Simulation Lab

Part 1

Task I.1

1)

World Frame: construct a reference frame. Any frame that is connected to the world frame can be identified with the world frame.

Solver Configuration: specifies parameters which are required to run simulation

Mechanism Configuration: specifies the gravity, its direction and linearization parameters for a mechanism connected to it.

2)

Rigid Transform1 rotates the base frame in +Y direction by 90 degrees

Rigid Transform2 rotates the follower frame Transform1 in +Z direction by 180 degrees

Why do we need them?

If we do not have them, then the direction of mass m1 is based on the world frame instead of the transformed relative frame, thus not making the orientation of mass m1 and forces in the correct direction.

3)

It is necessary to have Rigid Transform 3 because it offsets M2 from M1. If we do not have it, then the centers of M1 and M2 are overlapped and the two masses are into each other, making the system unrealistic.

4)

Springs and dampers are specified by the prismatic joint and prismatic joint2 parameters. We cannot see springs and dampers in the simulation as illustrated in the figure below. To change the parameters, double click the 'prismatic joint' block then inside **'Z Prismatic Primitive'** -> **'Internal Mechanics'** and set the parameters in the blocks.

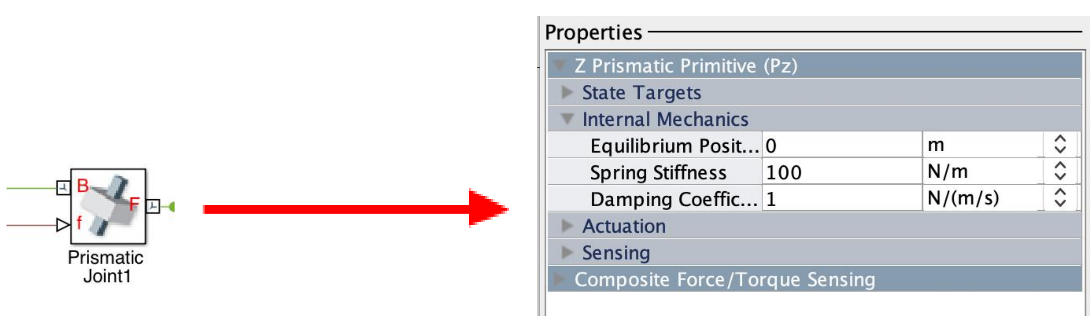


Figure 1.1.4: How to set spring stiffness and damping coefficient inside the prismatic Joint block

5)

The mass properties are inside the solid M1 and M2 block. To change them, just double click the block and we can set parameters for example geometry, inertia, frame, color and opacity.

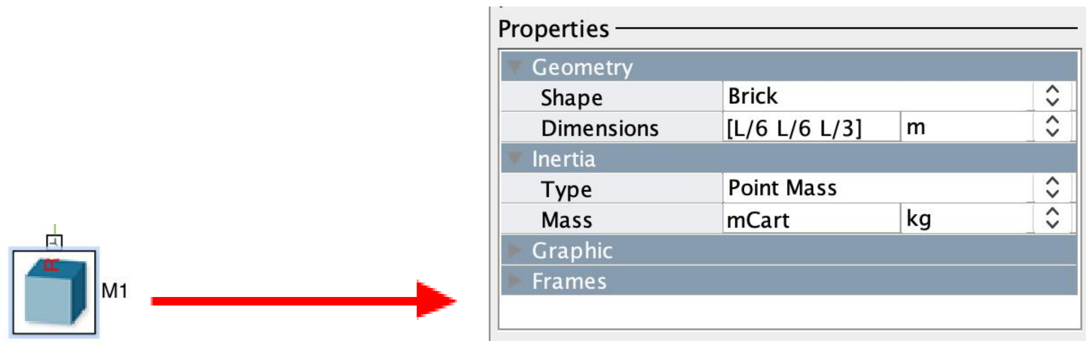


Figure 1.1.5: How to set mass properties: shape, size, mass, color, frame etc.

Task I.2

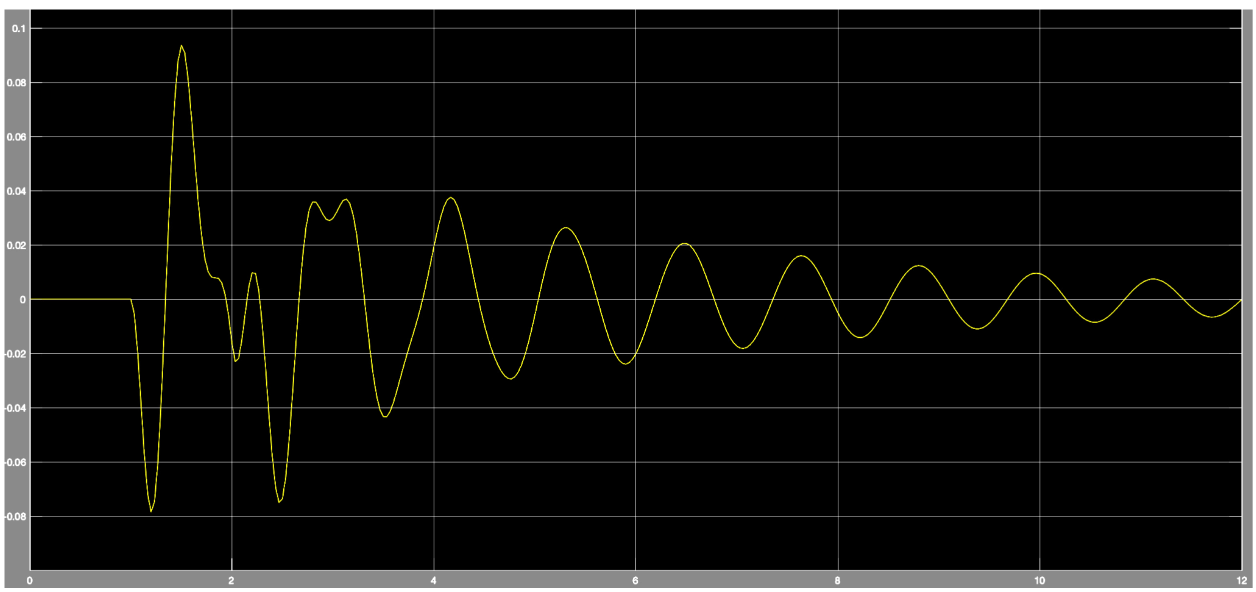


Figure 1.2: system response graph for a 10 unit-step input

Task I.3

The damped spring block connects to the same reference frame as the prismatic joint at one end and connects to the other end with mass M2. Therefore, we can modify the schematic figure by having only one spring and damper connecting between the wall and M2.

