Delft University of Technology

Control Engineering

SC42095

Assignment Report

Authors:
Chenghuai Lin (5703336)

January 8, 2023

1 Continuous-time Control

1.1 Plant Analysis

Before actually diving into controller design, we try to implement basic analysis to the plant. The transfer function of the system with the position of the extender robot arm as output is given as

$$G(s) = \frac{5}{s(5s+1)(0.5s+1)} = \frac{5}{2.5s^3 + 5.5s^2 + s}$$

and it turns out to be a third-order system with poles and no zeros. The poles-zeros plot, opened-loop bode graph plot, closed-loop step response without controller is shown in Figure 1

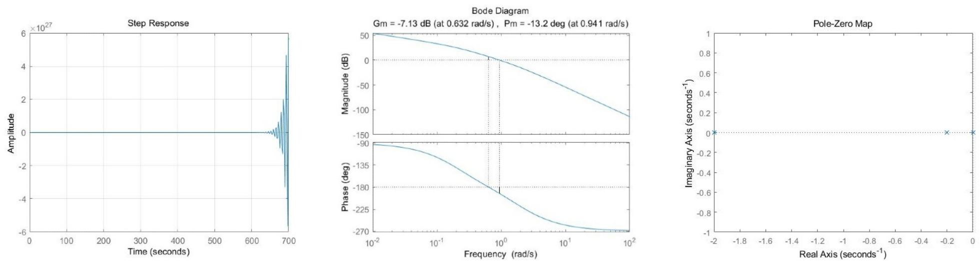

Figure 1: Plant Analysis: Open-loop Bode Plot(right) and Closed-loop Step Response without Controller(middle) and Pole-zeros Plot(right)

With the figures, several properties of the system can be distinguished.

1. The system is highly unstable due to negative phase margin. The first task should be to stabilize the system.
2. The crossover frequency of the system is around 0.632 rad/s, which could be improved.
3. The system has a generally has a bode plot with curve shapes similar to the first-order inertia term, which could be used for reference when shaping the loop function.
4. There are no zeros and 3 poles for this system. 2 poles are in the left half plane but there is one poles at the origin, which means a pure integral term.
5. The excess order of denominator is 3, which is greater than 2 and thus the system will not be minimum-phase after discretization due to the newly introduced zero outside the unit disk on the complex plane.

1.2 Question 1

1.2.1 Analysis

We are aiming to control the system in such a way that a reference set-point step has a fast response (i.e., minimal settling time in which the final output value is reached to at least 1% tolerance), and the response has an overshoot of maximum 5% with no steady-state error. Find the most suitable controller structure (from among P, PI, PD, PDD, PID, etc.) and tune it to achieve the following control objectives for a set-point step:

1. *minimal settling-time*
2. *overshoot $<$ 5%*
3. *steady-state error = 0*

The requirements could be interpreted in the frequency domain as

1. minimal settling-time
   To achieve small settling-time we need the system to achieve a high bandwidth, which means a quicker response. Also, we could improve the settling time by augmenting the gain magnitude in the lower frequency part.
2. overshoot $<$ 5%
   Reducing overshoot means increasing the phase margin.

3. steady-state error $= 0$
   We require the system to have integral term or higher system-type(i.e. higher order difference between denominator and numerator) to cancel the steady error.

Essentially, the goal of minimizing settling time and overshoot requires a higher phase margin and bandwidth. Eliminating steady-state error normally will introduce one or more integrator terms into the controller; these terms will also reduce phase margin, so their gain must be kept low. In creating our controller we focus on balancing these goals and limitations.

However, the given plant is actually a third-order system. For this reason, even if stabilized with controller like PID,PD,PI,PDD, the closed-loop system ($G_c(s) = \frac{GC}{1+GC}$) would still be at least a type-2 system, which means its steady error for step response would be zero. it turns out that the controller for step response doesn't necessarily need an integral term and PD controller is enough.

For PD controller with realizable D term, we have transfer function

$$G(s) = K_p + \frac{K_D s}{T_f s + 1} = K_p \frac{(T_f + \frac{K_p}{K_D})s + 1}{T_f s + 1} = K_p \frac{T_f + \frac{K_p}{K_D}}{T_f} \frac{T_f}{T_f + \frac{K_p}{K_D}} \frac{(T_f + \frac{K_p}{K_D})s + 1}{T_f s + 1} = K G_{lead}(s)$$

in which the right part $G_{lead}(s) = \frac{T_f}{T_f + \frac{K_p}{K_D}} \frac{(T_f + \frac{K_p}{K_D})s+1}{T_f s+1}$ could be taken as a standard phase-lead term and the left part $K = K_p \frac{T_f + \frac{K_p}{K_D}}{T_f}$ as a constant gain.

For standard phase-lead controller in the form of

$$G(j\omega) = (\frac{p}{z})\frac{z + j\omega}{p + j\omega} = \frac{1 + j\frac{\omega}{z}}{1 + j\frac{\omega}{p}}$$

some conclusions could be cited and used when tuning:

1. The phase-lead term generally has a bode plot with similar shape as Figure 2.
2. The central frequency of the lead-phase (where it improves the frequency the most)

$$\omega_c = \sqrt{pz} = \sqrt{\frac{1}{(\frac{K_P}{K_D} + T_f)T_f}} \tag{1}$$

3. The phase-lead term are generally used to improve the gain and phase stability margins and to increase the system bandwidth(but it also decreases the system response rise time)

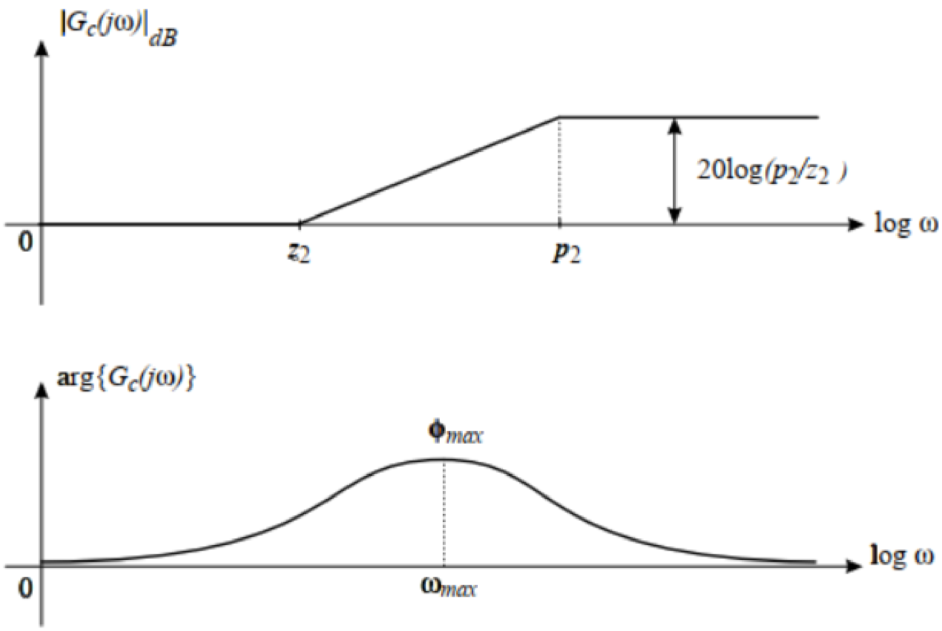

Figure 2: General Phase-lead Bode Plot

1.2.2 Tuning Process

The tuning process are shown as follows as milestones are captured when designing. The idea of loop shaping is used in the whole procedure. For Figure 3 to 9, from left to right the three plots in every figure are bode plot of controller, open-loop bode plot and closed-loop step response respectively.

To initialize, we first get an adequate result by tuning. In order to get an overshoot within 5%, a higher phase margin is needed, and thus a relatively high gain K_D for the derivative term was used as shown in Figure 3.

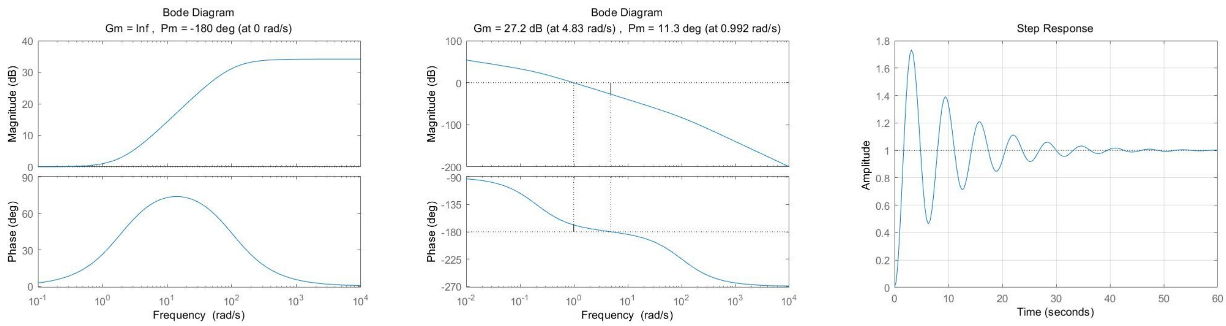

Figure 3: SettlingTime: 38.35s, Overshoot: 73.37%, $K_P = 1$, $K_D = 50$, $T_f = 100$

According to the equation 1 we can adjust the center frequency of the phase-lead term by changing the filtering length T_f as Figure 4 so as to obtain a higher phase margin by moving it towards the crossover frequency.

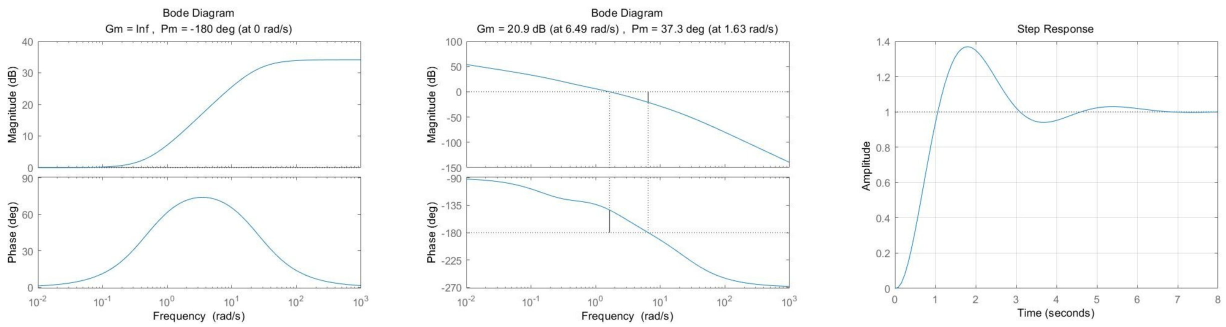

Figure 4: SettlingTime: 5.97s, Overshoot: 36.93%, $K_P = 1$, $K_D = 50$,$T_f = 25$

In Figure 4 there still remains a high overshoot, for which K_P is decreased to have a higher phase margin and the T_f is re-adjusted to move the central frequency of the phase-lead term as Figure 5.

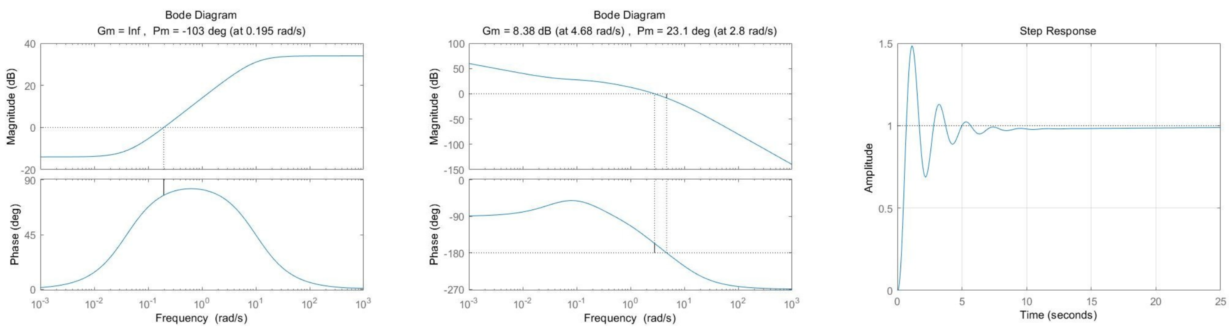

Figure 5: SettlingTime: 11.10s, Overshoot: 48.53%, $K_P = 0.2$,$K_D = 50$, $T_f = 10$

With following tuning, the overshoot is decreased with a lower K_D and the settling time is improved. This is interpreted as it moves the lower-frequency turning frequency $z = \frac{1}{T_f + \frac{K_P}{K_D}}$ of the phase-lead term to the right, meanwhile obtaining a higher gain as $K = K_P \frac{T_f + \frac{K_P}{K_D}}{T_f}$. For those two reason the crossover frequency would increase to have a smaller settling time. The thing is that the central frequency of the phase-lead is also moved towards right, which compensates the phase margin reduction that higher bandwidth should have brought.

Then the next step is to move the central frequency of the phase-lead term to around the crossover frequency to have a higher phase margin to further cancel overshoot as Figure 6 to 8.

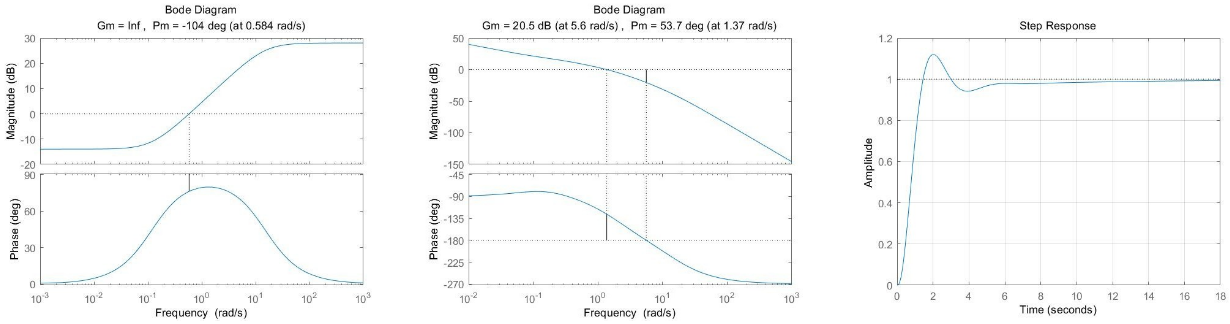

Figure 6: SettlingTime: 8.18s, Overshoot: 11.99%, $K_P = 0.2$, $K_D = 25$, $T_f = 15$

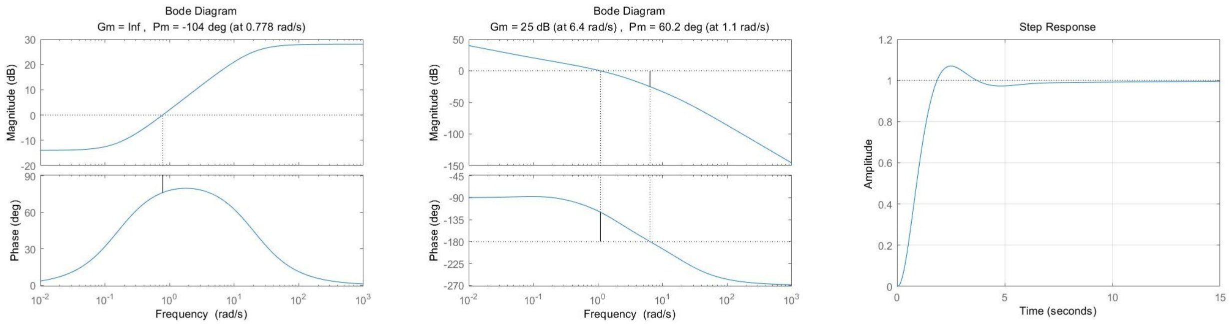

Figure 7: SettlingTime: 5.74s, Overshoot: 6.98%,$K_P = 0.2$, $K_D = 25$, $T_f = 20$

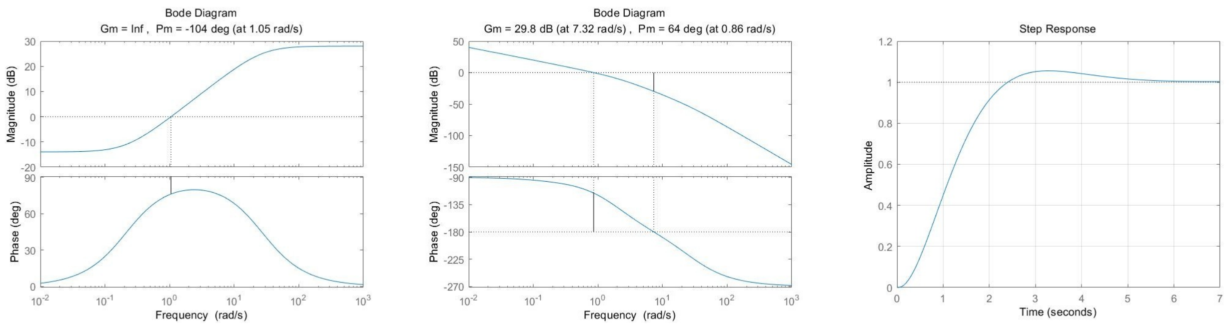

Figure 8: SettlingTime: 4.80s, Overshoot: 5.55%, $K_P = 0.2$, $K_D = 25$, $T_f = 27$

It turns out that $T_f = 27$ is the optimal value for the case where $K_P = 0.2, K_D = 25$ and the overshoot cannot be reduced further just by changing T_f so a marginal tuning to K_P is implemented and we have the final result in Figure 9.

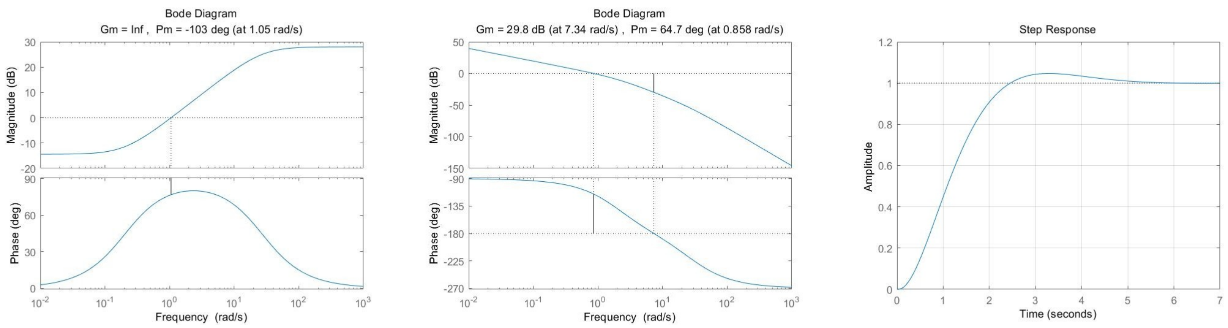

Figure 9: SettlingTime: 4.51s, Overshoot: 4.66%,$K_P = 0.19$,K_D= 25,$T_f = 27$

1.2.3 Results

As shown in Figure 9,the PD controller is eventually tuned to be

$$K_P = 0.19$$
$$K_D = 25$$
$$T_f = 27$$
$$C_{step}(s) = K_P + K_D \frac{s}{s + T_f} = \frac{25.19s + 5.13}{s + 27}$$

and it fits all the requirements given.

