

**ME5401 Linear System**

**Multivariable Controller Design for Diesel Engine Air System Control**

Submitted by

Fu Yanqging A0225413R

Email: e0576047@u.nus.edu

2020/11/17

Department of Mechanical Engineering

Faculty of Engineering

National University of Singapore

**1. Problem Statement**

By establishing a linear system model, we can solve the pollution problem of aviation engine air emissions. This paper aimed to established a controllable model which can be used to solve the engine air problem. The structure model is built as shown in Figure 1.1.

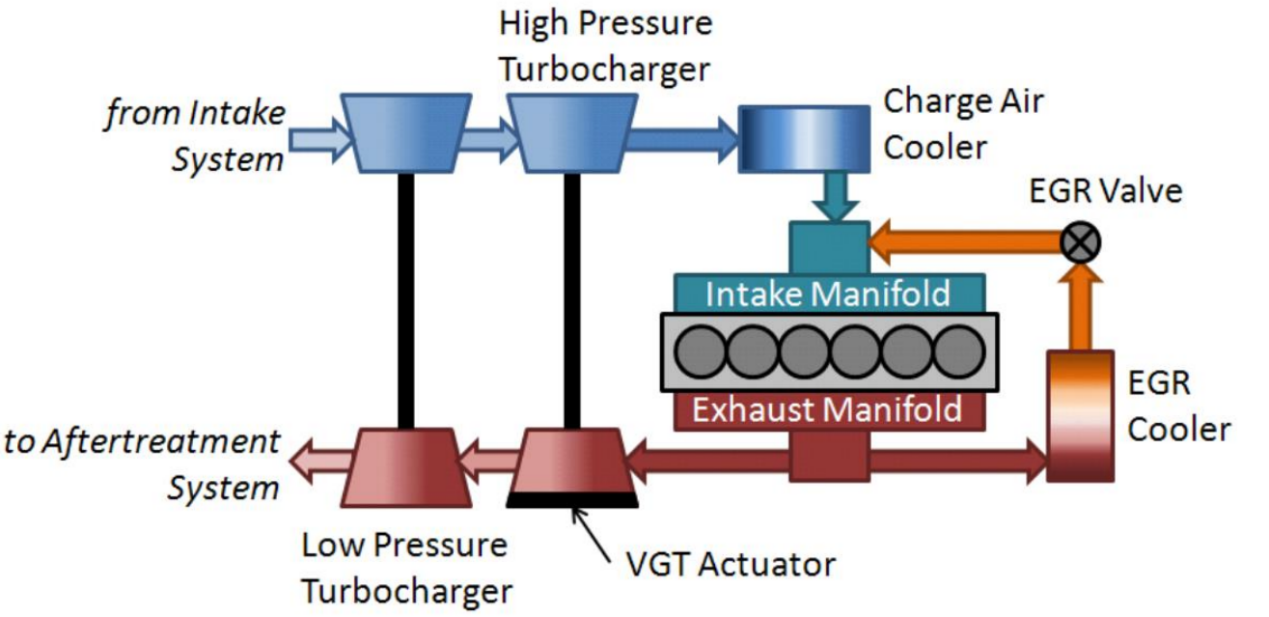
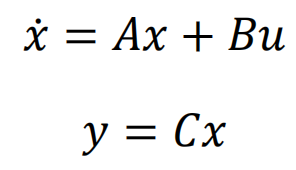


Figure 1.1 The Model of engine air emissions

This model can be described by using state space equation:



where the manipulated inputs u = [u1, u2]T are the VGT Vane position and EGR valve position respectively, The outputs y=[y1, y2]T represent the in-cylinder air/fuel ratio (AFR) and intake manifold EGR percentage respectively.

Solve state space parameters according to requirements:







Matrices A, B, C are derived by substituting matriculation number a=5, b=4, c=1, d=3.

1. **Solution to task 1:**

According to the question, state feedback is required to implement in the system. For this scenario, the feedback matrix K is a 2 by 4 matrix defined by the dimension of both A and B, such that the overall state matrix is a 4 by 4 matrix. The block diagram is shown in Figure 2.1.

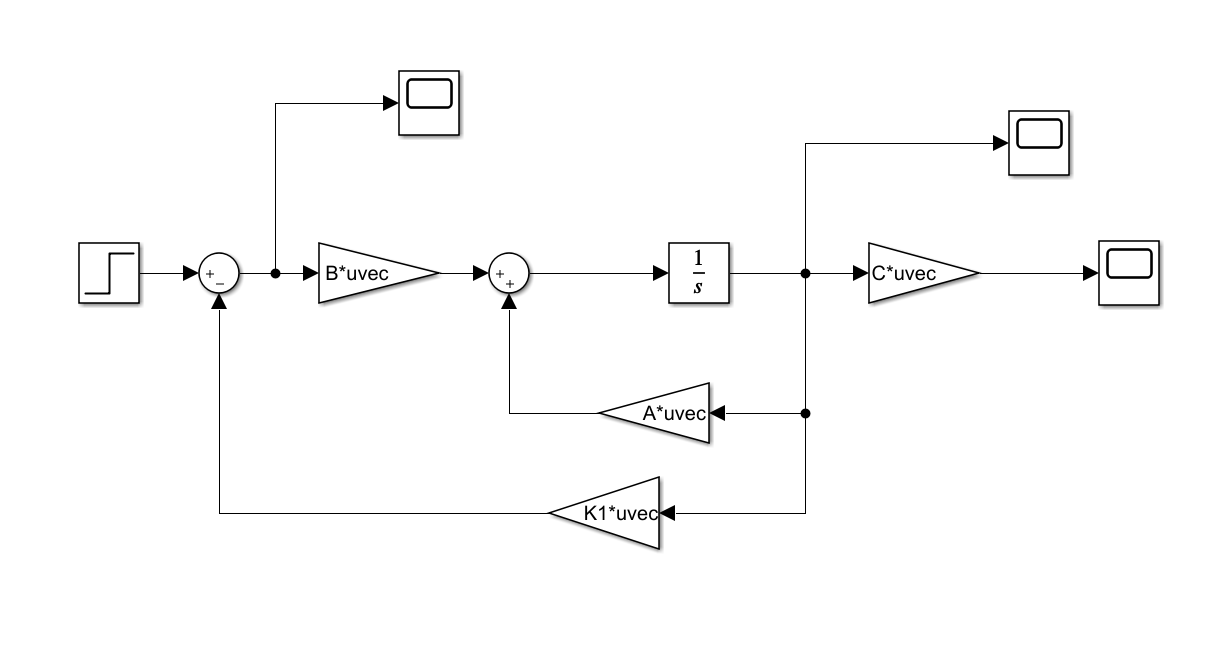
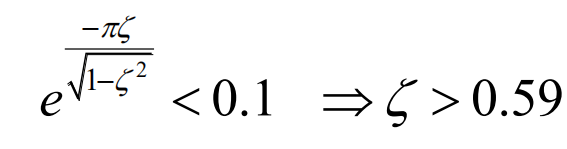


Figure 2.1 :State Feedback Diagram

* 1. **Design reference model:**

Since there are two performance criteria (i.e. overshoot less than 10% and 2% settling time less than 20 seconds) for all the below-mentioned tasks, I should choose a proper reference model for pole placement. If letting the damping coefficient =0.707, respectively, and placing two irrelevant poles to s=-0.48, s=-0.4.

the maximum overshoot should be less than 10%:



settling time meets  so that 

Set *t*s = 4s, damping = 0.707,



And the characteristic polynomial is derived as



Therefore, we can get the transfer function:



The simulation result of the desired model is shown in Figure 2.2. By inspection, the desire model can satisfy performance requirement:

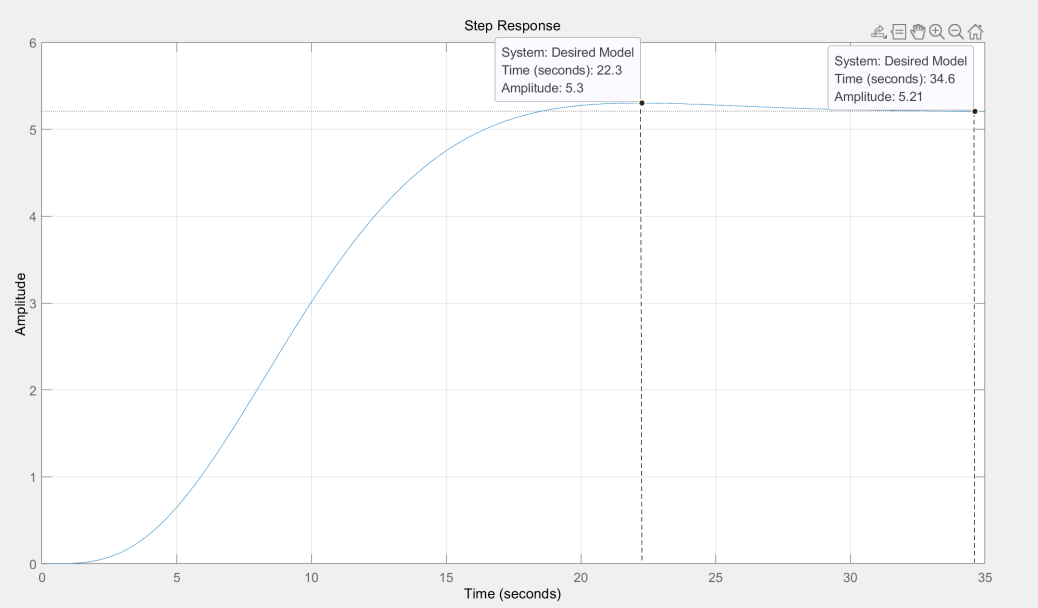


Figure 2.2 Desired Model Step Response



### 2.2 Computation Feedback Gain K Using Pole Placement Method

Firstly, we need to calculate the controllability matrix before carrying out any solution.



It can be verified by MATLAB that Rank() = 4, that means this process is controllable. After that, we need to select n independent vectors out of , strictly following the order from left to right. In my case the first six vectors are independent with each other. Then, the vectors associated with the same input are grouped together to form a new matrix , such that the MIMO canonical form can be calculated, and the inverse of is derived by MATLAB.

For the next step, take out the third and sixth rows from corresponding to the 2 inputs, and from the transformation matrix as



Therefore, the MIMO canonical form is obtained as

|  |  |
| --- | --- |
|  | (4a) |



where ;

The closed loop matrix = 

After we derive , the real feedback gain is obtained by transforming the canonical feedback gain into unit coordinate using

When the pole is placed at [-0.84, -0.8, -0.3535 - 0.3536i, -0.3535+0.3536i]

The feedback gain K is



So transient response and state trend can be plotted respectively:

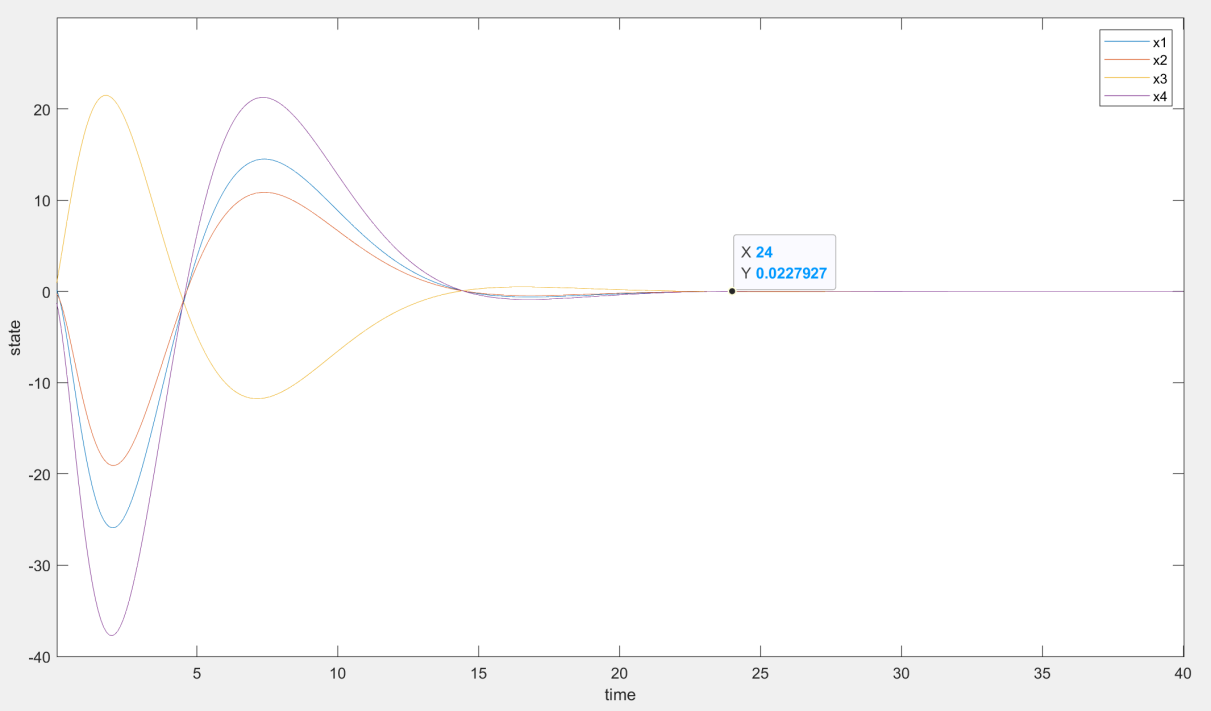


Figure 2.3 State Responses to non-zero initial state with zero external inputs

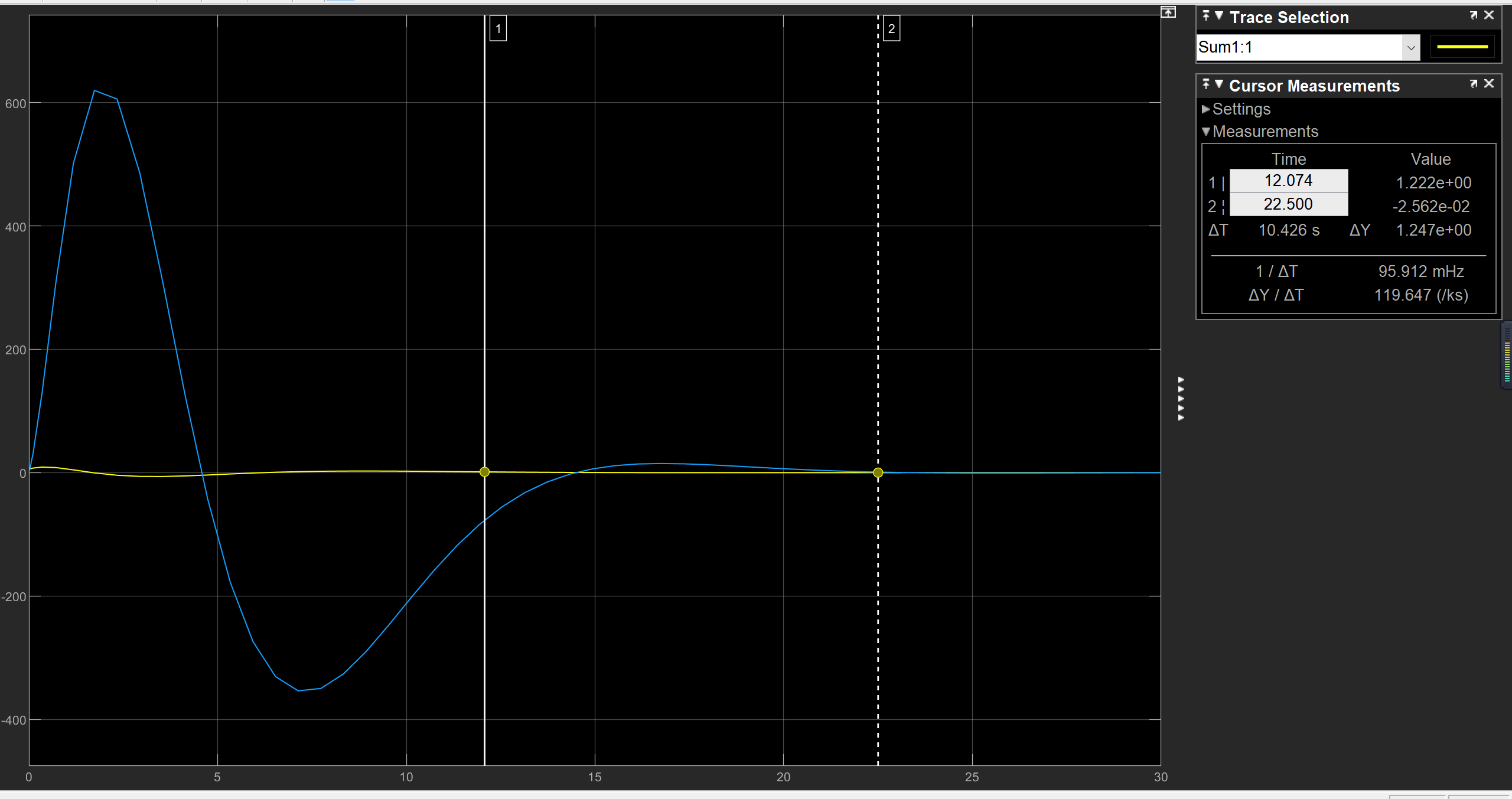


Figure 2.4 Control signal to non-zero initial state with zero external inputs

By observing the state response, it can be found that the overshoot of the x4 curve changes sharply at the beginning, and then starts to oscillate around the x-axis and finally stabilizes. And the figure below describes the out under step response and zero input with non-zero initial condition