Part 2

Task II.1 & Task II.2

Task II.1

obtain system's equations:

$$m_1\ddot{y}_1 + k_1 y_1 + b_1\dot{y}_1 + k_2(y_1 - y_2) + b_2(\dot{y}_1 - \dot{y}_2) = u \quad \text{---} (1)$$

$$m_2\ddot{y}_2 + k_2(y_2 - y_1) + b_2(\dot{y}_2 - \dot{y}_1) = 0 \quad \text{---} (2)$$

choose state variables $\rightarrow y_1, \dot{y}_1, y_2, \dot{y}_2$

arrange the system's equations in state-space form, $\dot{\underline{x}} = A\underline{x} + Bu$

$$\underline{y} = \underline{C}\underline{x}$$

$$\begin{bmatrix} \dot{y}_1 \\ \ddot{y}_1 \\ \dot{y}_2 \\ \ddot{y}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -(k_1+k_2)/M_1 & \frac{-(b_1+b_2)}{m_1} & \frac{k_2}{M_1} & \frac{b_2}{M_1} \\ 0 & 0 & 0 & 1 \\ \frac{k_2}{M_2} & \frac{b_2}{M_2} & -\frac{k_2}{M_2} & -\frac{b_2}{M_2} \end{bmatrix} \begin{bmatrix} y_1 \\ \dot{y}_1 \\ y_2 \\ \dot{y}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/M_1 \\ 0 \\ 0 \end{bmatrix} u_1$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ \dot{y}_1 \\ y_2 \\ \dot{y}_2 \end{bmatrix}$$

Task II.2

$k_1 = 100$ N/m
$b_1 = 1$ N/(m/s) } obtained from prismatic joint 1 block
$m_1 = $ mCart kg $\rightarrow$ from M_1 block

$k_2 = 50$ N/m
$b_2 = 1$ N/(m/s) } obtained from prismatic joint 2 block
$m_2 = $ mCart kg $\rightarrow$ from M_2 block

Note that the mass of M1 and M2 is assigned to be ‘mCart’. This variable can be found in Matlab Workspace. Its value is 1 kg.

Task II.3

The response signal for the system obtained by Simscape as in figure 2.3.1.

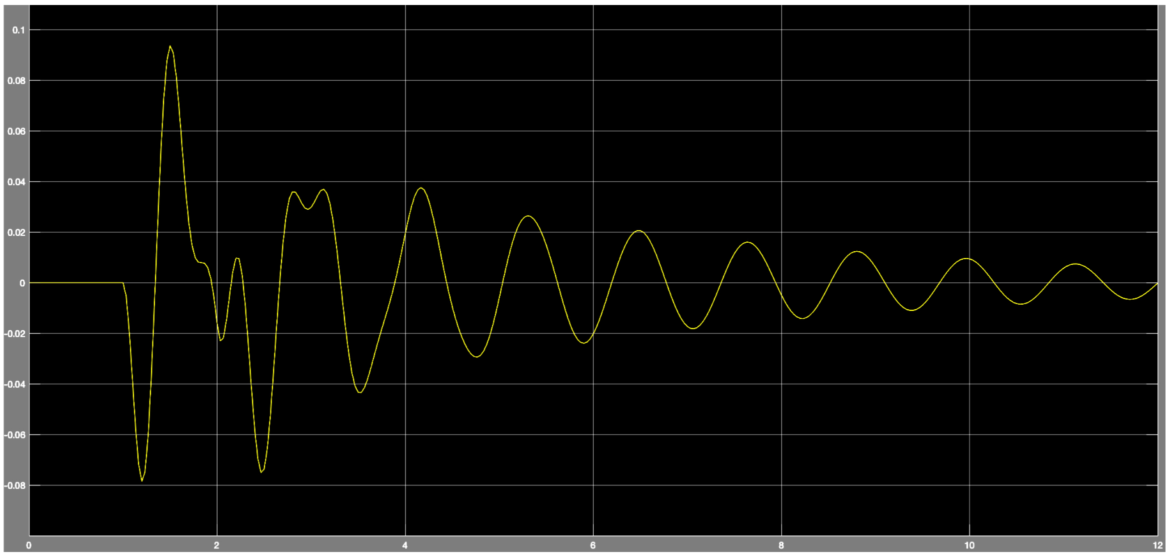


Figure 2.3.1: system response graph obtained from Simscape model

The output response is the relative displacement between y2 and y1. The initial reaction of the displacement decreases very quickly due to the input force acting directly on the mass m1 for only a short period(0-4 seconds), but the force does not transfer energy to m2. Between this interval, the erratic response shows that both movements of m1 and m2 are fighting against each other. However, after 4 seconds, the two carts begin to move in sync, thus making the response smooth.

In figure 2.3.2, the system response derived from state-space representation.

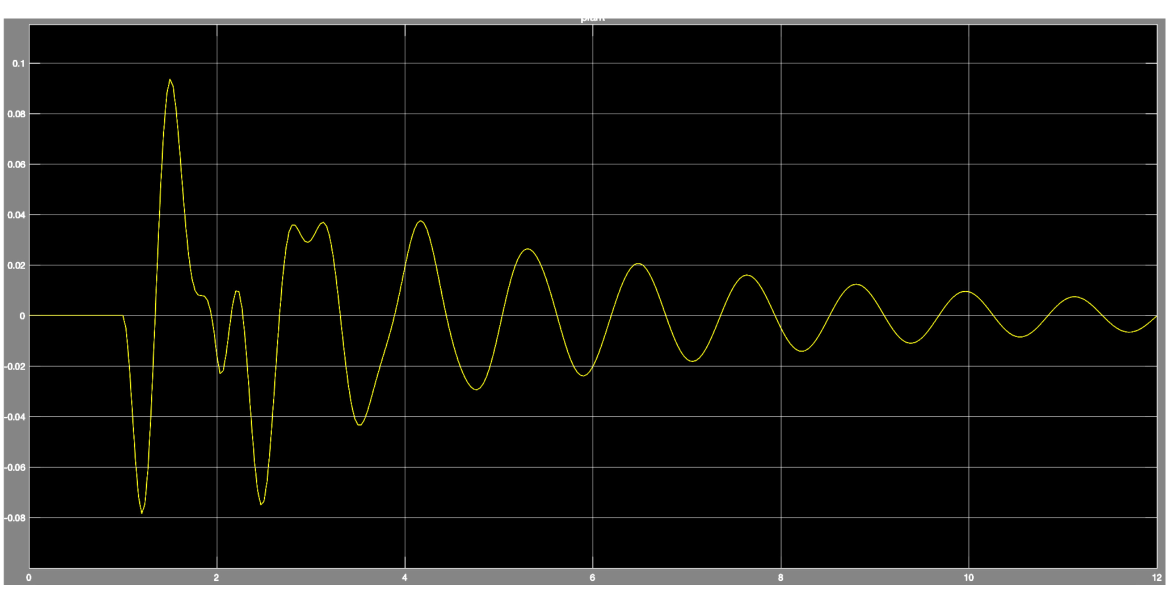


Figure 2.3.2: system response graph obtained from state-space model

Comparing the two graphs, we can verify that the two response signals are the same.

Task III.1

Task III.1

consider the system equation on M_1;

$$m_1\ddot{y}_1 + k_1 y_1 + b_1\dot{y}_1 = u(t) + v(t) \quad —(1)$$

$$u(t) = K_p(r - y_1) + K_d(\dot{r} - \dot{y}_1) \quad —(2)$$

substitute (2) in (1); $m_1\ddot{y}_1 + k_1 y_1 + b_1\dot{y}_1 = K_p(r - y_1) + K_d(\dot{r} - \dot{y}_1) + v(t)$

$$m_1\ddot{y}_1 + (K_d + b_1)\dot{y}_1 + (K_p + k_1)y_1 = \cancel{K_d\dot{r}}^{\,0} + K_p r + v(t)$$

Consider transient response

$$m_1\ddot{y}_1 + (K_d + b_1)\dot{y}_1 + (K_p + k_1)y_1 = 0$$

$$\downarrow$$

$$\ddot{y}_1 + 2\xi\omega_n\dot{y}_1 + \omega_n^2 y_1 = 0$$

Critically damped $\rightarrow \xi = 1$

$$\frac{K_d + b_1}{M_1} = 2\xi\omega_n \quad —(3)$$

$$\frac{K_p + k_1}{m_1} = \omega_n^2 \quad —(4)$$

Therefore, changing k_p would affect k_d.

