bmatrix} [B^* + C^{**}(sI - A)^{-1}B] \quad (2-22)$$

$$H(s) = C[I + K(sI - A)^{-1}B]^{-1}F$$

Where  $B^* = \begin{bmatrix} c_1^T B \\ c_2^T B \end{bmatrix} = \begin{bmatrix} -80 & 0 \\ 0 & -770 \end{bmatrix}$ , and we can place two poles,  $s_1 = -3$ ,  $s_2 = -10$ , so  $\phi_{f_1} = s + 3$ ,  $\phi_{f_2} = s + 10$ , then we can calculate  $C^{**}$ ,  $F$  and  $K$

$$\begin{aligned} C^{**} &= \begin{bmatrix} c_1^T (A + 3I) \\ c_2^T (A + 10I) \end{bmatrix} = \begin{bmatrix} 23 & -27 & 20 \\ 0 & -490 & -1130 \end{bmatrix} \\ K &= (B^*)^{-1} C^{**} = \begin{bmatrix} -0.2771 & 0.3253 & -0.2410 \\ 0 & 0.6364 & 1.4675 \end{bmatrix} \\ F &= (B^*)^{-1} = \begin{bmatrix} -0.0120 & 0 \\ 0 & -0.0013 \end{bmatrix} \end{aligned} \quad (2-23)$$

Take (2-23) into (2-22), we can get:

$$H(s) = \begin{bmatrix} \frac{1}{s+3} & 0 \\ 0 & \frac{1}{s+10} \end{bmatrix}$$

And the eigenvalue of  $A - BK$  is:  $s_1 = -0.3145$ ,  $s_2 = -3$ ,  $s_3 = -10$ , all of them are negative, so the system is internally stable.

## 2.5.2 Simulate

My simulate module is shown in the following figure:

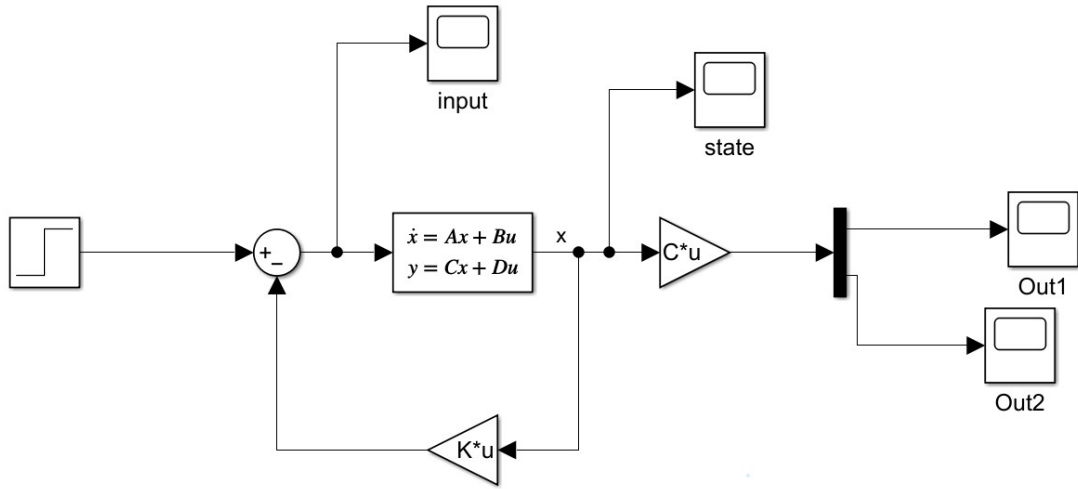


Figure 2.15 Decoupling controller

The step response with zero initial states are as follow:

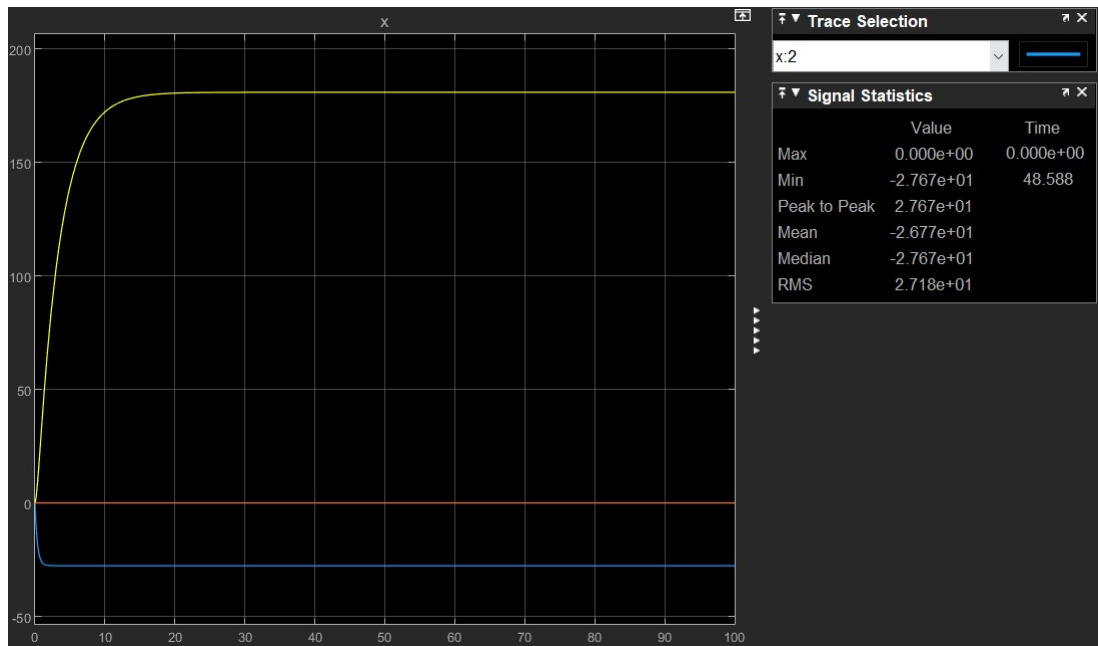


Figure 2.16 Step response( $u=[1 \ 0]^T$ )

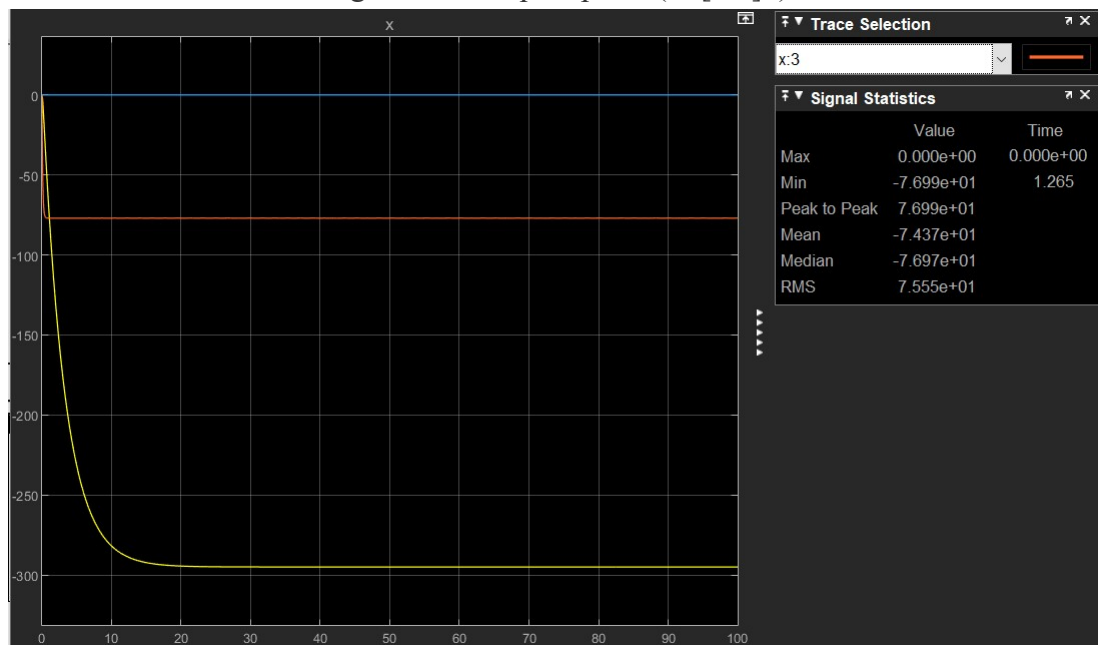


Figure 2.17 Step response( $u=[0 \ 1]^T$ )

The initial response with respect to  $x_0$  is as follow:



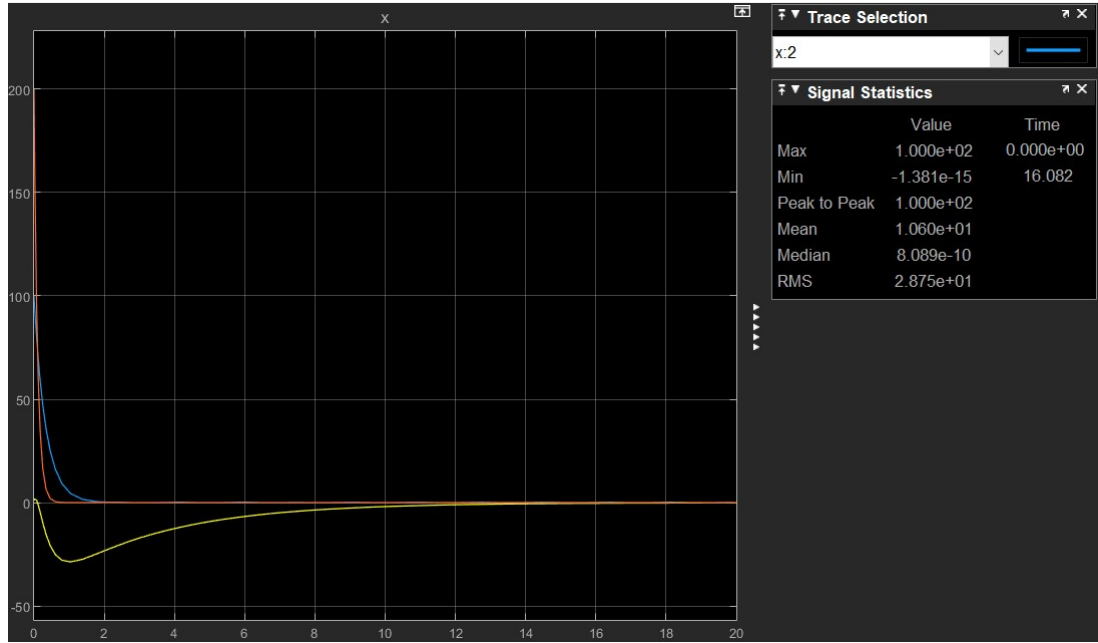


Figure 2.18 Initial response

## 2.6 Servo Controller

### 2.6.1 Controller Design

In an application, the operating set point for the two output is  $y_{sp} = [100, 150]^T$ , and we can only measure the output. The step disturbance  $w = [-2, 5]^T$  take effect from time  $t_d = 10$  s afterwards. Let

$$v(t) = \int_0^t e(\tau) d\tau \quad (2-24)$$

$$\dot{v}(t) = e(t) = r - y(t) = r - Cx(t)$$

Equation (1-1) and (2-24) form an augmented system:

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} B \\ 0 \end{bmatrix} w + \begin{bmatrix} 0 \\ I \end{bmatrix} r \quad (2-25)$$

$$\begin{aligned} \dot{\tilde{x}} &= \tilde{A}\tilde{x} + \tilde{B}u + \tilde{B}w + \tilde{B}_r r \\ y &= [C \quad 0] \begin{bmatrix} x \\ v \end{bmatrix} \end{aligned} \quad (2-26)$$

The rank of the controllability matrix is

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

So, the augmented system is controllable. Then we can use LQR to get the feedback gain  $K$ . Let  $Q = \text{diag}(10, 10, 10, 15, 15)$ ,  $R = \text{diag}(10, 10)$ , then calculate  $K$  via equation (2-15), (2-16) and (2-17).

$$K = \begin{bmatrix} 0.2968 & -0.7036 & 0.0017 & 1.1640 & -0.3811 \\ -0.1162 & 0.0088 & -0.3068 & 0.3811 & 1.1640 \end{bmatrix}$$

$$u = -K\bar{x} = -[K_1 \quad K_2] \begin{bmatrix} x \\ v \end{bmatrix} \quad (2-27)$$

And because we can not get the state  $x$  directly, we need to design an observer to estimate  $x$ . We can use full-order observer to estimate the state. The observer can be expressed as:

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + Bw(t) + L(y - \hat{y}) \\ \hat{y} &= C\hat{x} \end{aligned} \quad (2-28)$$

The poles of the controller are the eigenvalue of  $A - BK_1$ ,  $s_1 = -1.57$ ,  $s_2 = -97.61$ ,  $s_3 = -1368.6$ , we can choose the observer 3-5 times faster than controller poles, let  $s_1 = -5$ ,  $s_2 = s_3 = -10$ . We need to put these poles to  $A - LC$ , which has a similar form with  $A - BK$ . Because  $(A - LC)^T = A^T - C^T L^T$ , we can calculate  $L^T$  by using equation (2-6), (2-7), (2-8) and (2-9).

$$L^T = \begin{bmatrix} 0.7452 & -31.7 & -491 \\ -13.30 & 200 & -1115 \end{bmatrix}$$

$$L = \begin{bmatrix} 0.7452 & -13.30 \\ -31.7 & 200 \\ -491 & -1115 \end{bmatrix}$$

## 2.6.2 Simulate

My simulate module is shown in the following figure.