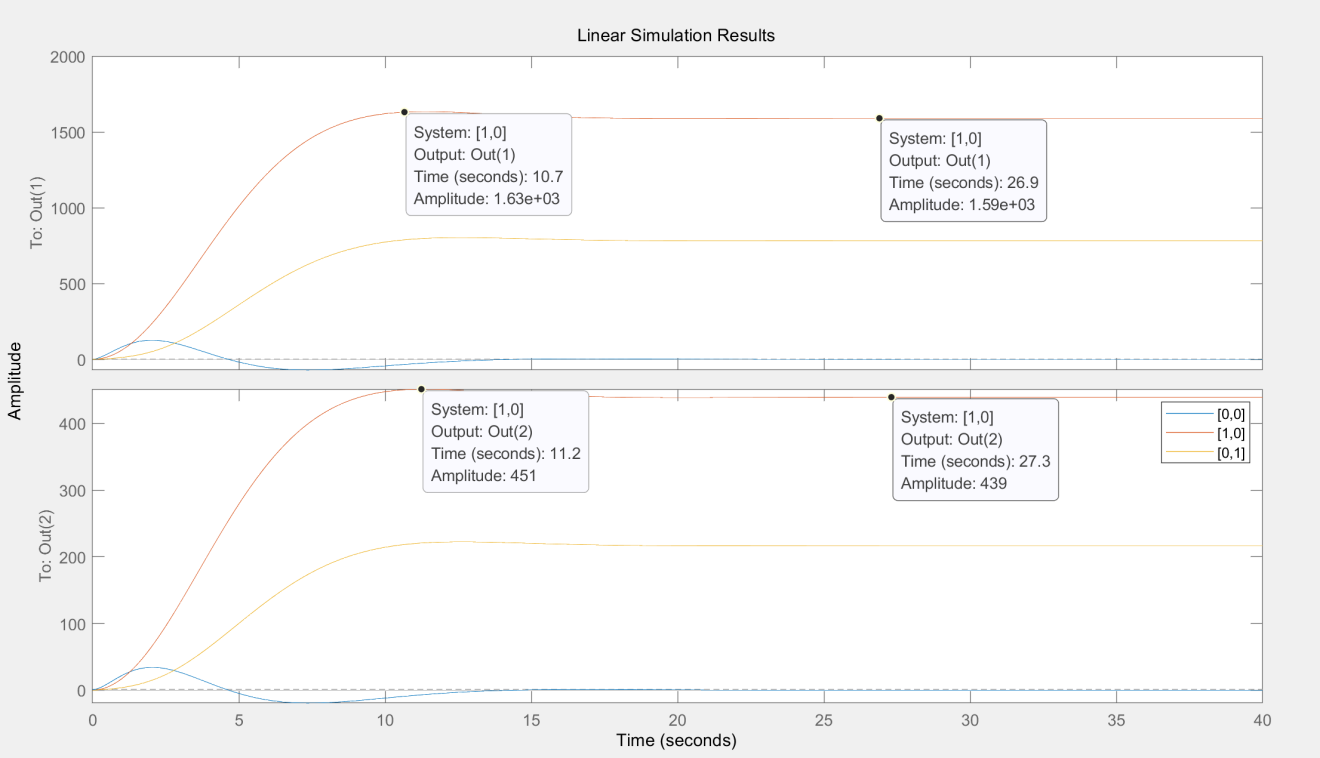


Figure 2.5 transient response

We can calculate the maximum overshoot of the two outputs based on the transient response graph:



### 2.3 Effect of Poles’ Position and Conclusion

During choosing reference model, the choice of  directly determines the setting time Ts and overshoot. The greater the , the shorter the setting time required by the system, and the greater may contribute to a better overshoot (exactly this is also related to the choice of  and the choice of irrelevant poles).

After choosing a good pair of , We get the two main poles of the desired model, and we choose irrelevant poles which little contribute to the model but only to fit the order of the actual system.

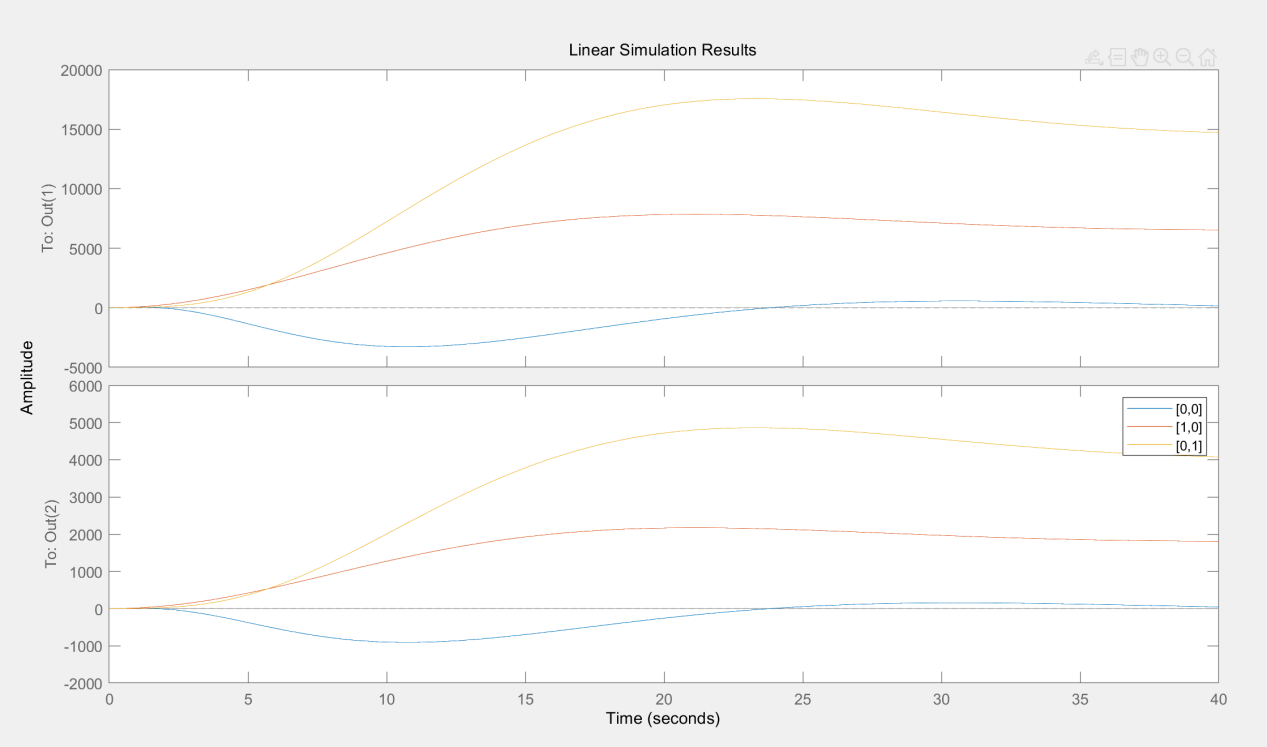


Figure 2.6 transient response when irrelevant poles:[0.13 0.13]

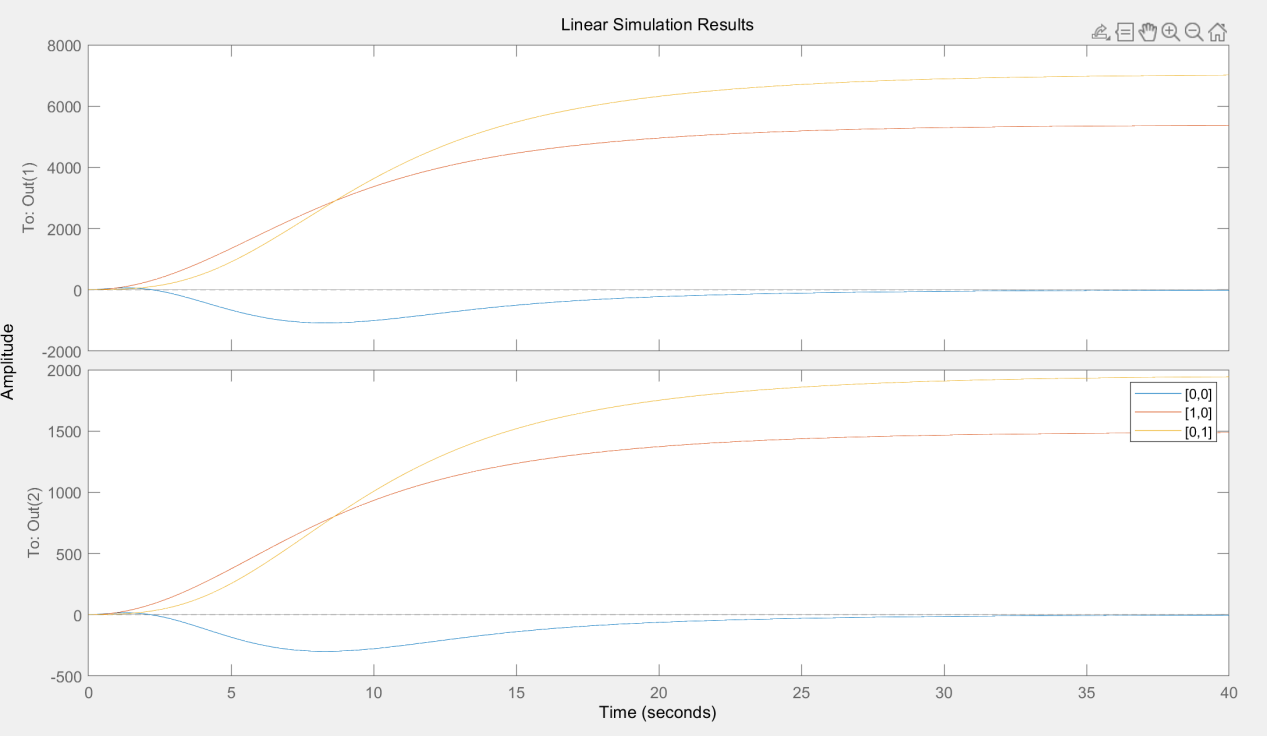
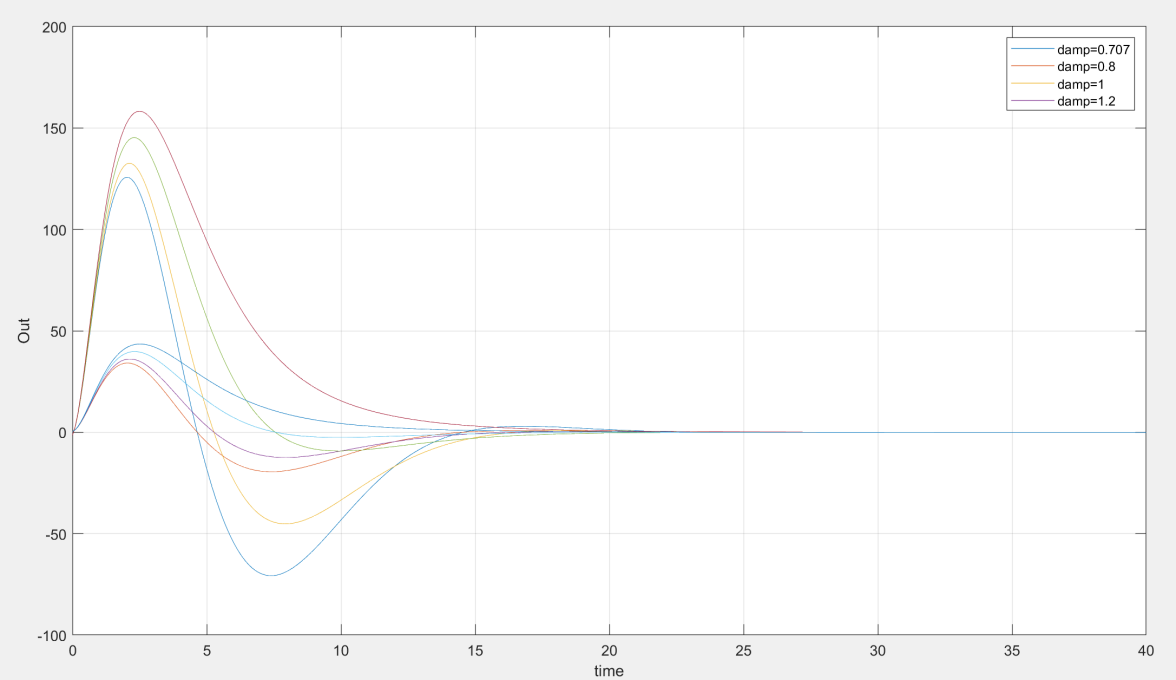


Figure 2.7 transient response when irrelevant poles:[0.3 0.3]

Compare figure 2.4, figure 2.5 and figure 2.6, it can be concluded that the positions of the poles would change the order of design model, with overshoot, setting time changed. When the gap between the two irrelevant poles and the main pole becomes larger and larger, its impact on the system becomes smaller and smaller.

Next I changed to observe how these two coefficients affect the system characteristics in the case of poles placement.

Figure 2.8 Output with Different

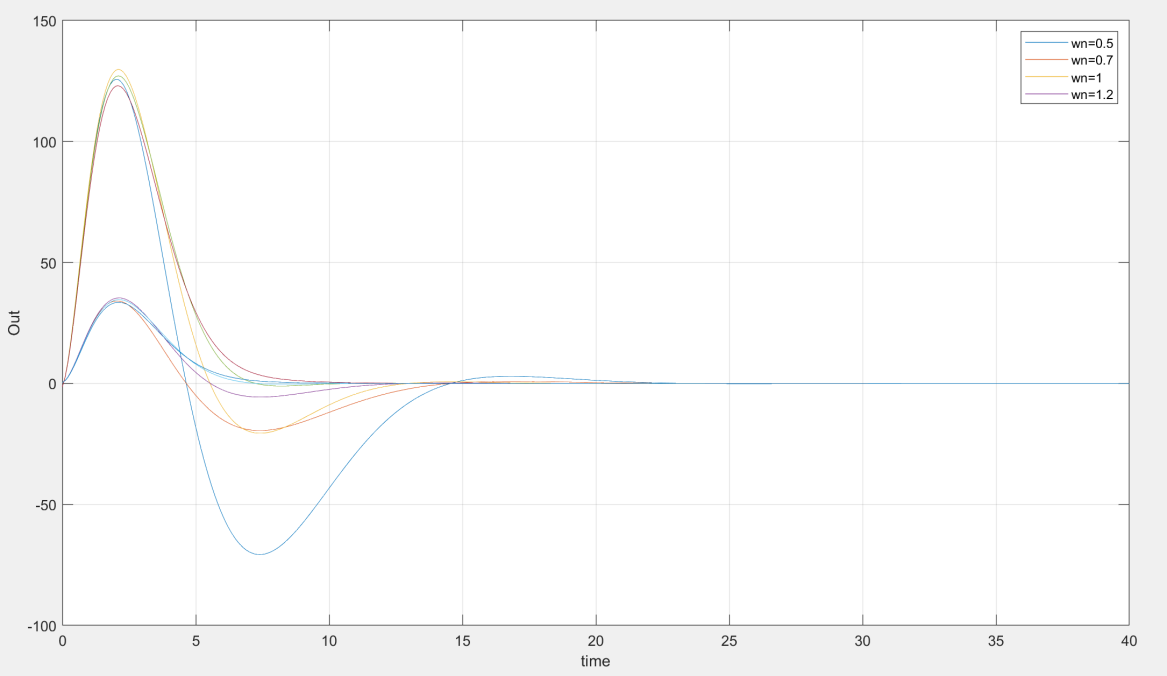


Figure 2.9 Output with Different

Through the comparison of the above two figures, it could be concluded that whenis small, the output oscillation range is obviously increased, indicating that the control ability of the system is weakened. But it may be due to the small difference in irrelevant poles placement, which has little effect on the order of the system.

As shown in the figure 2.9, the poles placement model designed by Matlab place() function is more rubust than model I designed. Because there are actually infinitely many solutions that can achieve the system goals we need, but different configurations of the poles will cause different system characteristics. Therefore, the pole should be configured according to the specific system requirements when designing the actual system.

