Multivariable Controller Design for Diesel Engine Air System Control

Assignment For

EE5101/ME5401 Linear systems

Submitted by

Fan Xiuqi A0285060J

Email:e1192618@u.nus.edu

Mechanical Engineering
Faculty of
National University of Singapore

Session 2023/2024

Abstract

Diesel Engine Air System Control plays an important role in diesel engines, which can monitor and control the emissions especially oxides of nitrogen (NOx) and particulate matter (PM). In this report, serval methods are used to realize such a dynamic system, such as pole placement, LQR, controller with observer, decoupling, etc. The dynamic system is built by Simulink and calculated by MATLAB. All the methods mentioned above are proved to be effective to solve certain problems in this controller.

Content

1. Introduction

With the development of the society, the environment issue has become a hot topic, and more and more attentions are paid to it. More severe actions are taken to reduce the emissions of environmentally harmful gas. Diesel energy is one of the important resources, thus more actions should be taken to limit the wasted gas, like NOx and PM, generated by the burn of diesel. Therefore, the diesel engine air system control is necessary to monitor and control the system. Based on the empirical data collected from the engine air system, a linear fourth order state space function is used to model the dynamic system. The state space function is as follows:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{1.1}$$

where the manipulated inputs $u = [u_1, u_2]^T$ are the VGT Vane position VGT_{vane} and EGR valve position EGR_{valve} respectively, which as mentioned earlier are common actuators used on diesel engines for air system management and emissions reduction. The outputs $y = [y_1, y_2]^T$ represent the in-cylinder air/fuel ratio (AFR) and intake manifold EGR percentage respectively.

Due the student ID, so

$$a = 5, b = 0, c = 6, d = 0 \tag{1.2}$$

Thus, the matrix A, B, C is:

$$A = \begin{bmatrix} -7.8487 & -0.0399 & -4.9500 & 3.5846 \\ -4.5740 & 1.1368 & -4.3662 & -1.0683 \\ 3.7698 & 14.9212 & -16.9603 & 4.4936 \\ -9.1895 & 8.3742 & -4.4331 & -16.9798 \end{bmatrix} \tag{1.3}$$

$$B = \begin{bmatrix} 0.0564 & 0.0319 \\ 0.0156 & -0.020 \\ 4.4939 & 1.9981 \\ -1.4269 & -0.273 \end{bmatrix} \tag{1.4}$$

$$C = \begin{bmatrix} -3.2988 & -1.7132 & 0.0370 & -0.0109 \\ 0.2922 & -2.1506 & -0.0104 & 0.0163 \end{bmatrix} \tag{1.5}$$

The initial state is that:

$$x_0 = [0.5 \quad -0.1 \quad 0.3 \quad -0.8]^T \tag{1.6}$$

Control theories are proved to be useful to control the dynamic system above. Pole-placement control with full rank method, LQR control, LQR control with observer, decoupling control with state-feedback method, and integral control by set point tracking are proved stable for this system in following sections.

Based on the requirements of the overshoot and settling time, the damping ξ, and the natural frequency w_n can be calculated by the following function:

$$M_p = e^{\frac{-\pi\xi}{\sqrt{1-\xi^2}}} < 10\%$$
$$t_s = \frac{4.0}{\xi\omega_n} < 20 \tag{1.7}$$

The damping ratio ξ and the natural frequency w_n is:

$$\xi = 0.89,\quad \omega_n = 1.12 \tag{1.8}$$

The desired poles are: $-1 \pm 0.5i$.

2. Pole Placement

2.1 Full Rank Method

Pole placement is a way to achieve state feedback control. Through pole placement, if the system is controllable with negative feedback, some certain properties can be realized, like rise time, settling time, overshoot, etc. Based on the desired poles from equation 8, the desired poles are assumed that:

$$p = [-1-0.5i,\ -1+0.5i,\ -4,\ -9] \tag{2.1}$$

The extra poles are determined by more than 2 times of the dominant poles.
The controllability matrix should be checked, making sure the controllability of the matrix:

$$W_c = \{B\ AB\ A^2B\ A^3B\} \tag{2.2}$$

Then, the assignment of the poles should be decided. In the project, two poles are assigned to one input and others are allocated to the other input, which means $d_1 = 2, d_2 = 2$.

Due to d_1, d_2, the matrix C can be gotten:

$$C = \{b_1, Ab_1, b_2, Ab_2\} \tag{2.3}$$

If the matrix C is full rank, C^{-1} can be gotten as well as the transformation matrix T:

$$C^{-1}=[q_1^T, q_2^T, q_3^T, q_4^T]^T \tag{2.4}$$

$$T=[q_2^T, q_2^T A, q_4^T, q_4^T A]^T \tag{2.5}$$

Based on the matrix T, the similar matrix of A, B are:

$$\overline{A}=TAT^{-1}, \overline{B}=TB \tag{2.6}$$

From the poles in equation 9, and d_1, d_2, one possible desired closed-loop matrix is:

$$A_d=\begin{pmatrix} 0 & 1 & 0 & 0 \\ -63 & -15 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -45 & -88 \end{pmatrix} \tag{2.7}$$

From the equation:

$$A_d=\overline{A}-\overline{B}\,\overline{K} \tag{2.8}$$

The matrix $\overline{K}$ can be obtained. Final K can be acquired:

$$K=\overline{K}T \tag{2.9}$$

2.2 Result and Experiments

A Simulink model is built to simulate the dynamic system and pole placement method.

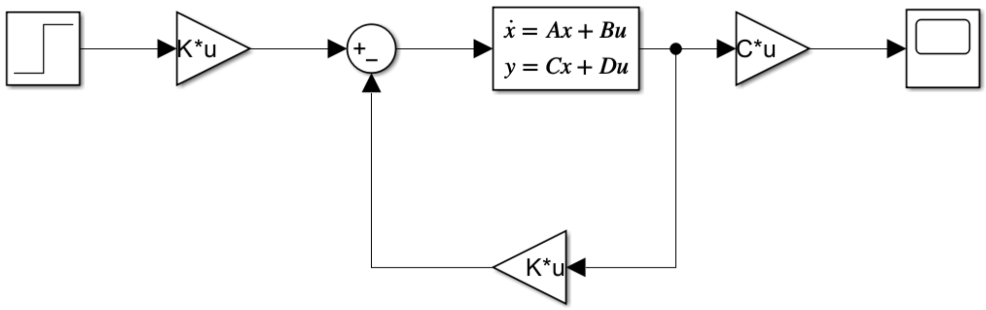

Figure 1 Model 1

Due to the requirement of the overshoot and settling time with the inputs $[1,0],[0,1]$ and zero initial state. Based on the pole placement in equation (2.1). The simulation result is shown below, satisfying the demand.

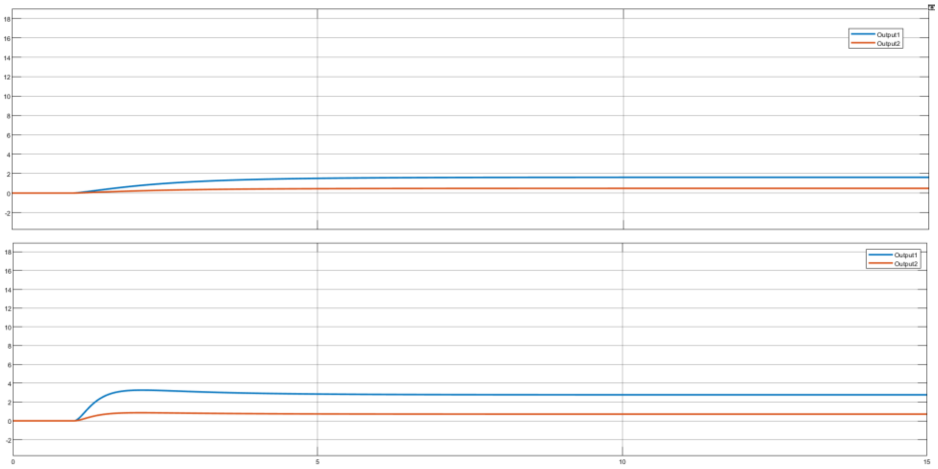

Figure 2 Results of Zero Initial State

If the initial input is zero with initial state (1.6), the result is that:

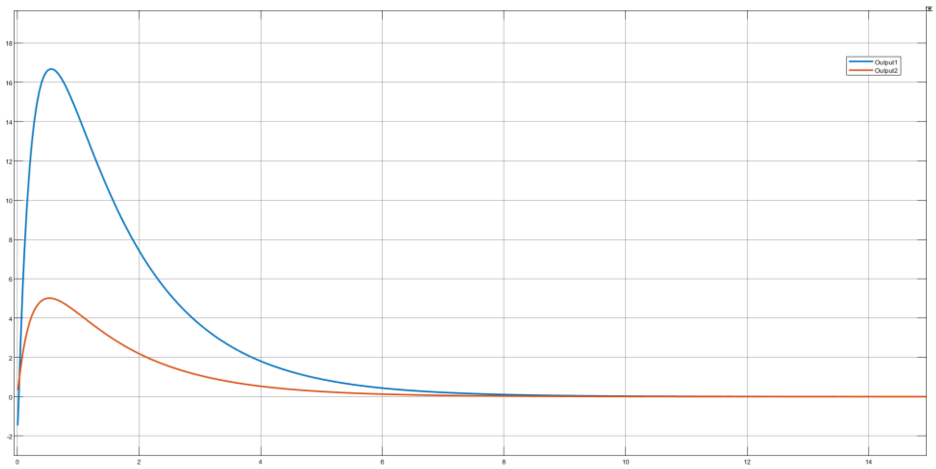

Figure 3 Results of Initial State

If the extra poles are located farther with unchanged dominant poles and the poles become $p=[-1-0.5i, -1+0.5i, -10, -20]$, the overshoot will decrease greatly.