Choose k_p so that $e_{ss} = 10\%\, r = 0.1(\cancel{0.01}^{\,0.1}) = 0.01$

$$e_{ss} = \lim_{s\to 0}\left(1 - \frac{K_p}{m_1 s^2 + (K_d + b_1)s + (K_p + k_1)}\right)\cancel{s} \times \frac{0.1}{\cancel{s}}$$

$$e_{ss} = \lim_{s\to 0} \frac{0.1 \times K_1}{m_1 s^2 + (K_d + b_1)s + (K_p + k_1)}$$

$$0.01 = \frac{0.1K_1}{K_p + K_1} \quad \leftarrow \text{substitute } K_1 = 100$$

$$K_p = 900$$

find K_d according to $K_p = 900$

$$K_d = 2\sqrt{1000m_1} - 1$$

The result from hand derivation shows that for the step reference signal(r = 0.1), we can find Kp and Kd so that the steady-state error is 0.01(10% of the input signal). To verify the PD controller model in Simulink, the system is set up as in figure 3.1.1. Note that we assume disturbance to be zero in this model so the error will be slightly different than the real system.

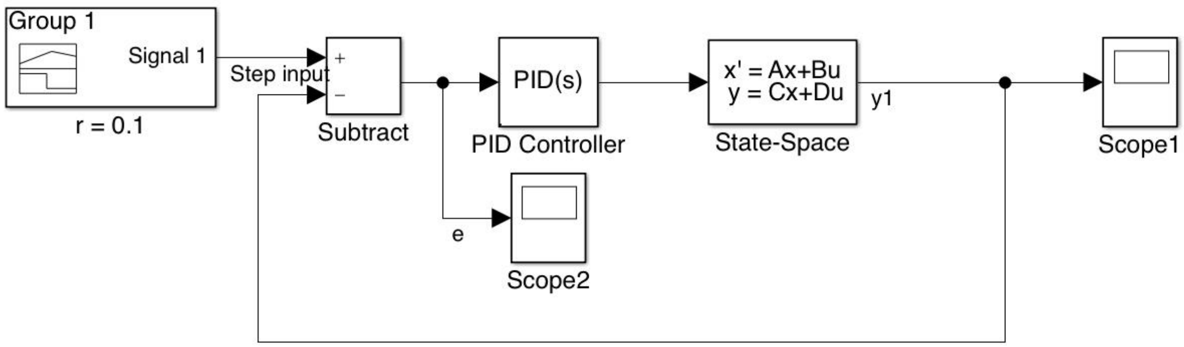


Figure 3.1.1: Closed-loop system for PD/PID controller

Figure 3.1.2 demonstrates that the error plot for a 100 seconds period is converged to 0.01.

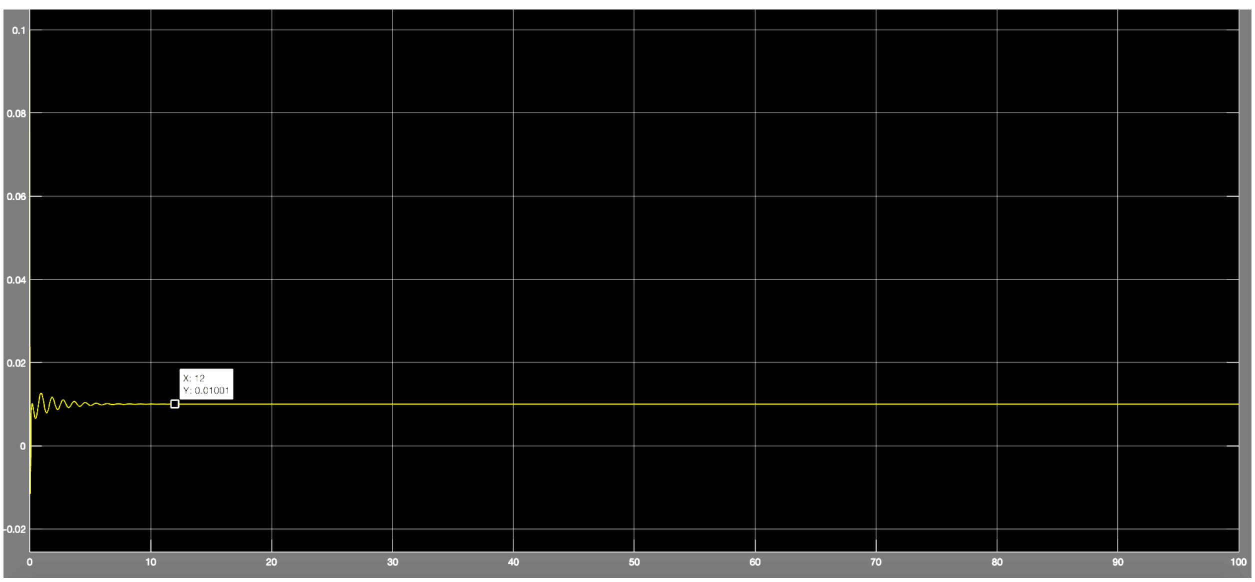


Figure 3.1.2: Error response of PD controller with Kp = 900, Kd = 62.245 for 100 seconds

Task III.2

Task III.2

according to the system equation

$$m_1\ddot{y}_1+(K_d+b_1)\dot{y}_1+(K_p+K_1)y_1 = K_d\dot{r}+K_pr+\cancel{v(t)}^{\,0}$$

$$\downarrow$$

$$(m_1s^2+(K_d+b_1)s+(K_p+K_1))Y_1(s)=(K_p+sK_d)R(s)$$

$$\frac{Y_1(s)}{R(s)} = \frac{K_p+sK_d}{m_1s^2+(K_d+b_1)s+(K_p+K_1)}$$

$$E(s) = R(s)-Y_1(s)$$

$$E(s) = \left(1 - \frac{K_p+sK_d}{m_1s^2+(K_d+b_1)s+(K_p+K_1)}\right)\cancel{R(s)} \quad R(s)=\frac{0.1}{s^2}$$

$$\lim_{s\to 0} E(s)\cdot s = \lim_{s\to 0}\left[\left(1 - \frac{(K_p+sK_d)}{m_1s^2+(K_d+b_1)s+(K_p+K_1)}\right)s\ast\frac{0.1}{\cancel{s^2}\,s}\right]$$

$$= \frac{0.1\,(\cancel{m_1s^2}^{\,0}+\cancel{(K_d+b_1)s}^{\,0}+\cancel{(K_p}+K_1)-\cancel{(K_p+sK_d)}^{\,0})}{s(m_1s^2+(K_d+b_1)s+(K_p+K_1))}$$

$$\lim_{s\to 0} E(s)\cdot s = \infty$$

The steady state error is ∞ (Ans)

consider the system equation

$$m_1\ddot{y}_1+k_1y_1+b_1\dot{y}_1 = u(t)+v(t) \quad -(1)$$

$$u(t) = K_p(r-y_1)+K_d(\dot{r}-\dot{y}_1)+K_i\int r\,dt-K_i\int y_1\,dt \quad -(2)$$

substitute (2) in (1);

$$m_1\ddot{y}_1+(K_d+b_1)\dot{y}_1+(K_p+K_1)y_1+K_i\int y_1\,dt = K_d\dot{r}+K_pr+K_i\int r\,dt+\cancel{v(t)}^{\,0}$$