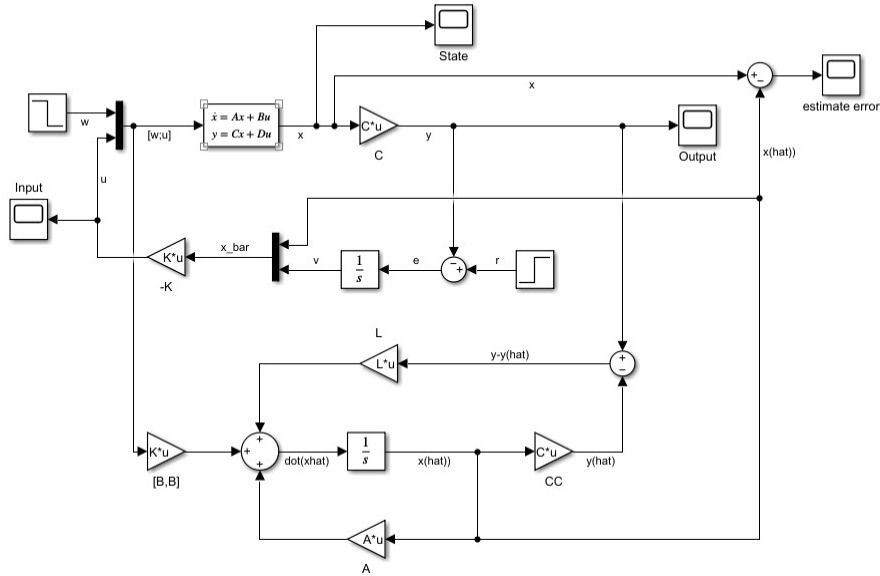


Figure 2.19 Servo controller and full-order observer

The control signal, output signal and the estimation is shown in the following figures.