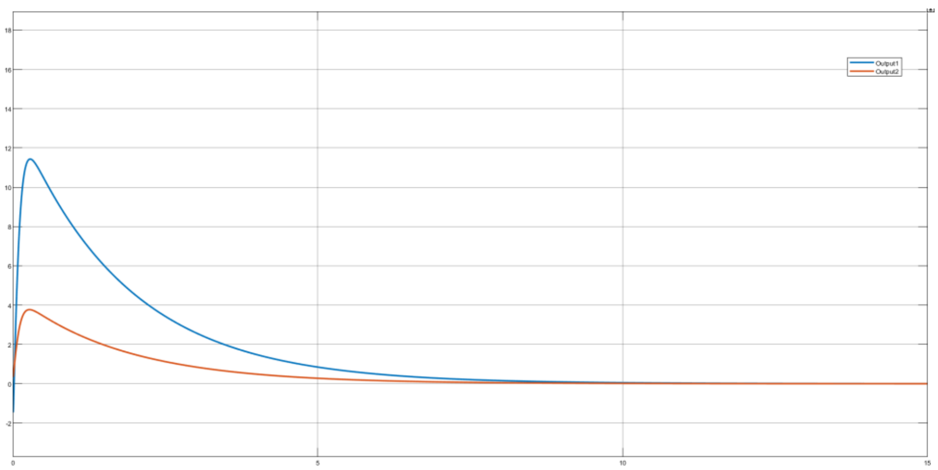

Figure 4 Results of Different Non-dominant Poles

If the extra poles are located nearer with unchanged dominant poles and the poles become $p = [-1-0.5i, -1+0.5i, -2, -4]$, the overshoot will decrease too, but the response will become rough and have big changes in short time.

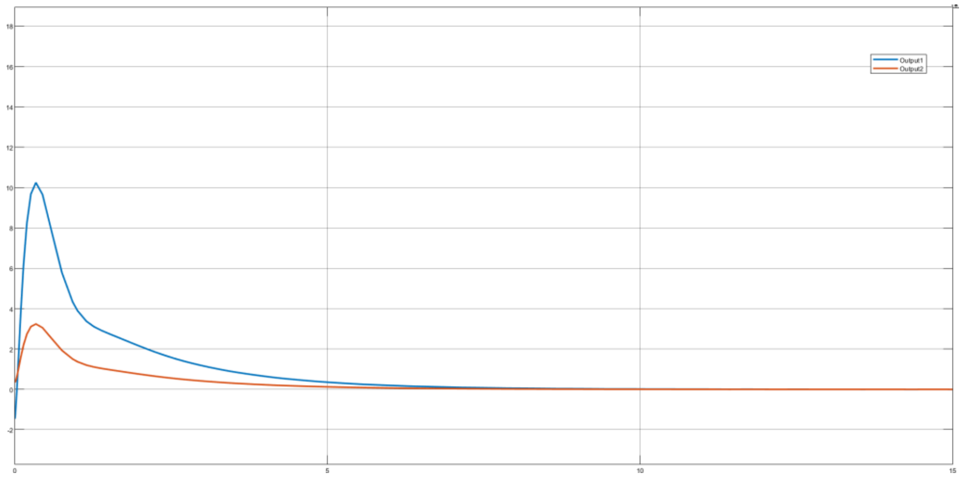

Figure 5 Results of Different Non-dominant Poles

If the real part of dominant poles is located nearer to zero with unchanged dominant poles and the poles become $p = [-0.5-0.5i, -0.5+0.5i, -4, -9]$, the overshoot will decrease greatly.

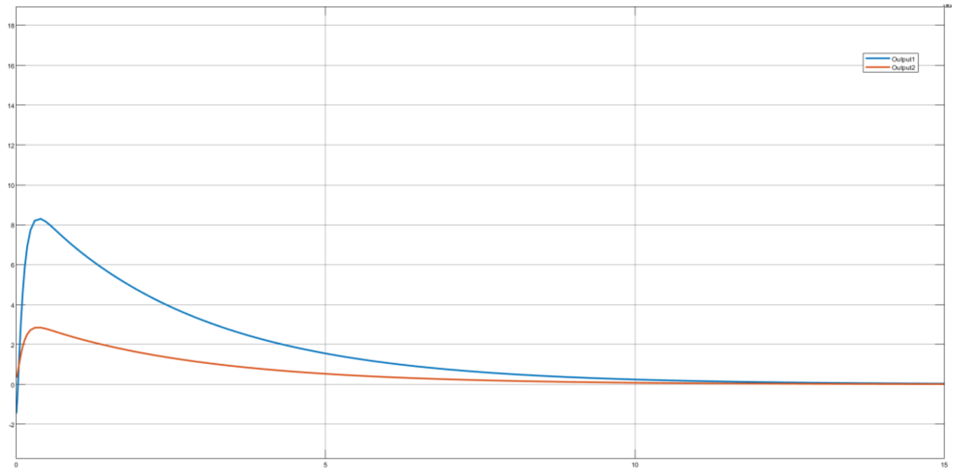

Figure 6 Results of Different Real Parts of Dominant Poles

If the real part of dominant poles is located farther to zero with unchanged dominant poles and the poles become $p = [-2-0.5i, -2+0.5i, -4, -9]$, the overshoot will decrease greatly.

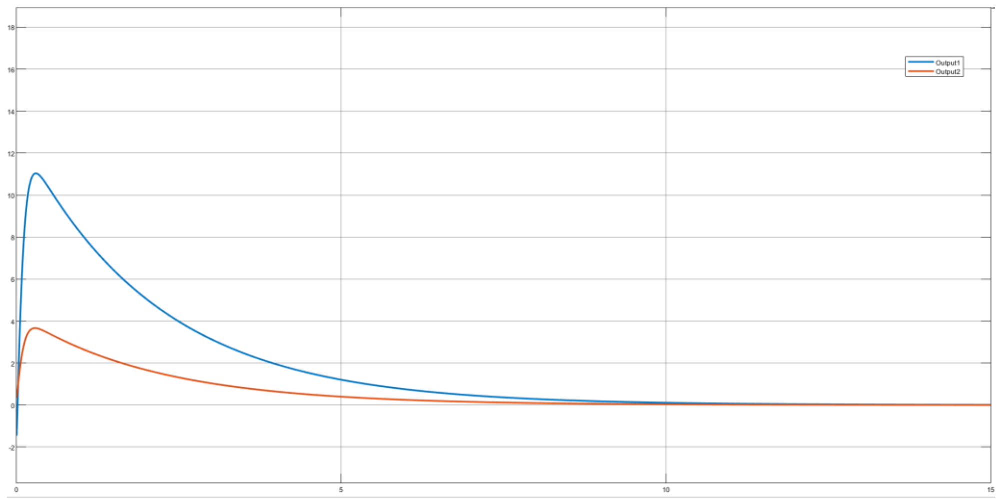

Figure 7 Results of Different Real Parts of Dominant Poles

2.3 Analysis

In a nutshell, different parts of poles play distinctive roles. The closer to zero of the real part of the dominant poles, the flatter and lower of the overshoot will be. With the increasing of the extra poles, the response will become smoother. Based on the above experiments, different parts should make a balance when using pole placement to design a state feedback controller so as to meet certain requirement.