$$\downarrow$$

$$\left(m_1s^2+(K_d+b_1)s+K_p+K_1+\frac{K_i}{s}\right)Y_1(s) = \left(sK_d+K_p+\frac{K_i}{s}\right)R(s)$$

$$\therefore\ Y_1(s) = \frac{\left(K_p+sK_d+\frac{K_i}{s}\right)}{m_1s^2+(K_d+b_1)s+K_p+K_1+\frac{K_i}{s}}\ast R(s)$$

$$E(s) = R(s)-Y_1(s) = \left(1 - \frac{K_p+sK_d+\frac{K_i}{s}}{m_1s^2+(K_d+b_1)s+K_p+K_1+\frac{K_i}{s}}\right)R(s)$$

$$e_{ss} = \lim_{s\to 0} sE(s)$$

$$= \lim_{s\to 0} s\left(1 - \frac{K_p + sK_d + K_i/s}{m_1 s^2 + (K_d + b_1)s + K_p + K_1 + K_i/s}\right)\cancel{R(s)} \qquad R(s) = \frac{0.1}{s^2}$$

$$= \lim_{s\to 0} \frac{0.1}{s}\left(\frac{m_1 s^2 + \cancel{K_d} + b_1)s + \cancel{K_p} + K_1 + \cancel{K_i/s} - \cancel{K_p} - \cancel{sK_d} - \cancel{K_i/s}}{m_1 s^2 + (K_d + b_1)s + K_p + K_1 + K_i/s}\right)$$

$$= \lim_{s\to 0} \frac{0.1(m_1 s^2 + b_1 s + \cancel{K_p} + K_1)}{m_1 s^3 + (K_d + b_1)s^2 + (K_p + K_1)s + K_i}$$

$$e_{ss} = \frac{0.1 K_1}{K_i} = \frac{10}{K_i}$$

If we choose $K_i = 1000 \rightarrow e_{ss} = 0.01$

Verify that using ramp input $r(t) = 0.1t \rightarrow e_{ss} = 0.01$

The hand derivation shows that the steady state error for input, r = 0.1t, is infinity when using the same PD gains. The derivation also explains that adding integral control to the system will reduce steady state error down to such a finite value that is a factor of integral gain, Ki. For example, when Ki is 1000, the error would be reduced to 0.01. Firstly, we set up the system as in the figure 3.2.1.

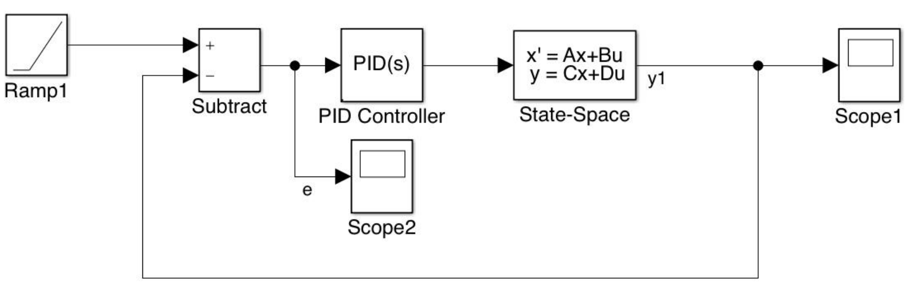


Figure 3.2.1: Error response of PD controller with Kp = 900, Kd = 62.245 for 100 seconds

The error plot when using PD gains with a ramp input is in figure 3.2.2. This verifies the derivation that the error diverges to infinity.

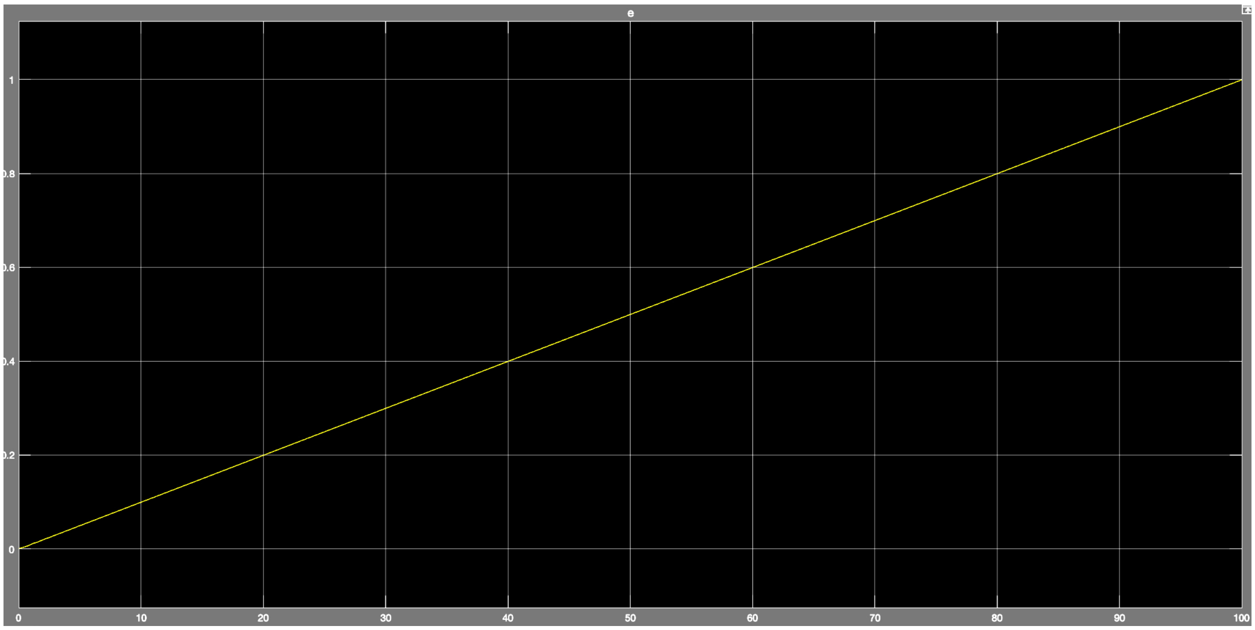


Figure 3.2.2: Error response of the same PD controller with a ramp input

However, using the PID controller the error can be limited to a finite value as illustrated in the following figures which illustrate the error behavior for varied Ki.