1.3 Question 2

1.3.1 Analysis

The control objective is to reduce and eliminate the effect of this disturbance as well as possible. This means:

1. *minimal amplitude of the system output y caused by disturbance q*
2. *minimal duration of the disturbance (i.e., time it takes to reduce to within 1% of its maximal value)*
3. *no offset caused by the disturbance*

The interpretation of the requirements in frequency domain are

1. minimal amplitude of the system output y caused by disturbance q
   We need the system to achieve a high bandwidth, which means a quicker response. It also requires the system to have a high phase margin.
2. minimal duration of the disturbance
   This is to require the system to have minimal settling time, which comes from high bandwidth.
3. no offset caused by the disturbance
   Typically, We require the system to have integral term or higher system-type(i.e. higher order difference between denominator and numerator) to cancel the steady error. Here specifically, even if the system type is high enough the steady output doesn't go to zero so actually it boils down to the requirement that the system should have a zero DC static gain.

For the disturbance rejection system, the transfer function from the disturbance to the output is calculated as follows

$$G_d(s) = \frac{G}{1 + GC}$$

in which $G(s)$ is the plant given and $C(s)$ is the controller we are going to design. This is the formula we are going to use in the matlab code for step disturbance response simulation.

However, it is observed that with PD controller closed-loop system would have a non-zero DC static gain, which leads to non-zero steady state error. Several tentative simulation as Figure 11.

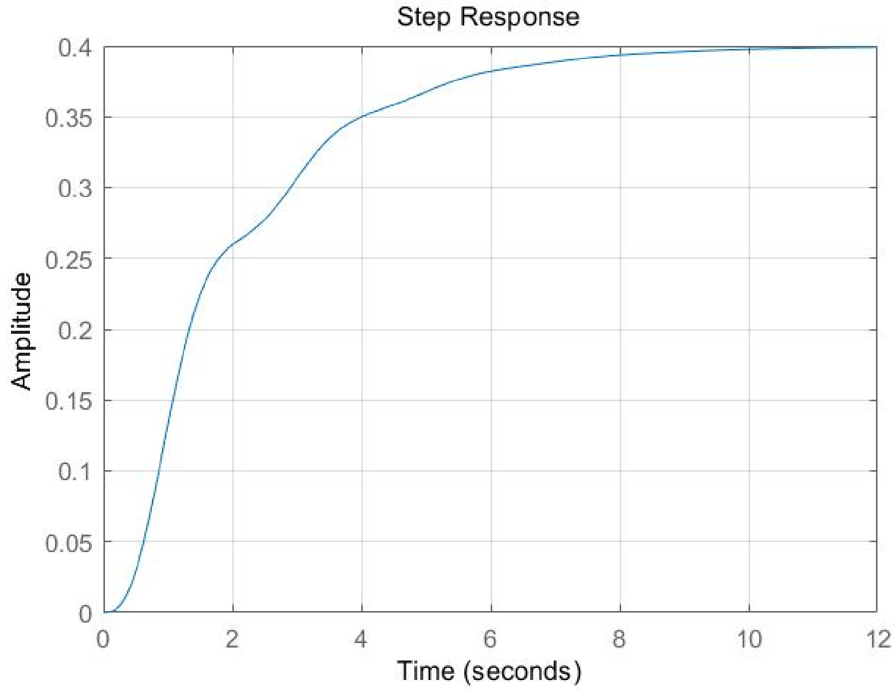

Figure 10: Disturbance Step Response with PD Controller, $K_P = 2.5$,K_D= 5,$T_f = 100$, showing a non-zero steady error, proving the necessity of newly introduced integral term

Therefore, an integral term is needed and the controller would become PID controller in the form as below

$$C_{dr}(s) = K_P + \frac{K_I}{s} + K_D \frac{T_f s}{s + T_f}$$

and it should be informed that the form used has a slightly difference on the numerator of the realistic D term with the typical PID transfer function discussed on class so as to agree with the transfer function given by the PID block in the simulink.

The whole tuning process is also going to be done with the loop shaping as general idea. One thing worth mentioning is the disturbance rejection tuning actually share the same open-loop function $G_O(s) = G(s)C(s)$ for loop-shaping as the step response tuning.

1.3.2 Tuning Process

The tuning process are shown as follows as milestones are captured when designing. The idea of loop shaping is used in the whole procedure. For Figure 11 to 16, from left to right the three plots in every figure are bode plot of controller, open-loop bode plot and closed-loop step response respectively.

In Figure 11, a tentative try is made by adding a K_I term into the previous controller.

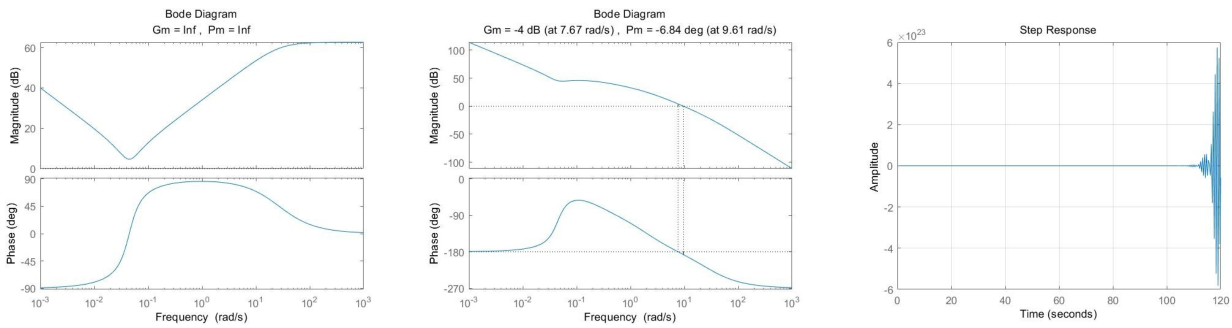

Figure 11: SettlingTime: NaN, PeakValue: Inf, $K_P = 1.7$,$K_I = 0.1$,$K_D = 50$, $T_f = 27$

The disturbance rejection response turns out to be unstable due to the negative phase margin. And then we try to move the phase augmentation peak to the right to increase the phase margin to make it stable as shown in Figure 12 to 13.

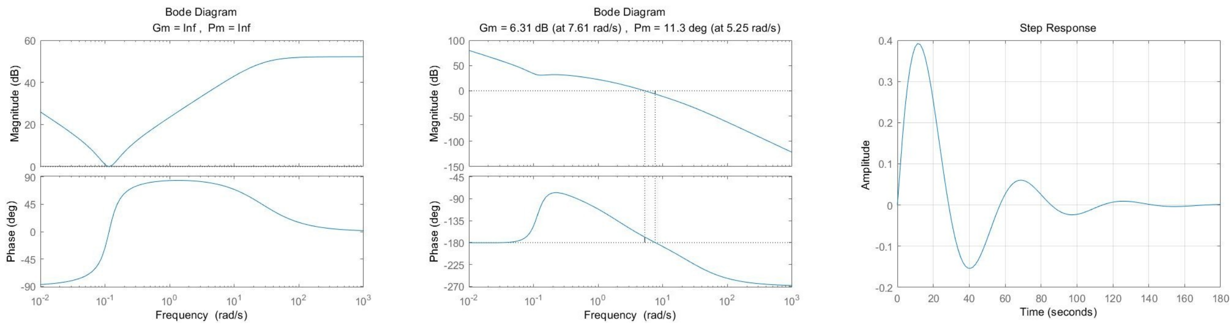

Figure 12: SettlingTime: 131.07s, PeakValue: 0.39, $K_P = 1$,$K_I = 0.2$,$K_D = 15$, $T_f = 27$

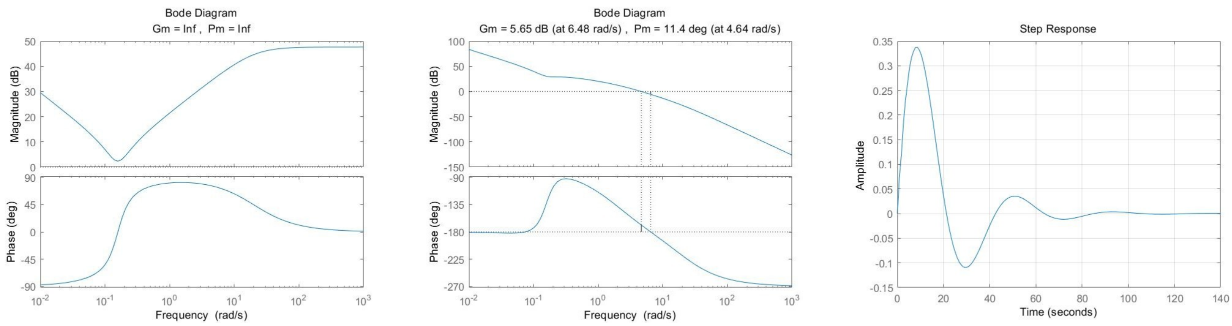

Figure 13: SettlingTime: 78.78s, PeakValue: 0.34, $K_P = 1.3$, $K_I = 0.3$, $K_D = 12$, $T_f = 20$

It is observed that increase of T_f moves the phase augmentation peak to the right greatly.

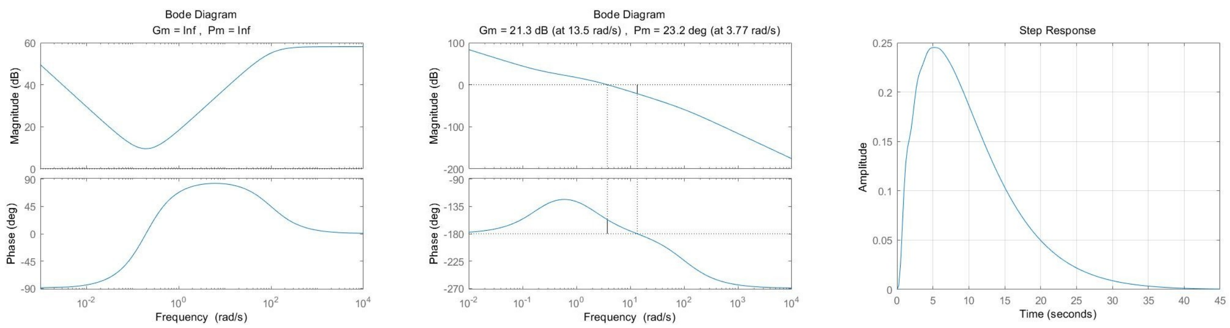

Figure 14: SettlingTime: 33.15s, PeakValue: 0.25, $K_P = 3$,$K_I = 0.3$,$K_D = 8$, $T_f = 100$

After Figure 14 just some minor adjustments are made as shown in Figure 15 and 16. Attempt is made to increase the K_P to have a higher bandwidth but it gives too high peak value in Figure 15. Then the K_P is reduced a little bit to have the final design, with an acceptable peak value.

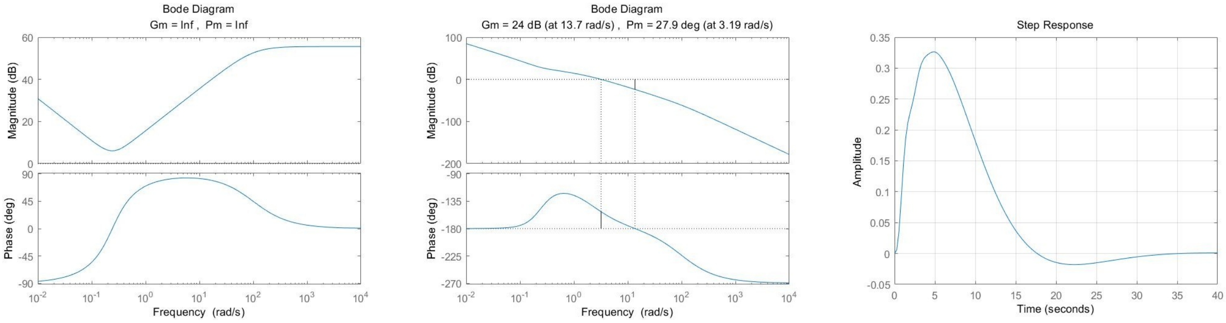

Figure 15: SettlingTime: 29.35s, PeakValue: 0.32, $K_P = 2$,$K_I = 0.35$,$K_D = 6$, $T_f = 100$

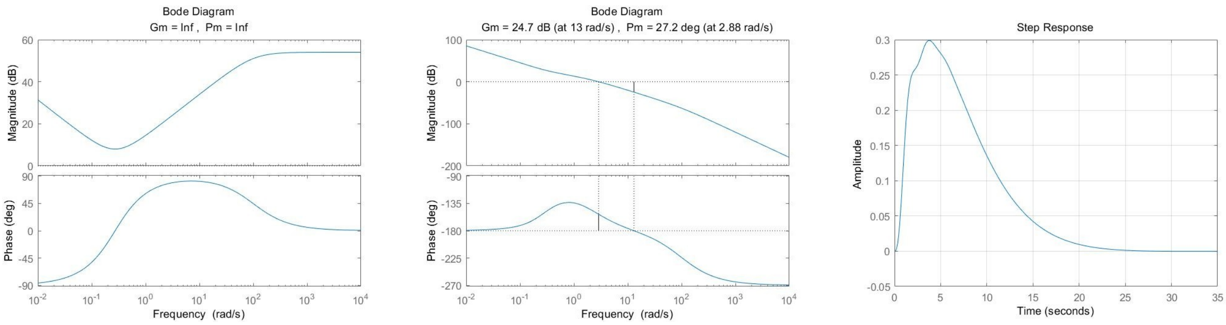

Figure 16: SettlingTime: 21.37s, PeakValue: 0.29, $K_P = 2.5$,$K_I = 0.37$,$K_D = 5$, $T_f = 100$

1.3.3 Results and Comparison

As shown in the Figure 16,the final PID controller for disturbance rejection is tuned to be

$$\begin{aligned} K_P &= 0.2 \\ K_I &= 0.1 \\ K_D &= 25 \\ T_f &= 27 \end{aligned}$$

$$C_{dr}(s) = K_P + \frac{K_I}{s} + K_D \frac{T_f s}{s + T_f} = \frac{502.5s^2 + 250.4s + 37}{s^2 + 100s}$$

2 Discrete-time Control

2.1 Question 3

Using the matlab function *tf2ss* we can have the state-space model of the given plant as

$$A = \begin{bmatrix} -2.2 & -0.4 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$
$$B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$
$$C = \begin{bmatrix} 0 & 0 & 2 \end{bmatrix}$$
$$D = 0$$

The next step is to choose the sampling time to discretize the system. The first attempt is try to use the rules of thumb regarding the rise time of the step response but it turns out to be impossible since the system is unstable itself thus the step response can't converge.

Then next try is to use the fastest natural frequency of the poles. The fastest poles inside the system is $s = -2$ with natural frequency $\omega_n = 2(rad/s)$ and based on the rules

$$\omega_n h \approx 0.2 - 0.6$$

I used $\omega_n h = 0.3 \implies h = \frac{0.3}{\omega_n} = 0.15$ and get sampling time h chosen to be $0.15s$ (i.e. $h = 0.1s$). And it also should be much greater than 10 times of the crossover frequency of the plant, which is $\omega_c = 0.632rad/s$, and we have

$$10 * \omega_c = 6.32(rad/s) = 1.0058592415(Hz) \approx 1(Hz) < 1/h \approx 6.67(Hz)$$

Therefore, with matlab function *c2d*, the discrete-time state-space model could be given as

$$A = \begin{bmatrix} 0.7153 & -0.05103 & 0 \\ 0.1276 & 0.996 & 0 \\ 0.0101 & 0.1498 & 1 \end{bmatrix}$$
$$B = \begin{bmatrix} 0.1276 \\ 0.0101 \\ 0.0005188 \end{bmatrix}$$
$$C = \begin{bmatrix} 0 & 0 & 2 \end{bmatrix}$$
$$D = 0$$

In the state-space model, according to the specific scene described in the questions (considering the output is said to be the position and the structure of the matrix C),the elements in the states vector $x = \begin{bmatrix} x_1, x_2, x_3 \end{bmatrix}^T$ are interpreted as acceleration, speed, position of the extender robot arm respectively for physical meaning.

2.2 Question 4

The sampling time for the controller is chosen so as to have enough numbers of samples within the transient states.

2.2.1 Step Controller Discretization

For the step controller

$$C_{step}(s) = K_P + K_D \frac{s}{s + T_f} = \frac{25.19s + 5.13}{s + 27}$$

we have the rise time shown in figure 9 as $T_r = 1.5673s$ and the sampling time will be chosen as

$$N = 10$$
$$T_s = \frac{T_r}{N} = 0.15673s$$

With the chosen sampling time T_s we discretize the plant and the step response controller respectively and then connect them to form a closed loop for step response simulation. For the plant we used zero-order hold method for discretization. For the controller we used Tustin method for discretization, which is

$$s = \frac{2}{T_s}\frac{1 - z^{-1}}{1 + z^{-1}}$$

and we have

$$C_{step}(z) = \frac{8.214z - 7.956}{z + 0.3581}$$

The result for step response is given as below in Figure 17

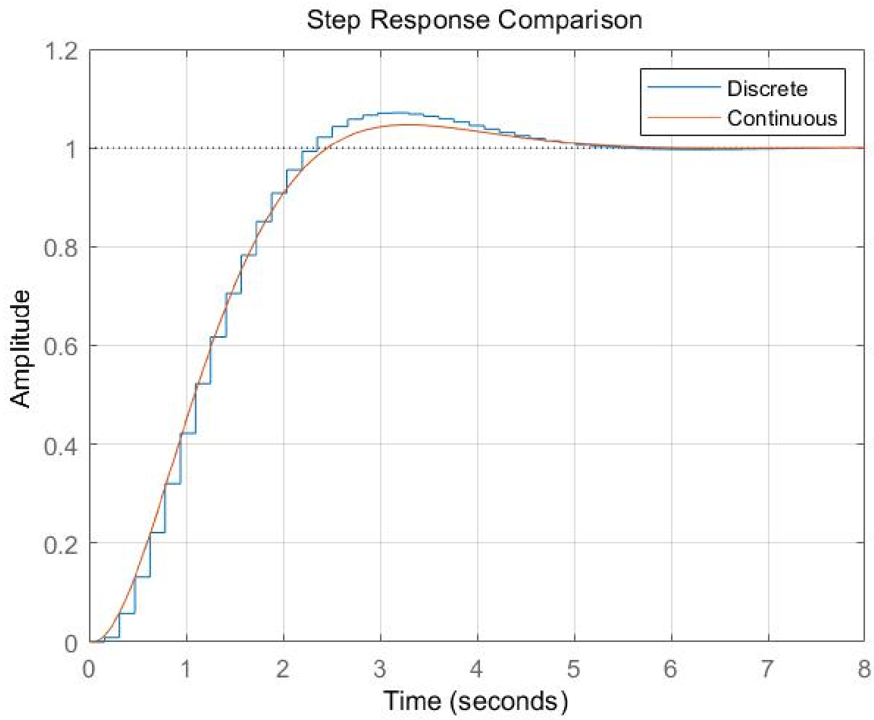

Figure 17: Step Response Comparison

As shown in Figure 17 it could be concluded that the step response with and without discretization generally agrees with slight difference, which is acceptable.