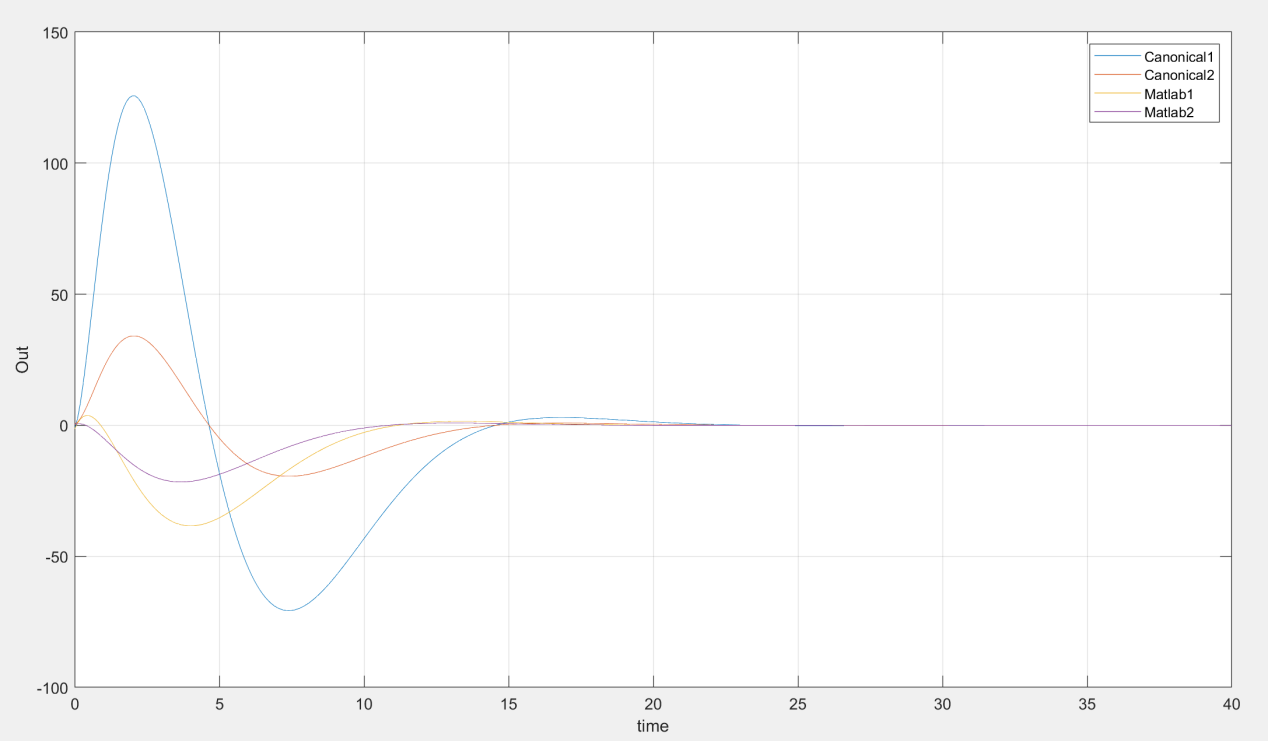


Figure 2.10 Compared with Matlab Function

1. Solution to Task2

In this task, we should design a state feedback controller using the LQR method. The LQR optimal control is to find the control law that minimizes.

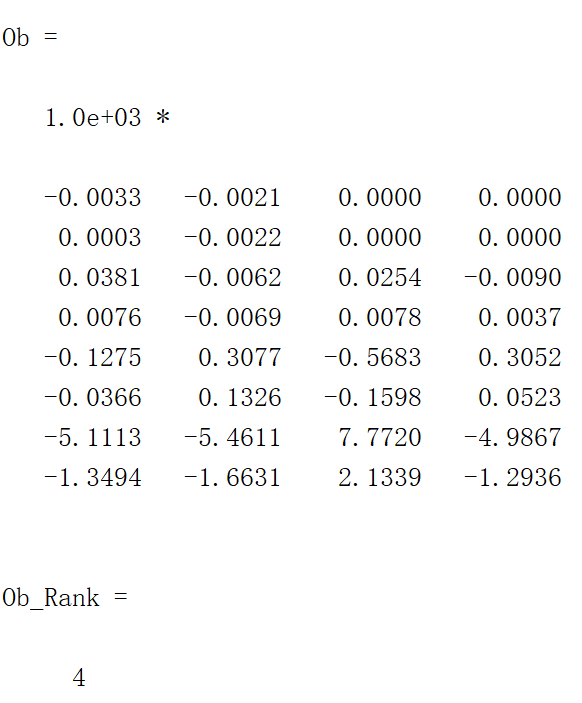
As quadratic cost function:



Therefore, to keep a balance between speed and cost, we want to minimize the following index. The weighting factors q and r express the relative importance of keeping x and u near zero. If we place more importance on x , then we select q to be large relative to r. In this case, the state x will converge to 0 faster, but the control effort will be bigger, and energy cost higher. If we care more about the energy cost rather than the response speed, then we should set higher r.

* 1. LQR Design

Before LQR design, system controllability should be verified: Rank() = 4



At the beginning of designing, assume the weight Q & R:

The minimized cost is derived by taking derivative of , i.e. , such that the optimal feedback gain follows:



Since taking the derivative of J is difficult in our 6-ordered-porcess scenario, we can borrow the concept of Lyapunov equation:



where P is a positive definite matrix satisfying ARE:



To use eigenvalue-eigenvector based method to solve the positive definite matrix P for ARE, we need to construct a 8 by 8 matrix:



And then find its eigenvalue



The next step is to pick out all stable eigenvalues, calculate their corresponding eigenvectors and then form a new matrix . After that P can be derived by . The feedback gain K is computed by using . The transient responses are respectively shown in figure 3.1 , figure 3.2 and figure 3.3.

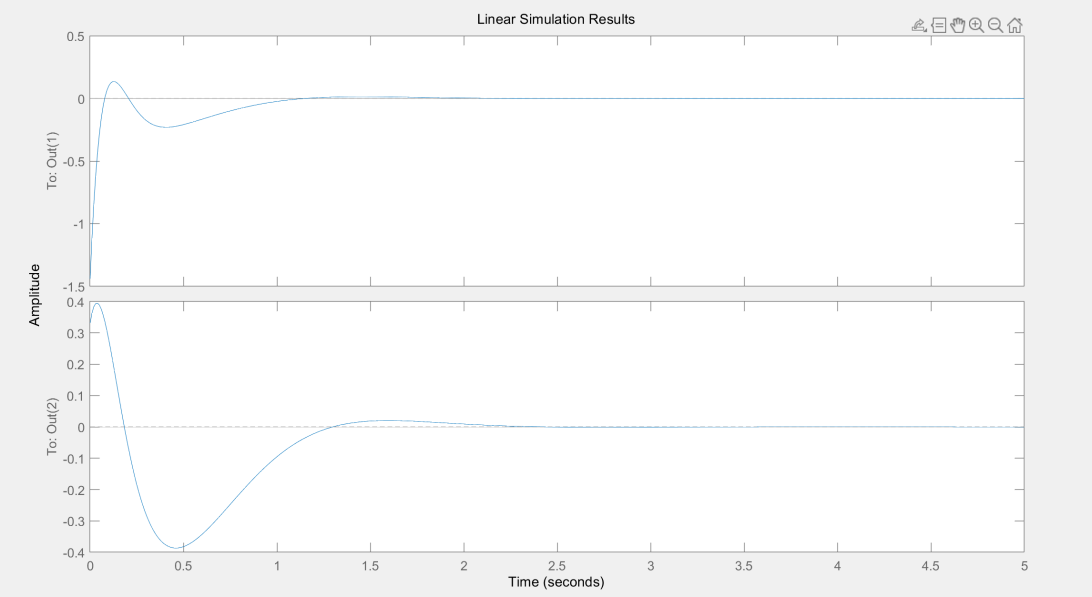
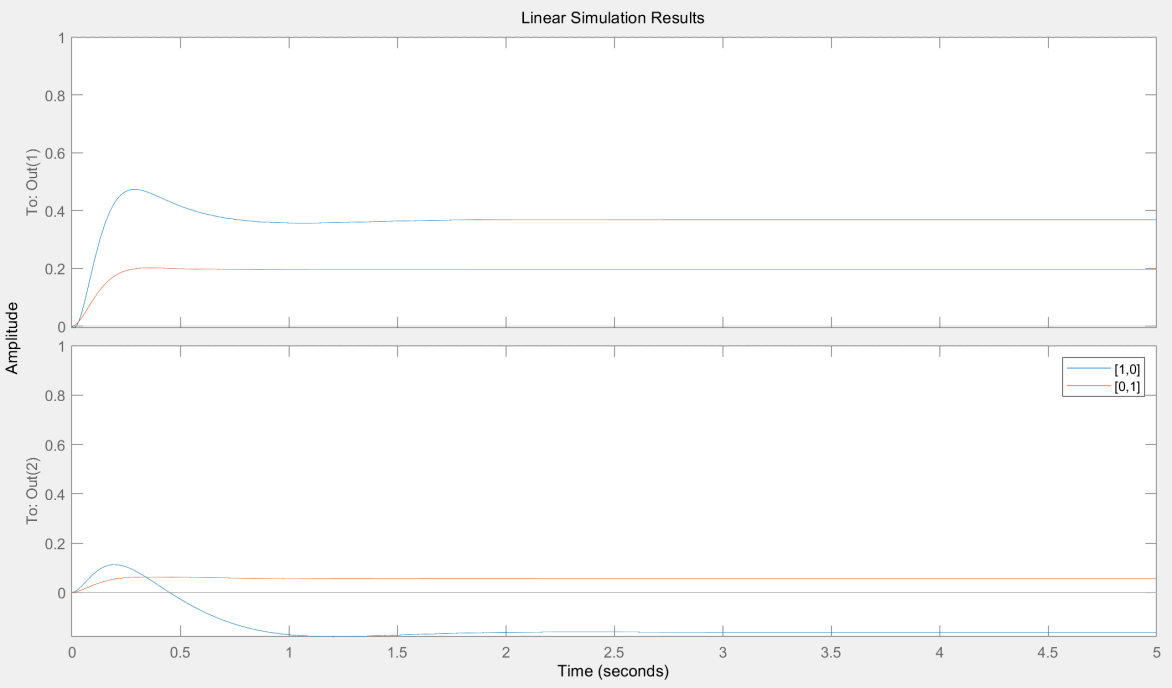


Figure 3.1 Zero input non-zero initial state response

Figure 3.2 [1 0] & [0 1] input step response

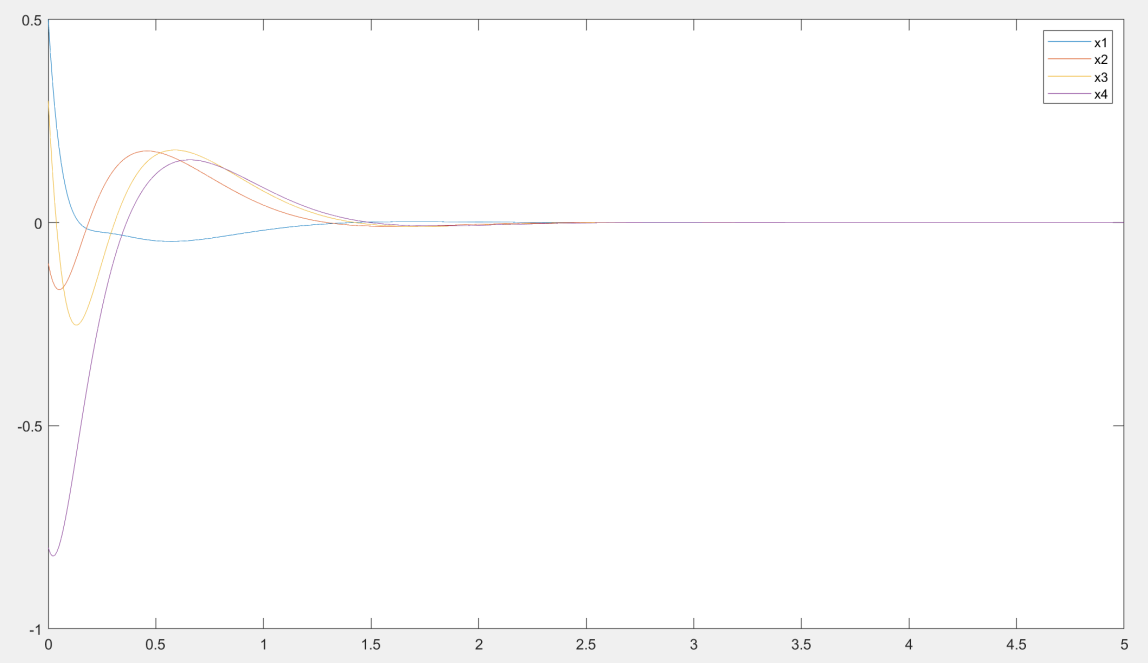


Figure 3.3 LQR: state trend under non-zero initial state

We can see that with such settings as Q and R, the overshoot condition cannot be met, so I changed Q and R to enhance the control ability of the system. The influence of Q and R on the system will be discussed in the next part.

* 1. Effect of Q & R

In order to meet the limitation of system overshoot and settling time, Q should be increased to obtain stronger control ability, and R should be reduced to obtain more energy.

Therefore I increased Q and decreased R respectively:

We can get a system that meets the conditions:

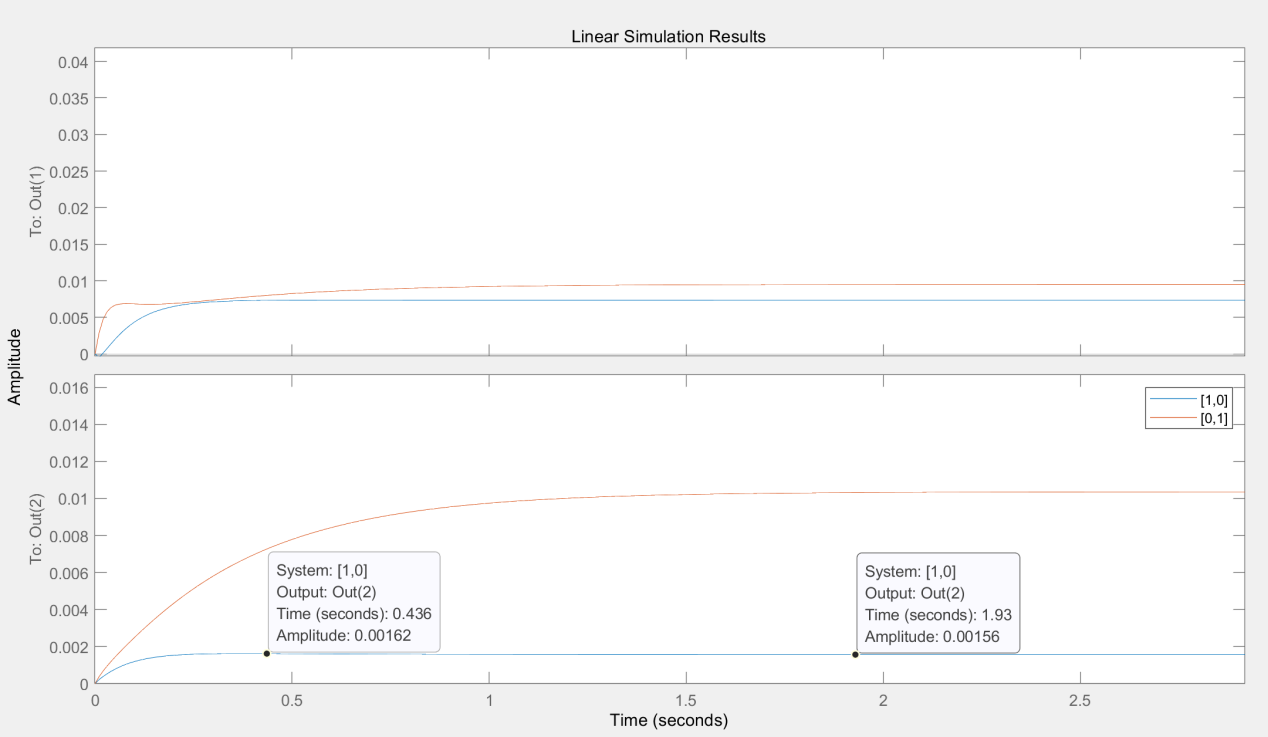


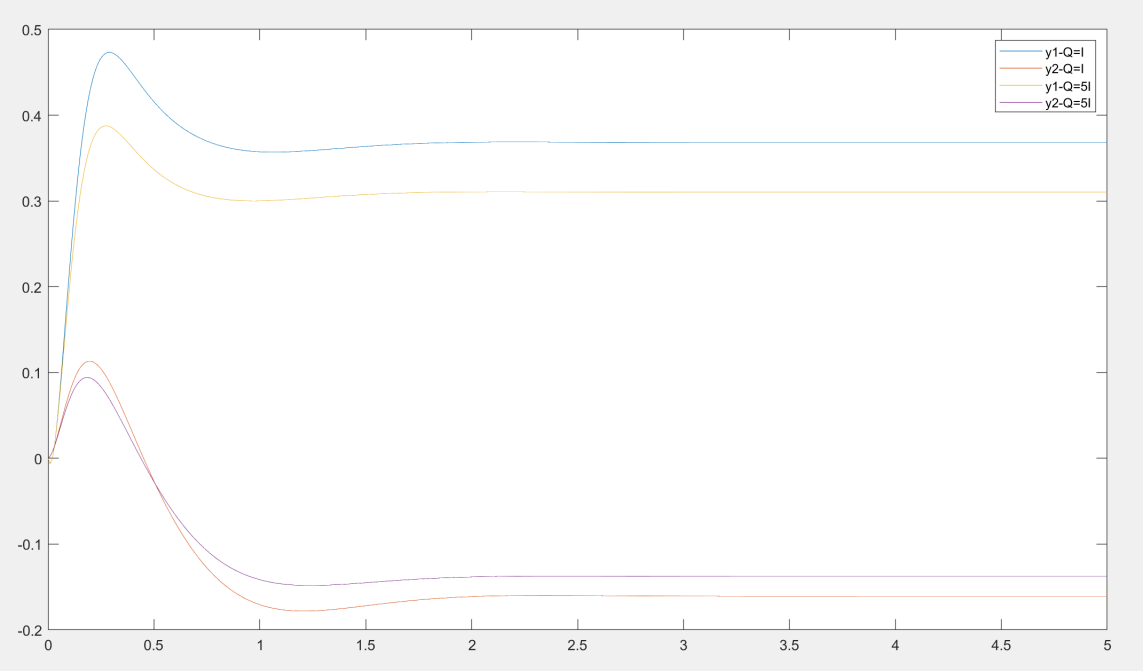
Figure 3.4 Output that meets the overshoot condition