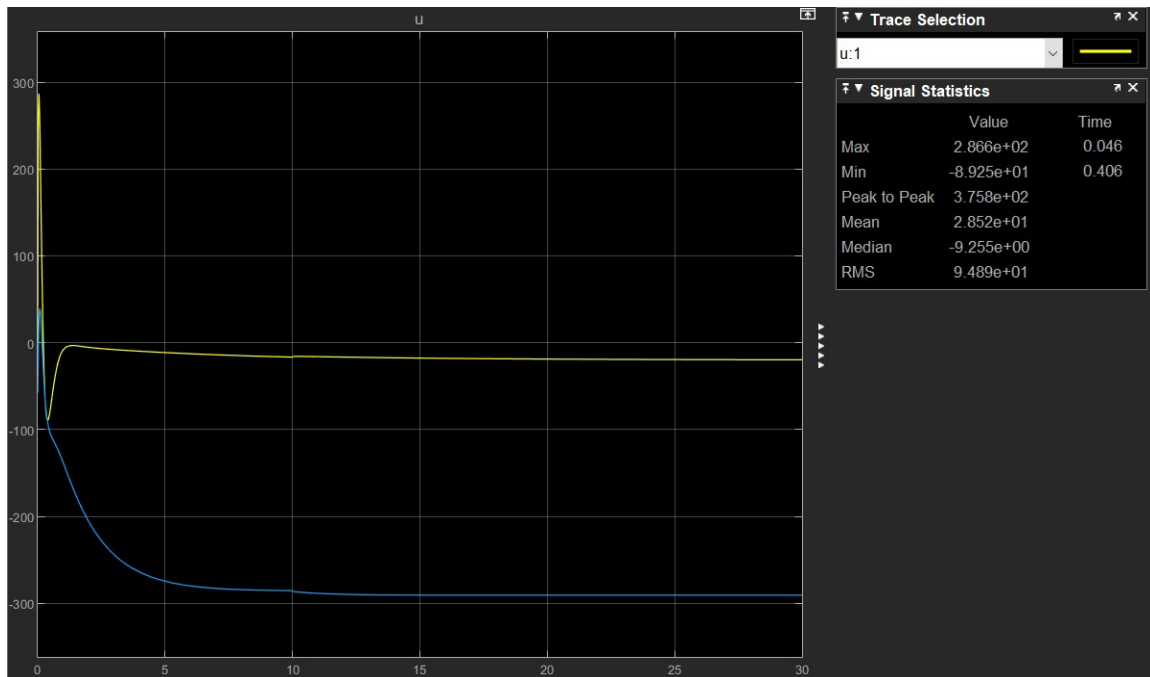


Figure 2.20 Control signal

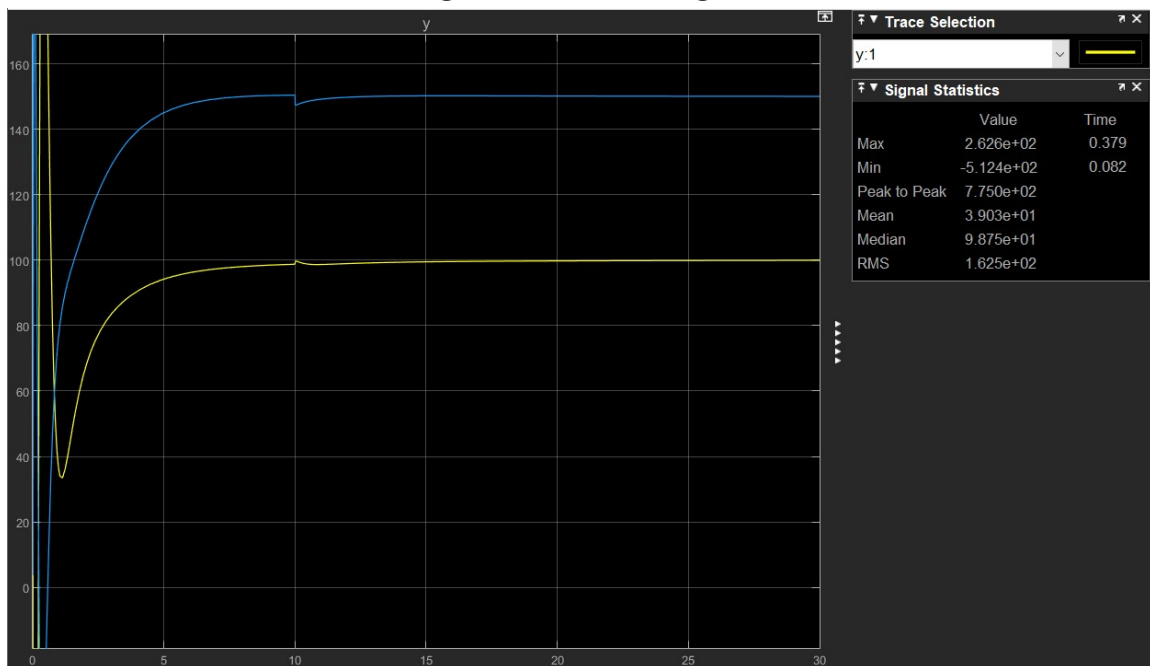


Figure 2.21 Output signal

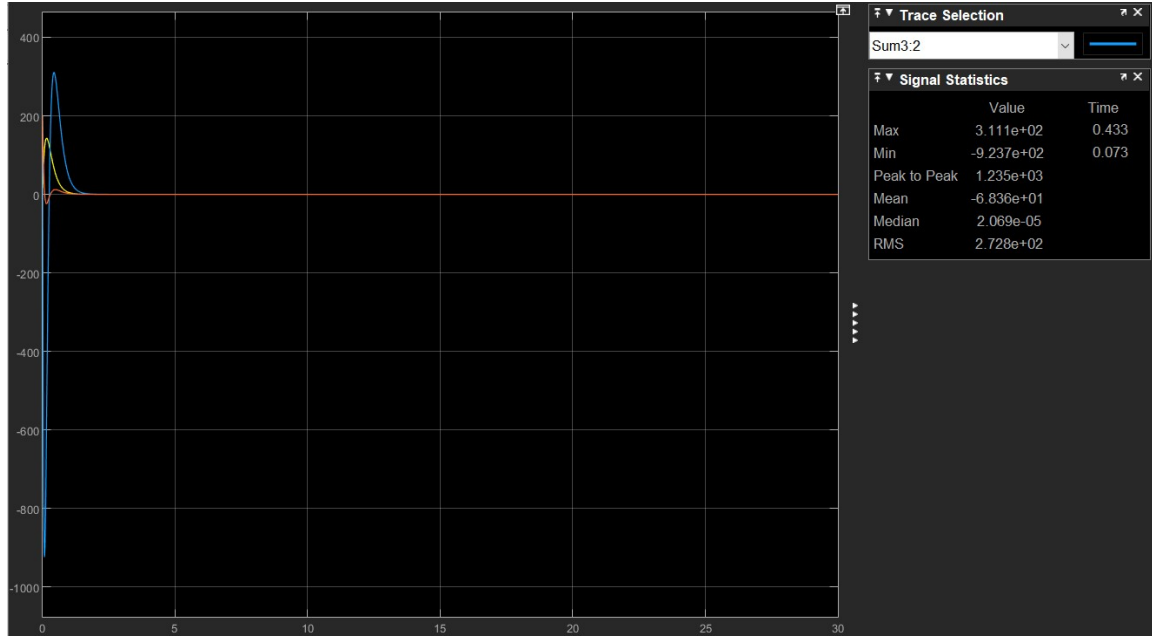


Figure 2.22 Estimation error

## 2.7 Minimize objective function

If the set point is  $x_{sp} = [5, 250, 300]^T$ , we can not achieve this point at steady. Because all the state can be measured, let  $C_1 = I$ , the state space will change to