2.2.2 Disturbance Rejection Controller Discretization

For disturbance rejection controller,

$$C_{dr}(s) = K_P + \frac{K_I}{s} + K_D\frac{T_f s}{s + T_f} = \frac{502.5s^2 + 250.4s + 37}{s^2 + 100s}$$

the sampling time choosing method regarding the rise time is kind of blur since the response curve doesn't explicitly contains a rise time as the steady output is zero in the end. But the general idea that there should be enough samples within the transient states of the system still applies. Therefore we read the timmestamp T_r when the response output reaches 90% of its peak output (see Figure 18)and divide it by factor N to get the sampling time T_s. The factor is tuned so that the discrete-time system and discrete-time controller give a properly sampled response, which is 17.

$$T_r = 3.17$$
$$N = 17$$
$$T_s = \frac{T_r}{N} = 0.1865$$

For the plant we used zero-order hold method for discretization. For the controller we used Tustin method for discretization, which is

$$s = \frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}}$$

and we have

$$C_{dr}(z) = \frac{50.97z^2 - 97.29z + 46.45}{z^2 - 0.1937z - 0.8063}$$

In the following part, by default, for all the controller discretization we used Tustin method and zero-order hold for all the plant discretizaiton.

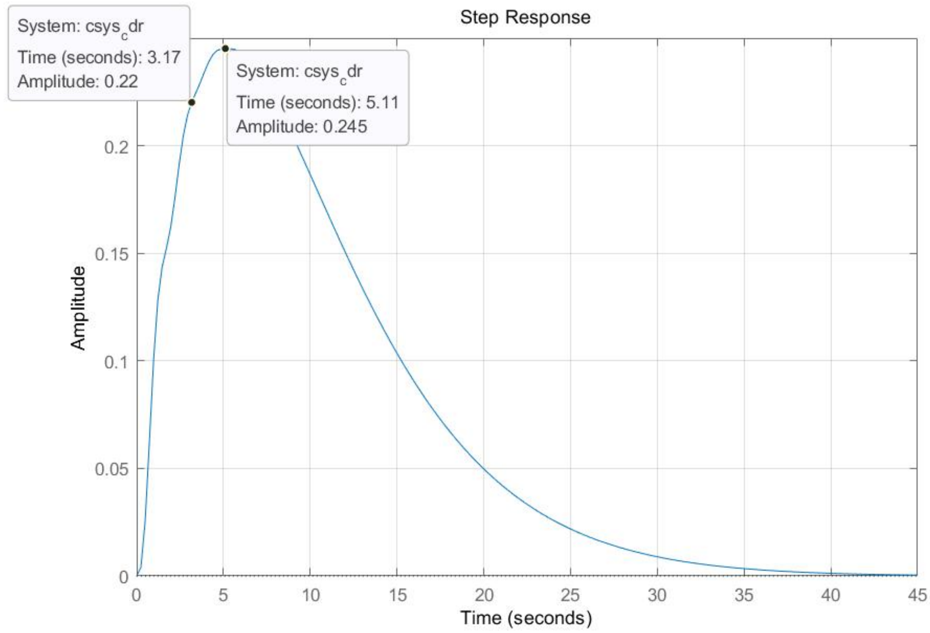

Figure 18: Disturbance Rejection Step Response with Timestamp to Read the "Rise Time"

With the chosen sampling time T_s we discretize the plant and the step response controller respectively and then connect them to form a closed loop for disturbance step response simulation. The result for disturbance rejection is given as below in Figure 19

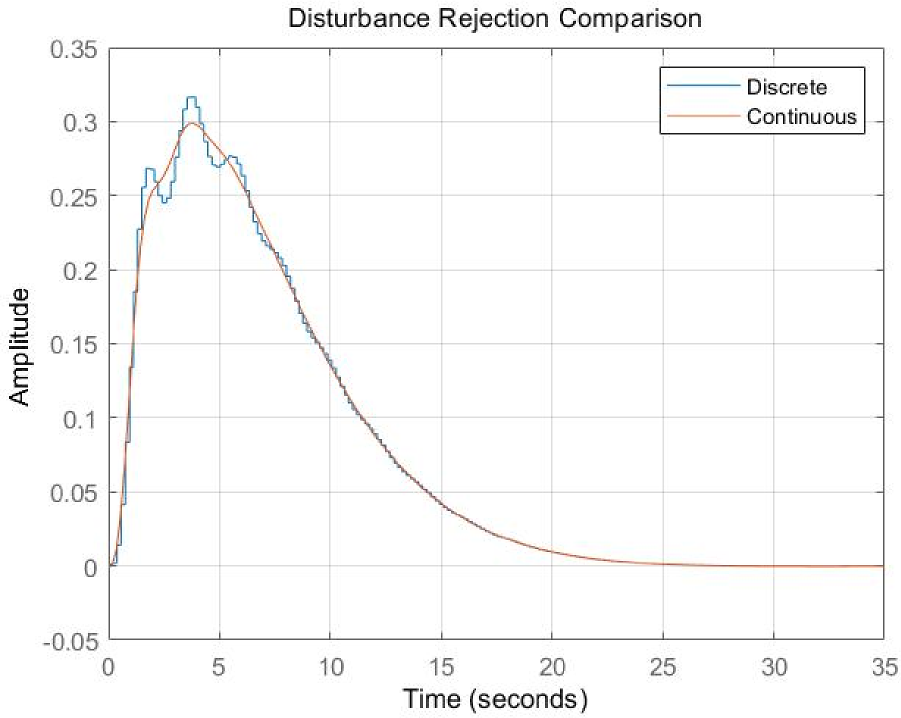

Figure 19: Disturbance Rejection Comparison

As shown in Figure 19 it could be concluded that the disturbance rejection response with and without discretization generally agrees with slight difference, which is acceptable.

2.3 Question 5

2.3.1 Analysis about Placing the Poles

The given system is a third-order system and it has 3 poles. The general idea for poles placement is to firstly settle the dominant poles pair(2 poles) like in a second-order system of the form $\frac{1}{s^2+2\zeta\omega s+\omega^2}$ such that the system property are mainly determined by the dominant poles pair to ensure the ideal performance.

For that, we need to make sure the poles are properly placed (not too far from the original poles to prevent controller gains becoming too big meanwhile making dominant influence to determine the system properties). After that we place the 1 more pole to be faster than the dominant poles pair (but also not too far from its original position).

To ensure that the system still meet the requirements in the continuous part, properties about the dominant have to be calculated. As the poles pair is placed in a form of second-order system, the overshoot for step response could be dropped down analytically

$$OS = \exp\left\{\frac{\zeta\pi}{\sqrt{1-\zeta^2}}\right\} \tag{2}$$

To ensure the overshoot smaller than 5%, the overshoot solving aim is selected as 4.5% and the damping ratio will be given by solving Equation 2, leading to

$$\zeta = \frac{-\ln OS}{\sqrt{\pi^2 + \ln^2 OS}} = 0.7025$$

The position of poles is a complex number determined by its modulus r and argument θ. With the geometric explanation of damping ratio we get to know that

$$\zeta = \cos(\pi - \theta) \implies \theta = \pi - \arccos\zeta = 134.6283°$$

and the only thing left to determine the dominant poles pair is to determine its modulus. However, there are still some limitations (some of them are actually already mentioned before) when deciding the modulus:

- Don't move poles too far from the original poles to prevent controller gains becoming too big meanwhile making dominant influence to determine the system properties
- The modulus of the poles pair cannot be greater than that of the one left pole since it actually represents the responding speed of the poles.

The plant possesses three poles located at

$$p_1 = 0$$
$$p_2 = -0.2$$
$$p_3 = -2$$

from which it could be told that the poles p_1 and p_2 are much closer than to p_3 and of smaller modulus and thus both slower than p_3. It's natural to think of pairing them up when designing to place the poles manually. And for the last left pole, we would try to put it on the negative real axis so that it would simply be the pole of a serial first-order system with the its modulus also representing the responding speed.

In addition, to reduce the resulting controller gain K, it's also a natural idea to maintain the original position of p_3.

2.3.2 Design Procedure

The general designing procedure is summarized as follows. Note the system is controllable.

1. Decide the position of dominant poles pair by choosing its modulus meanwhile choosing the modulus for the other pole on the negative real axis. The way we place the poles is already discussed in the previous section.
2. Choose the sampling time h based on the natural frequency of the fastest pole ω_f (it should be pole on the negative real axis) with the following rules

   $$\omega_f h = 0.3$$

   in which the 0.3 is chosen by tuning. For each new pole placement set, the plant has to be discretized with the new resulting sampling time h.
3. Use the mapping relationship between poles to get the poles location in discrete-time poles on the complex plane.

   $$z = e^{sh}$$

   in which z is the discrete-time pole and s is the continuous-time pole.
4. Discretize the system with the obtained sampling time h. It should be informed that the system needs to be sampled for every design.
5. Use the matlab command *place* to compute the feedback gain matrix K.
6. To construct a servo (tracking) controller, formulate a new state-space model $(\hat{\Phi}, \hat{\Gamma}, \hat{C}, \hat{D})$ as

   $$\hat{\Phi} = \Phi - \Gamma$$
   $$\hat{\Gamma} = \Gamma$$
   $$\hat{C} = C$$
   $$\hat{D} = D$$

   in which A, B, C, D are the matrices from the original plant.
7. Compute the static gain K_{DC} of the new system $(\hat{\Phi}, \hat{\Gamma}, \hat{C}, \hat{D})$ (of discrete transfer function $H(z)$) with formula (or matlab command *dcgain*)

   $$K_{DC} = \lim_{z \to 1} (1 - z^{-1}) H(z)$$

after which we add a gain for input to cancel the static gain K_{DC} by adding $L_C = \frac{1}{K_{DC}}$ to formulate the final new state-space model $(\overline{\Phi}, \overline{\Gamma}, \overline{C}, \overline{D})$

$$\overline{\Phi} = \Phi - \Gamma K$$
$$\overline{\Gamma} = \Gamma L_C$$
$$\overline{C} = C$$
$$\overline{D} = D L_C$$

and that is the final designed system we get.

8. Evaluate the system $(\overline{\Phi}, \overline{\Gamma}, \overline{C}, \overline{D})$ with step response using matlab.

2.3.3 Results with Different Poles

Four sets of poles placement are implemented to investigate the difference. The step response of the four designed systems are shown in Figure 20 and the parameters are shown in the Table 3.

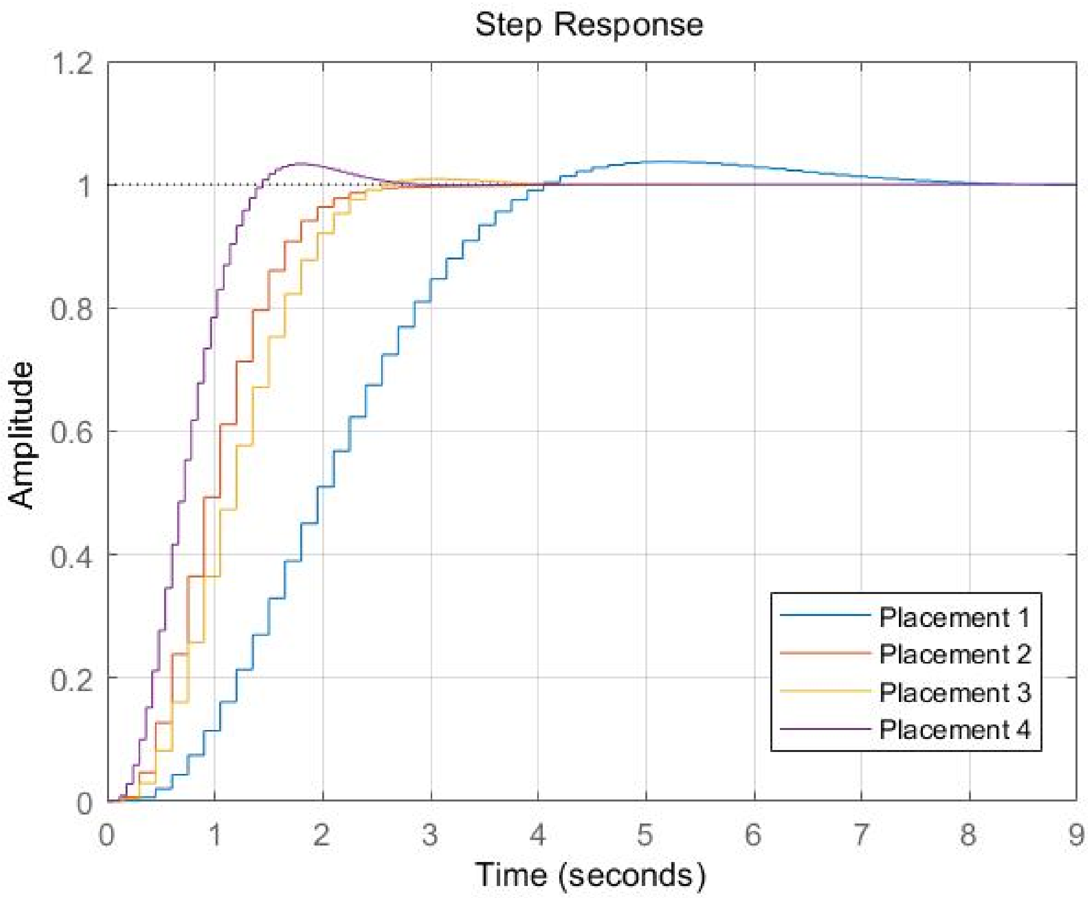

Figure 20: Step Tracking Response Comparison with Different Poles Placement

In Placement 1, the pole pair (p_1, p_2) has modulus of 1, which is smaller than the unchanged third pole $p_3 = -2$, responding slower and hence playing a major influence on the system property. Since the dominant pole pair in Placement 1 has the smallest modulus in all the design, it can be also observed in Figure 20 the response of it is also the poorest with the longest settling time, longest rise time and greatest overshoot.

In Placement 2, the pole pair (p_1, p_2) has modulus of 3, which means the natural frequency of it is greater than that of unchanged third pole $p_3 = -2$. Specifically, when comparing the responding speed, due to the existence of damping ratio, we need to compare the oscillation frequency $\omega_p = \zeta\omega_o$ (ω_o is the natural frequency

Placement	Pole Pair Modulus	Poles	Overshoot	Settling Time	Rise Time	Feedback Gain
1	$r = 1$	-0.7025 + 0.7117i -0.7025 - 0.7117i -2.0000 + 0.0000i	3.7038%	6.60s	2.40s	$K^T = \begin{bmatrix} 1.1541 \\ 3.2217 \\ 1.8271 \end{bmatrix}$
2	$r = 3$	-2.1075 + 2.1350i -2.1075 - 2.1350i -2.0000 + 0.0000i	0%	2.25s	1.20s	$K^T = \begin{bmatrix} 3.4680 \\ 13.5944 \\ 13.3167 \end{bmatrix}$
3	$r = 2/\zeta - 0.3 = 2.4470$	-1.7190 + 1.7414i -1.7190 - 1.7414i -2.0000 + 0.0000i	0.4631%	2.25s	1.20s	$K^T = \begin{bmatrix} 2.8770 \\ 10.4499 \\ 9.3916 \end{bmatrix}$
4	$r = 3$	-2.1075 + 2.1350i -2.1075 - 2.1350i -5.0000 + 0.0000i	3.3205%	2.22s	0.84s	$K^T = \begin{bmatrix} 6.0762 \\ 25.2174 \\ 36.5733 \end{bmatrix}$

Table 1: Poles Placement Parameters

of the pole, i.e. modulus of the pole) of the pole pair (p_1, p_2) in second-order system with the natural frequency of the pole p_3 in the first-order system. Here we have

$$\omega_p = \zeta\omega_o = 0.7025 * 3 = 2.1075 > 2 = \omega_{p_3}$$

Therefore, the pole pair has a quicker response than $p_3 = -2$. For that reason, p_3 becomes the dominant pole of the system and it is noticed that the response here actually doesn't have any overshoot and appears to be much more similar to response of a first-order system (e.g. $\frac{1}{Ts+1}$) rather than a typical second-order system.

In Placement 3, the pole pair (p_1, p_2) has modulus $r = 2/\zeta - 0.3 = 2.4470$, which is tuned to be a little smaller than the critical frequency at which the pole oscillation frequency of the second-order system is the same as the natural frequency of the first-order system. However, it should be informed that modulus r still remains greater than 2, which is the natural frequency of p_3. And the simulation result turns out to have overshoot, which cannot be seen in the first-order system, meaning here the pole pair (p_1, p_2) already begins to play stronger role. It also shows the oscillation frequency ω_p and damping ratio should be taken into consideration when evaluating the speed of the pole.

In Placement 4, the pole pair (p_1, p_2) and p_3 are both moved to the left and it brings the fastest response in the all four designs. This set of poles is just designed to show we could simply get a quicker response by moving the poles to the left.

In addition, it could be observed that the further we move the poles, the bigger the controller gain is.

2.3.4 Final Design

Since there are no limitations on the input or controller output, The performance of the system can be improved by placing the poles arbitrarily far on the left half plane. Here we just relatively casually realize a better result than any design in the last section just simply by moving the designed poles to the left.

The pole pair modulus is chosen to be 7 and p_3 is moved to -10 then the designed poles become

$$\begin{aligned} p_1 &= -4.9175 + 4.9818i \\ p_2 &= -4.9175 - 4.9818i \\ p_3 &= -10.0000 + 0.0000i \end{aligned}$$

and the step response of the system is shown in Figure 21. It's obvious that the response shows the system is better than any design in the previous section.