But in order to obtain such a system, the cost required to pay is huge, so in the actual design we should comprehensively consider how to set the values of Q and R to obtain the optimal solution. The parameter matrices Q and R are respectively the weight to determine how much the controller tends to improve performance or tends to save energy. Equation can be expanded as:



From the above formula, we can conclude that because RHS is equal to 0, the ratio of  to is the key to the impact of LQR on the system. Our goal is to minimize the energy function J, so both and  should be minimized. When t approaches infinity, x should approach 0. To ensure that  is a minimum constant value, then should be large. The larger the  is, the faster the attenuation speed of  will be; in the same way, if  is increased, u will become smaller, the control of the system will become weaker, and the attenuation speed of  will become slower. The figure below shows the impact of different Q and R on the system:

Figure 3.5 Step response with different Q

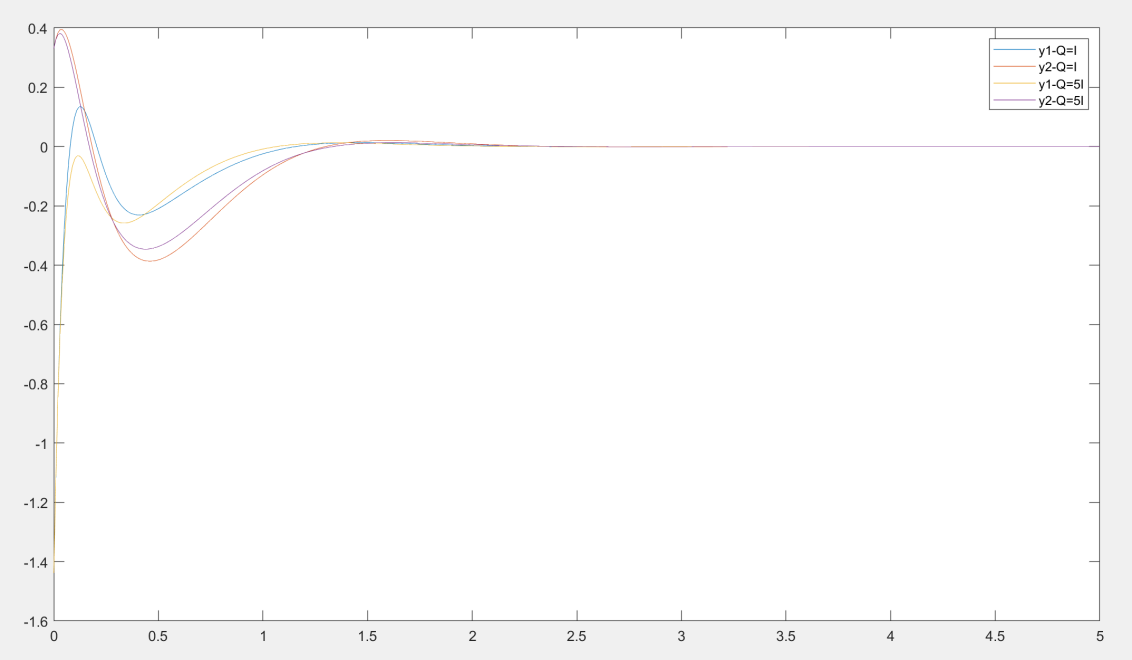


Figure 3.6 Non-zero initial state output with different Q

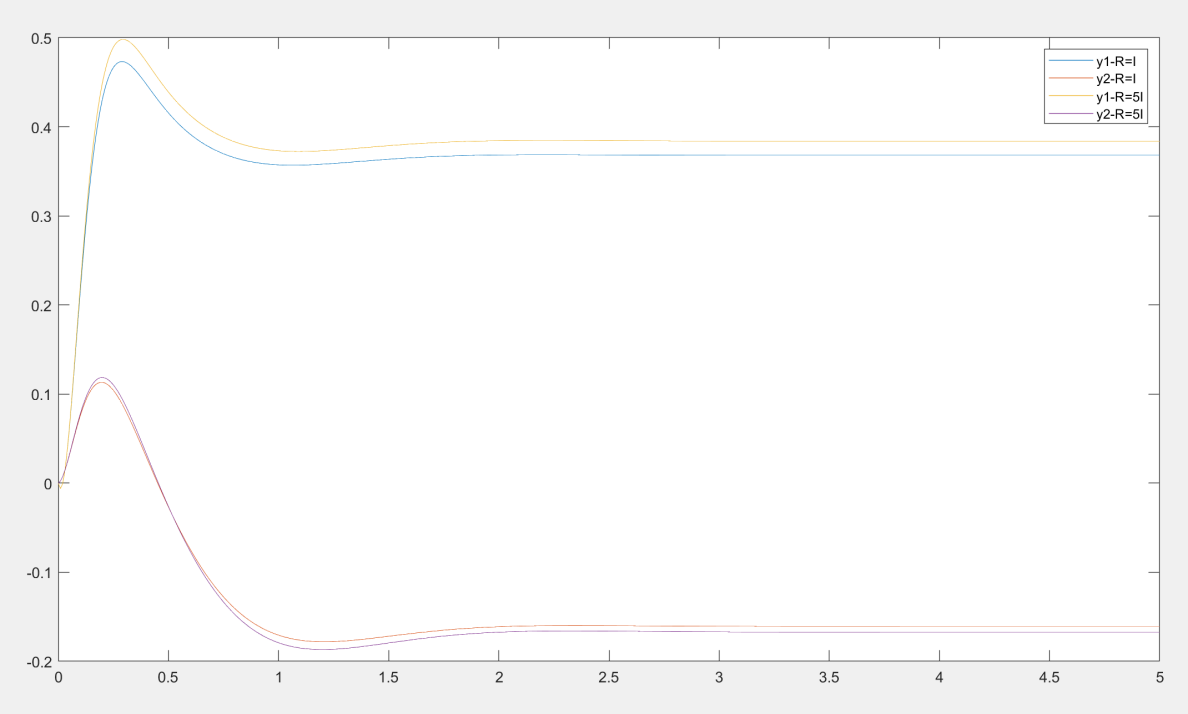


Figure 3.7 Step response with different R

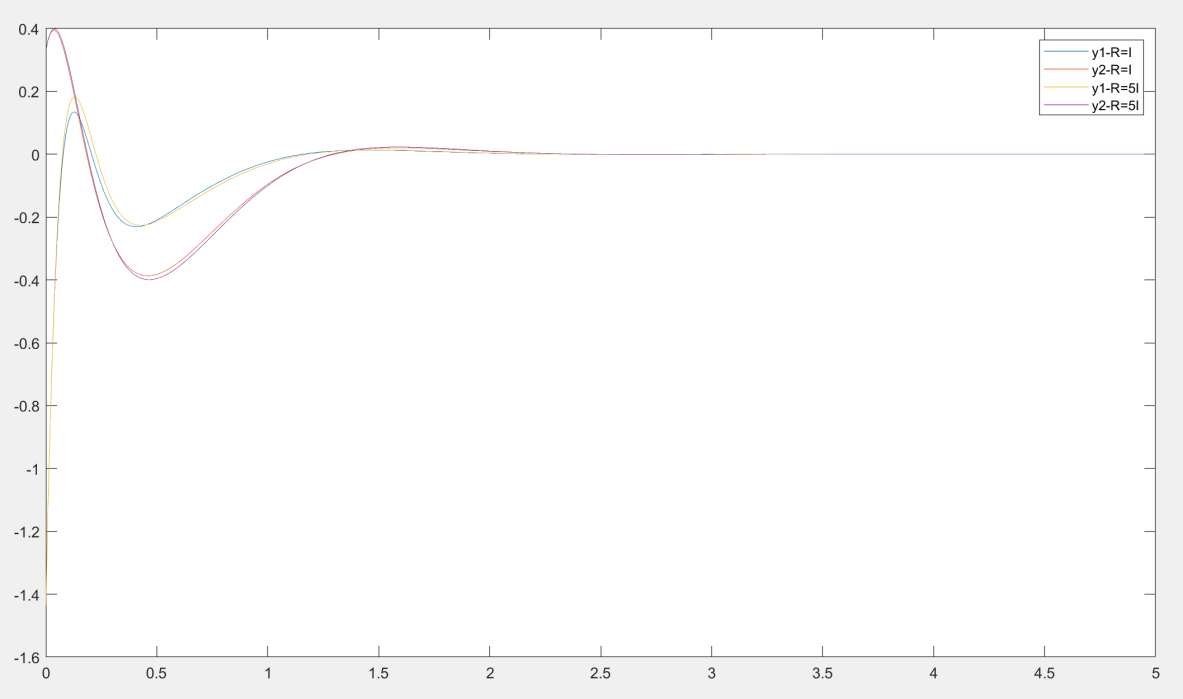


Figure 3.8 Non-zero initial state output with different R

As shown as above figures, the following conclusions can be drawn:

1. At the same time Q=>kQ, R=>kr, no effect on x1
2. Just take Q=>kQ or R=>R/k, the effect is the same
3. Increase Q, strengthen control over x1

About Q internal:

1. Increase the coefficient of x1, strengthen the control of x1
2. Increase the coefficient of x2, and x1 is basically unchanged
3. Solution to Task3
   1. Full-State Observer

When considering an estimator, the system would be:



Observer=Model + Feedback Correction Mechanism

In order to estimate some states that are not convenient for measurement, our system designs a method to estimate some states of the system, combining the states that have been measured, to obtain the full state in order to implement full state feedback. Let the estimation error in the state be:



If we choose L such that (A − LC) = A1 is stable, we have

As  will 'track' asymptotically

For the observer problem, write its error characteristic polynomial:

So we want to make  be stable and have arbitrary eigenvalues, for the observer problem, write its error characteristic polynomial:



we can derive an expanded state space whose state

vector contains both  and :



Then establish the simulink model of Full-state observer:

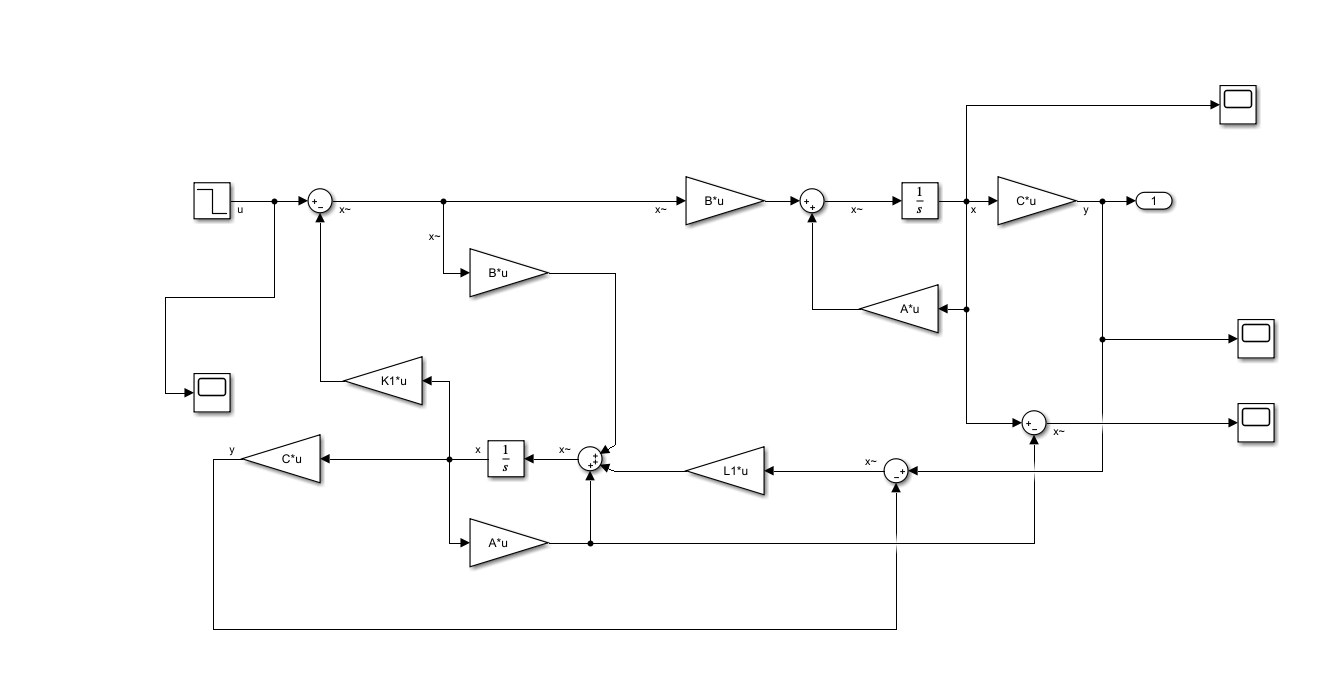
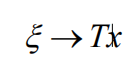


Figure 4.1 Full-State Observer feedback Control

* 1. Reduced-Order State Observer

We want to take advantage of the m state variables that are available through y and construct an observer of order n − m , lower than n, to estimate the remaining (n-m) state variables. Let ξ = Tx , where T is (n − m) × n with (n − m) rows that are linearly

independent of C. This is one of the keys to the design of a reduced-order observer.

Construct an observer as:

as t goes to infinity.

* First is to find the constraints on T such that is non-singular
* Second choose D such that its eigenvalues have negative real parts, or desired decay rates
* Solve DT −TA+ GC = 0 for T and G

The simulink model is showing:

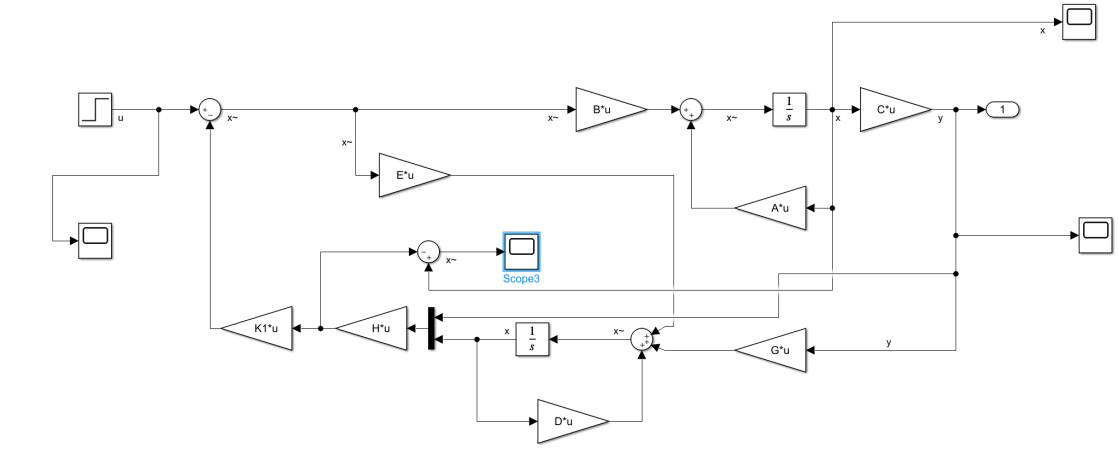


Figure 4.8 Reduced-Order State Observer

For the reduced-order observer, when some of the state variables in the control object are known, there is no point in predicting their size. Therefore, these state variables need to be regarded as known quantities, and only the state variables that need to be estimated and construct the observer, at this time the order of the state observer is reduced, For this system, because of ,

it is not convenient to compute the uncertain state variables, I chose to use a full-order observer to complete this task.