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= C_1 x\end{aligned}$$

Let

$$\begin{aligned}x^* &= \lim_{t \rightarrow \infty} x(t) \\ u^* &= \lim_{t \rightarrow \infty} u(t)\end{aligned} \tag{2-29}$$

For asymptomatic tracking,  $x^*$  and  $u^*$  must satisfy the equation

$$\begin{bmatrix} A & B \\ C_1 & 0 \end{bmatrix} \begin{bmatrix} x^* \\ u^* \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix} y_d \tag{2-30}$$

Equation (2-30) can be solved uniquely for  $\begin{bmatrix} x^* \\ u^* \end{bmatrix}$  if and only if  $\begin{bmatrix} A & B \\ C_1 & 0 \end{bmatrix}$  is nonsingular. But the matrix  $\begin{bmatrix} A & B \\ C & 0 \end{bmatrix}$  is not square matrix, we can not solve the equation (2-30).

Then, let minimize the following objective function:

$$J(x_s) = \frac{1}{2} (x_s - x_{sp})^T W (x_s - x_{sp}) \tag{2-31}$$

Where  $W = \text{diag}(12, 7, 10)$ . When the system is stable,  $\dot{x}$  will be zero. Let  $x = -Kx + Fr$ , where  $F$  is a Identity matrix,  $K$  is the feedback matrix from (2-17)

$$\dot{x}_s = (A - BK)x_s + Br = 0$$

$$x_s = (BK - A)^{-1}Br \quad (2-32)$$

Let take (2-32) into (2-31), the equation will change to

$$J(r) = \frac{1}{2}((BK - A)^{-1}Br - x_{sp})^T W ((BK - A)^{-1}Br - x_{sp}) \quad (2-33)$$

Where  $r = [r_1 \ r_2]^T$ , if we want minimize this function, the following equation should be zero

$$\begin{aligned} \frac{dJ}{dr_1} &= 0 \\ \frac{dJ}{dr_2} &= 0 \end{aligned} \quad (2-34)$$

Then the matrix  $r$  can be obtained:  $r = [-9589.8 \ -1126.3]^T$ . The simulation result is shown in following figure.

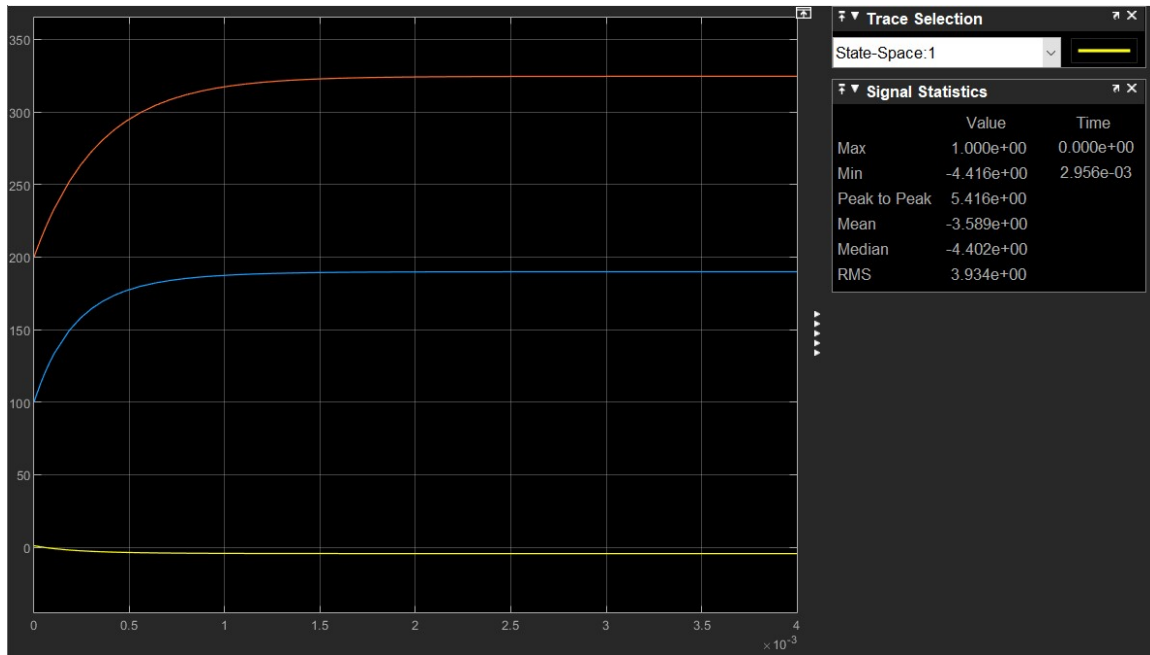


Figure 2.23 Output signal

### *Section III Conclusion*

In this project, we design the controller via different methods to satisfy the design specifications. First, we calculate the desired poles which satisfy the transient response performance: 1) The overshoot is less than 10%; 2) The 2% setting time is less than 30 seconds.