Ki = 1000

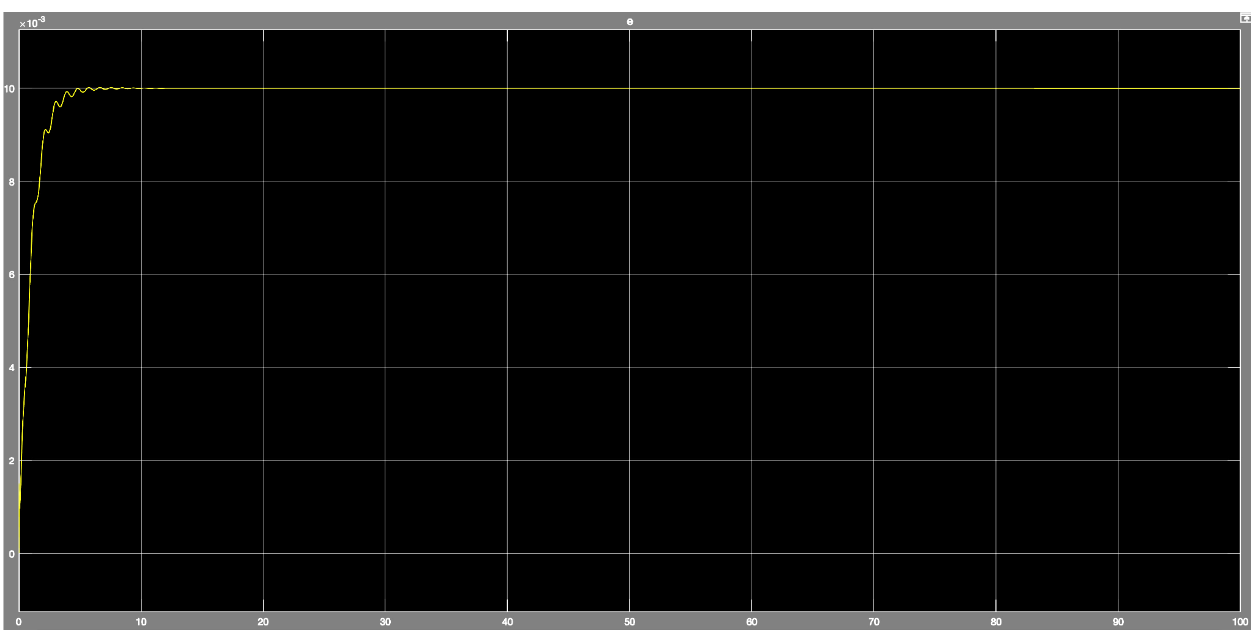


Figure 3.2.3: Error response of PID controller with a ramp input and Ki = 1000

Ki = 500

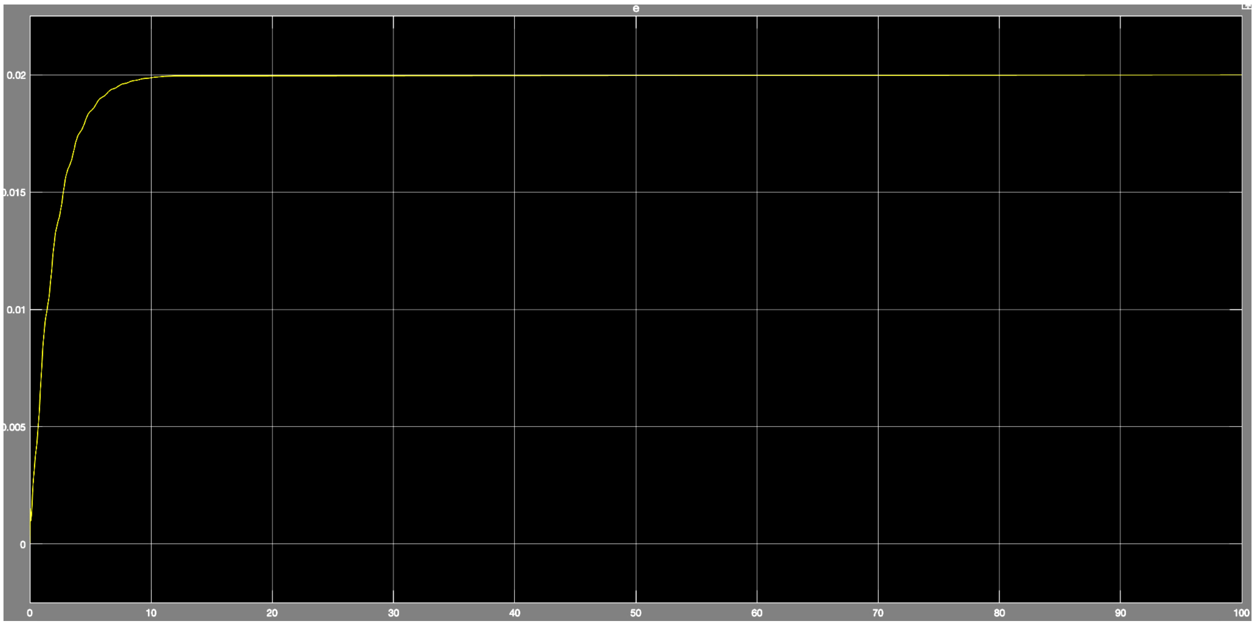


Figure 3.2.4: Error response of PID controller with a ramp input and Ki = 500

Ki = 200

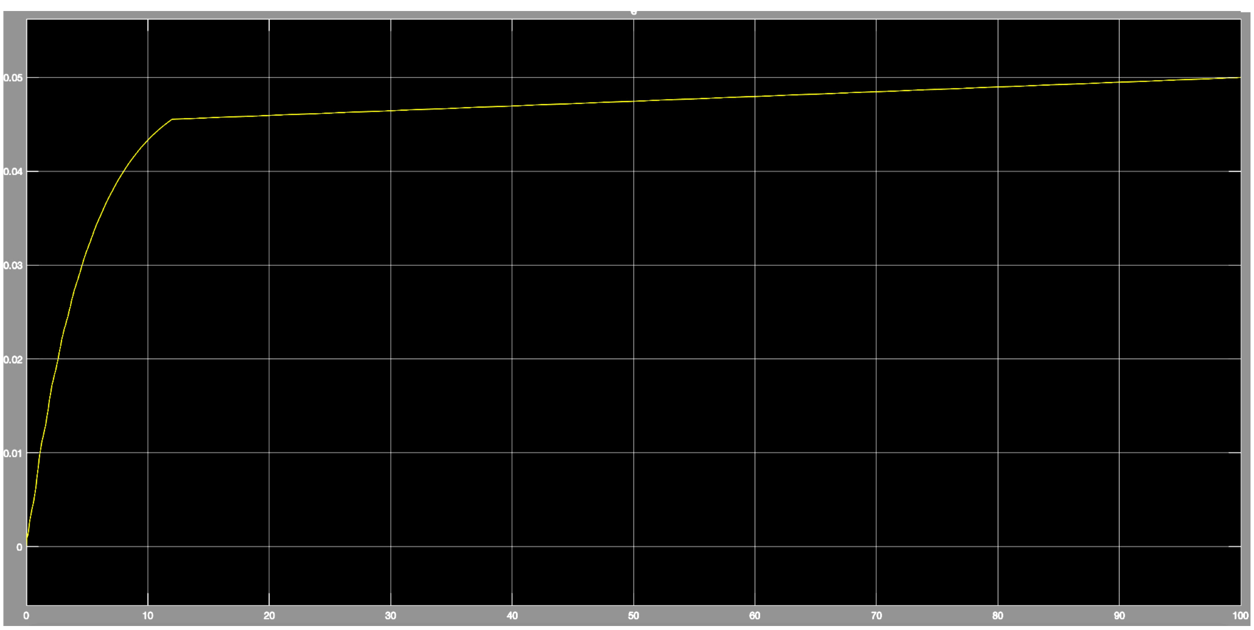


Figure 3.2.5: Error response of PID controller with a ramp input and Ki = 200

Ki = 100

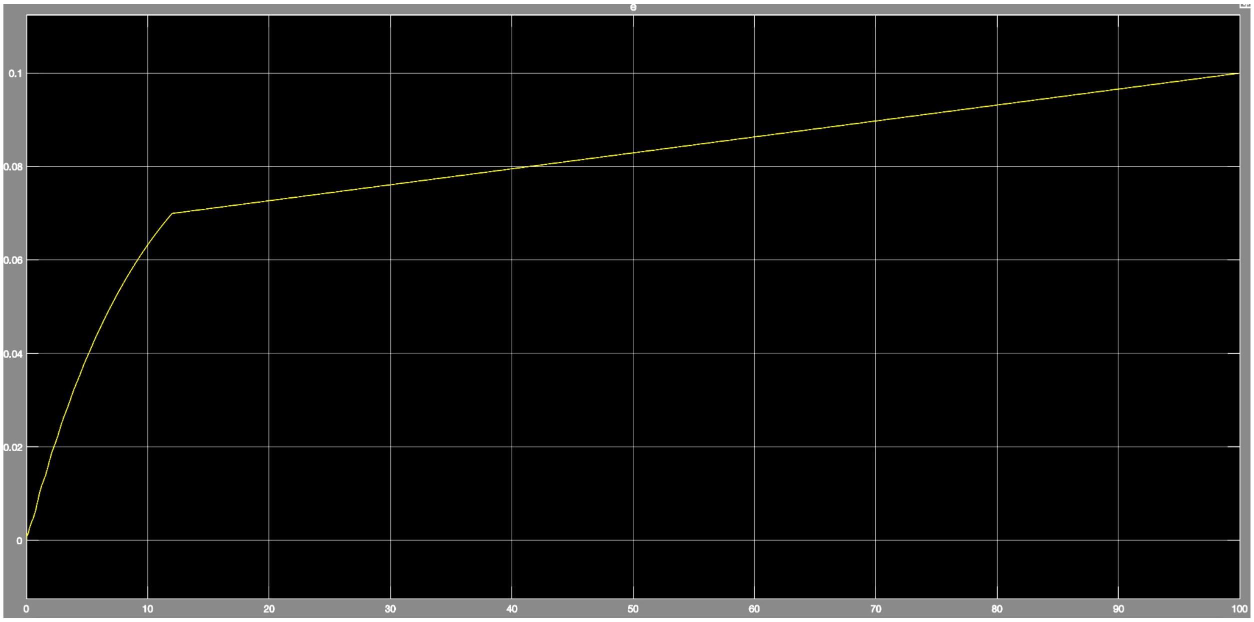


Figure 3.2.6: Error response of PID controller with a ramp input and Ki = 100

From the sample values of Ki, increasing Ki will decrease the steady state error and make the system converge to an asymptote more quickly.

However, keeping increasing Ki for a large value will make the system underdamped, for example, when Ki = 10000 as shown in figure 3.2.7.

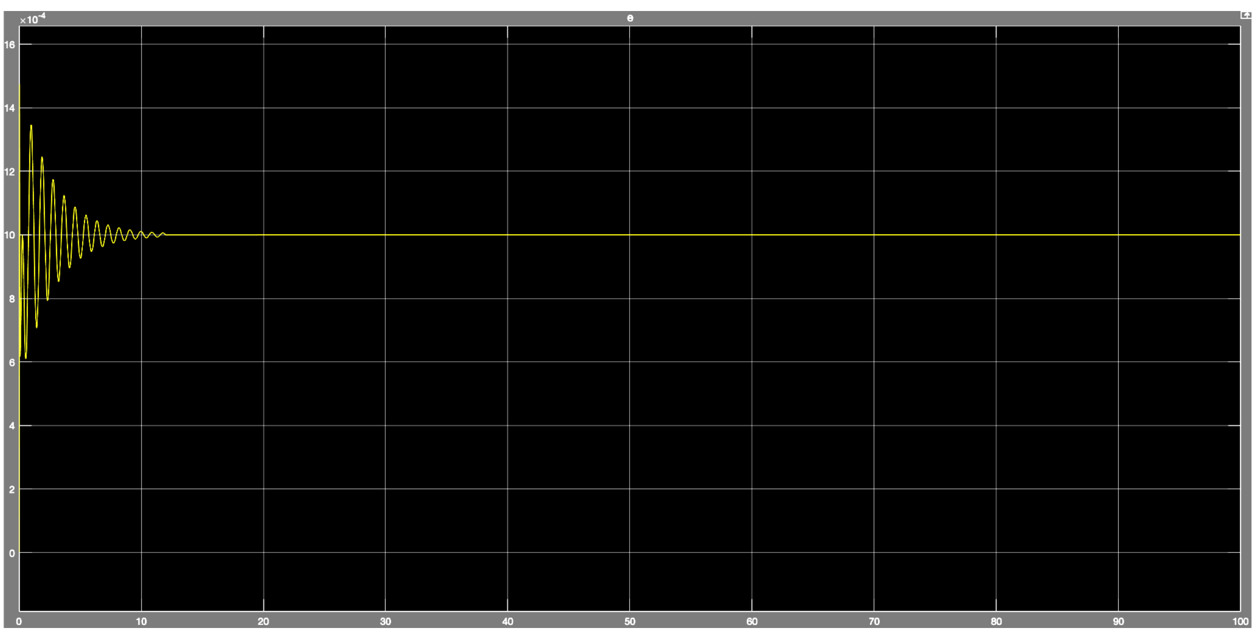


Figure 3.2.7: Error response of PID controller with a ramp input and Ki = 10000. The transient response is underdamped.

To check the damping ratio, we find the transfer function of the closed-loop system using Matlab.

```
clc;clear;
Kd = 62.245; Kp = 900; Ki = 1000;
num = [Kd, Kp, Ki];
den = [1, (Kd+1), (Kp+100), Ki];
sys = tf(num,den); %closed-loop tf
```

Command Window

```
>> sys

sys =

     62.24 s^2 + 900 s + 1000
  -------------------------------
  s^3 + 63.24 s^2 + 1000 s + 1000

Continuous-time transfer function.
```

Figure 3.2.8: Matlab script showing the system’s closed-loop transfer function

The system is using Ki = 1000. The following figure shows closed-loop poles of the system.

```
>> damp(sys)

   Pole          Damping          Frequency       Time Constant
                                (rad/seconds)       (seconds)