* 1. Full-Order State Observer

To make e converge to zero, we should design the observer feedback gain . It can be calculated using either pole placement method or LQR method. Because the original system uses the K of the LQR configuration, I also use the observer to configure K when designing the observer. To meet the requirements of overshoot and settling time, I set , step response is as shown as the picture:

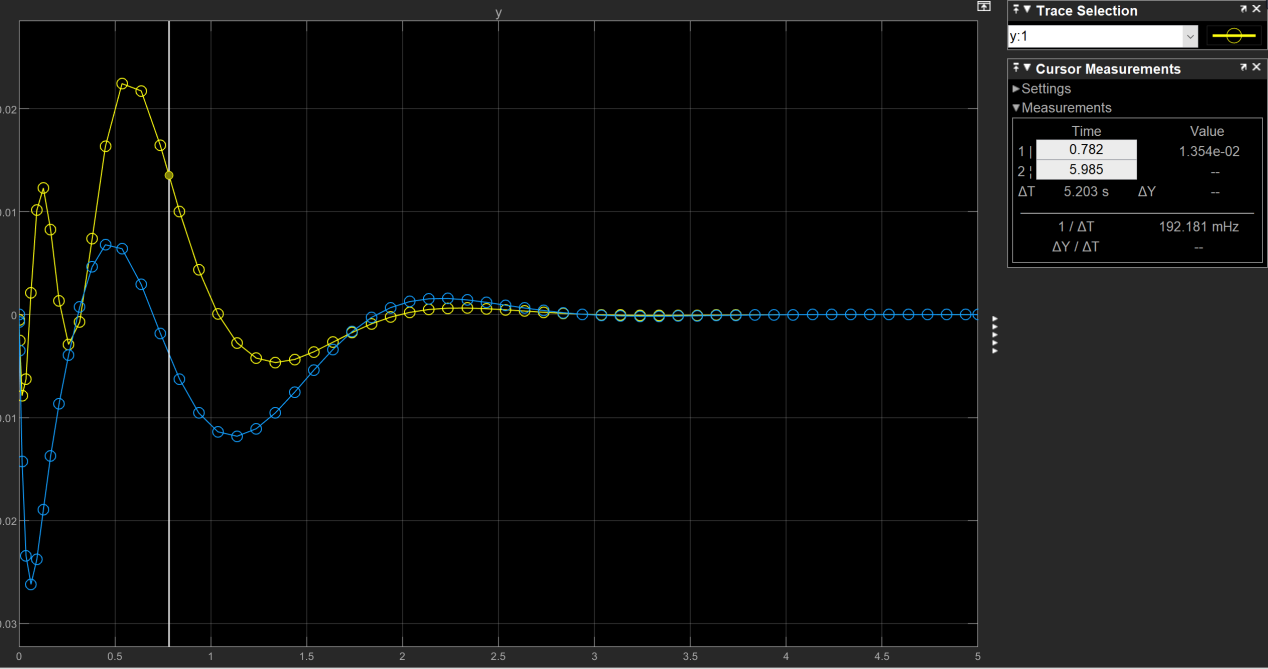


Figure 4.2 Step response with LQR method

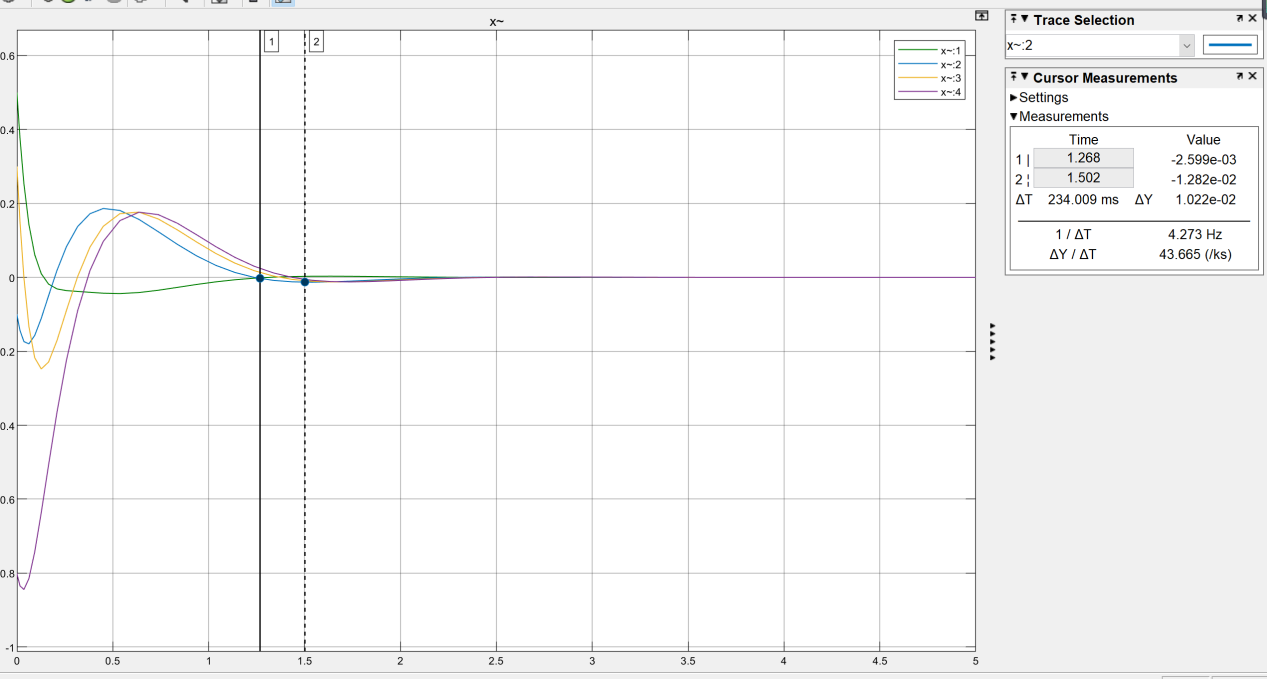


Figure 4.3 state response with LQR method

In addition to LQR, poles placement can also be used to find the gain of the observer L. Design procedure is summarized as follows:

1) Choose the observer poles 3-5 times faster than controller poles

2) Use a pole placement algorithm to obtain L

3) Implement the observer

The original system poles are: [-48.95 -13.088 -2.467+2.487i -2.467-2.487i]

So I set new observer poles are: [-0.707-0.707i -0.707+0.707i -2.4 -2.0]

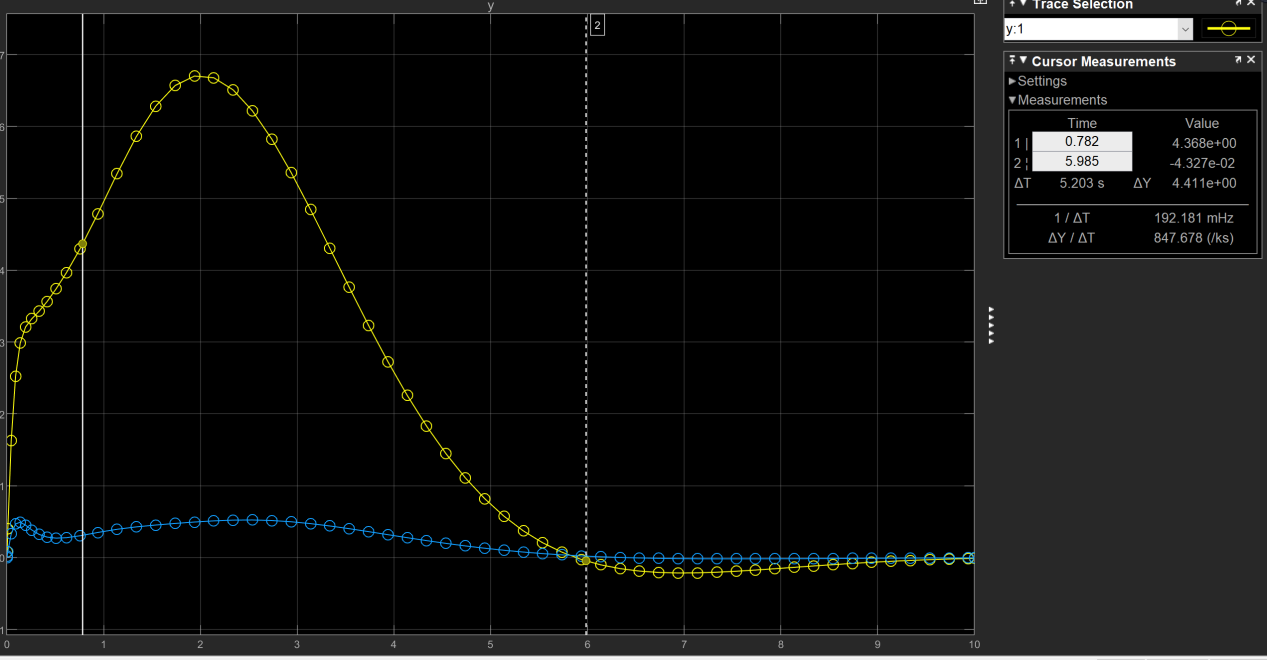


Figure 4.4 step response with poles placement method

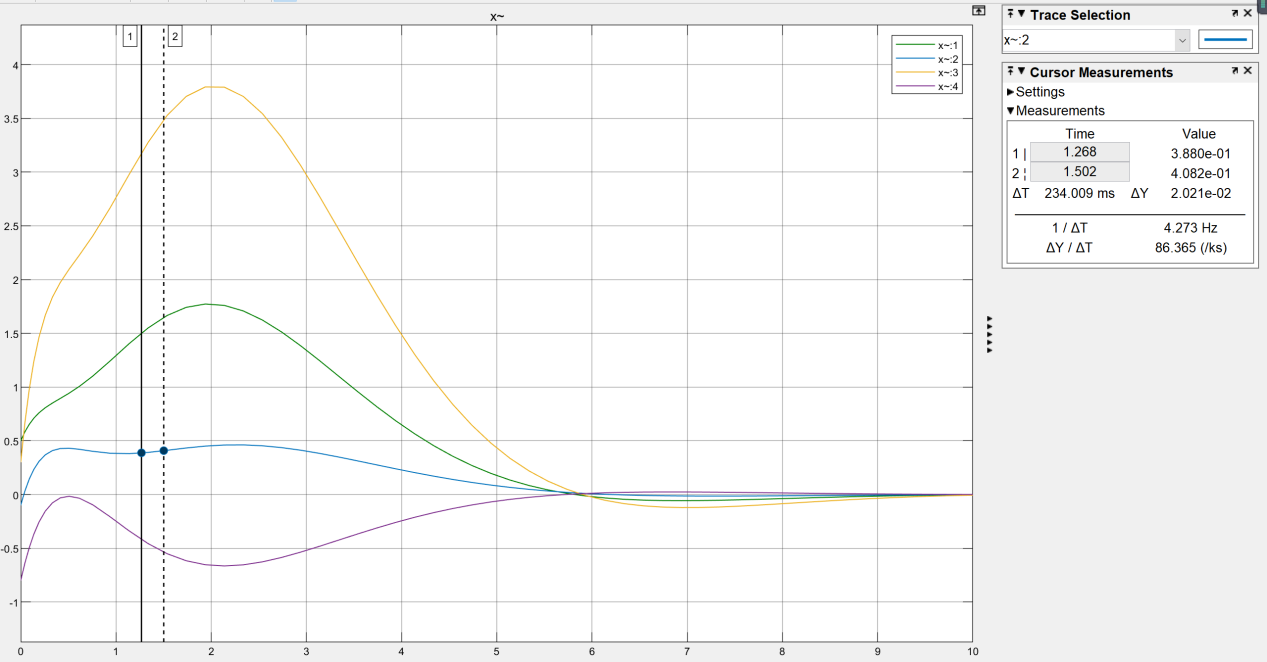
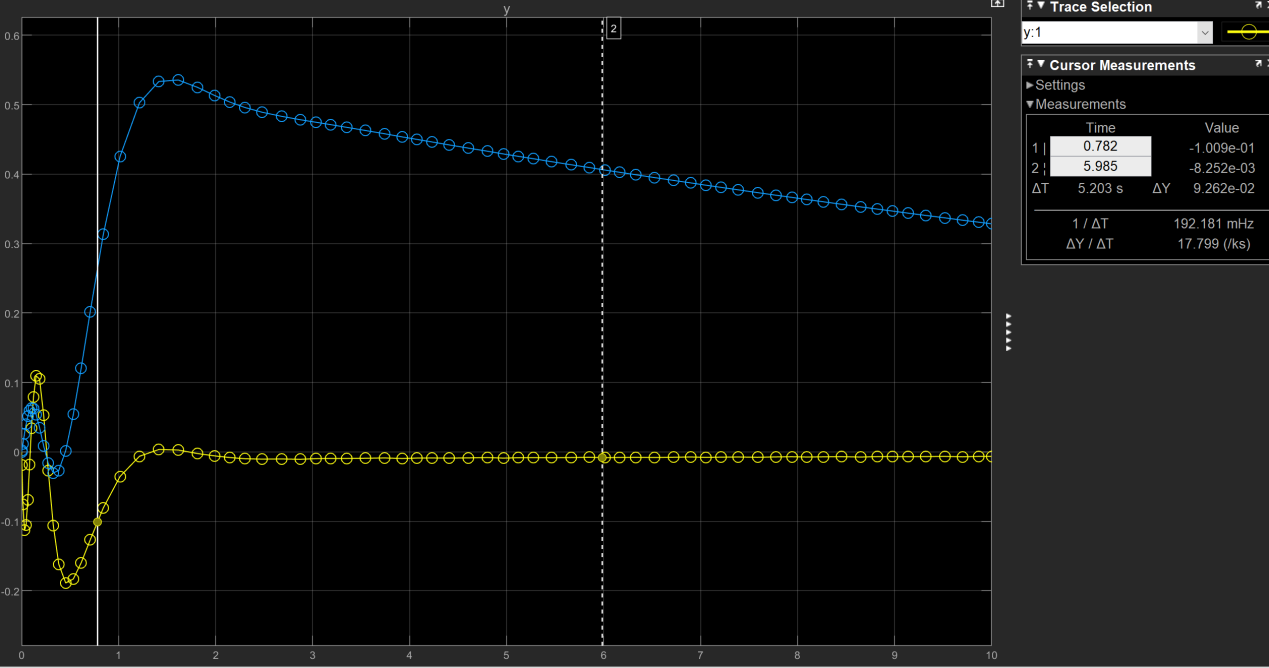


Figure 4.5state response with poles placement method

In order to compare the impact of changes in poles of the system, the coefficients Q and R are changed: , we can get the estimated state response and output response:

Figure 4.6 step response with new poles

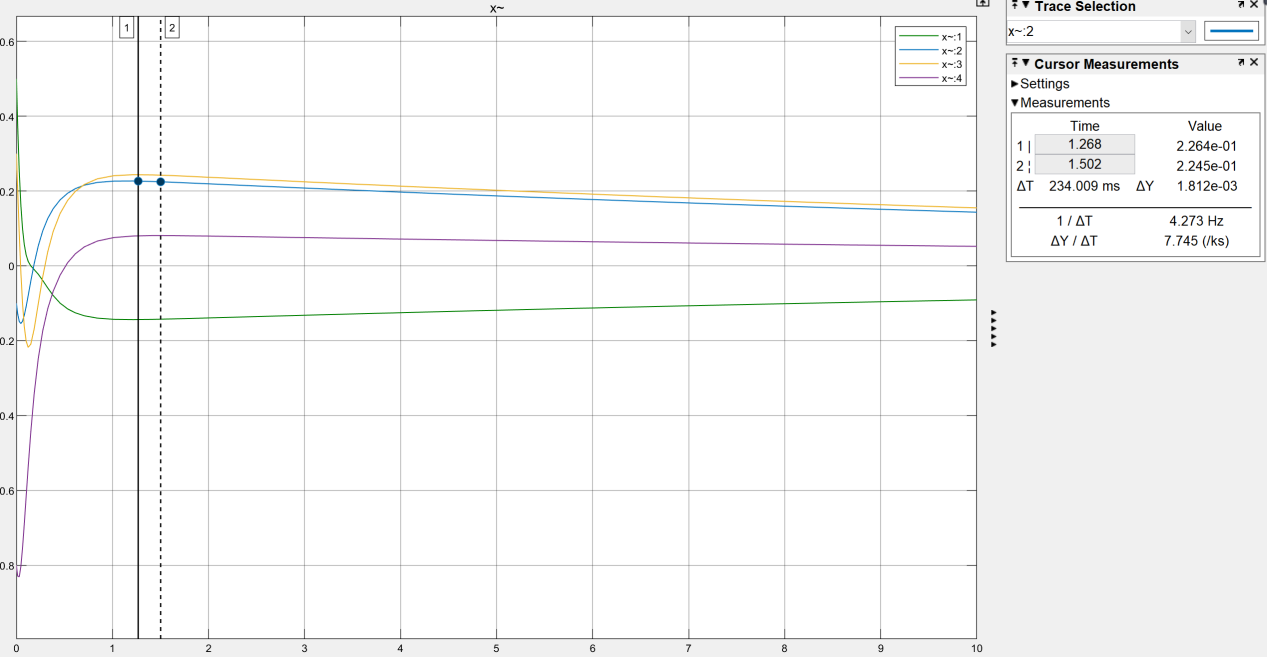


Figure 4.7state response with new poles

These three situations occur at the poles:







Through comparison, it can be concluded that when the negative real part of the main pole decreases, its state-estimation-error convergence speed becomes slower.