In task 1, we use pole place method design the controller. And discuss the effects if the position of the poles on system performance. If the real part of the domain poles goes away from original. The setting time of the system will be shorter and the input signal will be larger.

In task 2, we use LQR method design the controller, and design the weight matrix  $Q$  and  $R$ . After that, we discuss the effects of weighting  $Q$  and  $R$ . If we increase the number in  $Q$ , The setting time of the system will be shorter and the input signal will be larger. If we increase the number in  $R$ , The setting time of the system will be longer and the input signal will be smaller.

In task 3, we design a reduce-order observer, and investigate the effects of observer poles. If the real part of the domain poles goes away from original, the estimation error will go to zero much faster. Otherwise, it will slower.

In task 4, we design a decoupling controller with closed-loop stability. Because all the eigenvalues of  $A - BK$  are negative, the system is internally stable.

In task 5, we use servo controller to achieve the set point  $y_{sp} = [100, 150]^T$ . Because we can only measure two of the state, we design a state observer also.

In task 6, we want the state will go to  $x_{sp} = [5, 250, 300]^T$  at steady state, and prove that is impossible. So we try to minimize to function  $J(x_s)$ .

## Appendices

Matlab code:

### Task 1: Pole Placement

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%Author: Liu Weihao
%%Matriculation number: A0232935A
%%File:Q1_code
%%Date: 14 NOV 2021
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A=[-1.7,-0.25,0;23,-30,20;0,-200-290,-220-920];
B=[3+2,0;-30-53,0;0,-420-350];
C=[0,1,0;0,0,1];
D=zeros(2,2);
C1=eye(3);
D1=zeros(3,2);
x0=[1;100;200];
omega_n=0.24;
zeta=0.6;
s0=-zeta*omega_n-omega_n*sqrt(zeta^2-1);
s1=-zeta*omega_n+omega_n*sqrt(zeta^2-1);
s2=-0.64;
s=[s0,s1,s2];

b1=B(:,1);
b2=B(:,2);
Wc=[B,A*B,A^2*B];
%choose b1,A*b1,b2
C_1=[b1,A*b1,b2];
C_t=inv(C_1);
%d1=2,d2=1,T=[q2;q2*A;q3]
q2=C_t(2,:);
q3=C_t(3,:);
T=[q2;q2*A;q3];
Aba=T*A/T;
Bba=T*B;
Ad=[0,1,0;0,0,1;s0*s1*s2,-(s0*s1+s0*s2+s1*s2),s0+s1+s2];
% Ad=[s2,0,0;0,0,1;0,-(s0*s1),s0+s1];
Kba=Bba\ (Aba-Ad);
K=Kba*T;
```

## Task 2: LQR controller

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%Author: Liu Weihao
%%Matriculation number: A0232935A
%%File: Q2_code
%%Date: 14 NOV 2021
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A=[-1.7,-0.25,0;23,-30,20;0,-200-290,-220-920];
B=[3+2,0;-30-53,0;0,-420-350];
C=[0,1,0;0,0,1];
D=zeros(2,2);
C1=eye(3);
D1=zeros(3,2);
x0=[1;100;200];

qq=[10,500,1000];
rr=[0.1,100];
Q=diag(qq);
R=diag(rr);
Gamma=[A,-B/R*B';-Q,-A'];
[x,y]=eig(Gamma);
y=diag(y);
k=1;
for i=1:length(y)
    if(y(i)<0)
        v(:,k)=x(1:3,i);
        mu(:,k)=x(4:6,i);
        k=k+1;
    end
end
P=mu/v;
K=R\B'*P;
```