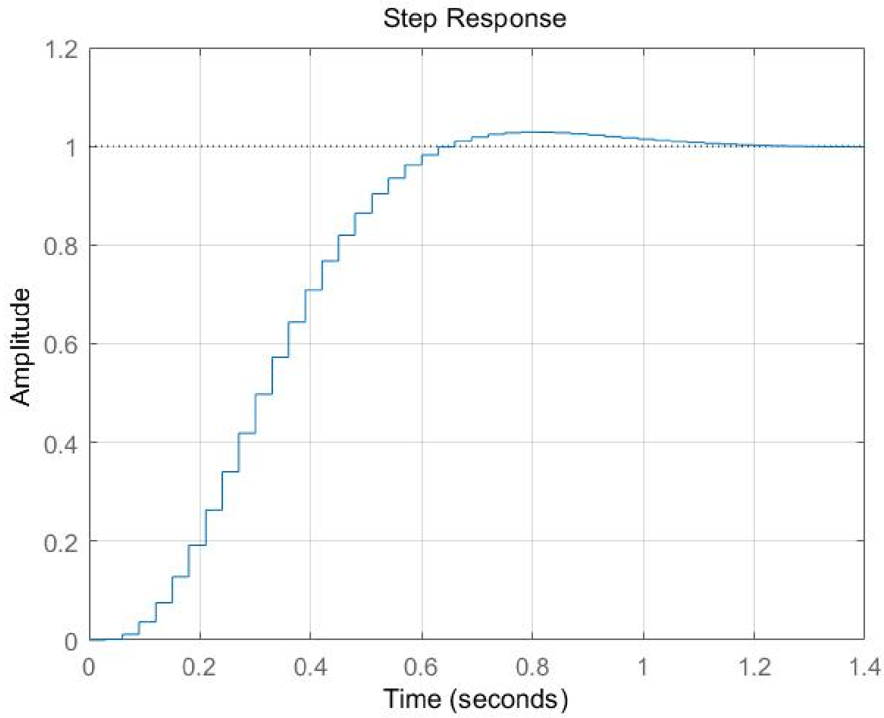

Figure 21: Step Response with State-feedback Controller of Designed Pole-Placement.
RiseTime: 0.36s, SettlingTime:0.96s, Overshoot:2.8967%

2.4 Question 6 - Observer

2.4.1 Observer-based State-feedback System

Assuming the discretized state-space model to be

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned}$$

in which Φ is 3x3 and Γ is 3x1, With the observer and the required scene of servo problem ,we have $u = -K\hat{x} + K_r r$ and the state-space model is changed to incorporate the observer

$$\begin{aligned} \hat{x}(k+1) &= \Phi\hat{x}(k) + \Gamma u(k) + L(y(k) - \hat{y}(k)) \\ &= \Phi\hat{x}(k) + \Gamma u(k) + L(Cx(k) - C\hat{x}(k)) \\ &= (\Phi - LC)\hat{x}(k) + \Gamma u(k) + LCx(k) \end{aligned}$$

and it turns out to be

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ \hat{x}(k+1) \end{bmatrix} &= \begin{bmatrix} \Phi & -\Gamma K \\ LC & \Phi - LC - \Gamma K \end{bmatrix} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix} + \begin{bmatrix} \Gamma K_r \\ \Gamma K_r \end{bmatrix} r(k) \\ &= \hat{\Phi} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix} + \hat{\Gamma} r(k) \\ y(k) &= Cx(k) - DK\hat{x} + DK_r r(k) \\ &= \begin{bmatrix} C & -DK \end{bmatrix} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix} + DK_r r(k) \\ &= \hat{C} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix} + \hat{D} r(k) \end{aligned}$$

in which K is the designed feedback gain obtained in the last question, K_r is the input gain to ensure the static gain from r to y to be 1 and with $\tilde{x} = x - \hat{x}$ we could also have

$$\begin{aligned}
\begin{bmatrix} x(k+1) \\ \tilde{x}(k+1) \end{bmatrix} &= \begin{bmatrix} \Phi - \Gamma K & \Gamma K \\ 0 & \Phi - LC \end{bmatrix} \begin{bmatrix} x(k) \\ \tilde{x}(k) \end{bmatrix} + \begin{bmatrix} K_r \Gamma \\ 0 \end{bmatrix} r(k) \\
&= \tilde{A} \begin{bmatrix} x(k) \\ \tilde{x}(k) \end{bmatrix} + \tilde{B} r(k) \\
y(k) &= Cx(k) + Du(k) \\
&= Cx(k) - DK\hat{x} + DK_r r(k) \\
&= Cx(k) - DK(x(k) - \tilde{x}(k)) + DK_r r(k) \\
&= (C - DK)x(k) + DK\tilde{x}(k) + DK_r r(k) \\
&= \begin{bmatrix} C - DK \\ DK \end{bmatrix} \begin{bmatrix} x(k) \\ \tilde{x}(k) \end{bmatrix} + DK_r r(k) \\
&= \tilde{C} \begin{bmatrix} x(k) \\ \tilde{x}(k) \end{bmatrix} + \tilde{D} r(k) \\
e(k) &= y(k) - \tilde{y}(k) = C\tilde{x}(k)
\end{aligned}$$

and this shows the dynamic of state estimation error is determined by the poles of subsystem $\Phi - LC$. As long as the poles of that is chosen properly, the estimation error will asymptotically goes to zero.

2.4.2 Poles Placement Analysis

The order of the observer is the same as the plant, which is 3. Therefore, we have 3 poles to place for the observer.

The general idea for choosing the observer poles is that they should be responding faster than the state-feedback poles to secure the accuracy of the state estimation in the sake of the quality of state-feedback. Besides, it is also required to have less overshoot and more stability (like less oscillation).

Here we basically just repeat the same design idea as the state-feedback pole design in the last question. That is, we place the first two poles as a dominant pole pair and the third pole as faster pole on the negative real axis, which represents a first-order system. We also require the response to have a overshoot less than 5% (because 5% is actually a good demand for any observer, neither too harsh nor too poor)and eventually comes up with the same damping ratio for the dominant pole pair. The main difference between the observer pole pair and the state-feedback pole pair is the modulus and that of the former is much bigger to secure a quicker response to the state changes than the state-feedback closed loop.

2.4.3 Design Procedure

The general procedure is generally the same as the state-feedback controller design except for the new-added first several steps in the front to design the observer. The general designing procedure is summarized as follows. Note that the system is observable.

1. For observer and state-feedback control respectively, decide the position of dominant poles pair by choosing its modulus meanwhile choosing the modulus for the other pole on the negative real axis.

2. Choose the sampling time h based on the natural frequency of the fastest pole ω_f of observer(it should be the third pole of the observer since the observer responds faster and that pole responds the fastest) with the following rules

$$\omega_f h = 0.3$$

   in which the 0.3 is chosen by tuning. For each new pole placement set, the plant has to be discretized with the new resulting sampling time h.

3. Use the mapping relationship between poles to get the poles location in discrete-time poles on the complex plane.

$$z = e^{sh}$$

   in which z is the discrete-time pole and s is the continuous-time pole.

4. Discretize the system with the obtained sampling time h. It should be informed that the system needs to be sampled for every design.
5. Use the matlab command *place* to compute the observer gain L and the feedback gain matrix K respectively.
6. Formulate a new state-space model $(\tilde{\Phi}, \tilde{\Gamma}, \tilde{C}, \tilde{D})$ as

$$
\begin{aligned}
\begin{bmatrix} x(k+1) \\ \hat{x}(k+1) \end{bmatrix} &= \begin{bmatrix} \Phi & -\Gamma K \\ LC & \Phi - LC - \Gamma K \end{bmatrix} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix} + \begin{bmatrix} \Gamma \\ \Gamma \end{bmatrix} r(k) \\
&= \tilde{\Phi} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix} + \tilde{\Gamma} r(k) \\
y(k) &= Cx(k) - DK\hat{x} + Dr(k) \\
&= \begin{bmatrix} C & -DK \end{bmatrix} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix} + Dr(k) \\
&= \tilde{C} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix} + \tilde{D} r(k)
\end{aligned}
$$

   in which A, B, C, D are the matrices from the original plant.
7. Compute the static gain K_{DC} of the new system $(\tilde{\Phi}, \tilde{\Gamma}, \tilde{C}, \tilde{D})$ (of discrete transfer function $H(z)$) with formula (or matlab command *dcgain*)

$$K_{DC} = \lim_{z \to 1} (1 - z^{-1}) H(z)$$

   after which we add a gain for input to cancel the static gain K_{DC} by adding $K_r = \frac{1}{K_{DC}}$ to formulate the final new state-space model $(\hat{\Phi}, \hat{\Gamma}, \hat{C}, \hat{D})$

$$
\begin{aligned}
\begin{bmatrix} x(k+1) \\ \hat{x}(k+1) \end{bmatrix} &= \begin{bmatrix} \Phi & -\Gamma K \\ LC & \Phi - LC - \Gamma K \end{bmatrix} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix} + \begin{bmatrix} \Gamma K_r \\ \Gamma K_r \end{bmatrix} r(k) \\
&= \hat{\Phi} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix} + \hat{\Gamma} r(k) \\
y(k) &= Cx(k) - DK\hat{x} + DK_r r(k) \\
&= \begin{bmatrix} C & -DK \end{bmatrix} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix} + DK_r r(k) \\
&= \hat{C} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix} + \hat{D} r(k)
\end{aligned}
$$

   and that is the final designed system we get.
8. Evaluate the system $(\hat{\Phi}, \hat{\Gamma}, \hat{C}, \hat{D})$ with step response using matlab.

2.4.4 Result

For each design, the sampling time is decided by the fastest pole in the designed observer.

In the simulation, we keep the state-feedback closed loop poles the same as in the last question.

$$
\begin{aligned}
p_1 &= -4.9175 + 4.9818i \\
p_2 &= -4.9175 - 4.9818i \\
p_3 &= -10.0000 + 0.0000i
\end{aligned}
$$

In order to show the effect of the observer, non-zero initial conditions are used and the initial state of the system is set to be

$$x_0 = \begin{bmatrix} -3 \\ -3 \\ -3 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$

which means the initial states of plant is $x_{0s} = \begin{bmatrix} -3 & -3 & -3 \end{bmatrix}^T$ and that of the observer is $x_{0o} = \begin{bmatrix} -1 & -1 & -1 \end{bmatrix}^T$.

With different poles settlement of the observer (See Table) we have the results as Figure 22 to 25 and Table 2. In each figure, the three images from left to right are state errors, 2-norm value of error vector($|E| = \sqrt{\Sigma_{i=1}^{3} x_i^2}$) and the step response comparison.

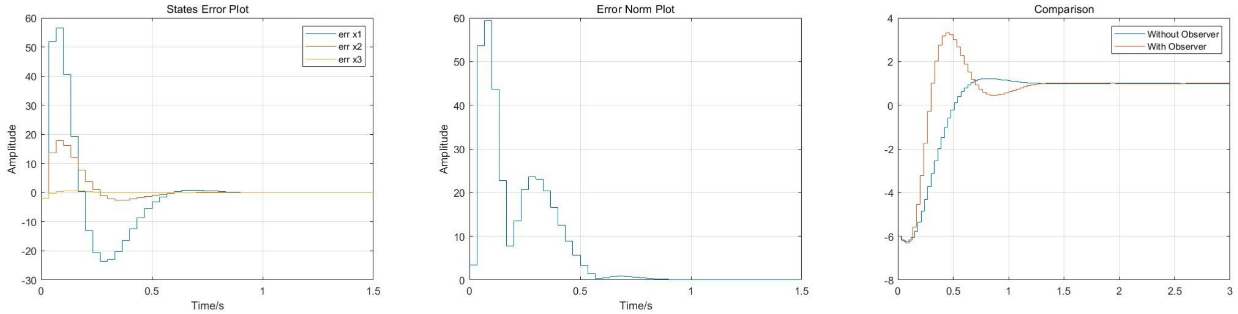

Figure 22: Observer Poles: -8.4301 + 8.5402i, -8.4301 - 8.5402i, -15.0000 + 0.0000i

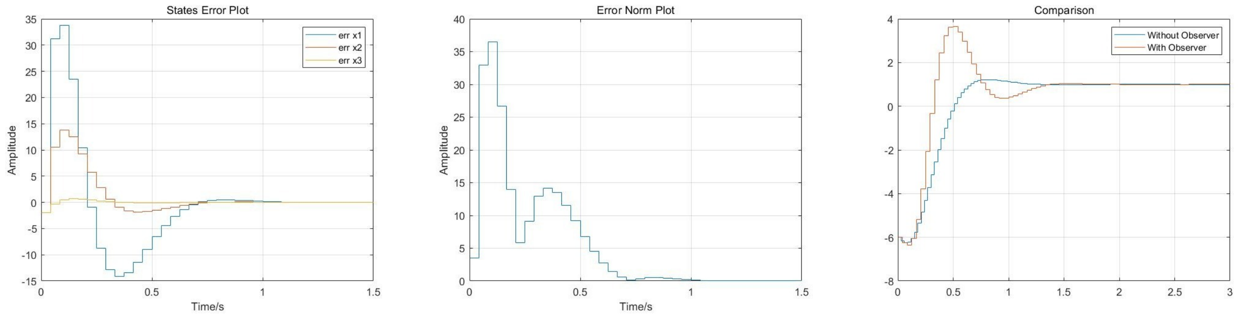

Figure 23: Observer Poles: -7.0250 + 7.1168i, -7.0250 - 7.1168i, -12.0000 + 0.0000i

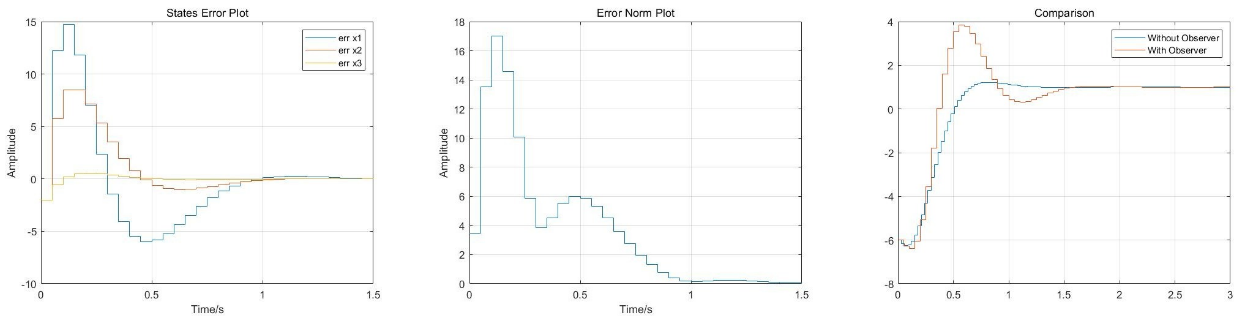

Figure 24: Observer Poles: -4.9175 + 4.9818i, -4.9175 - 4.9818i, -10.0000 + 0.0000i

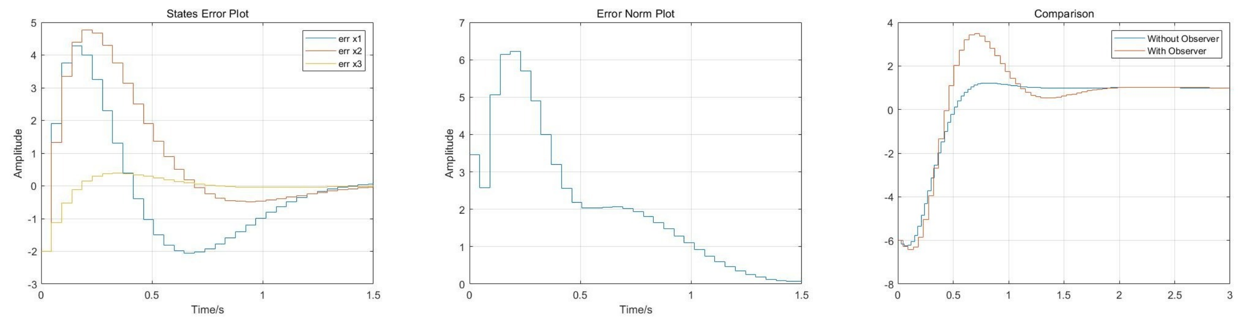

Figure 25: Observer Poles:-3.5125 + 3.5584i, -3.5125 - 3.5584i, -6.5000 + 0.0000i

Placement	Figure	Pole Pair Modulus	Poles
1	22	$r = 12$	-8.4301 + 8.5402i -8.4301 - 8.5402i -15.0000 + 0.0000i
2	23	$r = 10$	-7.0250 + 7.1168i -7.0250 - 7.1168i -12.0000 + 0.0000i
3	24	$r = 7$	-4.9175 + 4.9818i -4.9175 - 4.9818i -10.0000 + 0.0000i
4	25	$r = 5$	-3.5125 + 3.5584i -3.5125 - 3.5584i -6.5000 + 0.0000i

Table 2: Observer Poles Placement Parameters

Generally from all the placements, it can be drawn that with observer poles placed to the further left (which means faster), the observer can converge to the authentic states of the plant faster with less settling time (not only the respective error of each states will but also the norm of the error vector will.)

The placement 3 is actually the same scheme as the pole placement of the pre-designed feedback closed-loop poles. It is observed not only the observer performance but also the closed-loop step response deteriorate significantly once the observer poles is of the same speed as or slower than the state-feedback speed (Figure 24 to 25).

Also, from Figure 22 and 23, when observer poles are fast enough, the adjustment of tuning them to be even faster brings no significant but marginal effect on the system closed-loop step response, even it does bring a faster observer convergence.

2.5 Question 6 - Augmented Observer with Integral Action

2.5.1 System Augmentation

Now a step disturbance is added to the input of the plant, i.e. $u'(k) = u(k) + w(k)$. In order to eliminate the influence brought by the input disturbance, the system states as well as the observer are augmented to include the step disturbance. The dynamic of the disturbance is described as a constant, for which the observer of that turns out to be an integrator. Note that the system is observable.

The plant is augmented to become

$$\begin{bmatrix} x(k+1) \\ w(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \end{bmatrix} + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} u(k)$$
$$y(k) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \end{bmatrix}$$

and with control input $u(k) = -Kx(k) - K_w w(k)$ it leads to

$$\begin{bmatrix} x(k+1) \\ w(k+1) \end{bmatrix} = \begin{bmatrix} \Phi - \Gamma K & \Gamma - \Gamma K_w \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \end{bmatrix} + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} u(k)$$

So by choosing $K_w = 1$ we can cancel the impact of the disturbance(if we know w). The problem then boils down to how to estimate w. Then we need an augmented observer.

With the output error $\epsilon(k) = y(k) - Cx(k)$ the observer is

$$\begin{aligned} \hat{x}(k+1) &= \Phi\hat{x}(k) + \Gamma\hat{w}(k) + \Gamma u(k) + L\epsilon(k) \\ &= (\Phi - LC)\hat{x}(k) + \Gamma\hat{w}(k) + \Gamma u(k) + LCx(k) \\ \hat{w}(k+1) &= \hat{w}(k) + L_w(y(k) - C\hat{x}(k)) \\ &= \hat{w}(k) + L_w C(x(k) - \hat{x}(k)) \end{aligned}$$

Therefore, the dynamics of estimation error $e(k) = \begin{bmatrix} e_x(k) & e_w(k) \end{bmatrix}^T$ are

$$\begin{aligned}
e_x(k+1) &= x(k+1) - \hat{x}(k+1) \\
&= [\Phi x(k) + \Gamma w(k) + \Gamma u(k)] - [(\Phi - LC)\hat{x}(k) + \Gamma \hat{w}(k) + \Gamma u(k) + LCx(k)] \\
&= (\Phi - LC)e_x(k) \\
e_w(k+1) &= w(k+1) - \hat{w}(k+1) \\
&= w(k) - (\hat{w} + L_w C(x(k) - \hat{x}(k))) \\
&= e_w(k) - L_w C(x(k) - \hat{x}(k)) \\
&= e_w(k) - L_w C e_x(k)
\end{aligned}$$

Note that by choosing proper observer gain L, the $e_x(k)$ will asymptotically go to zero. But for disturbance observer, it is just a observer subsystem with fixed eigenvalue $\lambda = 1$ that cannot be changed by the observer gain L_w. Since here we have $C = \begin{bmatrix} 0 & 0 & 2 \end{bmatrix}$, it leads to

$$e_w(k+1) = e_w(k) - 2L_w e_{x3}(k)$$

In conclusion we have the overall closed-loop state-space model as follows.

$$\begin{aligned}
\begin{bmatrix} x(k+1) \\ w(k+1) \\ \hat{x}(k+1) \\ \hat{w}(k+1) \end{bmatrix} &= \begin{bmatrix} \Phi & \Gamma & 0 & 0 \\ 0 & 1 & 0 & 0 \\ LC & 0 & \Phi - LC & \Gamma \\ L_w C & 0 & -L_w C & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \\ \hat{x}(k) \\ \hat{w}(k) \end{bmatrix} + \begin{bmatrix} \Gamma \\ 0 \\ \Gamma \\ 0 \end{bmatrix} u(k) \\
&= \Phi_{aug} \begin{bmatrix} x(k) \\ w(k) \\ \hat{x}(k) \\ \hat{w}(k) \end{bmatrix} + \Gamma_{aug} u(k) \\
y(k) &= \begin{bmatrix} C & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \\ \hat{x}(k) \\ \hat{w}(k) \end{bmatrix} + Du(k) \\
&= C_{aug} \begin{bmatrix} x(k) \\ w(k) \\ \hat{x}(k) \\ \hat{w}(k) \end{bmatrix} + D_{aug} u(k)
\end{aligned}$$

With the feedback input $u(k) = -K\hat{x}(k) - \hat{w}(k) + K_r r(k)$, we have

$$\begin{aligned}
\begin{bmatrix} x(k+1) \\ w(k+1) \\ \hat{x}(k+1) \\ \hat{w}(k+1) \end{bmatrix} &= \begin{bmatrix} \Phi & \Gamma & -\Gamma K & -\Gamma \\ 0 & 1 & 0 & 0 \\ LC & 0 & \Phi - LC - \Gamma K & 0 \\ L_w C & 0 & -L_w C & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \\ \hat{x}(k) \\ \hat{w}(k) \end{bmatrix} + \begin{bmatrix} \Gamma K_r \\ 0 \\ \Gamma K_r \\ 0 \end{bmatrix} r(k) \\
y(k) &= \begin{bmatrix} C & 0 & -DK & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \\ \hat{x}(k) \\ \hat{w}(k) \end{bmatrix} + DK_r r(k)
\end{aligned}$$

in which the input K_r is introduced to ensure the static gain of 1.