In conclusion, when the negative real part of the main pole of the state observer is larger, the state observer estimate state converges faster.

1. Solution to Task4

If  and for all i ≠ j, the system is called decoupled. A

decoupled system has a diagonal and non-singular G(s):



As shown in the figure, a decouple system is to decouple a coupled plant such that

feedback system is decoupled.

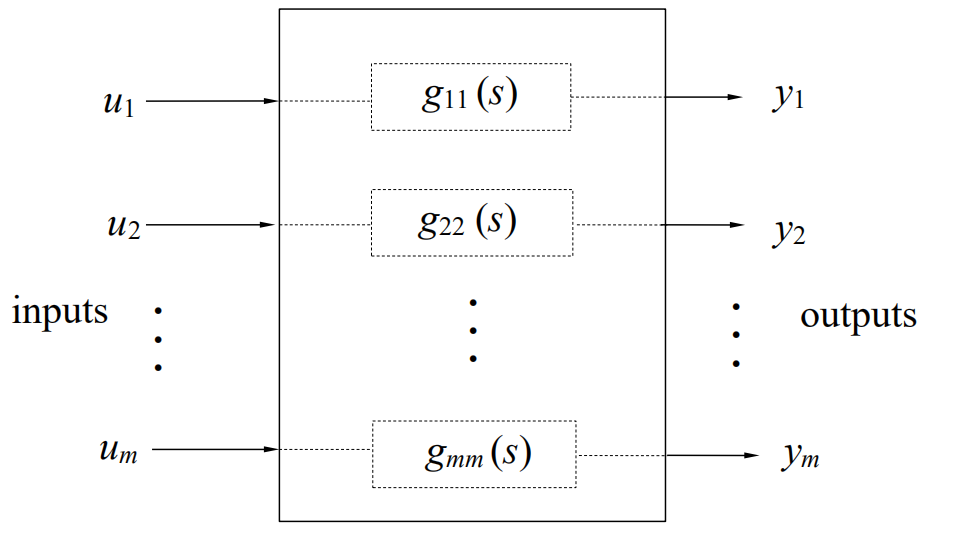


Figure 5.1 Decoupled system

* 1. Decouple with poles placement

The difference between the degree of the denominator (number of poles) and

degree of the numerator (number of zeros) is the relative degree of the transfer

function. Relevance is given by = given, where m and n are derived from



Hence, the whole transfer function can be written as



In our case, , so



The feedback gain , because , is the desired characteristic polynomial, in this task, I set = s+5, and then replace s by A, we can get . Because the total relative degree is , there must exist two poles that the controller cannot move. In the case of decoupling, we cannot configure all the poles. It can be computed that the four poles are  There is a positive pole which we can’t move, so it doesn’t exist a stable decoupling solution for this system.

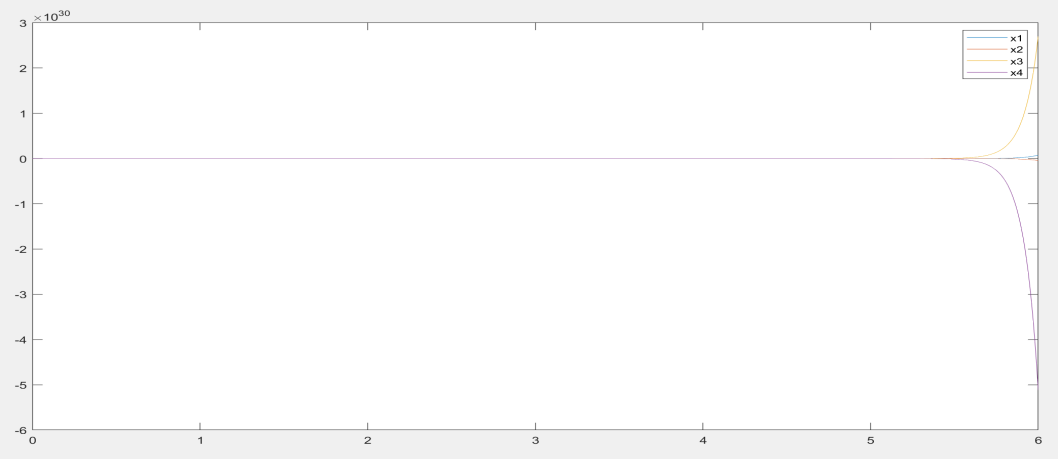


Figure 5.2 State response of decoupling system

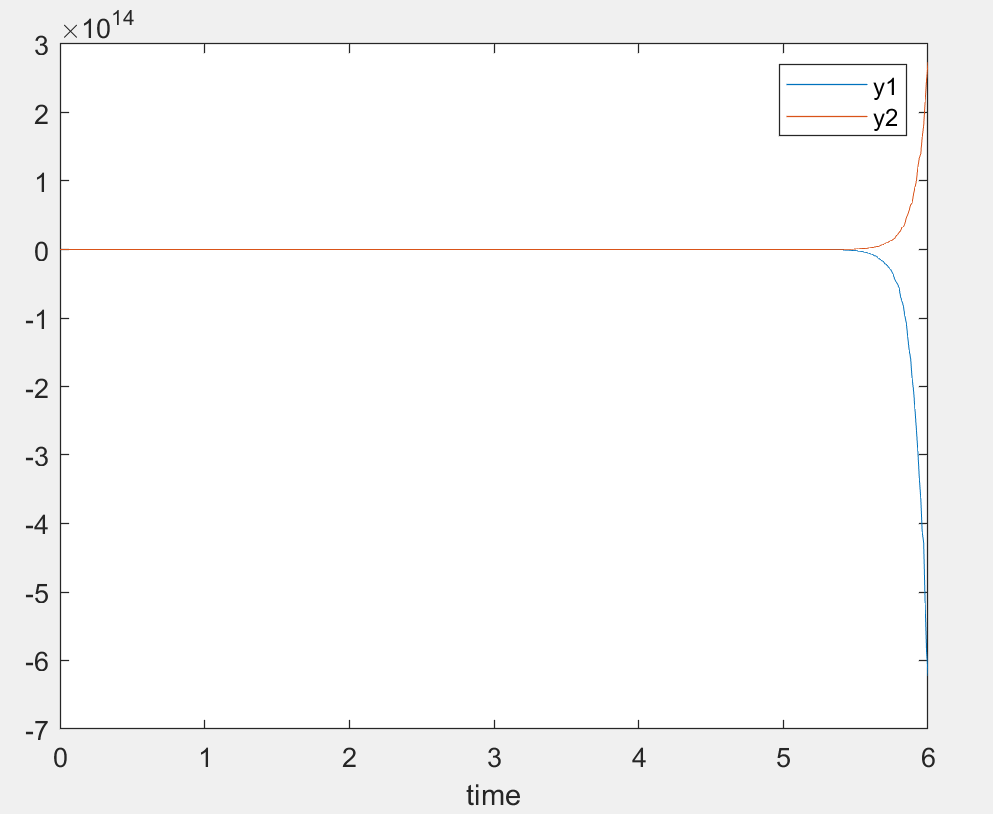


Figure 5.3 Step response of decoupling system

And then compute 

 It refers that this system has been decoupled but the system could not be stable, because its poles are .

As shown as the calculate results, this system is not internally stable. But according to our configured H(s), this system is BIBO stable.

1. Solution to Task 5

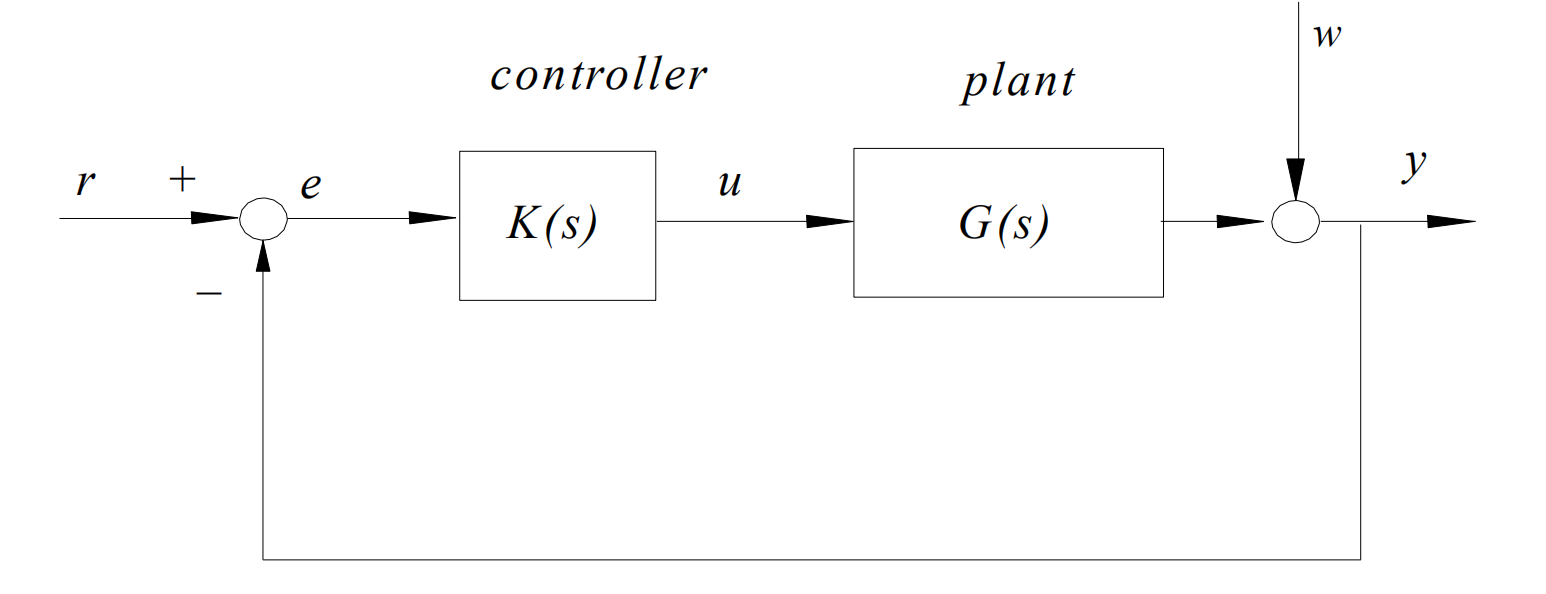


Figure 6.1 Unity output feedback system

For servo control systems, we hope，the feedback system is stable if and only if all the roots of its closed loop characteristic polynomial. The we want to use Q(s) to eliminate unstable poles. Suppose that the feedback system is stable. Let the reference r(t) and disturbance w(t) be: 

Hence, if Dr (s) and Dw (s) have zero or positive real parts. Take the least common denominator of the unstable modes of R(s) and W(s) , and assign it as a polynomial Q(s) of degree q.

In this task, because we only have two sensors to measure the output, it is necessary to

design a full-ordered state observer to estimate all states.

### **6.1 Servo Control**

Servo control design procedure:

1. Obtain plant coprime polynomial fraction as G = N(s) / D(s).
2. Determine Q from the given types of disturbance and reference, and include Q into the poles of the controller A(s). This is the servo mechanism. And the rest is to make the whole system stable.
3. Design to stabilize the generalized plant, N / (DQ).
4. Form the servo controller as

This task is a MIMO case, consider an m×m plant with m inputs and m outputs:



The objective is to achieve zero steady state error for all the outputs. Introducing the integral of error signal can cancel out unstable factor in input and disturbance signal, such that output can converge the reference set point in finite time. So establish the simulink model:

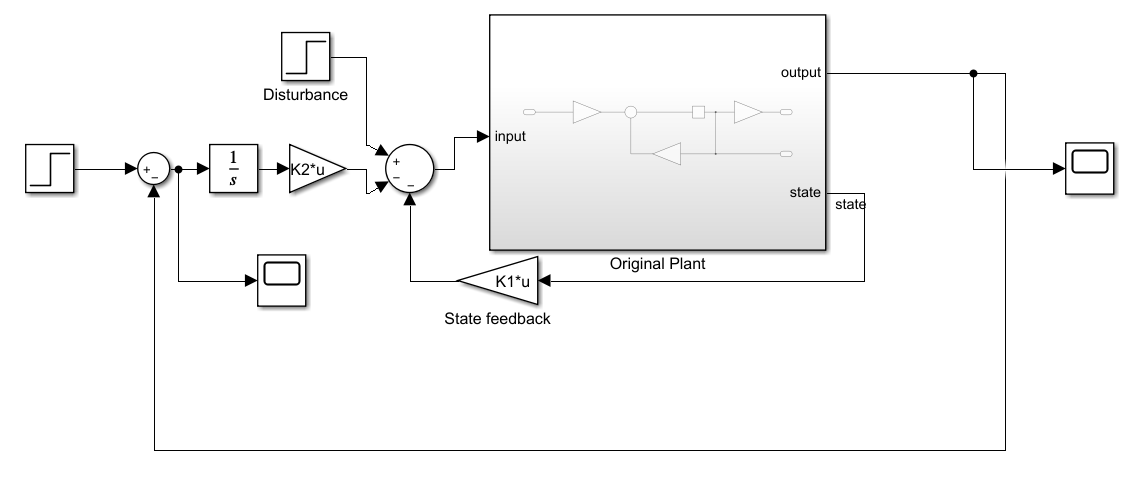


Figure 6.2 Servo controller simulink model

where K1 is the feedback control gain which is responsible for stabilization, K2 is

the integral gain in charge of set point tracking. It can be checked that this augmented system is controllable. To stabilize the closedloop augmented system, we can use pole placement or LQR method. In this task, I choose to use LQR method.

The optimal control minimizing , as the same as task 2,

I set Q = I, R=I, after computing: 



Use simulink to verify my design, we can get two figures:

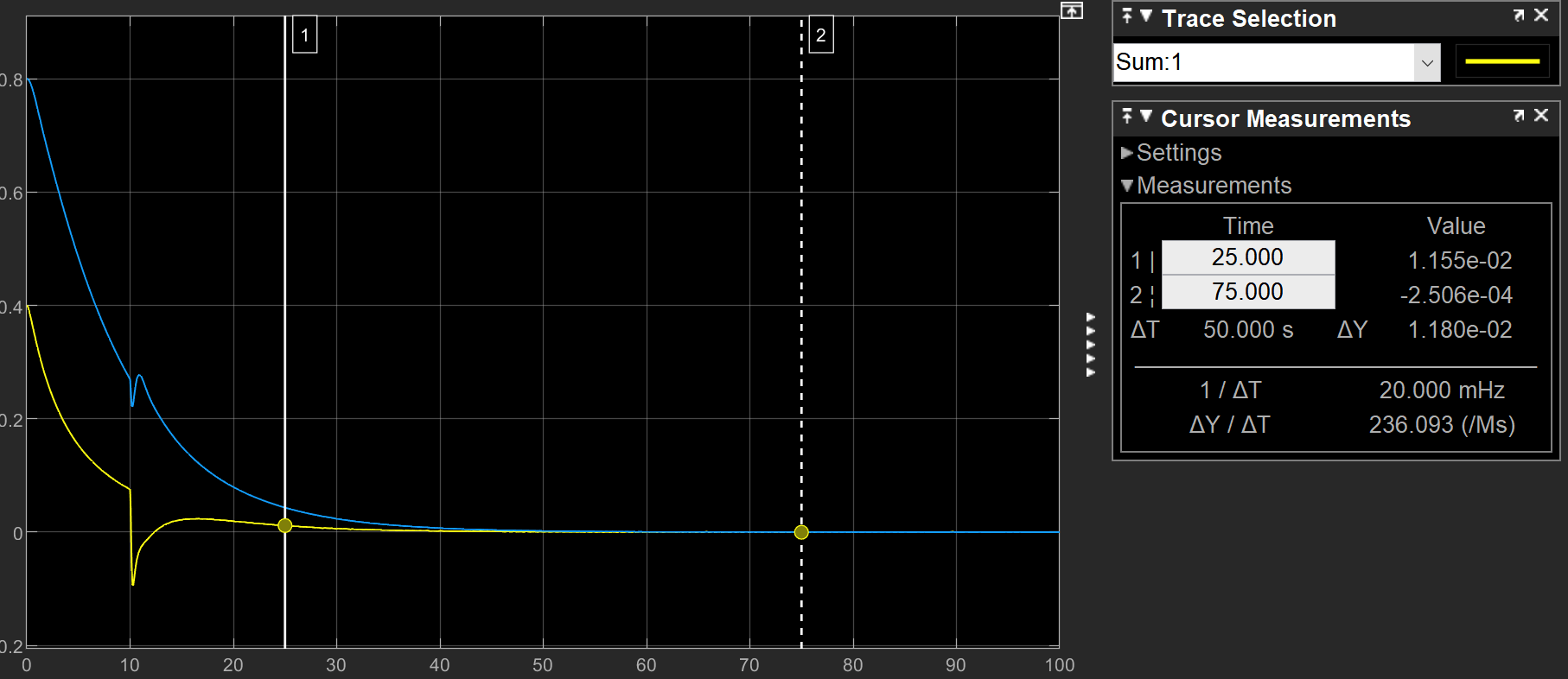
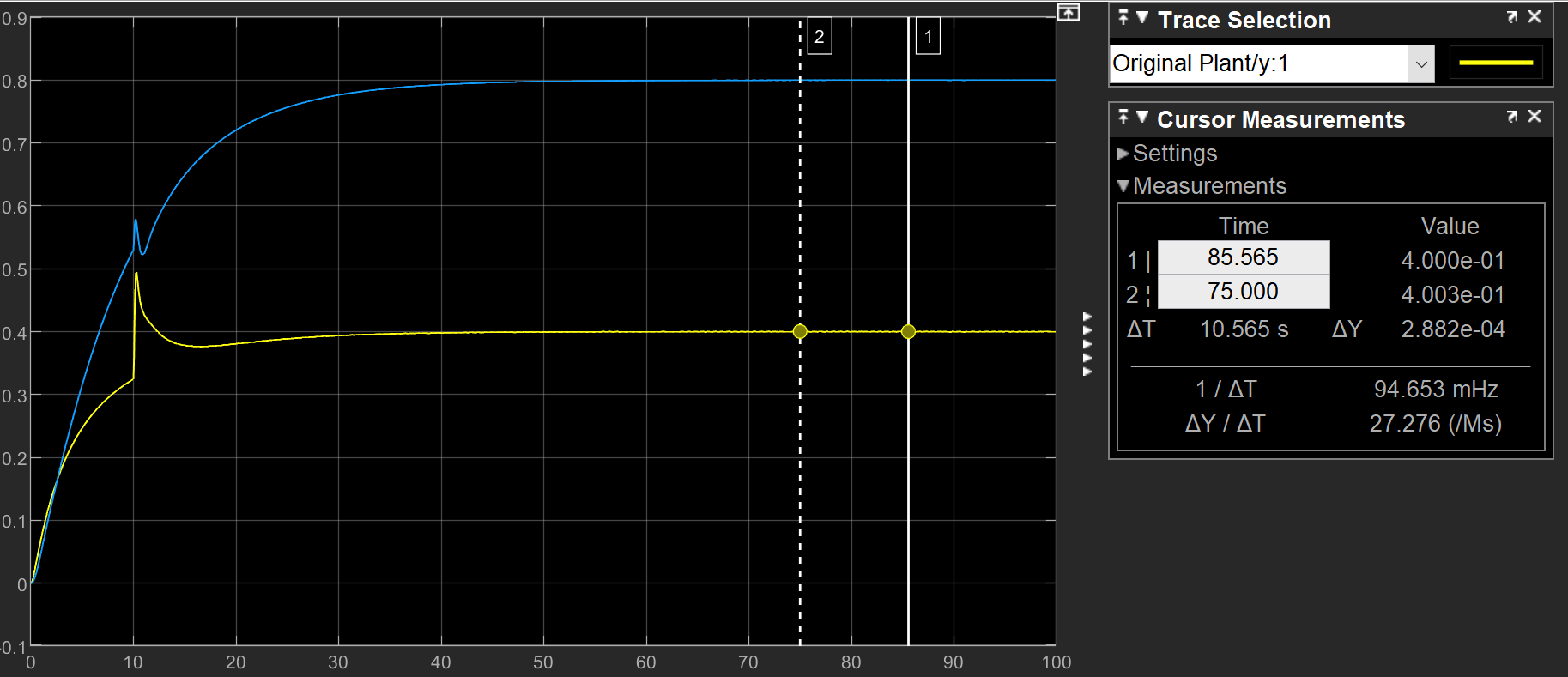