### Task 3: Reduce-order Observer

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%Author: Liu Weihao
%%Matriculation number: A0232935A
%%File: Q3_code
%%Date: 14 NOV 2021
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A=[-1.7,-0.25,0;23,-30,20;0,-200-290,-220-920];
B=[3+2,0;-30-53,0;0,-420-350];
C=[0,1,0;0,0,1];
D=zeros(2,2);
C1=eye(3);
D1=zeros(3,2);
x0=[1;100;200];

qq=[10,500,1000];
rr=[0.1,100];
Q=diag(qq);
R=diag(rr);
Gamma=[A,-B/R*B';-Q,-A'];
[x,y]=eig(Gamma);
y=diag(y);
k=1;
for i=1:length(y)
    if(y(i)<0)
        v(:,k)=x(1:3,i);
        mu(:,k)=x(4:6,i);
        k=k+1;
    end
end
P=mu/v;
K=R\B'*P;

d=-3;
G=[10,20];
left=A-d*eye(3);
right=G*C;
T=right/left;
E=T*B;
zeta=inv([C;T]);
```

#### Task 4: Decoupling System

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%Author: Liu Weihao
%%Matriculation number: A0232935A
%%File: Q4_code
%%Date: 14 NOV 2021
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

syms s
A=[-1.7,-0.25,0;23,-30,20;0,-200-290,-220-920];
B=[3+2,0;-30-53,0;0,-420-350];
C=[0,1,0;0,0,1];
D=zeros(2,2);
C1=eye(3);
D1=zeros(3,2);
x0=[1;100;200];

C1T=C(1,:);
C2T=C(2,:);

s0=-3;
s1=-10;

B_s=[C1T*B;C2T*B];
C_s=[C1T*A;C2T*A];
C_ss=[C1T*(A-s0*eye(3));C2T*(A-s1*eye(3))];
K=B_s\C_ss;
F=inv(B_s);
H=C*(s*eye(3)-(A-B*K))^( -1)*B*F;

[x,y]=eig(A-B*K)
```

## Task 5: Servo Control

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%Author: Liu Weihao
%%Matriculation number: A0232935A
%%File: Q5_code
%%Date: 14 NOV 2021
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A=[-1.7,-0.25,0;23,-30,20;0,-200-290,-220-920];
B=[3+2,0;-30-53,0;0,-420-350];
C=[0,1,0;0,0,1];
BB=[B,B];
x0=[1;100;200];

Abar=[A,zeros(3,2);-C,zeros(2,2)];
Bbar=[B;zeros(2,2)];

qq=[10,10,10,15,15];
rr=[10,10];
Q=diag(qq);
R=diag(rr);
Gamma=[Abar,-Bbar/R*Bbar';-Q,-Abar'];
[x,y]=eig(Gamma);
y=diag(y);
k=1;
for i=1:length(y)
    if(y(i)<0)
        v(:,k)=x(1:5,i);
        mu(:,k)=x(6:10,i);
        k=k+1;
    end
end
P=mu/v;
K=R\Bbar'*P;

s0=-5;
s1=-10;
s2=-10;

At=A';
Ct=C';

b1=Ct(:,1);
b2=Ct(:,2);
Wc=[Ct,At*Ct,At^2*Ct];
%choose b1,A*b1,b2
C_1=[b1,At*b1,b2];
C_t=inv(C_1);
%d1=2,d2=1,T=[q2;q2*A;q3]
q2=C_t(2,:);
q3=C_t(3,:);
T=[q2;q2*At;q3];
```

```
Aba=T*At/T;  
Bba=T*Ct;  
Ad=[0,1,0;0,0,1;s0*s1*s2,-(s0*s1+s0*s2+s1*s2),s0+s1+s2];  
Lba=Bba\ (Aba-Ad);  
Lt=Lba*T;  
L=Lt';
```

## Task 6: Minimize objective function

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%Author: Liu Weihao
%%Matriculation number: A0232935A
%%File: Q6_code
%%Date: 14 NOV 2021
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A=[-1.7,-0.25,0;23,-30,20;0,-200-290,-220-920];
B=[3+2,0;-30-53,0;0,-420-350];
C=[0,1,0;0,0,1];
D=zeros(2,2);
x0=[1;100;200];
W=diag([2+9+1,3+4,5+5]);

syms r1 r2

qq=[10,500,1000];
rr=[0.1,100];
Q=diag(qq);
R=diag(rr);
Gamma=[A,-B/R*B';-Q,-A'];
[x,y]=eig(Gamma);
y=diag(y);
k=1;
for i=1:length(y)
    if(y(i)<0)
        v(:,k)=x(1:3,i);
        mu(:,k)=x(4:6,i);
        k=k+1;
    end
end
P=mu/v;
K=R\B'*P;

R=[r1;r2];
Xs=(B*K-A)\B*R;
Xsp=[5;250;300];
J=0.5*(Xs-Xsp)'\*W*(Xs-Xsp);
J1=diff(J,r1);
J2=diff(J,r2);
[R1,R2]=vpasolve([J1==0,J2==0],[r1,r2]);
R=[R1;R2];
```