2.5.2 Analysis

The general design process is the same as the observer design in the last question, i.e. we use the state-feedback gain K and observer gain L obtained in the last question where the system hasn't been augmented yet. The only difference is that we need to tune the newly added disturbance observer gain L_w, which is related to the dynamics of the error. Here we just take it as a gain that could be tuned.

2.5.3 Results

We used the observer gain L and state-feedback gain K from the last question such that the poles of the state-feedback loop are placed at

$$p_1 = -4.9175 + 4.9818i$$
$$p_2 = -4.9175 - 4.9818i$$
$$p_3 = -10.0000 + 0.0000i$$

and the ones of observer at

$$p_1 = -4.9175 + 4.9818i$$
$$p_2 = -4.9175 - 4.9818i$$
$$p_3 = -10.0000 + 0.0000i$$

And tune the L_w to get different performance result. Here the initial condition is chosen to be

$$x_0 = \begin{bmatrix} 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 \end{bmatrix}^T$$

such that observer and plant have different initial value meanwhile initializing the constant input disturbance with 1. The results are shown in the Figure 26 to 30.

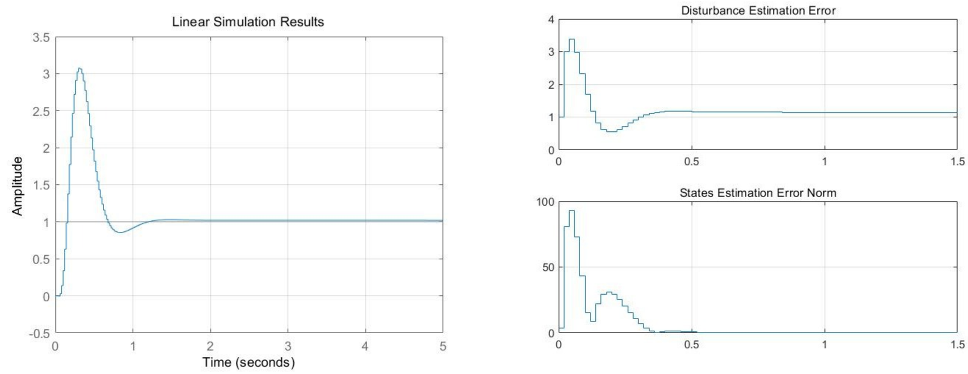

Figure 26: $L_w = 1$

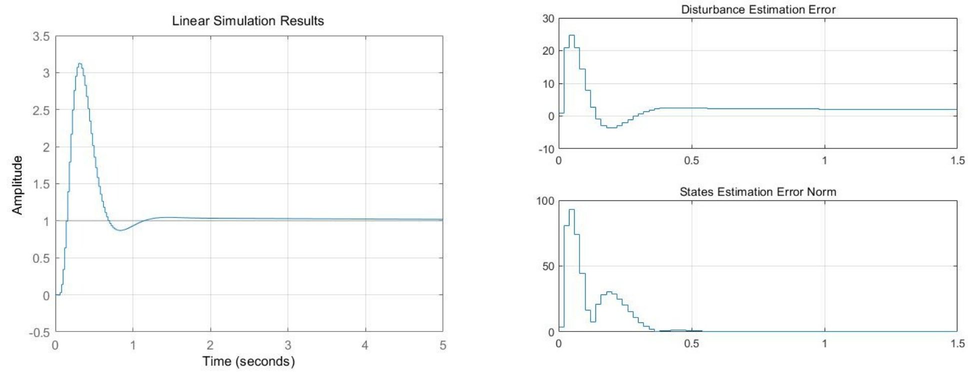

Figure 27: $L_w = 10$

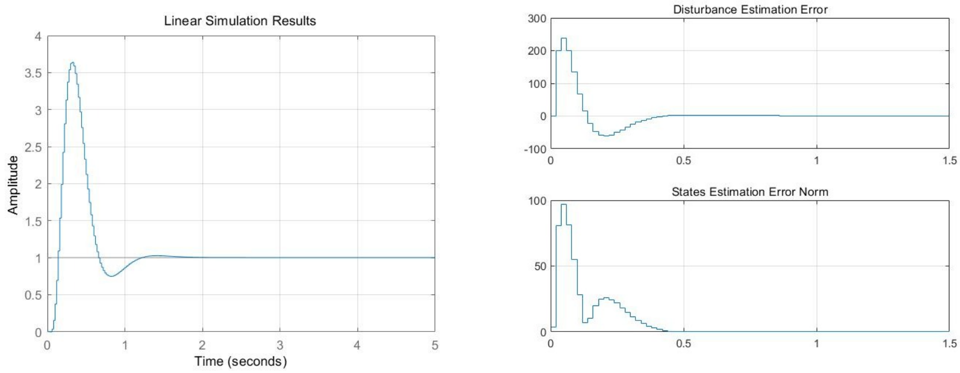

Figure 28: $L_w = 100$

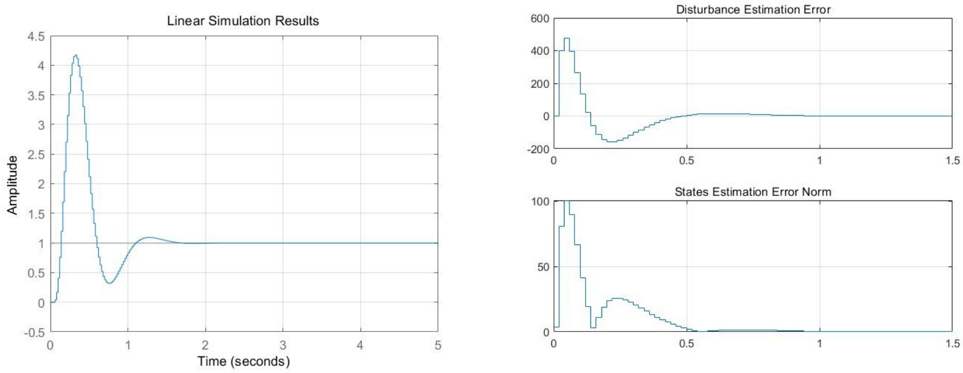

Figure 29: $L_w = 200$

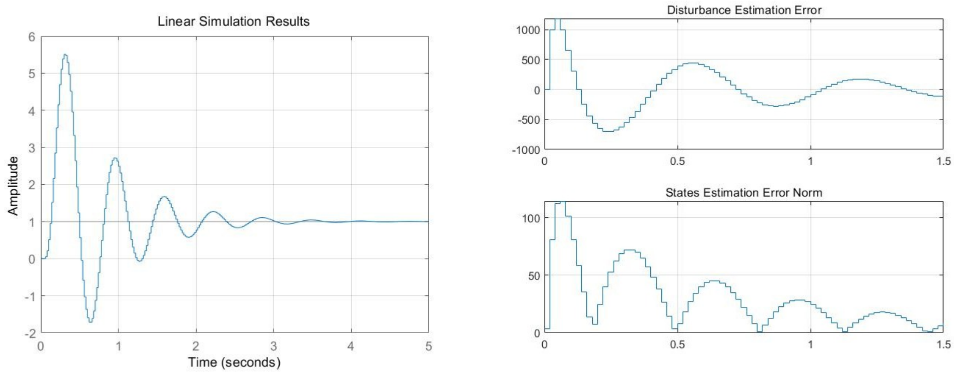

Figure 30: $L_w = 500$

When L_w goes too high the system begins to fluctuate and performs poorly. The final L_w is chosen to be 200. It is observed that the settling time for disturbance observer generally agrees no matter what value big L_w is (but not too big to make system unstable) and large L_w only change the peak value disturbance estimation error as well as decrease the steady error, but it doesn't have much influence on the step response of the overall system.

2.6 Question 7

For LQ controller we only need to tune the weighting matrix Q_1 and Q_2 (Here as there is no explicit necessity we assume the weighting matrix for cross-term to be zero) for the following form of cost function

$$J = \Sigma(x^T(k)Q_1x(k) + u^T(k)Q_2u(k))$$

Also for Q and R we assume them to be diagonal matrix so that we could evaluate the penalty for each state variable and input element respectively.

For the choice of weighting matrix, we have a general formula as belows.

$$q_{ii} = \frac{1}{(allowed \quad deviation)^2}$$

Then here with different weighting matrix we have the results as Figure 31.

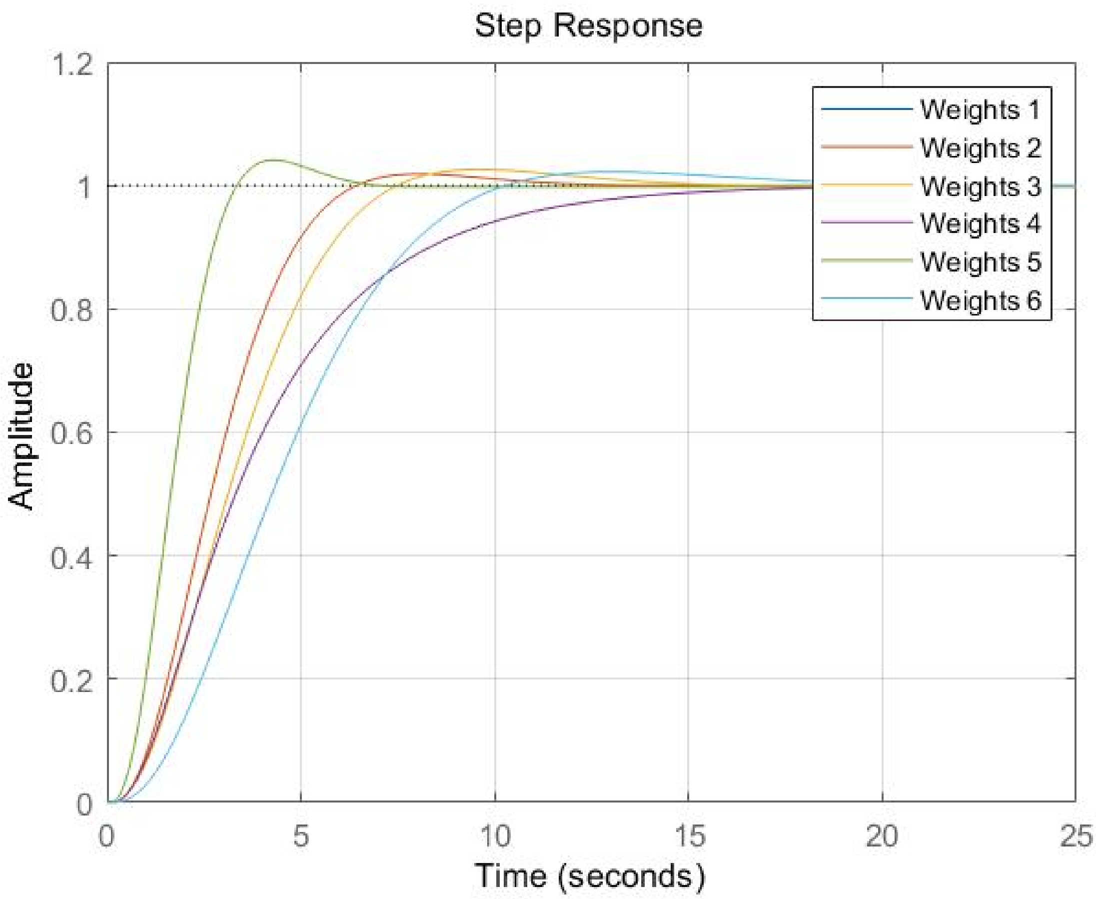

Figure 31: LQ Control Step Response with Different Weights

Weights NO.	Q_{11}	Q_{12}	Q_{13}	Q_2
1	1	1	1	1
2	10^3	10^3	10^3	10^3
3	10^4	10^3	10^3	10^3
4	10^3	10^4	10^3	10^3
5	10^3	10^3	10^4	10^3
6	10^3	10^3	10^3	10^4

Table 3: Weights Parameters

As shown in Figure 31, the results Weight 1 and 2 totally overlap each others and it shows that the absolute value of the gain doesn't matter that much instead it is the relative greatness between gains that works.

From Weight 3 to 6, we respectively increase the gain of the three states and the input to see its effect by comparison. It shows that the increase of the weight of the first two states leads to slower response.

The weight increase on the third state would brings a more significant impact to improve the system performance. This is might be interpreted as the output matrix $C = \begin{bmatrix} 0 & 0 & 2 \end{bmatrix}$ means the output is linear dependant on the third state x_3(and is positively correlated).

Last but not the least, the weight increase on input leads to slower response. This is easy to understand as the input would be penalized much more with higher weight.

As there is not limitation for controller output here, we could continuously achieve the better performance with higher weight difference.

Here we have the final matrix as below(Figure).

$$Q_1 = \begin{bmatrix} 10^3 & 0 & 0 \\ 0 & 10^5 & 0 \\ 0 & 0 & 10^8 \end{bmatrix}$$
$$Q_2 = 1$$

and it gives back the feedback gain $K =$

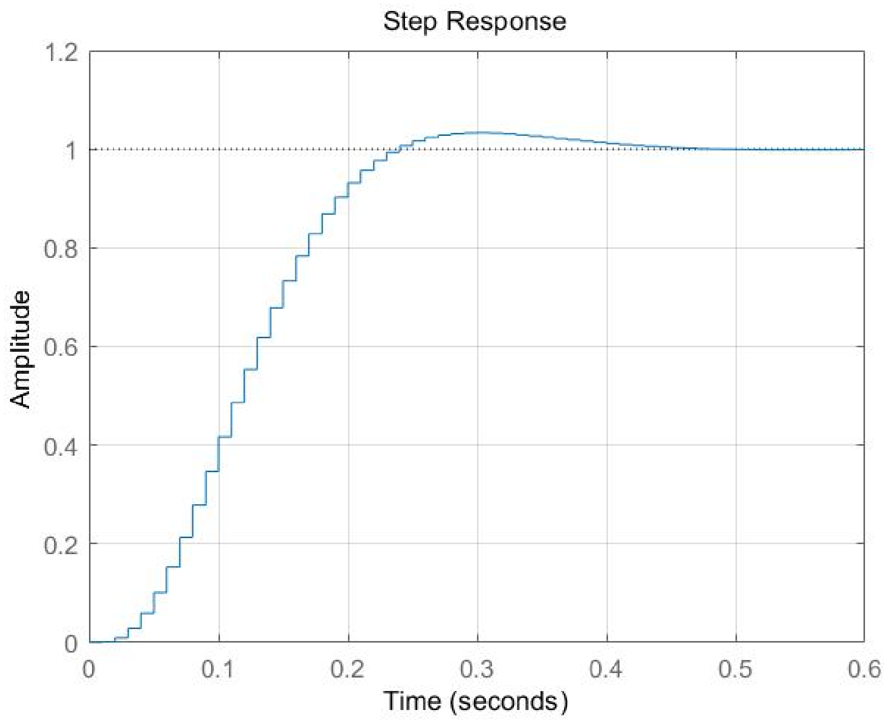

Figure 32: Final Designed LQ Controller Step Response(Left), Response Comparison with Other Design(Right)

and it fits the requirements given in the first quesiton (overshoot less than 5%).

3 Controller Limitation

3.1 Question 8

For different controllers designed above, different ways are used to get the input signal.

For discrete-time PID controller for step response, the transfer function from reference to the input signal is calculated by

$$G_u(z) = \frac{C(z)}{1 + C(z)G(z)}$$

in which $C(z)$ is the discretized controller and $G(z)$ is the discretized plant (with the same sampling time T_s). The result is shown in Figure 33.

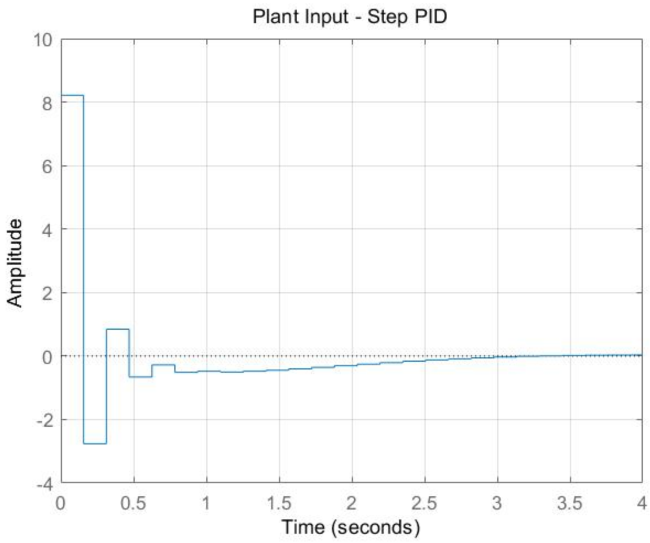

Figure 33: Plant Input - Step PID

For discrete-time PID controller for step response, the transfer function from reference to the input signal is calculated by

$$G_u(z) = \frac{1}{1 + C(z)G(z)}$$

in which $C(z)$ is the discretized controller and $G(z)$ is the discretized plant (with the same sampling time T_s). The result is shown in Figure 33.

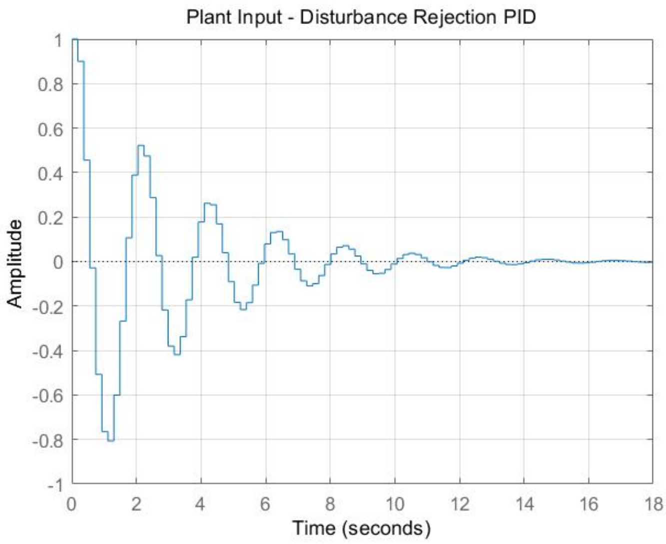

Figure 34: Plant Input - Disturbance Rejection PID

For pole-placement state-feedback controller, input is calculated by

$$u(k) = -Kx(k) + K_r r(k)$$

in which the system states are obtained from simulation and K_r is the calculated gain to ensure the static from r to output to be 1. The input is shown in Figure 35.