 -1.07e+00       1.00e+00         1.07e+00          9.33e-01
 -2.53e+01       1.00e+00         2.53e+01          3.95e-02
 -3.68e+01       1.00e+00         3.68e+01          2.72e-02
```

Figure 3.2.9: Demonstration for poles and damping ratio of the closed-loop system when Ki = 1000

In figure 3.2.9, the damping ratio is 1(critically damped). Then we will adjust Ki to see how the damping ratio and its corresponding poles change.

Ki = 100,

```
>> damp(sys)

   Pole        Damping       Frequency      Time Constant
                           (rad/seconds)      (seconds)

 -1.01e-01     1.00e+00       1.01e-01         9.94e+00
 -2.98e+01     1.00e+00       2.98e+01         3.36e-02
 -3.33e+01     1.00e+00       3.33e+01         3.00e-02
```

Figure 3.2.10: Demonstration for poles and damping ratio of the closed-loop system when Ki = 100

Ki = 10000,

```
>> damp(sys)

         Pole              Damping       Frequency      Time Constant
                                       (rad/seconds)      (seconds)

 -8.46e+00 + 1.20e+01i     5.76e-01       1.47e+01         1.18e-01
 -8.46e+00 - 1.20e+01i     5.76e-01       1.47e+01         1.18e-01
 -4.63e+01                 1.00e+00       4.63e+01         2.16e-02
```

Figure 3.2.11: Demonstration for poles and damping ratio of the closed-loop system when Ki = 10000

According to the results above, adjusting Ki to a small value will not change the damping ratio. In other words, the damping ratio is equal to 1. However, as Ki increases the damping ratio is reduced, thus making the system underdamped. This is because new poles contain imaginary parts which cause the system to oscillate.