3 LQR

3.1 LQR Method

In last chapter, different matrix T will lead to different results. One method can generate infinite answers. Whether the answer is the best cannot be determined. Linear Quadratic Regulator (LQR) is one of the optimal control methods to achieve the best trade-off among all the desired objectives, which is a balance between the speed and cost.
The problem of LQR is cast into the following quadratic cost function:

$$J=\frac{1}{2}\int_0^\infty (x^T Qx+u^T Ru)dt \tag{3.1}$$

where Q and R are diagonal matrices. Different weights of q_i, r_i show different weights of the state x_i. The key part of calculating J is that:

$$\frac{dJ}{dK}=0 \tag{3.2}$$

Based on the concept of Lyapunov method, the stability of the system can be assured when and only when the derivative of Lyapunov function is less than zero.
From the Lyapunov function, an Algebraic Riccati Equation can be gotten:

$$A^T P + PA + Q = PBR^{-1}B^T P \tag{3.3}$$

,where the matrix P remains to be solved.
In order to solve the equation, a matrix should be formed at first:

$$\Gamma = \begin{pmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{pmatrix} \tag{3.4}$$

Get the stable eigenvalues and compose a new matrix $\begin{pmatrix} \upsilon_i \\ \mu_i \end{pmatrix}, i = 1, 2, ..., n$ corresponding to them.
After that, P is given by:

$$P = [\mu_1, ..., \mu_n]\,[\nu_1, ..., \nu_n]^{-1} \tag{3.5}$$

Finally, the K can be achieved:

$$K = R^{-1}B^T P \tag{3.6}$$

3.2 Results and Experiments

A model in SIMULINK is built to simulate the LQR control.

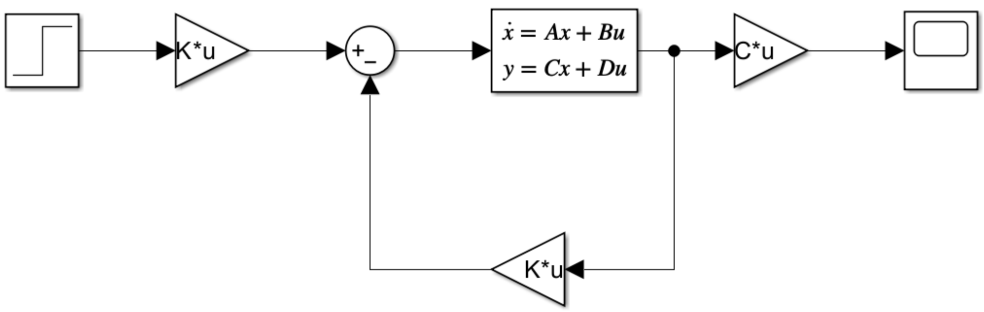

Figure 8 Model 2

Based on the requirements of overshoot and settling time. When all elements in Q and R are all 1, these can meet the demands. The output is showing below:

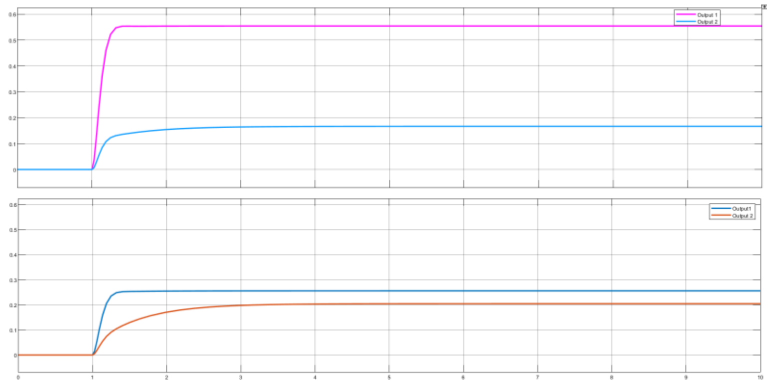

Figure 9 Results of Zero Initial State

Keep the matrix Q and R. Set the initial state without input. The output is:

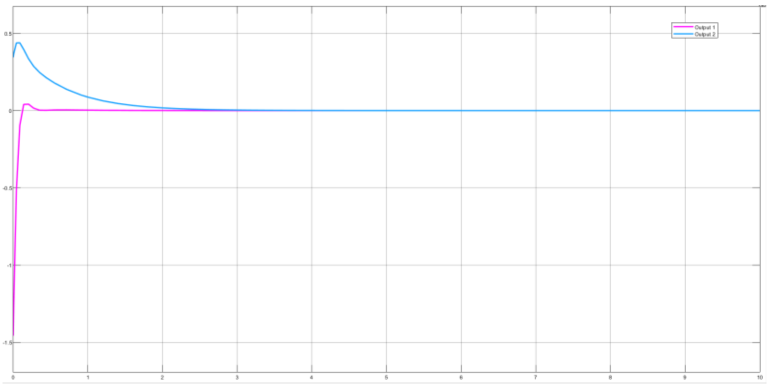

Figure 10 Results of Initial State without Input

The state is:

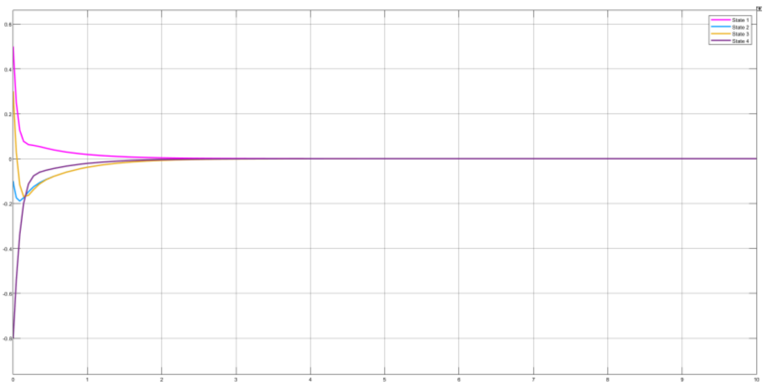

Figure 11 Results of the Changing Initial State

If the matrix Q is changed to $Q = \mathrm{diag}[100\ 100\ 100\ 100]$. The output response and state response will become faster. The output response is:

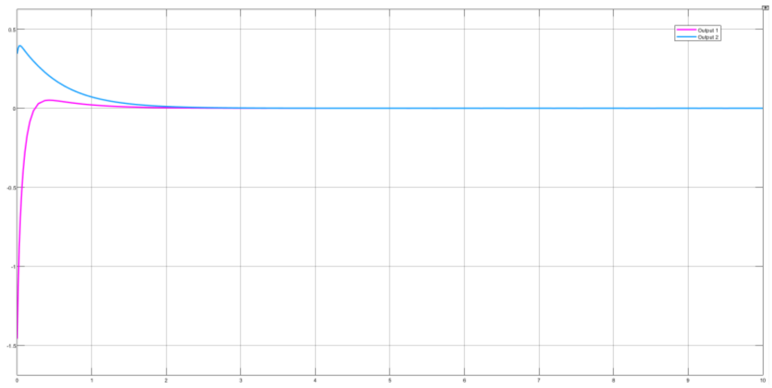

Figure 12 Results of Different Q

The state response is:

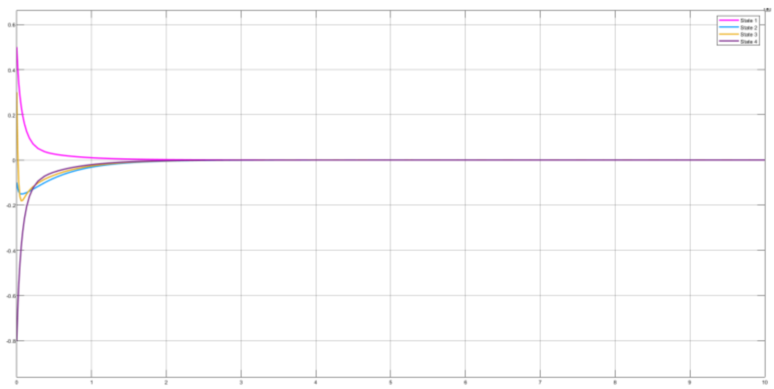

Figure 13 State Results of Different Q

If the Q is set to be $Q = \mathrm{diag}[1\ 1\ 1\ 1]$, and the matrix R to be $R = [100\ 100]$, the output will be:

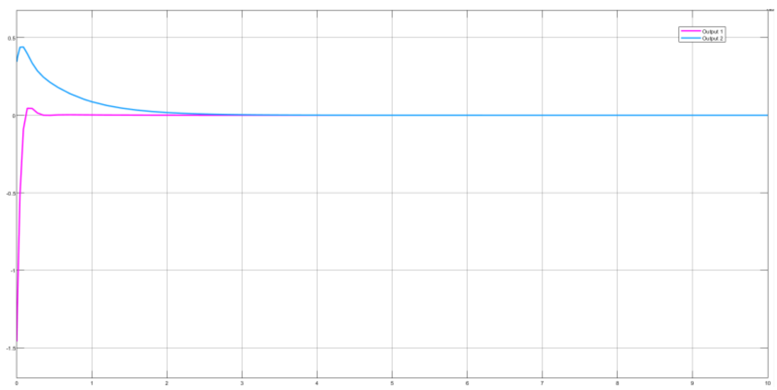

Figure 14 Results of Different R

If the Q is set to be $Q = \mathrm{diag}[1\ 1\ 1\ 1]$, and the matrix R to be $R = [0.01\ 0.01]$, the output will be:

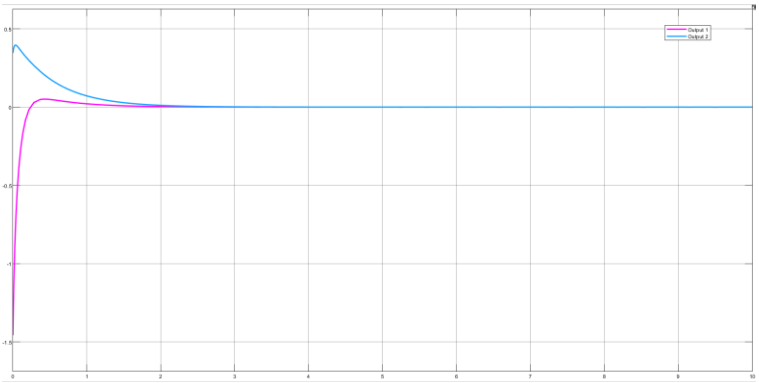

Figure 15 Results of Different R

3.3 Analysis

All in all, when it comes to LQR, larger Q will lead to faster response approaching zero. Larger R will slow the response and low the energy cost. If the Q and R change, only when order of magnitude changes, obvious differences will occur. The choice of Q and R varies from different tasks and aims.

4 LQR Controller with Observer

In this chapter, only two outputs can be measured. If the state feedback controller is still tried to implement, the values of all the state variables should be known. Without sensors, an observer should be built to sense them.

4.1 Method

The observability should be checked to ensure the system can be observed:

$$\begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix} is\ full\ rank \tag{4.1}$$

If the system is observable, an estimator will be built:

$$\begin{gathered} \dot{\hat{x}} = A\hat{x}(t) + Bu(t) + L[y - \hat{y}] \\ \hat{y} = C\hat{x} \end{gathered} \tag{4.2}$$

, where $\hat{x}, \hat{y}$ are the estimated state and output. L is the observer. Let the estimation error in the state be:

$$\tilde{x} = x - \hat{x} \tag{4.3}$$

Put the error in the estimator, the function will be gotten:

$$\dot{\tilde{x}} = (A - LC)\tilde{x} \tag{4.4}$$

Based on the characteristic polynomial of $\det[sI - (A - LC)]$, the equation can be found, which is similar to the pole placement method.

$$\det[sI - (A - LC)] = \det[sI - (A^T - C^T L^T)] \tag{4.5}$$

From the equation, the pole placement method can be used to get L.
From the equation above, L is:

$$L = \begin{pmatrix} -12 & 0 & 7 & -53 \\ 5233 & -9830 & -2610 & -50089 \end{pmatrix}^T \tag{4.6}$$

4.2 Result and Experiments

A model in Simulink will be built to simulate the results of the output. According to the requirements, the results in LQR will be heritage to realize the control. Thus, the K2 in the model serves as a state feedback controller.

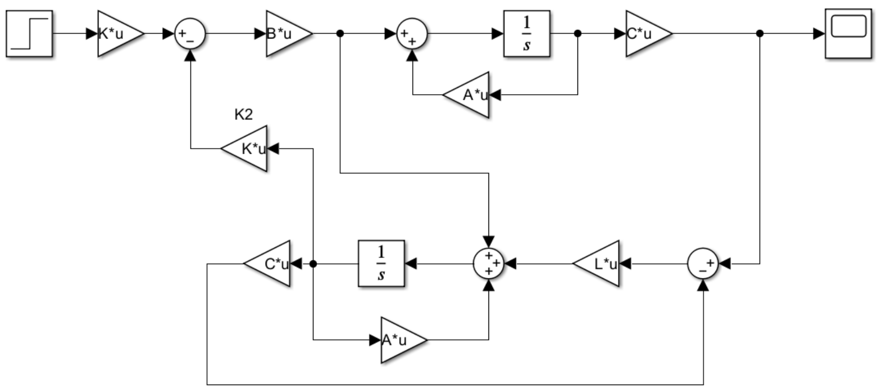

Figure 16 Model 3

First, checking the demand of the overshoot and settling time. Due to the LQR controller comes from last chapter, this part will still satisfy the requirements.

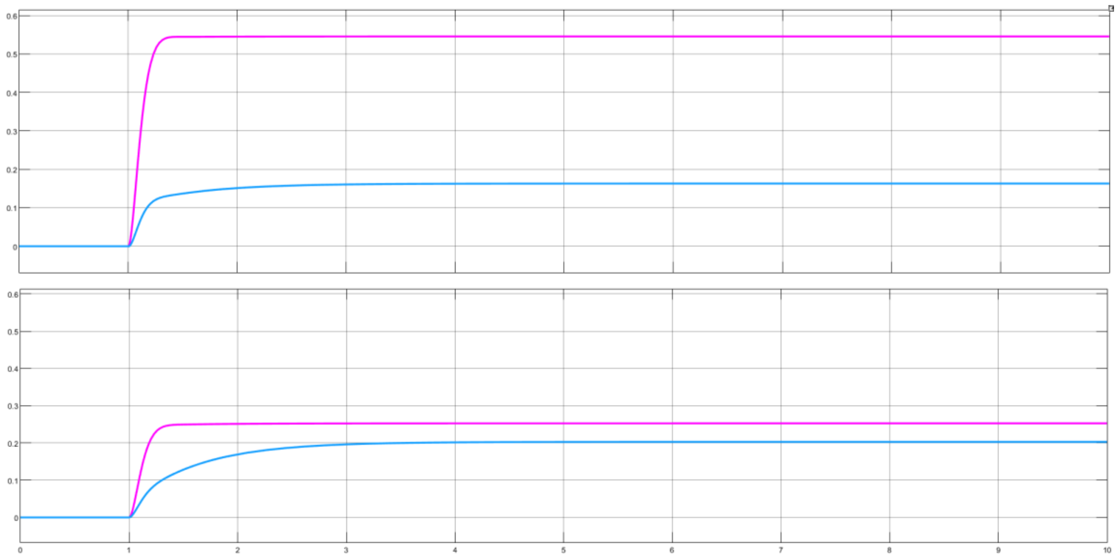

Figure 17 Results of Step Input without Initial State

Even though the four states are unknown, the fluctuations are very small and get close to zero from the initial state.

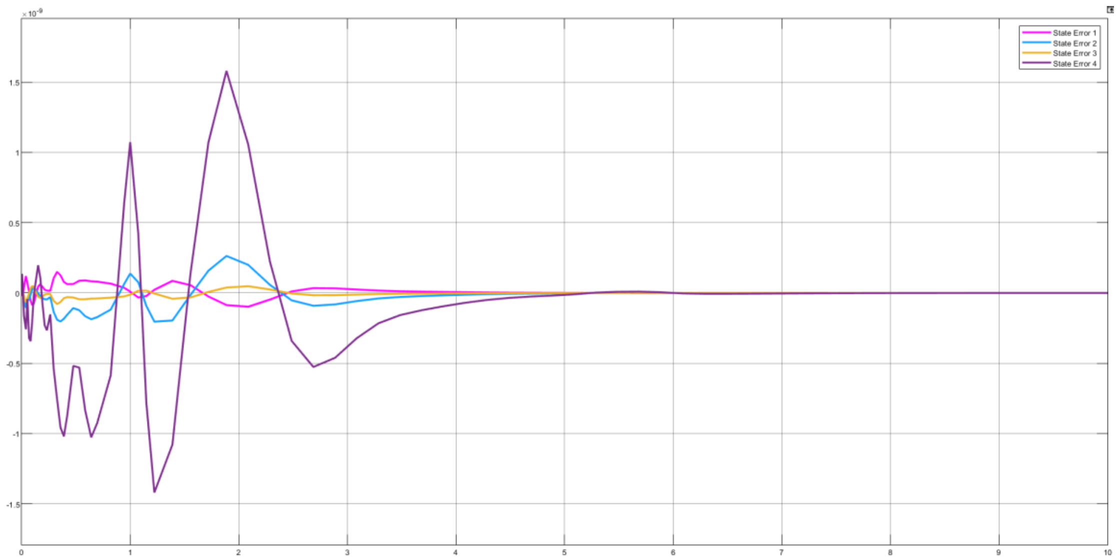

Figure 18 State Results of Initial State

The outputs do not show any difference between the situation where the states can be observed after their fluctuations. There will be differences at first, but the errors will disappear in a short time.