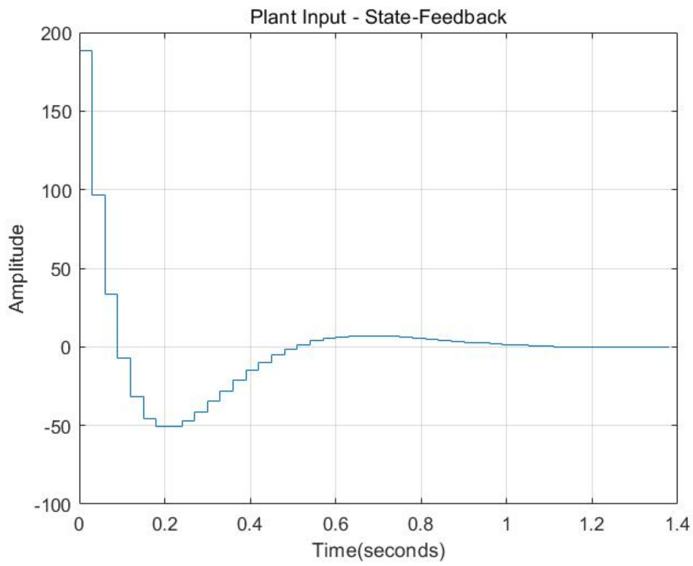

Figure 35: Plant Input - State-Feedback

For pole-placement observed-based output-feedback controller, input is calculated by

$$u(k) = -K\hat{x}(k) + K_r r(k)$$

in which the system states are obtained from simulation and K_r is the calculated gain to ensure the static from r to output to be 1. The input is shown in Figure 36.

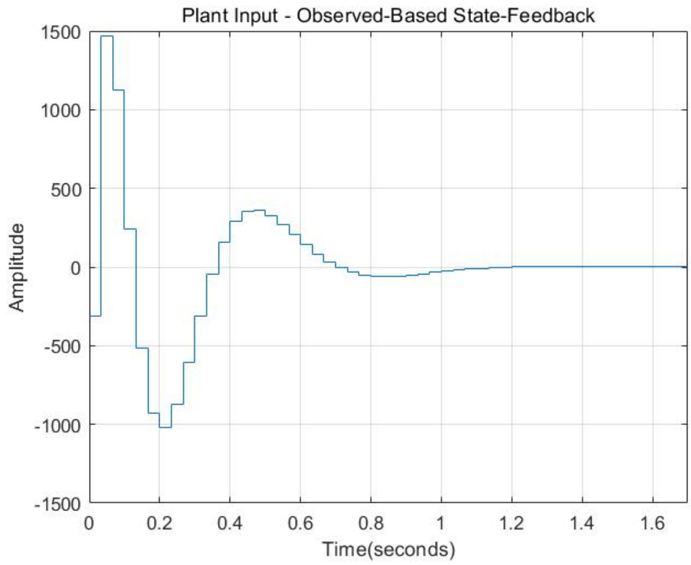

Figure 36: Plant Input - Observed-Based State-Feedback

For the augmented pole-placement observed-based output-feedback controller with input integral action, input is calculated by

$$u(k) = -K\hat{x}(k) + \hat{w}(k) + K_r r(k)$$

in which the system states are obtained from simulation and K_r is the calculated gain to ensure the static from r to output to be 1. The input is shown in Figure 37.

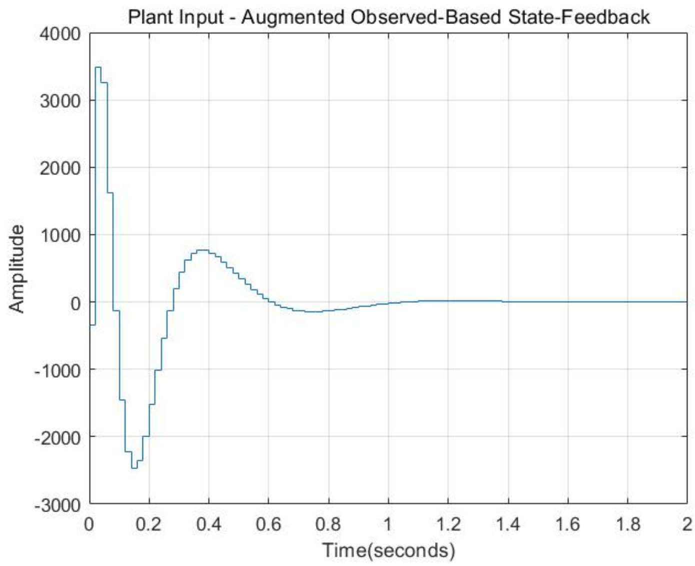

Figure 37: Plant Input - Augmented Observed-Based State-Feedback

For LQ controller, input is calculated by

$$u(k) = -Kx(k) + K_r r(k)$$

in which the system states are obtained from simulation and K_r is the calculated gain to ensure the static from r to output to be 1. The input is shown in Figure 38.

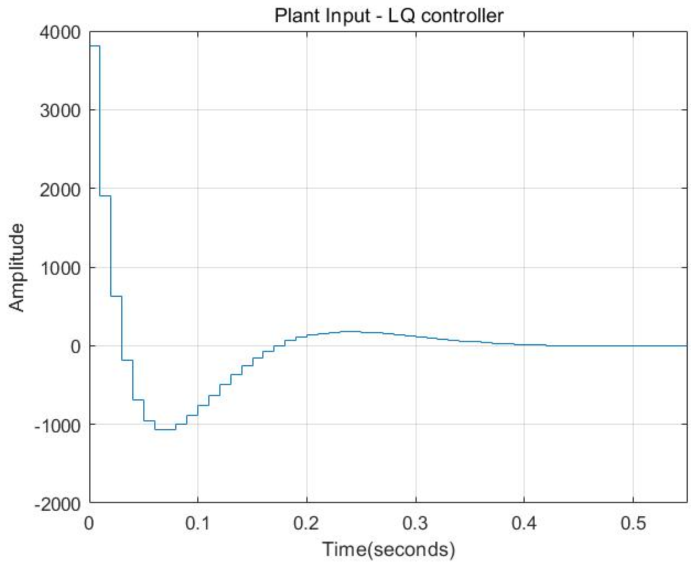

Figure 38: Plant Input - LQ controller

3.2 Question 9

Here we retune the PID controller to have the input signal within the limitations.

For the step response, it is observed that it generally reaches the highest at the very beginning and could be significantly reduced by lowering the K_D. Meanwhile, K_P must be also lowered to give more phase margin to prevent the system from having too much overshoot and being unstable. T_f is generally unchanged. the PID controller is re-tuned to become

$$K_P = 0.05$$
$$K_D = 5$$
$$C_{step}(s) = K_P + K_D \frac{s}{s + T_f} = \frac{5.05s + 1.35}{s + 27}$$

and after discretization of Tustin method it becomes

$$C_{step}(z) = \frac{0.4989z - 0.4079}{z + 0.8204}$$

The retuned controller is shown as below in Figure 39 and 40.

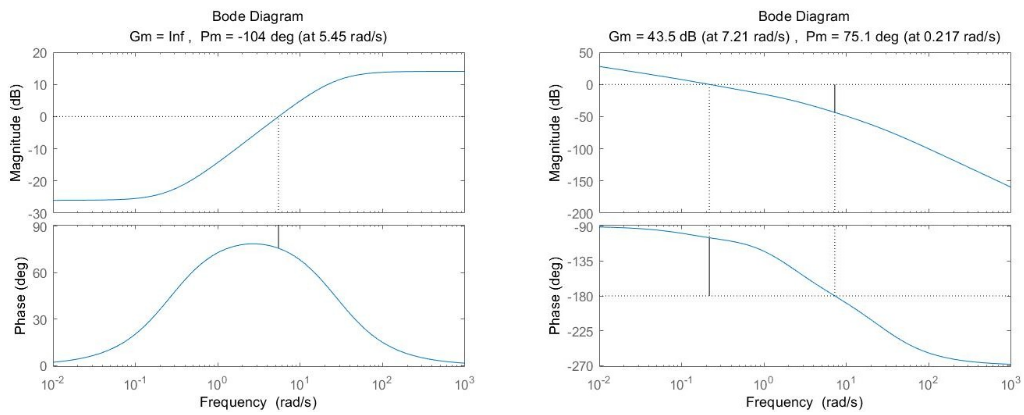

Figure 39: Continuous-time Controller Bode Plot(Left), Continuous-time Open-loop Bode Plot(Right)

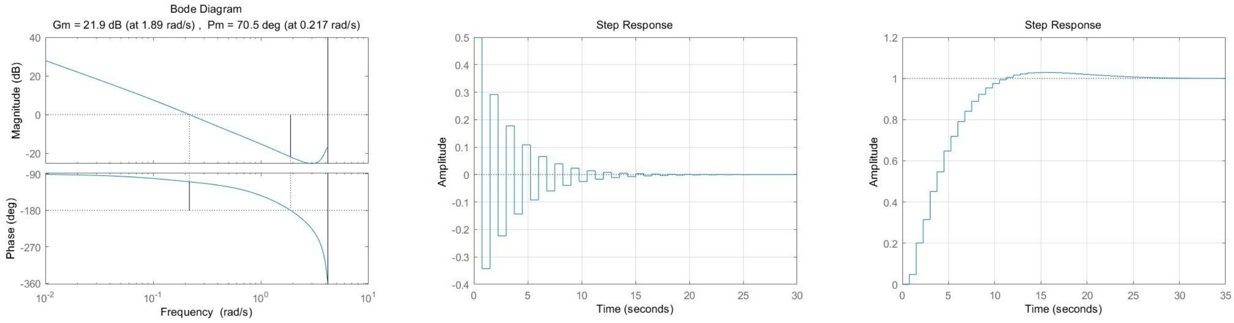

Figure 40: Discrete-time Open-loop Bode Plot(Left), Discrete-time Plant Input with Peak Value of 0.4989(middle), Discrete-time Step Response(Right)

For disturbance rejection controller, the input requirements are questionable as it requires input to be within $[-0.5, 0.5]$ while input must starts from 1 for the disturbance is directly fed on the plant input ($u = w + u_{con}, u_{con}$ is the controller output). If we switch to interpret it as just the controller instead of the plant input, the controller output actually converges to -1 to counter the effect of the step disturbance as time goes to infinity, which is also out of the required range.

Therefore, we stick with the original interpretation that what we aim at is the plant input and take the requirements as to restrict the second peak(the first wave trough) greater than -0.5. The generally re-tuning idea is the same as what we discussed before for the step PID controller re-desgining. And we have the re-tuned PID disturbance rejection controller as

$$\begin{aligned} T_f &= 100 \\ K_P &= 2.5 \\ K_I &= 0.37 \\ K_D &= 5 \\ C_{dr}(s) &= \frac{502.5s^2 + 250.4s + 37}{s^2 + 100s} \\ \implies C_{dr}(z) &= \frac{39.39z^2 - 73.97z + 34.75}{z^2 - 0.1474z - 0.8526} \end{aligned}$$

The results are shown in the Figure 41 and 42.

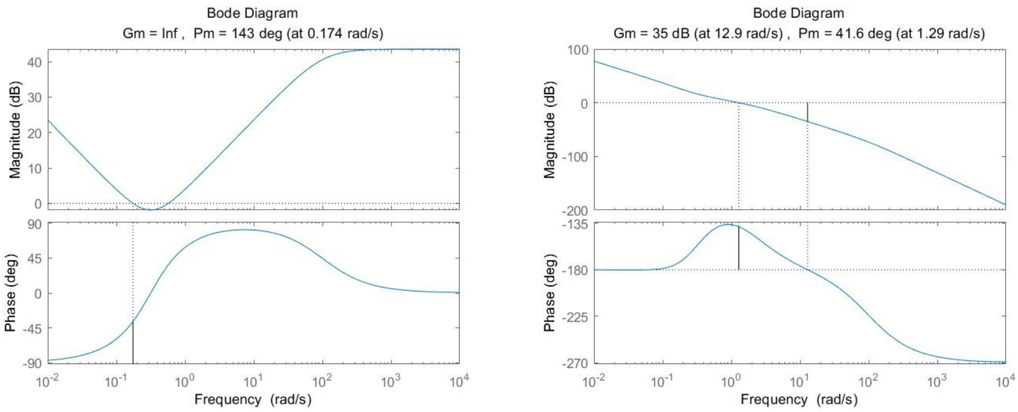

Figure 41: Continuous-time Controller Bode Plot(Left), Continuous-time Open-loop Bode Plot(Right)

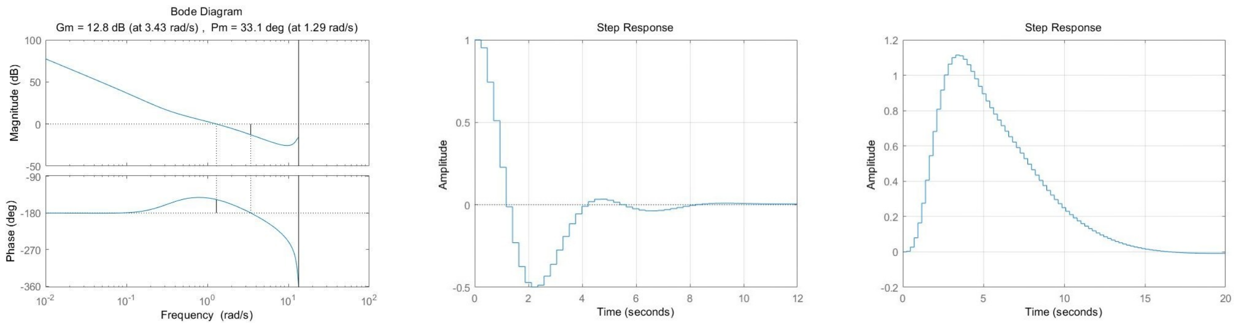

Figure 42: Discrete-time Open-loop Bode Plot(Left), Discrete-time Plant Input with Trough Value of 0.499(middle), Discrete-time Step Response(Right)

3.3 Question 10

Here we re-design the poles locations to fit the input requirements. The general idea is to reduce the magnitude of the poles (pull them to the right) and to get them closer to the original positions. As we try to achieve the best performance within the input limitations, also because the performance is mainly determined by the dominant poles, we try to fit into the requirements meanwhile moving the dominant pole pair to see how left it could be to speed up the system. The input increase brought by increasing the magnitude of the dominant poles could be compensated by decreasing the one of third pole (i.e. moving it to the right). However, if the dominant pole pair gets too close to the third pole, the dominance of the pole pair will be diminished and the system would slow down since the third pole begins to play a stronger role. In conclusion, in the re-design of pole-placement state-feedback controller, we try to balance all the limitations above.

The re-designed poles in continuous-time domain are

$$
\begin{aligned}
p_1 &= -0.5620 + 0.5693i \\
p_2 &= -0.5620 - 0.5693i \\
p_3 &= -1.6000 + 0.0000i \\
|p_1| &= |p_2| = 0.8
\end{aligned}
$$

and the result is shown in Figure 43

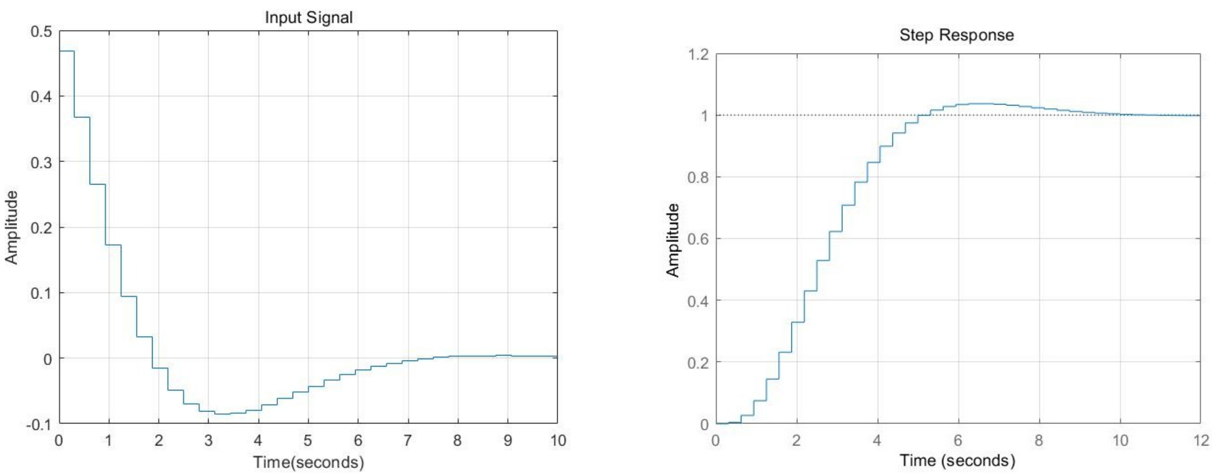

Figure 43: Re-tuned Input Signal with Peak Value of 0.4689(Left), Re-tuned System Step Reponse(Right)

Since there is no limitations for observer gain L we could still keep the original observer pole design. Since the the observer could be improved infinitely, the feedback will get infinitely close to the full-state feedback behavior. The only thing that leads to difference in the input signal is the different initial states between the true plant and the observer, but as long as the initial difference grows up the input would soon explodes out of the given limitation. Therefore, the observer case as well as the augmented observer case won't be neither discussed nor re-tuned here.

3.4 Question 11

The LQ controller could be simply re-tuned by choosing small weights on the states x_1, x_2, x_3 while increasing the weight imposed on the input u. Here the new weighting matrices are

$$Q_1 = \begin{bmatrix} 10^4 & 0 & 0 \\ 0 & 10^4 & 0 \\ 0 & 0 & 10^4 \end{bmatrix}$$
$$Q_2 = 6.01 * 10^3$$

And the result is shown in Figure 44.

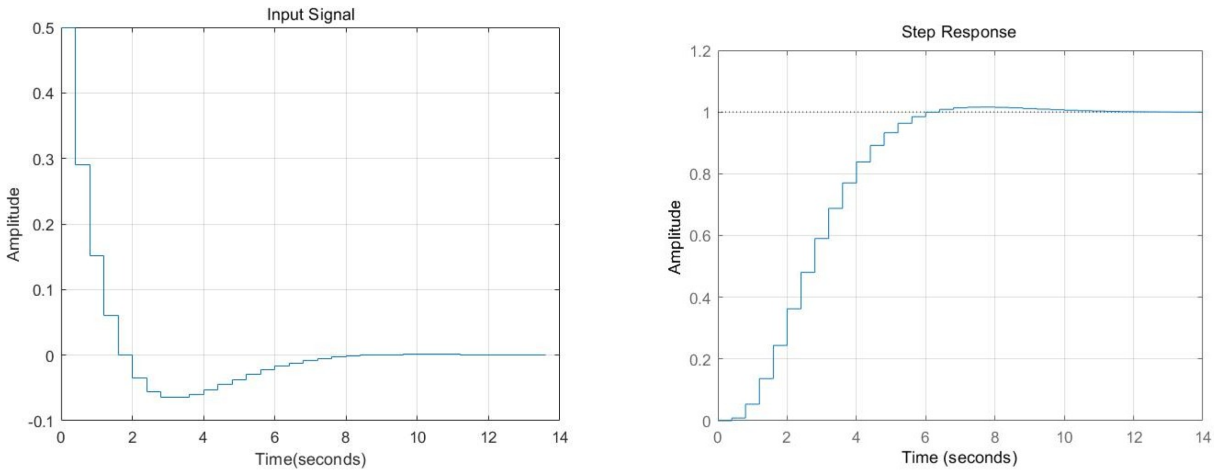

Figure 44: Re-tuned Input Signal with Peak Value of 0.4998(Left), Re-tuned System Step Reponse(Right)

4 Steady State Error

4.1 Question 12

4.1.1 PID Controller

The PID controller designed in the previous parts could be directly used to cancel the effect of the input disturbance. The simulation is done with simulink as Figure 45.