Task III.3

We set up the system as in the figure 3.3.1. The errors behavior for PD and PID system are shown in the table

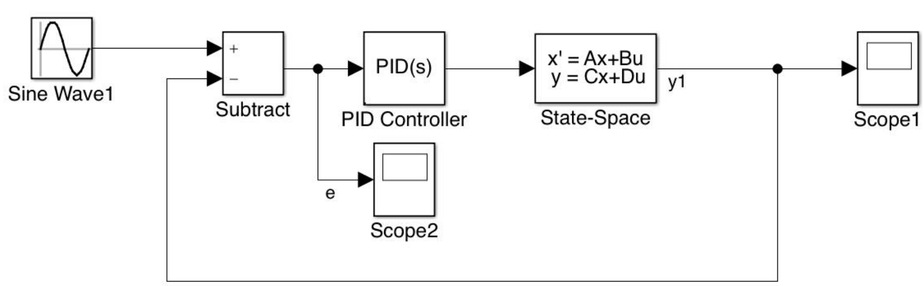
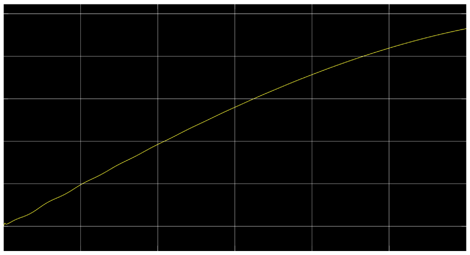
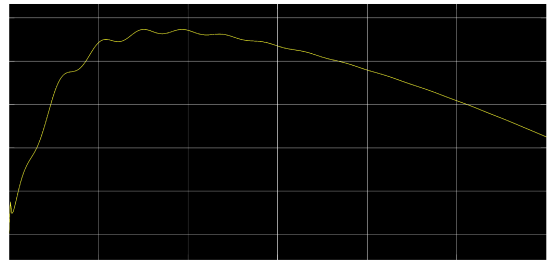
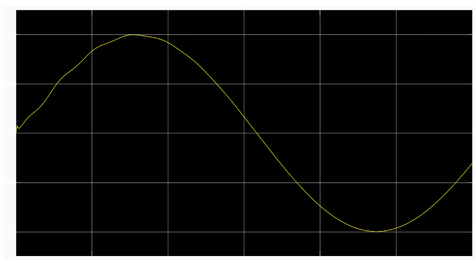
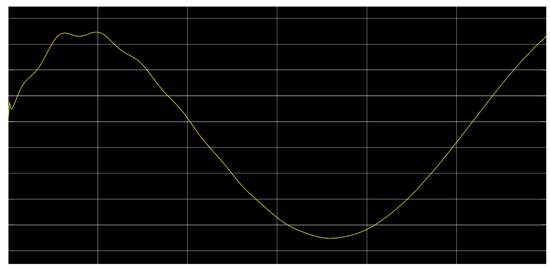
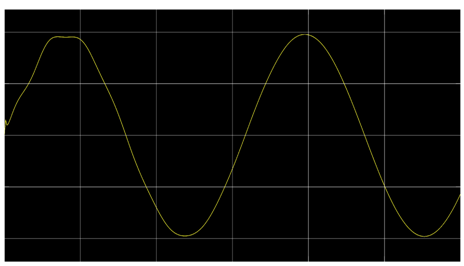
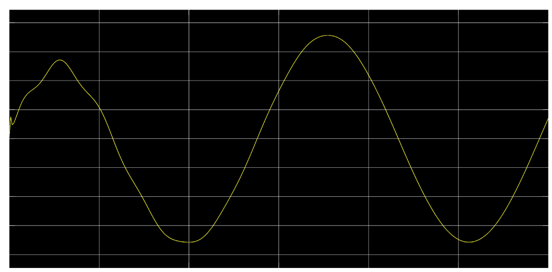
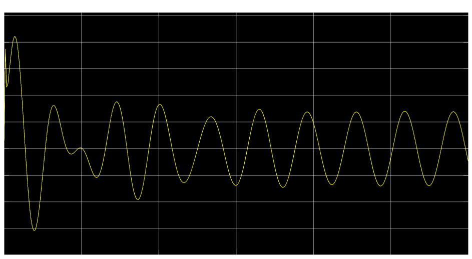
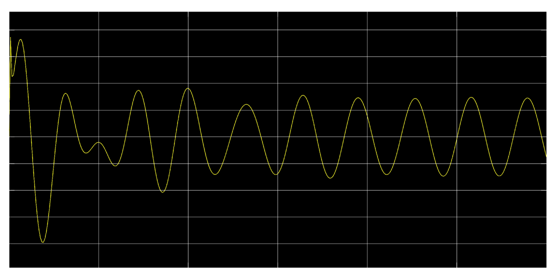
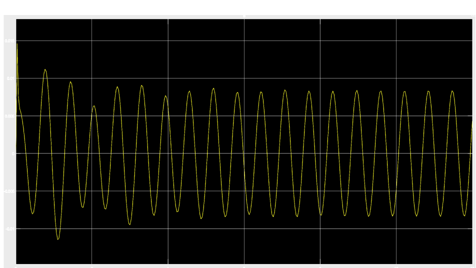
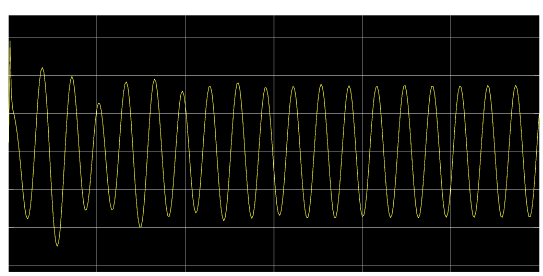


Figure 3.3.1: Closed-loop system with sine wave input

Input Frequency (Hz)	Error from PD controller (12 secs period)	Error from PID controller (12 secs period)
0.1		
0.5		
1		
5		
10		

From the observation in the table, the error tracements from PD and PID controllers are very similar. This is because the input signal is sinusoidal so the controllers cannot track the error properly. Moreover, increasing frequency will also increase the error magnitude.

The Bode plot of the system closed-loop transfer function is shown below.