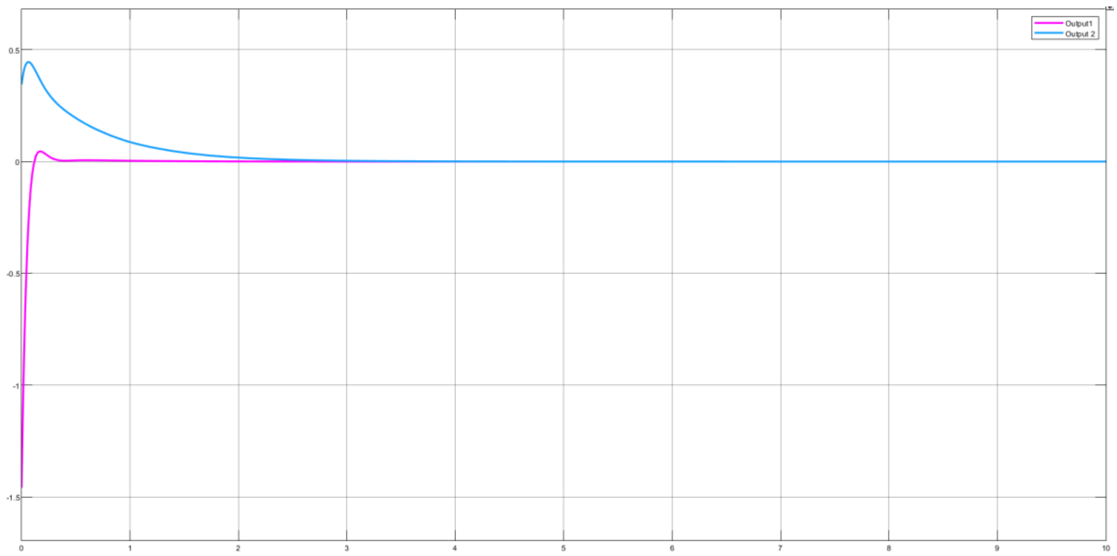

Figure 19 Results of LQR with Observer

4.3 Analysis

In general, if some states or all states cannot be measured, using the observer is a good choice to estimate the states. Based on the observed states, other control methods can be carried out, which can take the place of many sensors greatly.

5 Decoupling Control

Decoupling is a way to separate MIMO system into many SISO systems. Decoupling is usually required in practice for easy operations.

5.1 Method

In order to decouple the system, the open loop transfer function matrix should be calculated:

$$G(s) = C(sI - A)^{-1}B \tag{5.1}$$

The transfer function matrix of the feedback system is:

$$H(s) = C(sI - A + BK)^{-1}BF \tag{5.2}$$

Separate C and G into 4 parts:

$$G(s) = \begin{bmatrix} c_1^T \\ c_2^T \end{bmatrix} (sI - A)^{-1}B = \begin{bmatrix} g_1^T(s) \\ g_2^T(s) \end{bmatrix} \tag{5.3}$$

Define σ as an integer by

$$\sigma_i = \begin{cases} \min(j|c_i^T A^{j-1} B \neq 0, j = 1, 2, 3, 4) \\ 4,\ c_i^T A^{j-1} B \equiv 0 \end{cases},\quad i = 1, 2 \tag{5.4}$$

The number of σ can be calculated: $\sigma_1 = 1$, $\sigma_2 = 1$

G(s) can be written as:

$$G(s) = \mathrm{diag}[s^{-\sigma_1}, s^{-\sigma_2}][B^* + C^*(sI - A)^{-1}B] \tag{5.5}$$

, where B^* and C^{**} are defined as:

$$B^* = \begin{bmatrix} c_1^T A^{\sigma_1 - 1} B \\ c_2^T A^{\sigma_2 - 1} B \end{bmatrix},\ C^{**} = \begin{bmatrix} c_1^T \phi_{f1}(A) \\ c_2^T \phi_{f2}(A) \end{bmatrix} \tag{5.6}$$

, where $\phi_{fi}(A) = A^{\sigma_i} + \gamma_{i1} A^{\sigma_{i-1}} + \ldots + \gamma_{i\sigma_i} I$.

Due to $\sigma = 1$, the $\phi_1(s) = s + 2, \phi_2(s) = s + 4$ based on pole placement.

In this task, B^* and C^* are:

$$B^* = \begin{bmatrix} -0.03 & 0.006 \\ -0.087 & 0.027 \end{bmatrix},\quad C^{**} = \begin{bmatrix} 20.77 & -8.21 & 23.37 & -9.68 \\ 7.94 & -6.77 & 8.03 & 3.05 \end{bmatrix} \tag{5.7}$$

Based on B^* and C^*, K and F can be gotten:

$$F = B^{*-1}, K = B^{*-1} C^* \tag{5.8}$$

Thus, the K and F are:

$$K = \begin{bmatrix} 1604 & 566 & 1822 & 873 \\ -4860 & 1570 & -5557 & 2916 \end{bmatrix},\quad F = \begin{bmatrix} -84 & 18.74 \\ -270 & 92.22 \end{bmatrix} \tag{5.9}$$

However, the model crushes when running in the Simulink showing there is a singularity problem. Thus, the σ are changing to $\sigma_1 = 2$, $\sigma_2 = 2$. The polynomials become $\phi_1(s) = (s+2)^2, \phi_2(s) = (s+4)^2$. Then, the K and F are:

$$K = \begin{bmatrix} 11.08 & -10.91 & 8.47 & 16.37 \\ -23.84 & 30.61 & -26.79 & -28.12 \end{bmatrix},\quad F = \begin{bmatrix} 0.063 & -0.2 \\ -0.13 & 0.48 \end{bmatrix}$$

5.2 Result and Experiments

From the Simulink, a model is built:

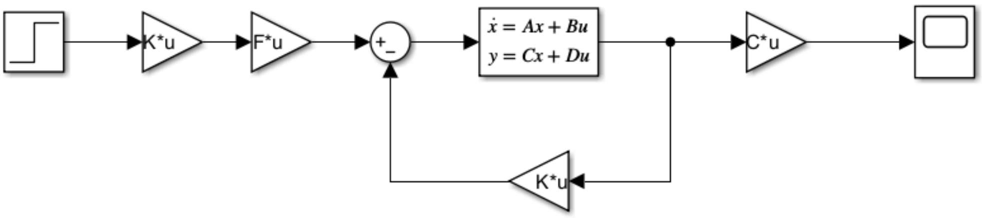

Figure 20 Model 4

First, set the inputs are $[1,1],[1,0],[0,1]$ from the top to the end without initial state.

The outputs are shown below:

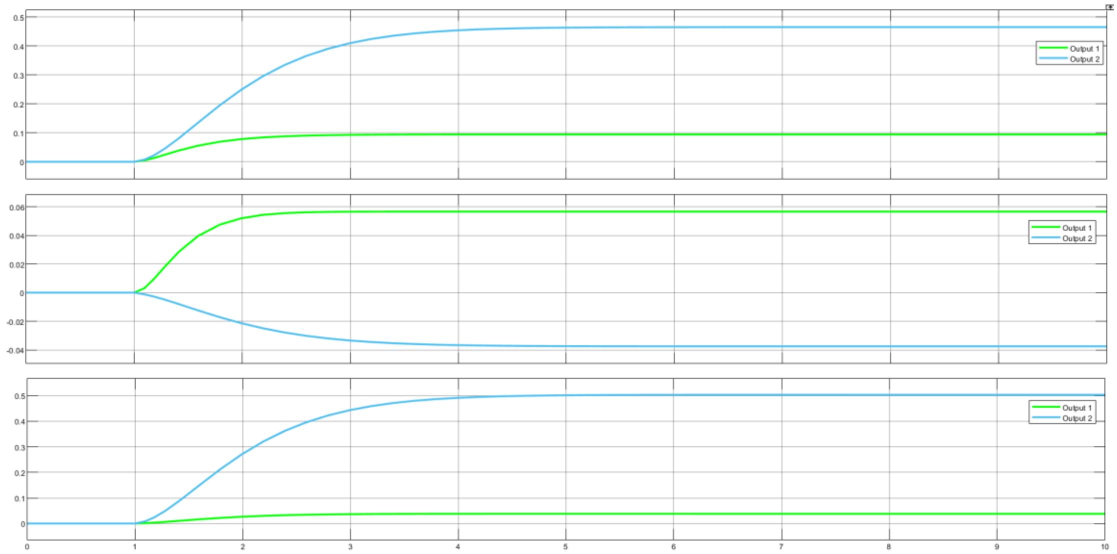

Figure 21 Results of Initial State

From this figure, when changing the inputs, the outputs seem to be decoupled. From the first and second figure, the output 2 changes greatly. From the first and third figure, the output 1 changes.
Set the initial state and apply the zero input and all the state variables are shown below:

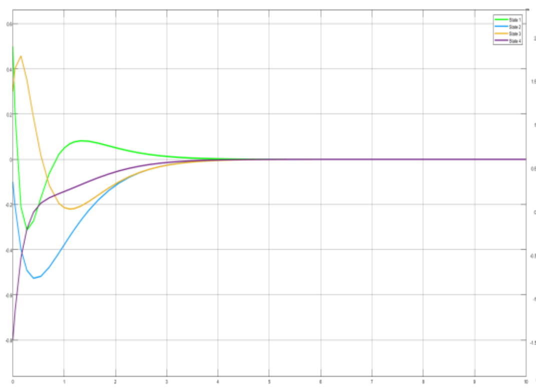

Figure 22 State Results of Initial State

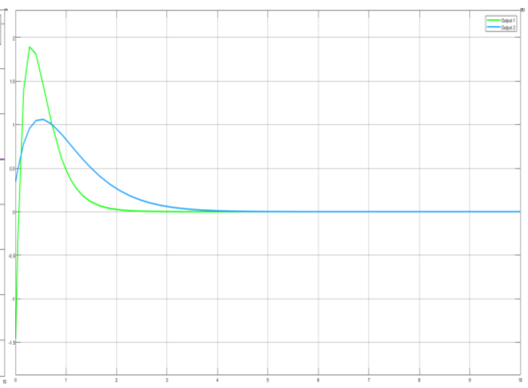

Figure 23 Results of Initial State

All the state variables go to zero very fast. If the outputs are observed, they go to zero rapidly too, showing the decoupled system is internally stable.

5.3 Analysis

The decoupling can transfer the MIMO system into SISO system, making the control become simple and separated.

6 Servo Control

In task 5, the set point of two outputs has been set: $y_{sp}=[0.4,0.8]^T$. A step disturbance $w=[0.3,0.2]^T$ will take effects from $T_d=10s$ afterwards. A servo control should be carried out to realize the aim that the outputs will get as close as possible to the set point regardless of the disturbance.

6.1 Description

The idea of servo control is to extend the state and take the error into consideration. Based on the state space function with disturbance, an augmented system is built:

$$\begin{pmatrix}\dot{x}\\ \dot{v}\end{pmatrix}=\begin{pmatrix}A & 0\\ -C & 0\end{pmatrix}\begin{pmatrix}x\\ v\end{pmatrix}+\begin{pmatrix}B\\ 0\end{pmatrix}u+\begin{pmatrix}B_w\\ 0\end{pmatrix}w+\begin{pmatrix}0\\ I\end{pmatrix}r$$
$$y=(C\ \ 0)\begin{pmatrix}x\\ v\end{pmatrix} \tag{6.1}$$

, where v is:

$$\dot{v}(t)=e(t)=r-y(t) \tag{6.2}$$

The controllability matrix should be checked at first:

$$Q_c=\begin{pmatrix}B & AB & A^2B & A^3B\\ 0 & -CB & -CAB & -CA^2B\end{pmatrix} \tag{6.3}$$

The inputs become:

$$u=-K\overline{x}=-[K_1\ \ K_2]\begin{bmatrix}x\\ v\end{bmatrix} \tag{6.4}$$

The resultant feedback system is:

$$\begin{pmatrix}\dot{x}\\ \dot{v}\end{pmatrix}=\begin{pmatrix}A-BK_1 & -BK_2\\ -C & 0\end{pmatrix}\begin{pmatrix}x\\ v\end{pmatrix}+\begin{pmatrix}B_w\\ 0\end{pmatrix}w+\begin{pmatrix}0\\ I\end{pmatrix}r \tag{6.5}$$

To stabilize the closed-loop augmented system, the LQR method is used to optimize the control, finding out K. the result of K is shown below:

$$K=\begin{bmatrix}-0.3 & -0.1 & 0.24 & -0.07 & -0.99 & 0.11 \\ 0.57 & -1.39 & 0.18 & 0.23 & -0.11 & -0.99\end{bmatrix} \tag{6.6}$$

However, there are only two sensors to measure the outputs. Therefore, an observer should be built to observe the states by pole placement. The observer poles usually are 3-5 times faster than the closed loop dominant poles designed by the controller. The poles here are $p_0=[-12, -16, -20, -24]$, and the L is:

$$L=\begin{bmatrix}-12 & 0 & 7 & -53 \\ 5233 & -9830 & -2610 & -50089\end{bmatrix} \tag{6.7}$$

6.2 Results and Experiments

Based on the calculation in previous part, a model is built in Simulink to simulate the results of the augmented system.

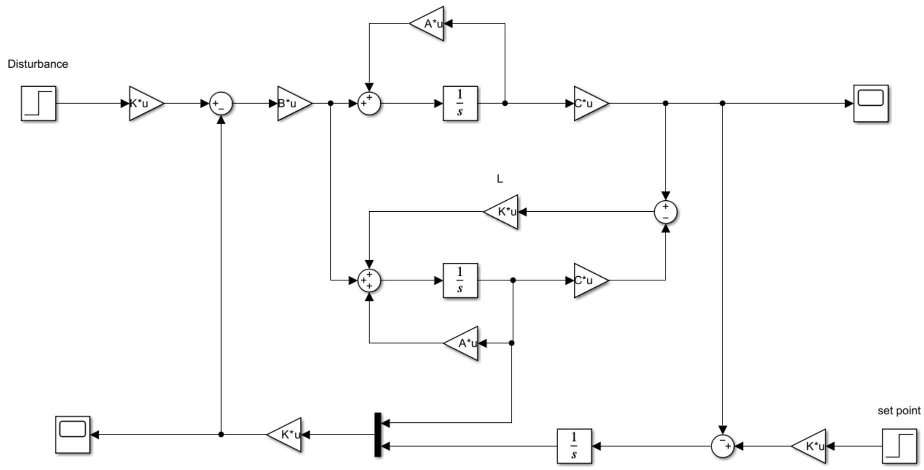

Figure 24 Model 5

If there is no disturbance in the system, the outputs are:

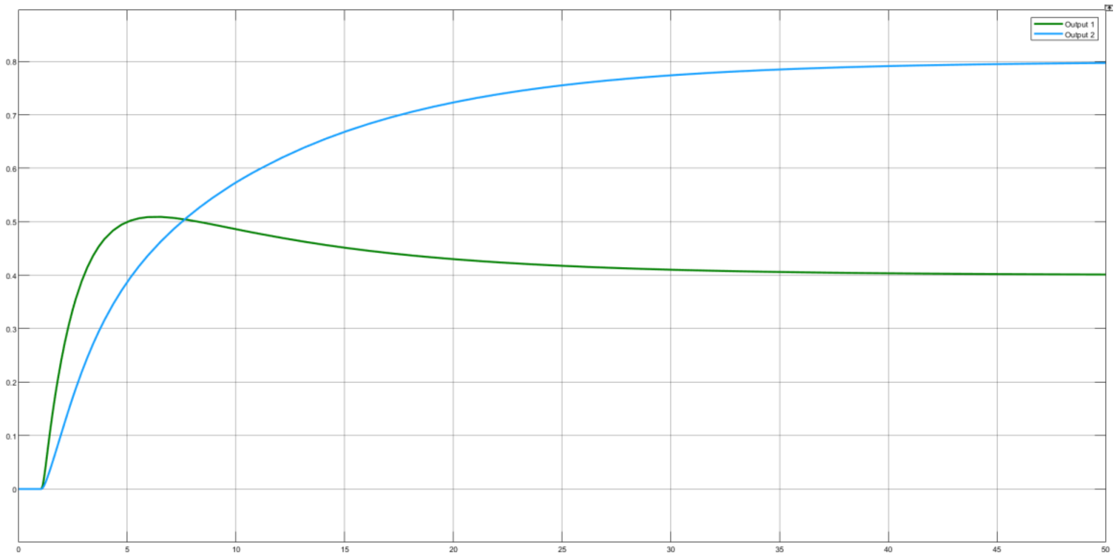

Figure 25 Results of Initial State without Disturbance

If there is a disturbance at 10s afterward, the outputs are:

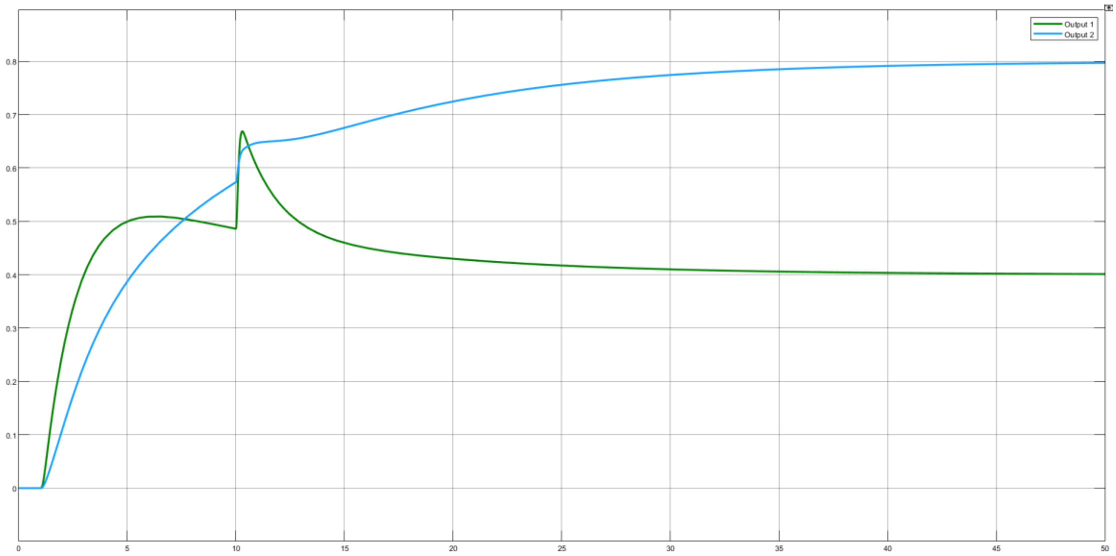

Figure 26 The Response of Initial State with Disturbance

From two situations, both the outputs go to the set point and become stable at the points. A disturbance will interrupt the response of the original system, but the state feedback will adjust it very fast and go back to set point.