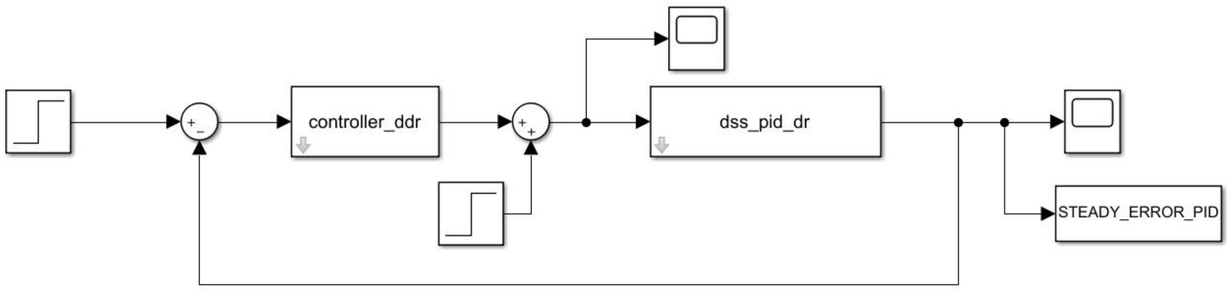

Figure 45: Simulation for PID Controller Servo Step Tracking with Input Disturbance

and the result is shown in Figure 46.

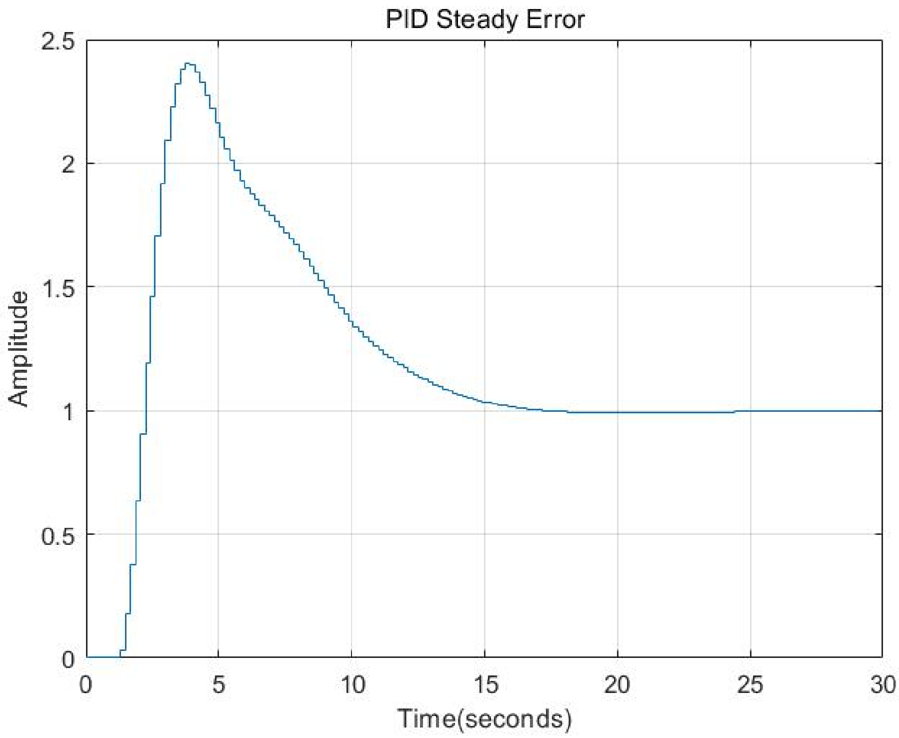

Figure 46: Simulation Result for PID Controller Servo Step Tracking with Input Disturbance

4.1.2 Pole Placement State-Space Feedback

For Pole Placement Controller, we implement the augmented output-feedback controller to cancel the effect of input disturbance (which is exactly the same as what we did in Question 6). Therefore here we just show the result of that. We have the augmented system as

$$\begin{bmatrix} x(k+1) \\ w(k+1) \\ \hat{x}(k+1) \\ \hat{w}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma & -\Gamma K & -\Gamma \\ 0 & 1 & 0 & 0 \\ LC & 0 & \Phi - LC - \Gamma K & 0 \\ L_w C & 0 & -L_w C & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \\ \hat{x}(k) \\ \hat{w}(k) \end{bmatrix} + \begin{bmatrix} \Gamma K_r \\ 0 \\ \Gamma K_r \\ 0 \end{bmatrix} r(k)$$

$$y(k) = \begin{bmatrix} C & 0 & -DK & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \\ \hat{x}(k) \\ \hat{w}(k) \end{bmatrix} + DK_r r(k)$$

and with input step disturbance (by initializing the state $w(k)$ with 1) we have the simulation result as Figure 47 (here L_w is chosen to be 100 as before).

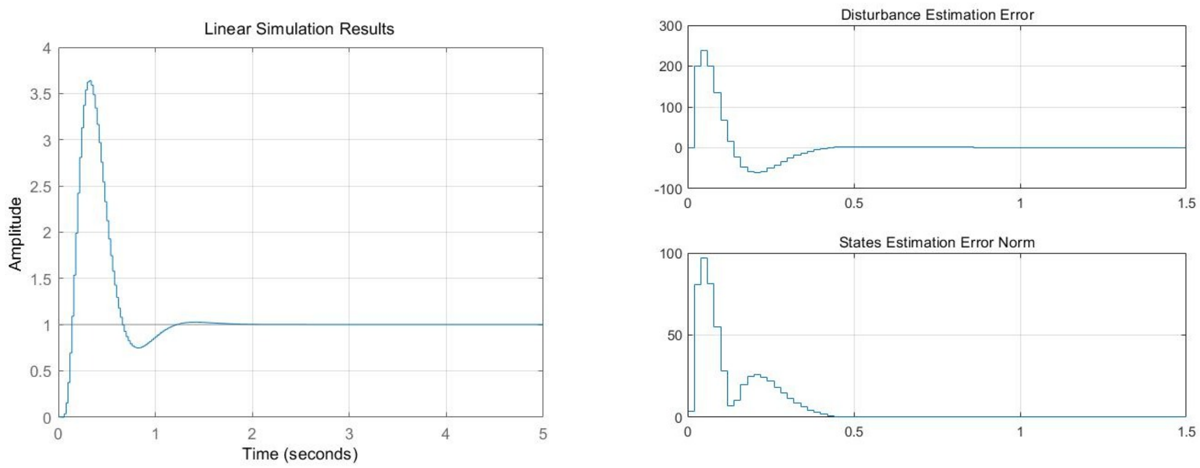

Figure 47: Simulation Result for Pole-Placement Controller Servo Step Tracking with Input Disturbance

4.1.3 LQ Controller

Here it is verified that simply LQ controller cannot eliminate the steady error brought by the input disturbance. Therefore we also augment the system as last question in the form

$$\begin{bmatrix} x(k+1) \\ w(k+1) \\ \hat{x}(k+1) \\ \hat{w}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma & -\Gamma K_l q & -\Gamma \\ 0 & 1 & 0 & 0 \\ LC & 0 & \Phi - LC - \Gamma K_l q & 0 \\ L_w C & 0 & -L_w C & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \\ \hat{x}(k) \\ \hat{w}(k) \end{bmatrix} + \begin{bmatrix} \Gamma K_r \\ 0 \\ \Gamma K_r \\ 0 \end{bmatrix} r(k)$$

$$y(k) = \begin{bmatrix} C & 0 & -K_l q & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \\ \hat{x}(k) \\ \hat{w}(k) \end{bmatrix} + DK_r r(k)$$

and the major difference is the feedback gain we used, which comes from the LQ design.

However, we have to verify the poles locations. Since when designing the LQ controller, we only obtain the poles in the discrete-time domain, now we have to compare the speed of poles also just in discrete-tme domain. With the poles mapping rules $z = e^{sh}$ we know in discrete-time domain the closer poles are placed to the edge of the unit circle, the slower the poles response are. And with LQ controller design, we get to know the state-feedback poles are

$$z_1 = 0.7379 + 0.2046i$$
$$z_2 = 0.7379 - 0.2046i$$
$$z_3 = 0.6065 + 0.0000i$$

which are much faster than the observer poles. And it shouldn't be. One more thing is to consider is the change of sampling time, which inevitably requires the system to be re-designed. Therefore, there's necessity to re-design the observer and it leads to the re-designed observer poles

$$\begin{aligned} p_1 &= -42.1503 + 42.7008i \\ p_2 &= -42.1503 - 42.7008i \\ p_3 &= -80.0000 + 0.0000i \\ |p_1| &= |p_2| = 60 \end{aligned}$$

which in discrete-time domain turns out to be

$$\begin{aligned} z_1 &= 0.7412 + 0.2026i \\ z_2 &= 0.7412 - 0.2026i \\ z_3 &= 0.6065 + 0.0000i \end{aligned}$$

After that we have the simulation result of that in Figure 48 (with the initial value $x_0 = \begin{bmatrix} 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 \end{bmatrix}^T$).

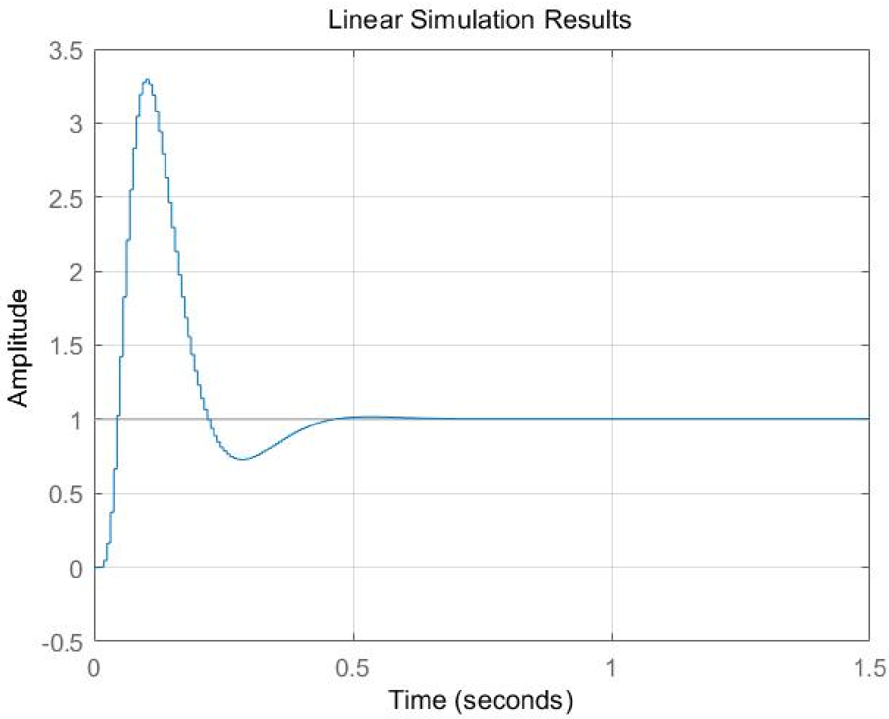

Figure 48: Simulation Result for Augmented LQ Controller Servo Step Tracking with Input Disturbance

5 Time Delay

5.1 Question 13

The general formation for the problem would be

$$x(k+1) = \Phi x(k) + \Gamma u(k-1)$$
$$y(k) = Cx(k) + Du(k-1)$$

For transfer function model (i.e. the first two questions), to add one-step delay for the transfer function system here, we just need to multiply the plant with $\frac{1}{z}$ term(Note that it should be of the same sampling time as the system).Note that it's the plant we decide to the multiply $\frac{1}{z}$ with as it means that the system will process the input given in the last time step while the controller will works perfectly and obeys $u(k) = C(z)e(k)$instead of $u(k) = C(z)e(k-1)$.

For state-space model, we generally incorporated the delay into our model by augmenting the system in the form:

$$\begin{bmatrix} x(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(k+1) \\ u(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$

and then for different system we have different feedback law, for which it would turns out to have diverse state-space model.

5.1.1 Delayed PID Step Response

With what we discussed above for the transfer function system, we have the new plant $G_{delay}(z)$

$$G_{de}(z) = \frac{1}{z} G(z)$$

with the original discretized plant $G(z)$ and the closed-loop transfer function from the reference to the system output would still be

$$G_{cl,de}(z) = \frac{C(z)G_{de}(z)}{1 + C(z)G_{de}(z)}$$

The step response comparison is shown in Figure 49.

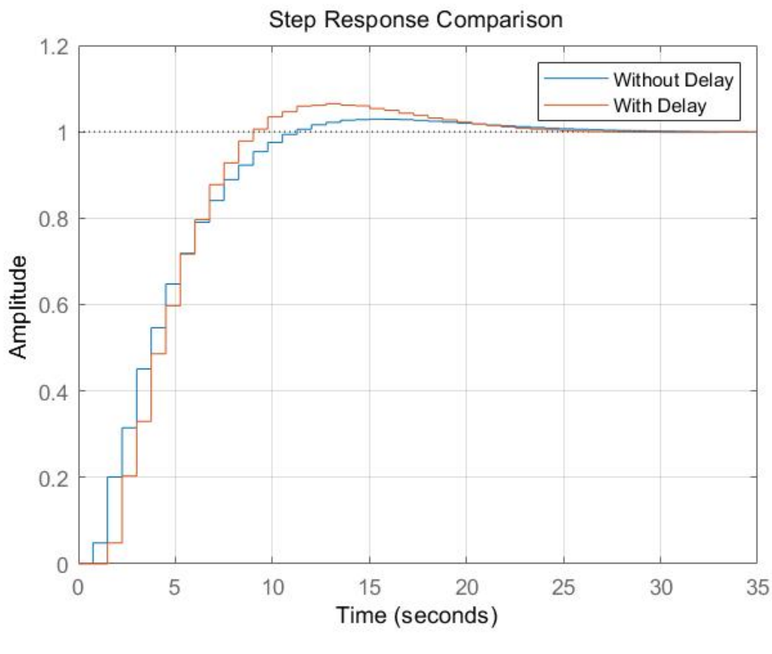

Figure 49: Comparison between with Delay and without Delay

A one-step delay could be clearly observed in the plot and the response of the delayed system has a higher overshoot, which indicates a smaller phase margin.

5.1.2 Delayed PID Disturbance Rejection

With what we discussed above for the transfer function system, we have the new plant $G_{delay}(z)$

$$G_{de}(z) = \frac{1}{z}G(z)$$

with the original discretized plant $G(z)$ and the closed-loop transfer function from the disturbance to the system output would still be

$$G_{cl,de}(z) = \frac{G_{de}(z)}{1 + C(z)G_{de}(z)}$$

The disturbance rejection response comparison is shown in Figure 50.

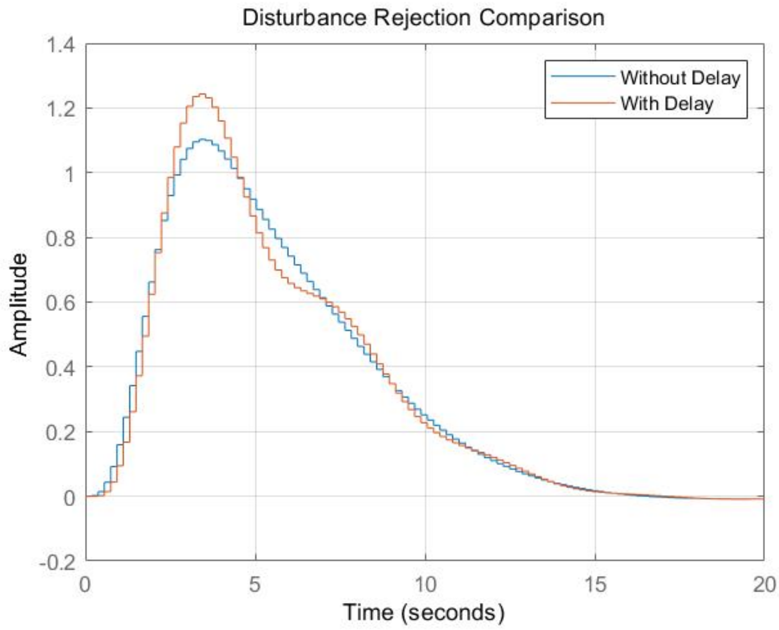

Figure 50: Comparison between with Delay and without Delay

The disturbance rejection has more fluctuation than the original system, which indicates a smaller phase margin.

5.1.3 Delayed State-feedback Controller

The state-feedback controller system is obtained by substituting in the controller law

$$u(k) = -Kx(k) + K_r r(k)$$

And the system turns out to be

$$\begin{bmatrix} x(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma \\ -K & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} 0 \\ K_r \end{bmatrix} r(k)$$
$$y(k) = \begin{bmatrix} C - DK & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + DK_r r(k)$$

For the system above we have the step response (as a servo tracking step simulation) comparison as Figure 51.

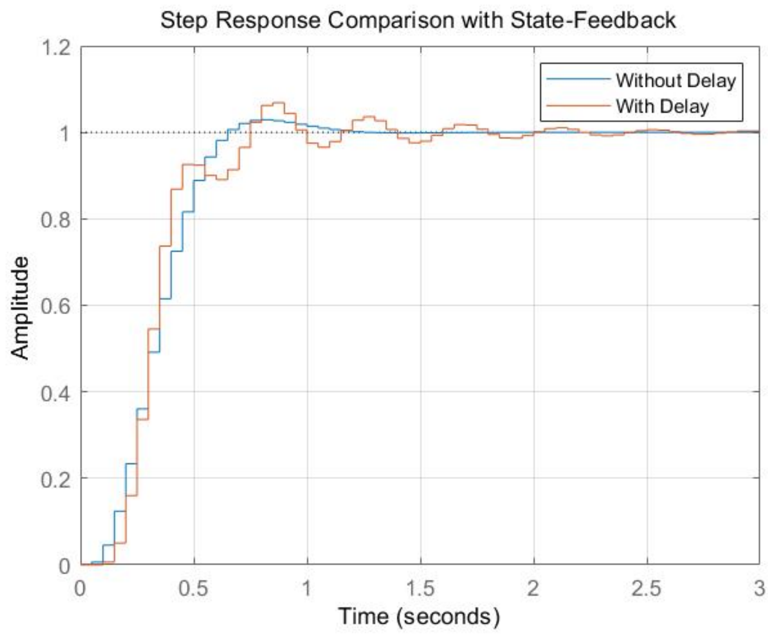

Figure 51: Comparison between with Delay and without Delay

The step response of the delayed system has significantly more fluctuation than one of original system and it indicates a smaller phase margin after delay is added.

5.1.4 Delayed Observer-based Feedback Controller

We firstly have augment the states to be $\begin{bmatrix} x(k+1) & u(k) & \hat{x}(k+1) \end{bmatrix}^T$ as what we normally do for adding an observer and we have

$$\begin{bmatrix} x(k+1) \\ u(k) \\ \hat{x}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma & 0 \\ 0 & 0 & 0 \\ LC & \Gamma & \Phi - LC \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \\ \hat{x}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u(k)$$
$$y(k) = Cx(k) + Du(k)$$

By substituting in the controller law

$$u(k) = -K\hat{x}(k) + K_r r(k)$$

the system turns out to be

$$\begin{bmatrix} x(k+1) \\ u(k) \\ \hat{x}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma & 0 \\ 0 & 0 & -K \\ LC & \Gamma & \Phi - LC \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \\ \hat{x}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ K_r \\ 0 \end{bmatrix} r(k)$$
$$\begin{aligned} y(k) &= Cx(k) - DK\hat{x}(k) + DK_r r(k) \\ &= \begin{bmatrix} C & 0 & -DK \end{bmatrix} \begin{bmatrix} x(k+1) \\ u(k) \\ \hat{x}(k+1) \end{bmatrix} + DK_r r(k) \end{aligned}$$

Note that here when doing comparison we try to keep the initial value the same for system with and without delay even if there's a dimension difference between the systems.Here for observer-based feedback system we have the in

$$x_0 = \begin{bmatrix} -3 & -3 & -3 & -1 & -1 & -1 \end{bmatrix}^T$$

And for the augmented one we have the initial state

$$x_0' = \begin{bmatrix} -3 & -3 & -3 & 0 & -1 & -1 & -1 \end{bmatrix}^T$$

Note that we initialize the augmented state $u(k)$ with 0 as at the beginning there's no input due to the one-step delay. Then response comparison is shown in Figure 52.