Figure 6.3 Error status change with disturbance

Figure 6.4 Output step response with disturbance

In order to get stable faster, I increase the Q to 10I, and the error should be:

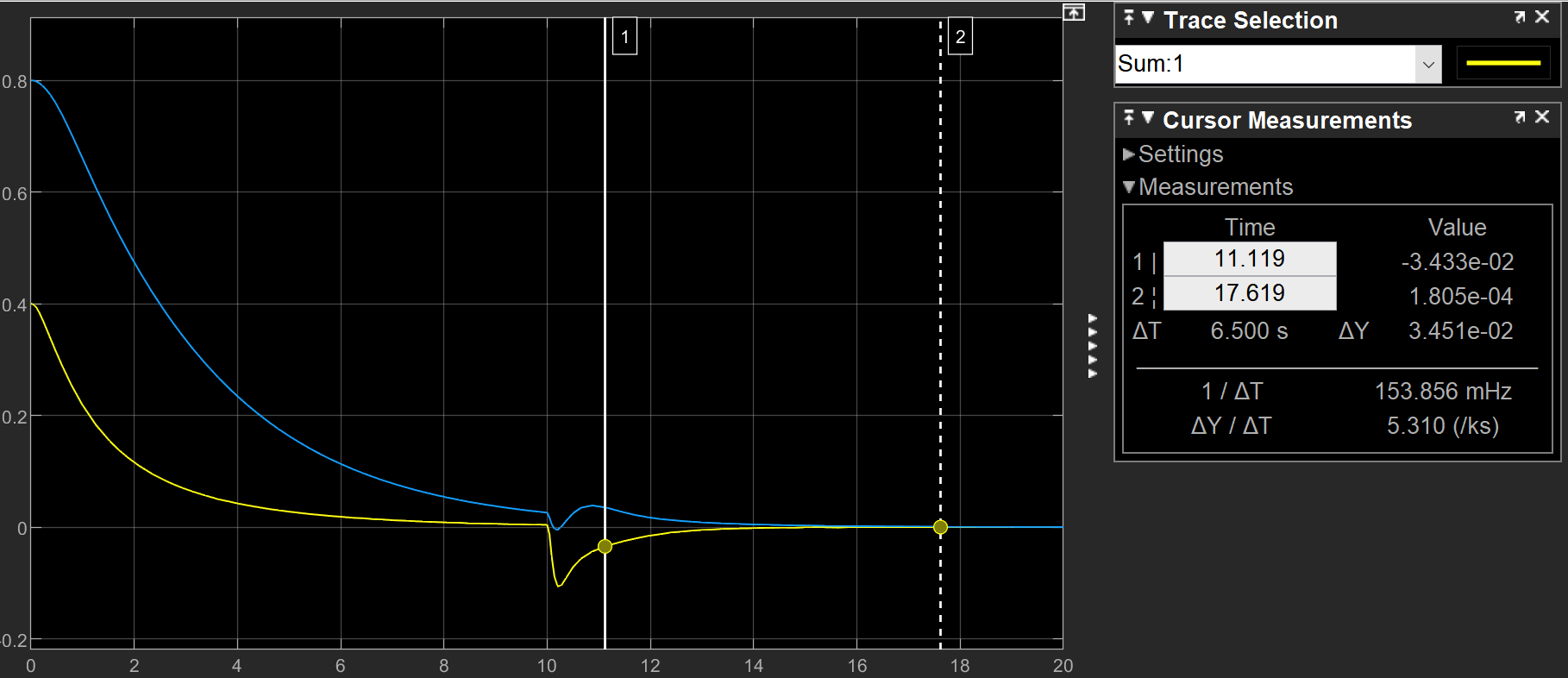


Figure 6.5 Error status with Increased Q

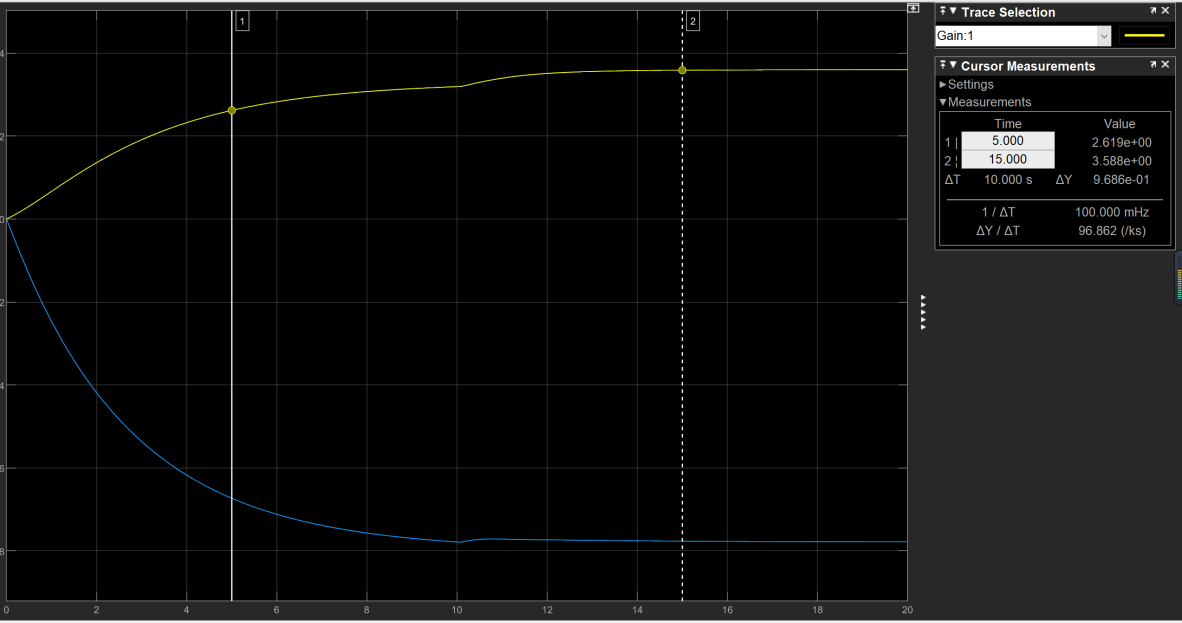


Figure 6.6 Controller signal with Increased Q

From these figure, we can find the output of the system was traced to YSP=[0.4 0.8] at the end, and the disturbance does not affect the stability and tracking ability of the system.

6.2 Add full-ordered state observer

In reality, we only have 2 cheap sensors, so an LQR based full-ordered observer is built to estimate all states. The whole block diagram is shown in Figure 6.6.

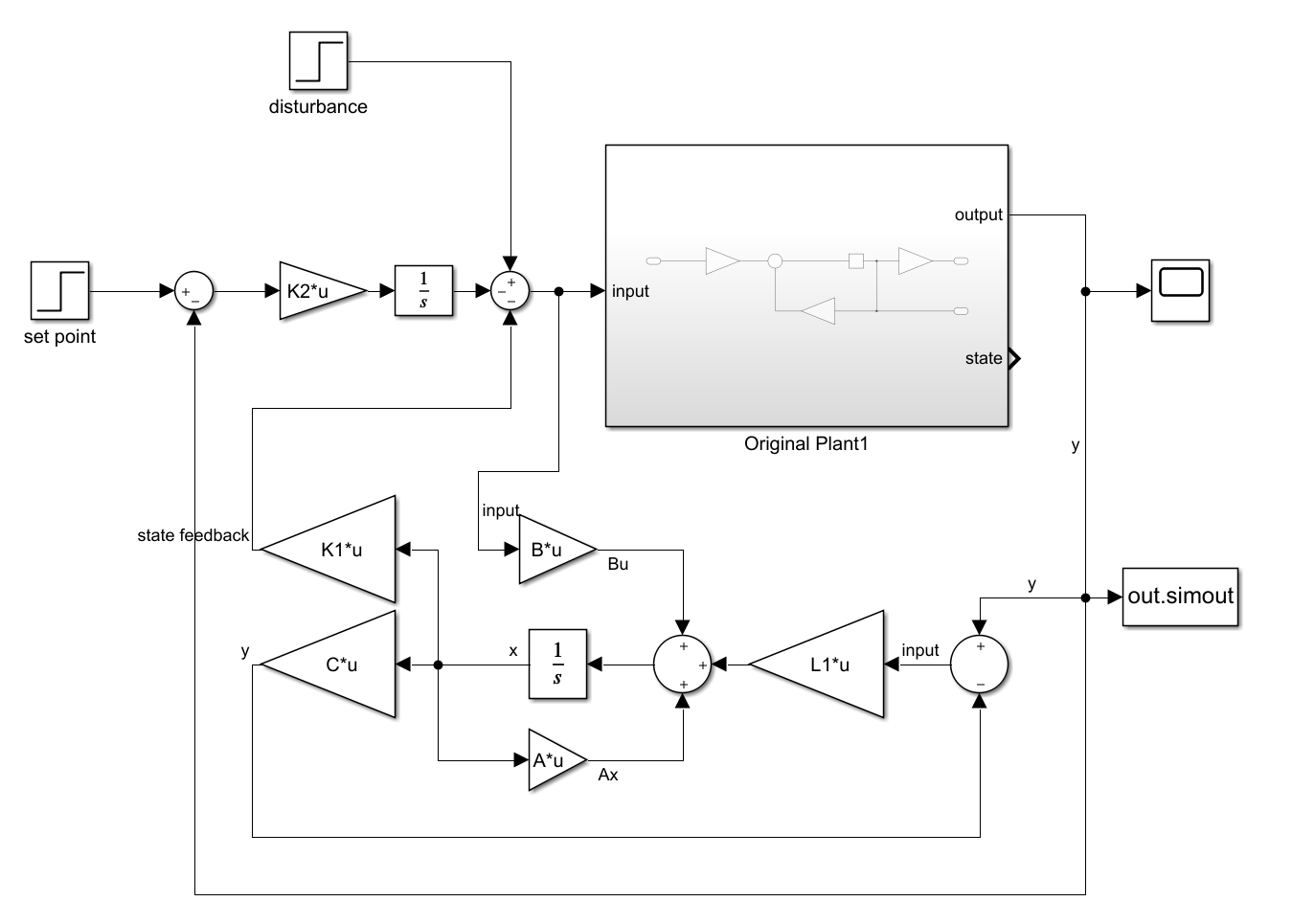


Figure 6.7 Servo controller with observer model

I set Q=10I, R=I to design this observer, and the output is the same as the original system.

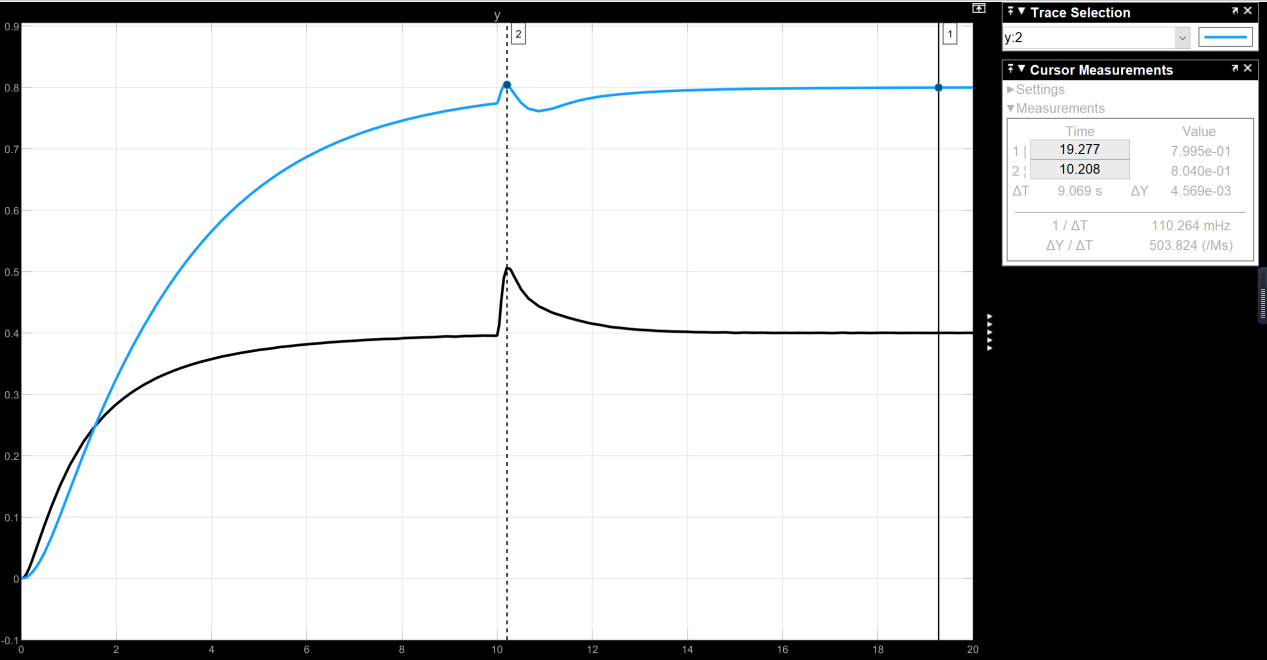


Figure 6.8 Output signal with observer

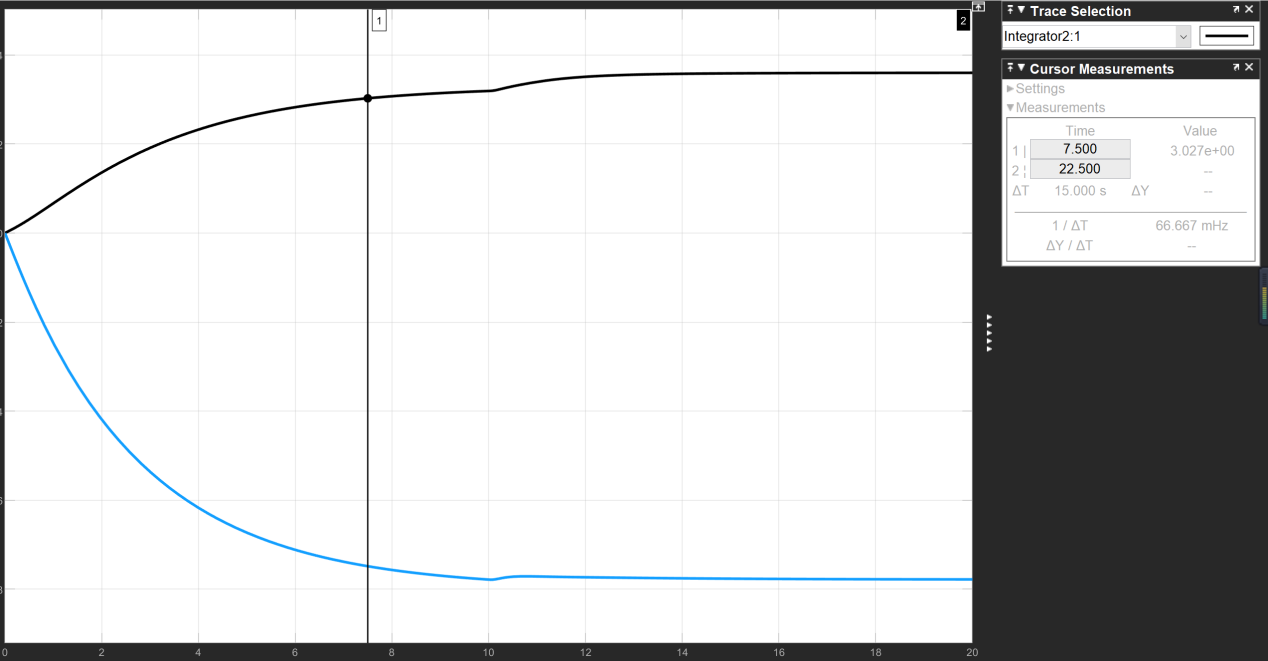


Figure 6.9 Controller signal with observer model

Compared with the system without observer, it is found that there is no difference between the output signal and the control signal.