The control signals are:

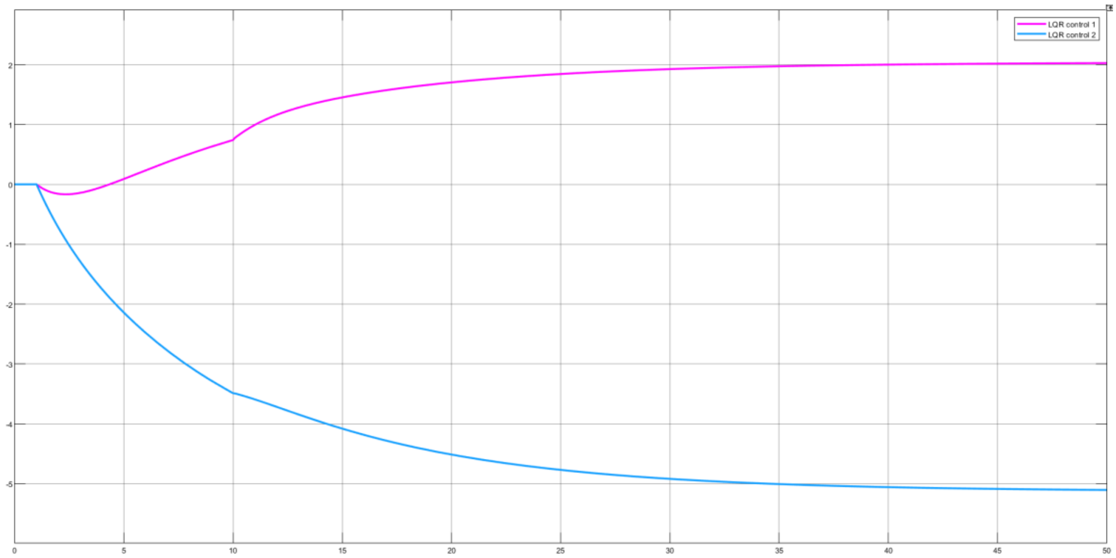

Figure 27 The Response of Control Signals

From the control signals, they become stable at about 30s. When the disturbance is enforced, sudden changes happen at moment, but no obvious change occur. The responses are fluent and fast.

6.3 Analysis

The servo control method can effectively enable the outputs to trace the set point regardless of the disturbance.

7 State Error

7.1 Description

In this part, the four states are intended to regulate directly instead of two outputs. A set point is set $x_{sp} = [0, 0.5, -0.4, 0.3]^T$. The aim of this project is to eliminate the steady state error as close as possible to the set point. From my perspective, the aim is doable, and the following part shows how to achieve that.
Based on the servo control method, two inputs enable two outputs to achieve the target set point. Thus, two separate and similar systems are built to realize the goal.

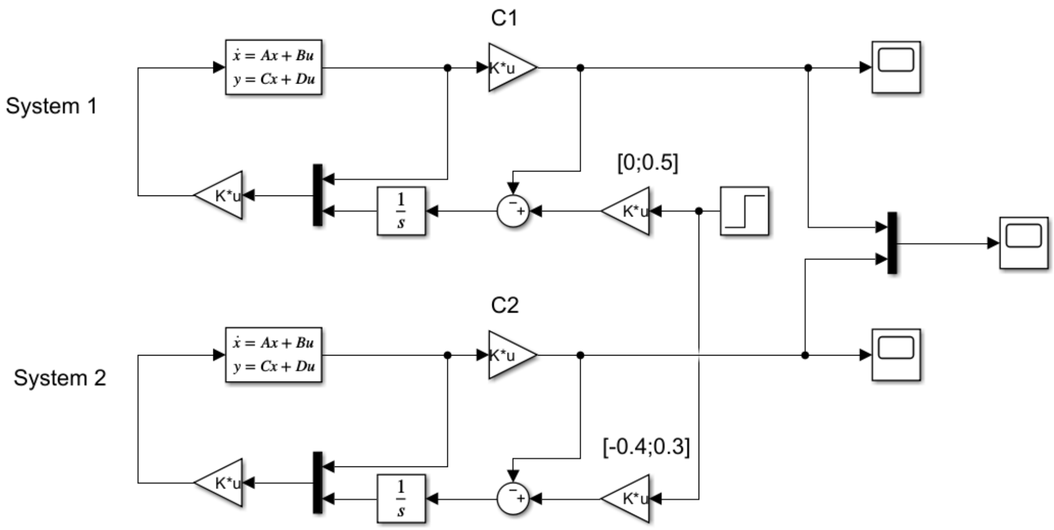

Figure 28 Model 6

In order to transfer the state into output, two matrices C are set:

$$C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, C_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.1}$$

The outputs of systems are also the two states of the systems. The target states are separated into two matrices:

$$x_{sp1} = [0; 0.5], x_{sp2} = [-0.4; 0.3] \tag{7.2}$$

A LQR method is used to optimize the state feedback K. To make the response faster to reach the target goals, the Q, R, and resultant K are:

$$Q = \mathrm{diag}([500\ 500\ 10\ 100\ 150\ 50]), R = \mathrm{diag}([1\ 1]) \tag{7.3}$$

$$K = \begin{bmatrix} -10.30 & -4064 & 3.64 & -3.95 & 12 & 1.41 \\ 3.01 & -16.86 & 2.37 & 1.73 & -2.45 & 6.93 \end{bmatrix} \tag{7.4}$$

7.2 Result

The output is shown below:

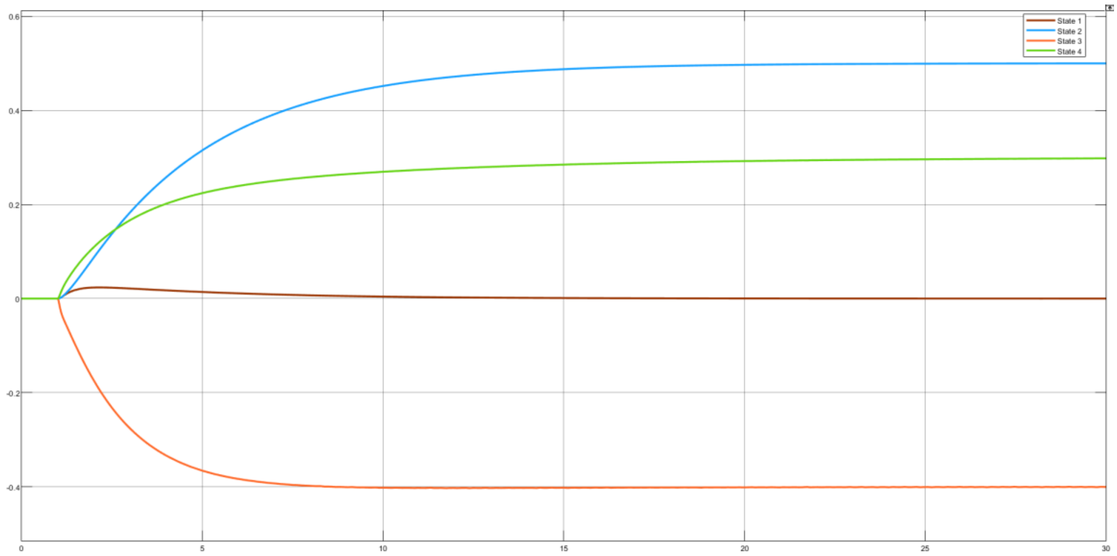

Figure 29 The Response of Four states

All the four states go to their target goals and maintain themselves at the steady states.