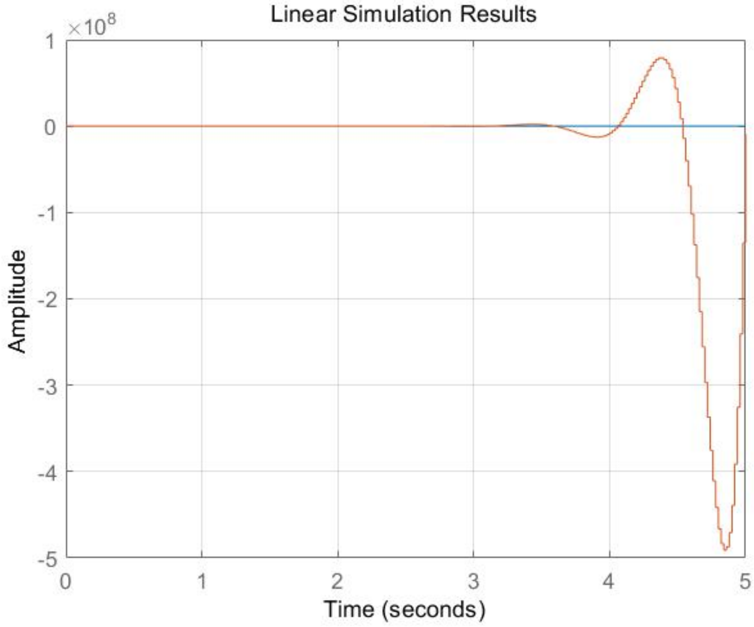

Figure 52: Comparison between with Delay and without Delay

It is clear that the response of the delayed system explodes to infinity. It shows the power of the delay. With matlab we have the poles of the delayed system

$$\begin{aligned}
p_1 &= 1.0707 + 0.1430i \\
p_2 &= 1.0707 - 0.1430i \\
p_3 &= 0.8168 + 0.0000i \\
p_4 &= 0.0078 + 0.5784i \\
p_5 &= 0.0078 - 0.5784i \\
p_6 &= -0.5480 + 0.0000i \\
p_7 &= 0.0000 + 0.0000i
\end{aligned}$$

in which the first 2 are placed outside the unit circle, indicating instability.

5.1.5 Delayed Augmented Observer-based Feedback Controller

Here the noise is included in the state $\begin{bmatrix} x(k+1) & u(k) & w(k+1) & \hat{x}(k+1) & \hat{w}(k+1) \end{bmatrix}^T$ and we have

$$\begin{bmatrix} x(k+1) \\ u(k) \\ w(k+1) \\ \hat{x}(k+1) \\ \hat{w}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma & \Gamma & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ LC & \Gamma & 0 & \Phi - LC & \Phi \\ L_w C & 0 & 0 & -L_w C & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \\ w(k) \\ \hat{x}(k) \\ \hat{w}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} C & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \\ w(k) \\ \hat{x}(k) \\ \hat{w}(k) \end{bmatrix} + Du(k)$$

By substituting in the controller law

$$u(k) = -K\hat{x}(k) - \hat{w}(k) + K_r r(k)$$

we have

$$\begin{bmatrix} x(k+1) \\ u(k) \\ w(k+1) \\ \hat{x}(k+1) \\ \hat{w}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma & \Gamma & 0 & 0 \\ 0 & 0 & 0 & -K & -1 \\ 0 & 0 & 1 & 0 & 0 \\ LC & \Gamma & 0 & \Phi - LC & \Phi \\ L_w C & 0 & 0 & -L_w C & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \\ w(k) \\ \hat{x}(k) \\ \hat{w}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} C & 0 & 0 & -DK & -1 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \\ w(k) \\ \hat{x}(k) \\ \hat{w}(k) \end{bmatrix} + DK_r r(k)$$

Note that here when doing comparison we try to keep the initial value the same for system with and without delay even if there's a dimension difference between the systems.Here for observer-based feedback system we have the in

$$x_0 = \begin{bmatrix} 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 \end{bmatrix}^T$$

And for the augmented one we have the initial state

$$x_0' = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 2 & 3 & 1 & 0 \end{bmatrix}^T$$

Note that we initialize the augmented state $u(k)$ with 0 as at the beginning there's no input due to the one-step delay. And the disturbance rejection comparison is shown in Figure 53.

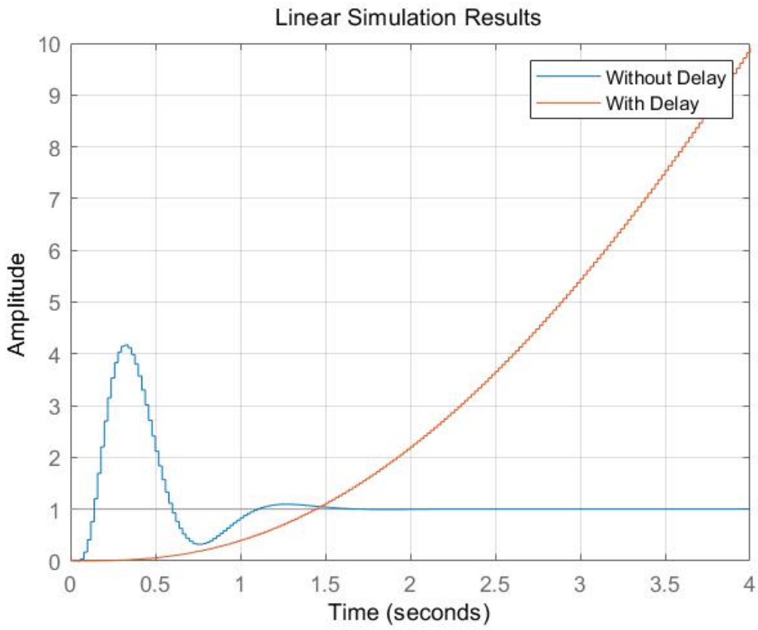

Figure 53: Comparison between with Delay and without Delay

It is clear that the response of the delayed system explodes to infinity. It shows the power of the delay. With matlab we have the poles of the delayed system

$$
\begin{aligned}
p_1 &= 1.0000 + 0.0000i \\
p_2 &= 0.9960 + 0.0000i \\
p_3 &= 0.9608 + 0.0000i \\
p_4 &= 0.0000 + 0.0000i \\
p_5 &= 0.8896 + 0.0998i \\
p_6 &= 0.8896 - 0.0998i \\
p_7 &= 0.6515 + 0.1411i \\
p_8 &= 0.6515 - 0.1411i \\
p_9 &= 1.0000 + 0.0000i
\end{aligned}
$$

of which 2 are placed outside on the edge unit circle, indicating marginal stability.

5.1.6 LQ Controller

For LQ controller, the state-feedback controller system is obtained by substituting in the controller law

$$u(k) = -Kx(k) + K_r r(k)$$

in which the K is the gain we get from the LQ design and the system turns out to be

$$
\begin{aligned}
\begin{bmatrix} x(k+1) \\ u(k) \end{bmatrix} &= \begin{bmatrix} \Phi & \Gamma \\ -K & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} 0 \\ K_r \end{bmatrix} r(k) \\
y(k) &= \begin{bmatrix} C - DK & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + DK_r r(k)
\end{aligned}
$$

For the system above we have the step response (as a servo tracking step simulation) comparison in Figure 54.

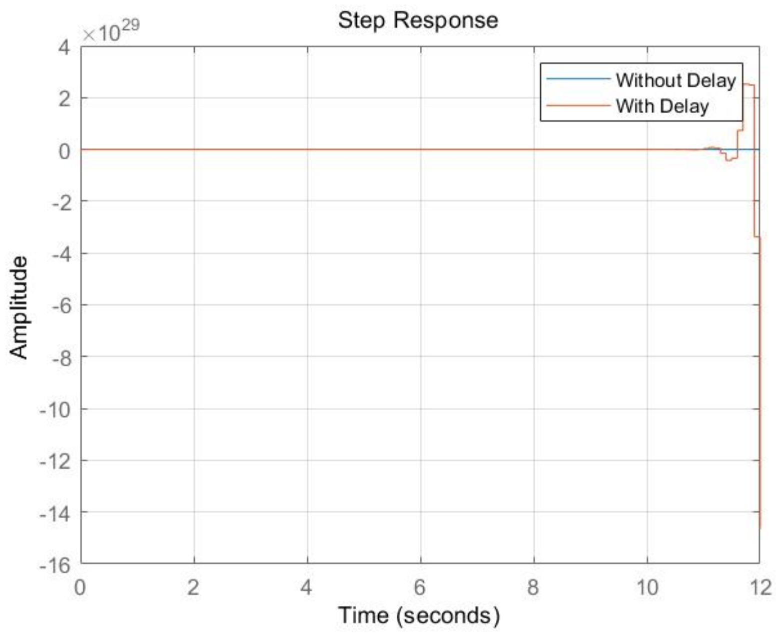

Figure 54: Comparison between with Delay and without Delay

It is clear that the response of the delayed system doesn't converge. It shows the power of the delay. With matlab we have the poles of the delayed system

$$
\begin{aligned}
p_1 &= 0.9692 + 1.5229i \\
p_2 &= 0.9692 - 1.5229i \\
p_3 &= 0.4303 + 0.2423i \\
p_4 &= 0.4303 - 0.2423i
\end{aligned}
$$

of which 2 are placed outside outside the unit circle, indicating instability.

5.1.7 Redesigned PID Controller

Here we retuned the PID to be

$$
\begin{aligned}
C(s) &= \frac{180.9s^2 + 90.3s + 30}{s^2 + 100s} \\
C(z) &= \frac{18.36z^2 - 35z + 16.73}{z^2 - 0.1937z - 0.8063}
\end{aligned}
$$

We accomplished it by re-designing it in continuous-time domain then transforming it into the discrete-time domain. The results for step reference and input step disturbance in Figure 55.

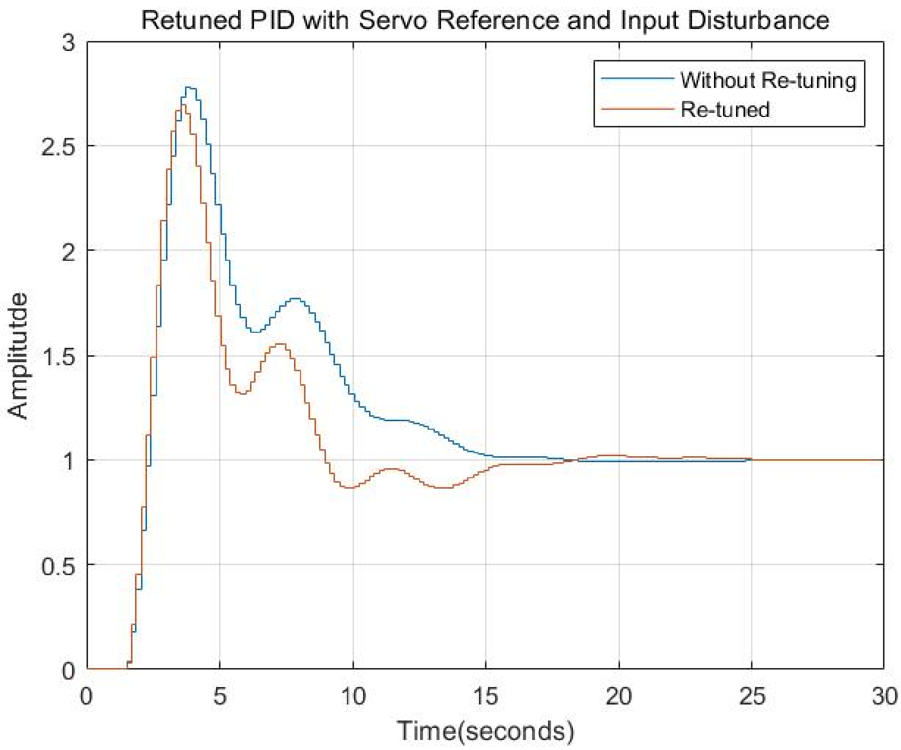

Figure 55: Re-Tuned PID Comparison with Step Reference and Step Input Disturbance

It could be observed that the re-tuned controller has a lower overshoot and a quicker response.

5.1.8 Redesigned Pole-Placement State-Feedback Controller

We redesigned the controller by doing the pole-placement to the fourth-order delayed system, which means now we need to place four poles. Here we design the new four poles at

$$
\begin{aligned}
p_1 &= -4.9175 + 4.9818i \\
p_2 &= -4.9175 - 4.9818i \\
p_3 &= -10.0000 + 0.0000i \\
p_4 &= -12.0000 + 0.0000i \\
|p_1| &= |p_2| = 7
\end{aligned}
$$

and it leads to

$$
\begin{aligned}
z_1 &= 0.7579 + 0.1928i \\
z_2 &= 0.7579 - 0.1928i \\
z_3 &= 0.6065 + 0.0000i \\
z_4 &= 0.5488 + 0.0000i
\end{aligned}
$$

And we have the system response shown in Figure 56.

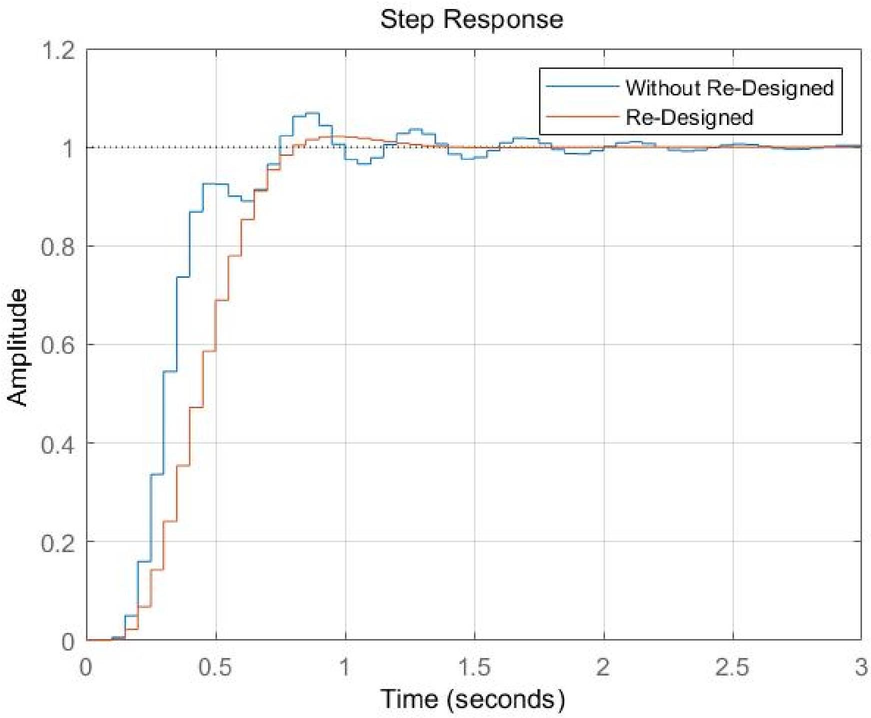

Figure 56: Re-Designed State-feedback Controller Comparison

It is observed the re-design get the delayed system rid of the undesirable fluctuation.

5.1.9 Redesigned LQ Controller

For LQ controller, we redesigned it by tuning the weight for the delayed system. The major difference is now there are four weights for the states instead of three.

We did the same LQ design for the delayed fourth-order system and chose the weight to be

$$
\begin{aligned}
Q_1 &= \begin{bmatrix} 10^3 & 0 & 0 & 0 \\ 0 & 10^3 & 0 & 0 \\ 0 & 0 & 10^6 & 0 \\ 0 & 0 & 0 & 10^4 \end{bmatrix} \\
Q_2 &= 1
\end{aligned}
$$

and the result is shown in Figure 57.

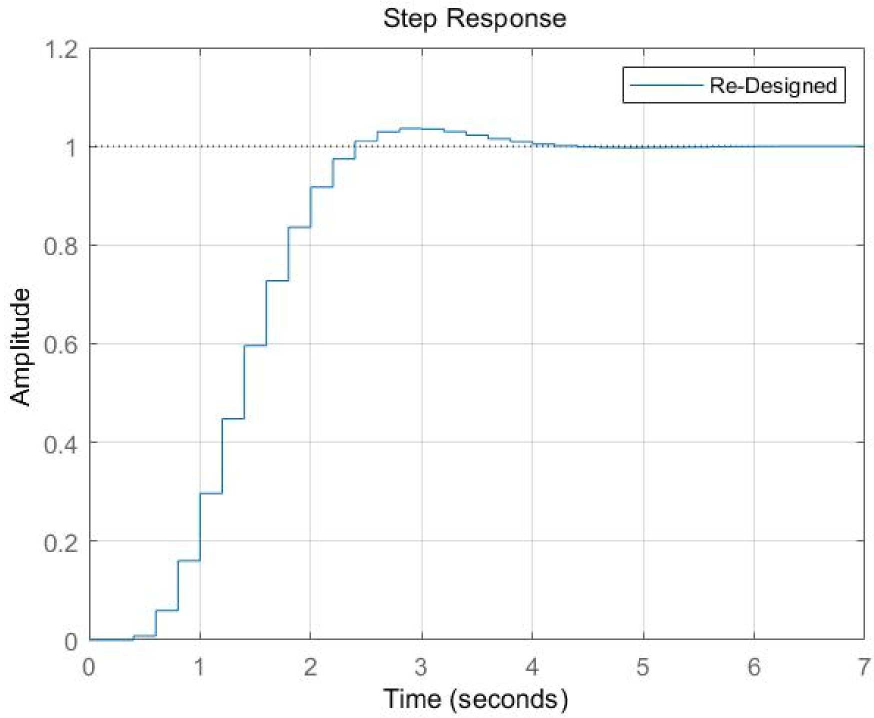

Figure 57: Re-Designed LQ Controller Comparison

The original delayed LQ-controlled system cannot be plotted in the figure since it is unstable. Here we can see that the re-design of the system significantly brings the system back to stable.

6 Conclusion

In this report, many situations are considered to fit into the real engineering problems: delay, different controller, discretization, input limitations, steady error cancellation etc. General good results are obtained by all the design above and could be used to control. However, in practical engineering problem, especially for this robot arm system, sometimes problems regarding safety need to be considered much more when tuning the controller, like the limitations of each states should be also considered in addition to limitations of input due to their physical meaning. Research about stability should be done with more depth too.

Personally the most interesting part is that by modeling the input delay into the state-space model, we could actually re-design the controller for the higher-order new system using regular methods to compensate the influence of the delay, especially about stability. It's quite exciting to know that we could actually do something to counter the delay that exists everywhere in the practical engineering problem.

In conclusion,for the given robot arm system, we are able to design three different controllers with input limitation, zero steady error, one-step delay in discrete-time domain. And they are proved to be performing well.