1. Solution to Task 6
   1. Use two Input signals

Suppose the system



Take Laplace Transform on both sides, we get



Where 

From this formula, we can see that the final value of X(s) is related to U(s). With the original system unchanged, we enter a vector u with dimension 2×1, so we can only control two state variables. So I thought of a way to increase the input dimension,

Suppose  

The equation 

By solving this equation we get 

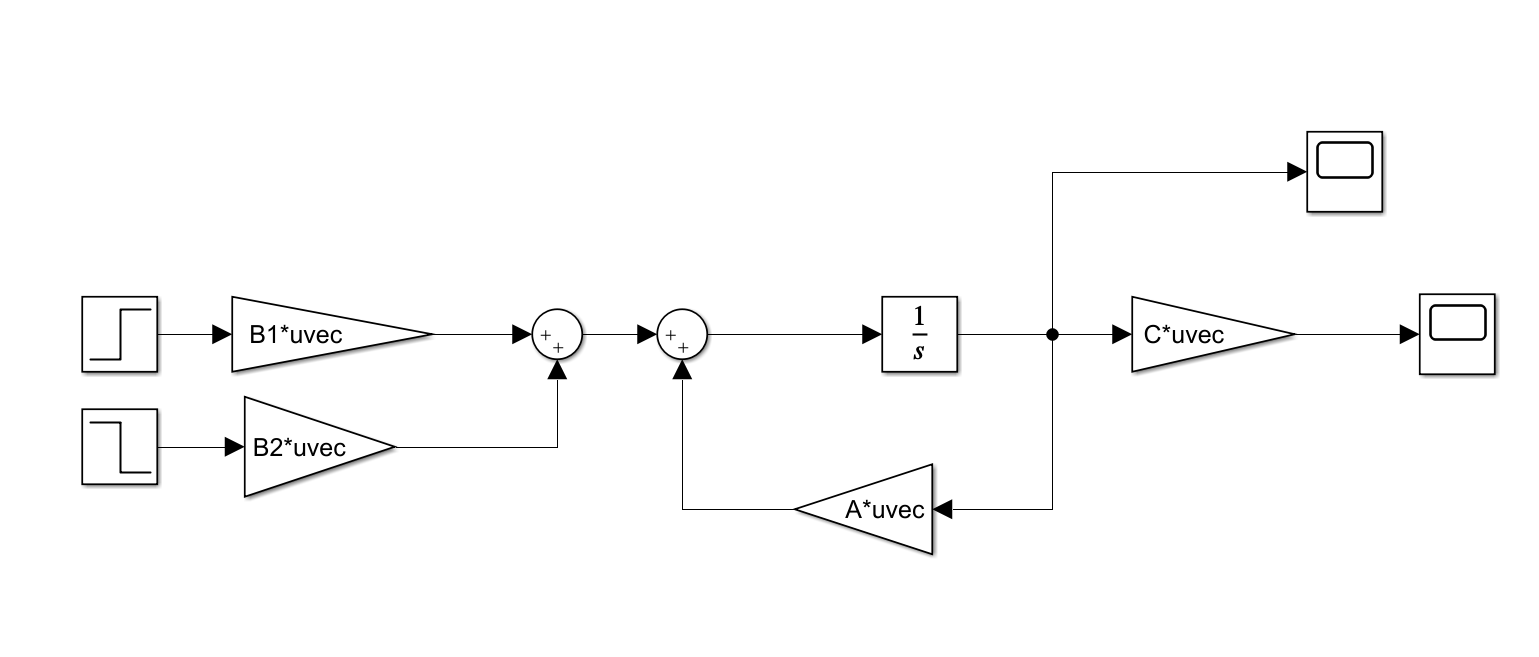


Figure 7.1 Dual input control model

Then check whether the method is correct through the status response:

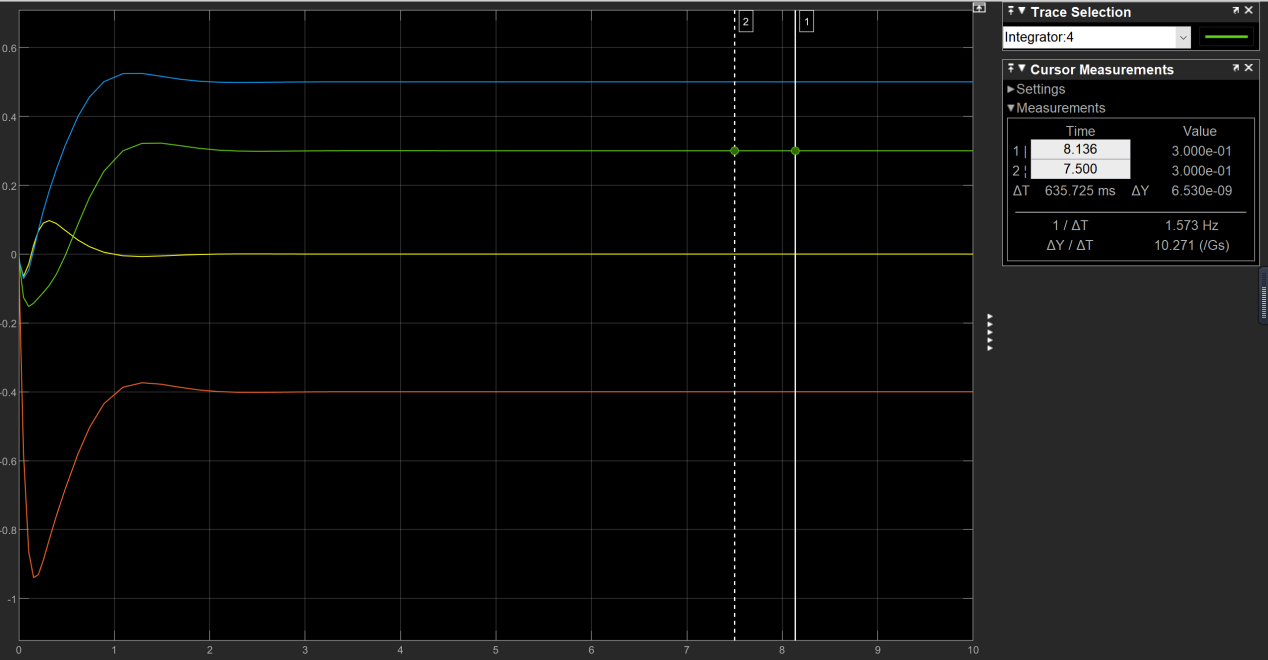


Figure 7.2 State response with dual input

The target is to maintain the states around a given set point = [0, 0.5, −0.4, 0.3]at steady state, and as shown in this figure we can find the target has been achieved.

* 1. Use feedback control

Similar to feedback control, we design a 

Suppose ,

We can get the equation Take Laplace Transform on both sides, we get

 where 

At the same time, we should make sure the system is stable, so (A-BK) should stable.

But there are 10 unknown numbers, and four equations and some inequalities, it’s complicated to solve the equation. Next, discuss how to minimize J when there are only two inputs.

7.3 Minimize the objective function

In this case , we only control one input with dimension 2×1，and minimize the following objective function:



The main idea is that make dJ/du = 0 by controlling u.

Based on task1, suppose the system:



For 22a, take Laplace Transform on both sides, we get

 where 

So we suppose the final  , which is relative to u=[u1;u2]

Objective function becomes to J(u), and solve the equation dJ/du = 0 we can get the u.

And use equation , we can calculate final state:

 with one input 

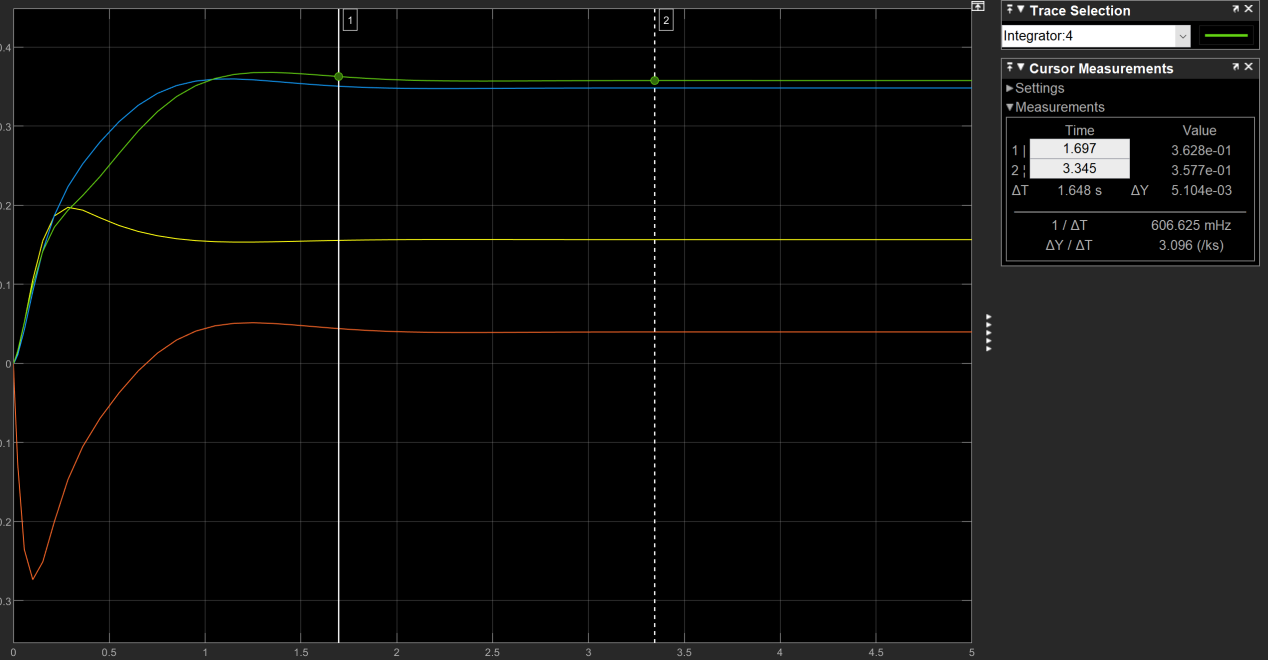


Figure 7.3 State response with one input

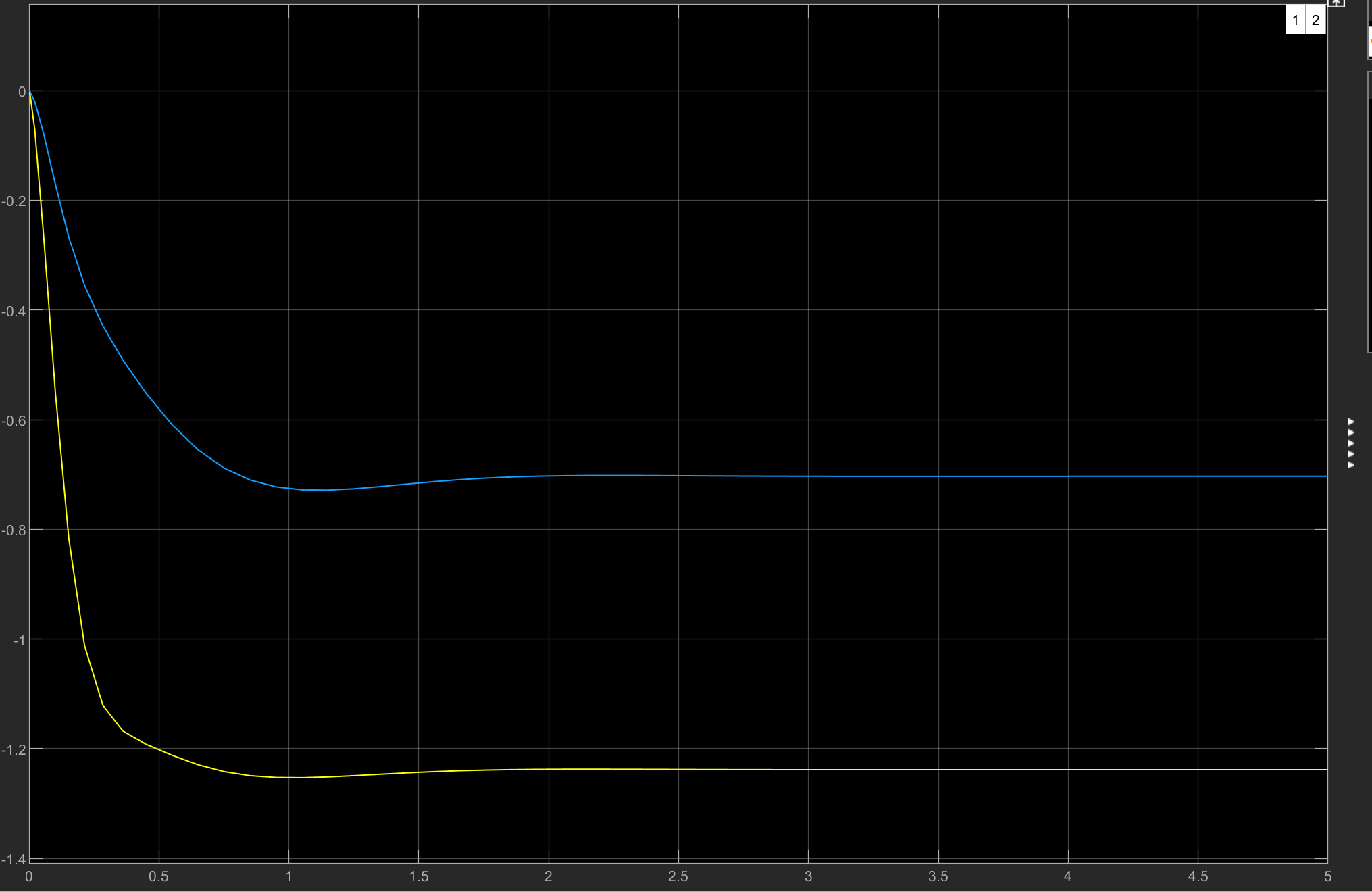


Figure 7.4 Output signal with one input

With 

1. Conclusion

For task 1, I use the pole placement method to configure a K for state feedback to stabilize the system and meet the overshoot and settling time requirements, and the placed poles will affect the characteristic of the system. For task 2, I use LQR to configure the K. The LQR method can change the cost and time of the system, experiment result shows that increasing the corresponding coefficient of Q can enhance the control ability of the state space, thereby reducing the overshoot and the settling time. For task 3, I compared the method of using LQR and pole placement method to construct the observer, and it concludes that the poles change of the observer will affect the convergence speed of the error. Because the C of this system is complicated, I only chose the full-order observer for observation. For task 4, after computing I found that the , which indicates I can only place two poles of this system. However there is a positive pole I can’t move, so this system is not internally stable but BIBO stable. For task 5, I design a servo controller so that the output could track to the set point, and add the step disturbance. For task 6, I use the final value theorem to design u to make the state variables finally stabilize at a set point.