7.3 Analysis

Based on the above system design tragedy, the states can be ensured to reach x_{sp} and maintain steady in a short period of time.

8 Conclusion

In pole placement part, different poles are tried in the model. The changes of different parts of the poles will lead to different results. The closer to zero of the real part of the dominant poles, the flatter and lower of the overshoot will be. With the increasing of the extra poles, the response will become smoother. Based on the ideal results, poles should be adjusted to satisfy the goals.

In LQR part, the larger elements of the matrix Q, the response will become faster, but it will have larger energy cost. The larger elements of the matrix R, the response will become slower with less energy cost. However, compared to pole placement, small changes in Q and R do not show obvious differences. Only when the Q and R are changed in order of magnitude, big changes can be found. The LQR can be more effective than pole placement, but with more elements of Q, it will become harder to adjust to a very ideal model.

In LQR with observer part, an observer is used when there is no sensor. By setting an observer, it can greatly measure the response of the observed states. Even though there will be obvious differences at first, the controller will adjust them back to the ideal response. From this dynamic system, because this controller does not have high demand of short period, it can be used to take the place of sensor, but when it comes to a situation

where immediate response is needed, observers cannot substitute the sensors directly.
In decoupling part, by decoupling, the MIMO system can be transferred into SISO system. Based on my matrix, the system is internally stable.
In servo control, the dynamic system can greatly follow the set point and become steady at these points regardless of the disturbance.
In the last part, the problem is to identify whether four states can be directly regulated instead of two outputs. I succeed in doing that by separating the four target states into two states in two matrices and controlling them with two parallel systems. This method comes from the servo control method. Thus, the original system only has two inputs and two outputs. To control 4 states as well as four outputs when the matrix C is a kind of unit matrix (no real unit matrix), there are two ways to realize that. One way is to add two more inputs to achieve the target, in which situation the matrix needs to be changed. The second way is to divide the target state into two target states. I use the second way, which is used in the report and the results show the effectiveness.

Appendices

Main_function.m:

```
clc
clear all
a = 5;
b = 0;
c = 6;
d = 0;
A = [-8.8487+(a-b)/5,-0.0399,-5.55+(c+d)/10,3.5846;
     -4.574,2.5010*(d+5)/(c+5),-4.3662,-1.1183-(a-c)/20;
     3.7698,16.1212-c/5,-18.2103+(a+d)/(b+4),4.4936;
     -8.5645-(a-b)/(c+d+2),8.3742,-4.4331,-7.7181*(c+5)/(b+5)];
B = [0.0564+b/(10+c),0.0319;
    0.0165-(c+d-5)/(1000+20*a),-0.02;
    4.4939,1.5985*(a+10)/(b+12);
    -1.4269,-0.2730];
C = [-3.2988,-2.1932+(10*c+d)/(100+5*a),0.037,-0.0109;
    0.2922-a*b/500,-2.1506,-0.0104,0.0163];
D = zeros(size(C,1), size(B,2));
x0 = [0.5;-0.1;0.3;-0.8];
%% question1, kesi = 0.7, wn = 1/1.4
p = [-1-0.5i, -1+0.5i, -4, -9]; % kesi = 0.89  wn = 1.12
K1 = pole_placement(A,B,p);
K = place(A, B, p);
%% question2
Q = diag([1 1 1 1]);
R = diag([1 1]);
K2 = lqr(A,B,Q,R);
%% question 3
O = obsv(A,C);
isObservable = (rank(O) == size(A, 1));
po = [-12,-16,-20,-24];
L = pole_placement(A',C',po);
L = L';
question3_draw(A,B,C,D,x0,L,K2)
%% qustion4
syms s
G = C*inv(s*eye(4)-A)*B;
[rowG,columnG] = size(G);
B_star = [];
C_star2 = [];
```

```
B_star(1,:) = C(1,:)*A*B;
B_star(2,:) = C(2,:)*A*B;
C_star2(1,:) = C(1,:)*(A+4*eye(4))^2;
C_star2(2,:) = C(2,:)*(A+2*eye(4))^2;
K_4 = inv(B_star) * C_star2;
F = inv(B_star);
H = C*inv(s*eye(4)-(A-B*K_4))*B*F;
%% question5 servo control
po = [-12,-16,-20,-24];
[K5,L5] = servo_control(A,B,C,po);
%% question6
xsp = [0;0.5;-0.4;0.3];
xsp1 = [0;0.5];
xsp2 = [-0.4;0.3];
C1 = [1 0 0 0;
     0 1 0 0;];
C2 = [0 0 1 0;
     0 0 0 1;];
Q1 = diag([500 500 10 100 150 50]);
R1 = diag([1 1]);
K61 = servo_control_state(A,B,C1,Q1,R1);
K62 = servo_control_state(A,B,C2,Q1,R1);
```

Pole Placement.m

```
function K = pole_placement(A,B,p)
    %x0 = [0;0;0;0];
    syms s;
    polynomial = (s-p(1))*(s-p(2))*(s-p(3))*(s-p(4));
    pol_cof = double(coeffs(polynomial));
    Ad = [0, 1, 0, 0;
          -pol_cof(3:4) ,0, 0;
          0 ,0, 0, 1;
          0, 0 ,-pol_cof(1:2)];
    %calcuate the Controllable matrix
    Wc = ctrb(A, B);
    Wc = Wc;
    is_controllable = (rank(Wc) == length(A));
    %change the sequence of Qc
    Con = [];
    temp = [];
    j = 1;
    jj= 5;
    for i = 1 : length(Wc)
```

```
        if mod(i,2) == 0
            Con(:,jj) = Wc(:,i);
            jj = jj + 1;
        else
            Con(:,j) = Wc(:,i);
            j = j + 1;
        end
    end
    temp = Con(:,5);
    Con(:,5) = Con(:,3);
    Con(:,3) = temp;
    temp = Con(:,6);
    Con(:,6) = Con(:,4);
    Con(:,4) = temp;
    is_controllable1 = (rank(Con(:,1:4)) == length(A));
    inv_Con = inv(Con(:,1:4));
    T = [];
    for i = 1 : length(A) - 2
        T(i,:) = inv_Con(2,:) * A^(i-1);
    end
    for i = 3 : length(A)
        T(i,:) = inv_Con(4,:) * A^(i-3);
    end
    %T(4,:) = inv_Con(4,:);
    A_bar = T*A/T;
    B_bar = T*B;
    A_bar = round(A_bar);
    B_bar = round(B_bar);
    syms k11 k12 k13 k14 k21 k22 k23 k24;
    K_bar = [k11 k12 k13 k14;
             k21 k22 k23 k24];
    A_last = A_bar - B_bar * K_bar;

    eqn = solve(Ad == A_last);
    S = double(struct2array(eqn));
    K_bar = [S(1:4);S(5:8)];
    K = K_bar * T;
    K = round(K);
end
```

LQR.m

```
function K = lqr(A,B,Q,R)
    syms P;
    %AREeqn = P * A + A' * P - P * B / R * B' * P + Q == 0;
    matrix = [A ,-B/R*B';
             -Q , -A'];
    [V, DD] = eig(matrix);
    [row, col] = size(DD);
    d = diag(DD);
    dd = [];
    record_vector = [];
    j = 1;
    for i = 1 : length(d)
        if d(i) < 0
            dd(j) = d(i);
            record_vector(j) = i;
            j = j + 1;
        end
    end
    final_vector = [];
    for i = 1 : length(dd)
        final_vector(:,i) = V(:,record_vector(i));
    end
    v = [];
    mu = [];
    width = size(final_vector,1);
    v = final_vector(1:width/2,:);
    mu = final_vector(width/2+1:width,:);
    P = mu / v;
    K = real(inv(R) * B' * P);
end
```

Servo_Contol.m

```
function [K L] = servo_control(A,B,C,po)
Aba = [A zeros(4,2)
      -C zeros(2,2)];
Bba = [B;zeros(2,2)];
Cba = [C zeros(2,2)];
Qc = ctrb(Aba,Bba);
isControllable = (rank(Qc) == size(Aba, 1));
Q = diag([1 1 1 1 1 1]);
R = diag([1 1]);
K = lqr(Aba,Bba,Q,R);
```

```
K1 = K(:,1:4);
K2 = K(:,5:6);
L = pole_placement(A',C',po);
L = L';
end
```

Servo_Contol_State.m (the function used in task 6)

```
function K = servo_control_state(A,B,C,Q,R)
Aba = [A zeros(4,2)
      -C zeros(2,2)];
Bba = [B;zeros(2,2)];
Cba = [C zeros(2,2)];
Qc = ctrb(Aba,Bba);
isControllable = (rank(Qc) == size(Aba, 1));
K = lqr(Aba,Bba,Q,R);
K1 = K(:,1:4);
K2 = K(:,5:6);
end
```