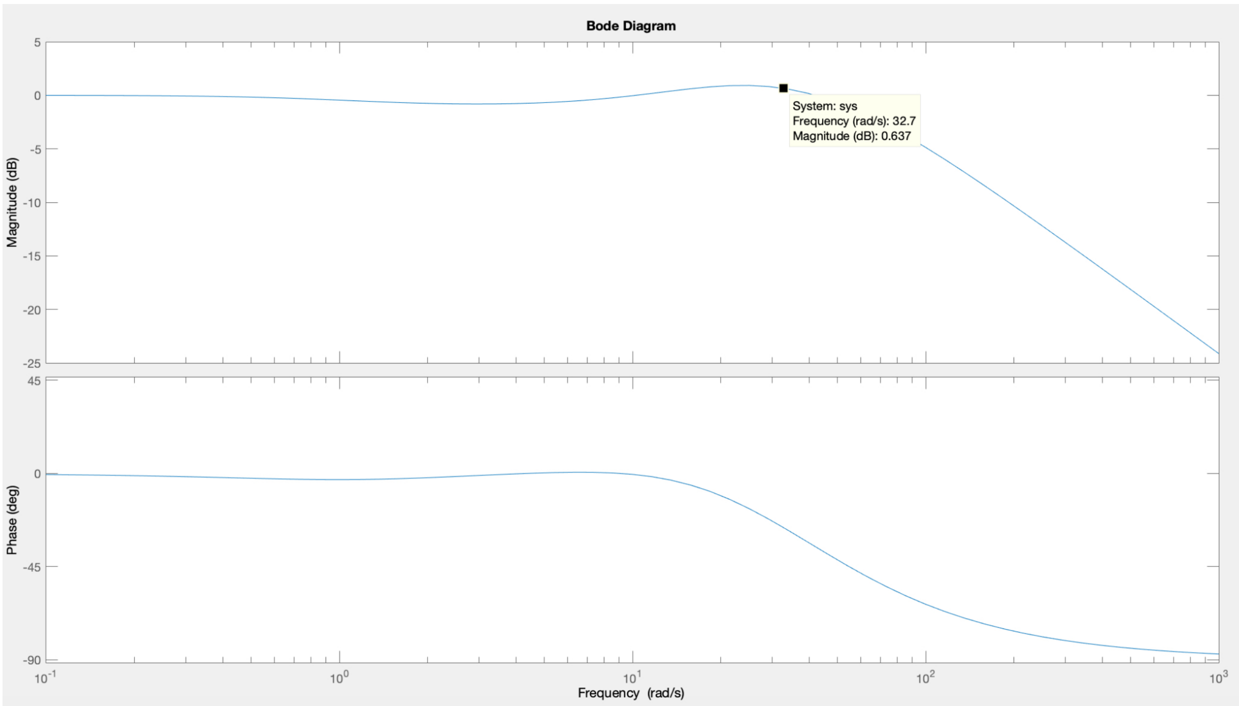


Figure 3.3.2: Bode plot for the closed loop system

From figure 3.3.2, the cutoff frequency is about 30 rad/s which is equivalent to 5 Hz approximately. This is the transition from the vibrations affecting the system to be too quick to properly affect it.

Part IV

First, set up the state-feedback controller system.

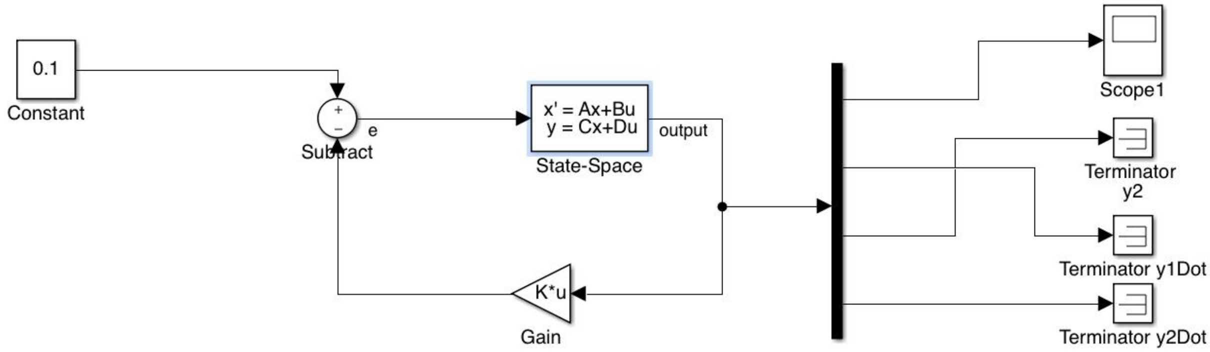


Figure 4.0.1: Closed-loop system with state-feedback controller

Task IV.1

Select poles according to the following requirements

1) **Damping ratio is greater than 0.707**

This means that the selected poles should have less than -45 degree placed on the left-half of s plane. In other words, its real part's magnitude should be larger than its imaginary part.

2) **The imaginary parts are lower than 70**

Because the imaginary part comes in pairs, one on the upper plane and the other on the lower plane. Therefore, these two imaginary parts' magnitude must be lower than 70.

3) **The settling time of each pole is smaller than 0.1 seconds**

To set the settling time of each pole lower than 0.1 seconds, the real components must be lower than 40.

Select poles based on the requirements

s1,2 = -50+-28.7i (Zeta = 0.866, Wn = 57.7)
s3,4 = -60+-21.9i (Zeta = 0.939, Wn = 63.8)

Compute K according to the desired poles using Matlab place() method. *Let kr = 1 in this case*

K = [8490 220 262600 10490], kr = 1

```
mCart = 1;
A = [0 1 0 0; -150/mCart -2/mCart 50/mCart 1/mCart;...
    0 0 0 1;50/mCart 1/mCart -50/mCart -1/mCart];
B = [0;1/mCart;0;0];
C = [1 0 0  0];
D = 0;

poles = [-50+28.7i -50-28.7i -60+21.9i -60-21.9i];
K = place(A,B,poles);
```

Figure 4.1.1: Matlab command for pole placement

Figure 4.1.2: system response for output y1 in 12 seconds interval

According to Figure 4.1.2, the settling time is 0.16 seconds, higher than 0.1. This is because the disturbance caused by the second cart. However, the response comes to a steady state very fast compared to part III and has a very high overshoot.

Task IV.2

The output y2 can be measured and compared to y1 using the same K and system's poles.

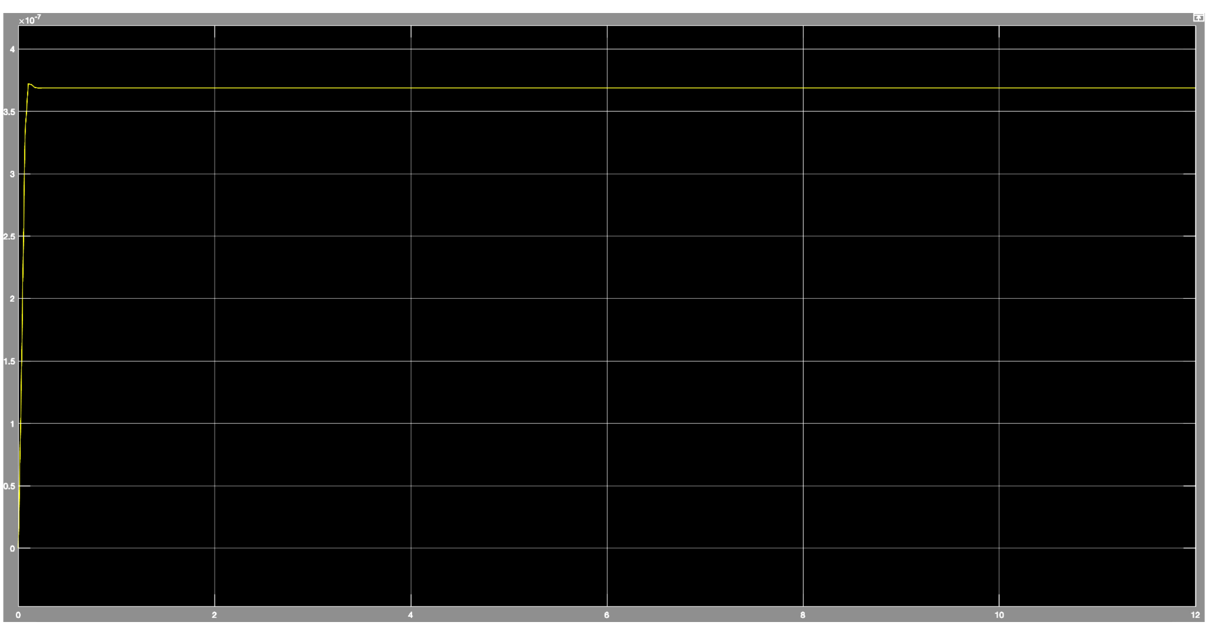


Figure 4.1.3: system response for output y2 in 12 seconds interval

Output y2 is different from y1 in that unlike y1, it has only a small overshoot and it is less sensitive to the input than y1. Also the steady state value is smaller than y1, supporting that it has less disturbance than y1.