Final Project Report

Magnetic Levitation System

Prepared for:

Prof. Khorasani

Department of Aerospace

University of Concordia

Submitted by:

Sourena Morteza Ghasemi (DEPARTMENT OF MECHANICAL ENGINEERING**)**
(40171622)

Mona Arjmandi (DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**)**
(40231290)

Course:
ENGR 6131

Fall 2022

Abstract

This report presents controllers for a magnetic levitation system this system is described with a nonlinear equation. After linearization, controllers use to improve system performance. To investigate this project, we used MATLAB software.

Figure1- Magnetic Levitation System

Contents

1. Introduction:

In this project, we want to consider Magnetic Levitation System as a system that can reduce Friction between moving surfaces and can be reduced substantially using the magnetic levitation principle. Many industrial applications have been found to have used magnetic levitation in order to increase the overall efficiency of the system. For instance, Kaplan and Regev (1976) utilized the magnetic levitation principle for high-speed train suspension. The design of magnetic bearings can be found in superconductor rotor suspension of gyroscopes by Bencze, and wind turbines have been reported by Kumbernuss. The magnetic levitation principle has been discussed by Kaloust for launch assistance in space missions. [1]

To less Friction and satisfy industrial goals, controllers for such a system are designed primarily to maintain the levitated object at the desired height. In this regard, we used the Jacobian formula for converting nonlinear equations to linear to investigate several items such as input-output equations, transfer function, controllable, observable, and Jordan canonical forms, Bode plot, PID controller, etc.

2. Magnetic Levitation System Dynamics and Modeling

The magnetic levitation system consists of two electromagnetic coils and position sensors. The electromagnetic forces are varying, and magnet disks move on the horizontal axis. If the disk distance from the electromagnetic coil crosses the specified value, then the falling chances of disks increase. In figure 1 the electromagnetic levitation system with two disks is shown.

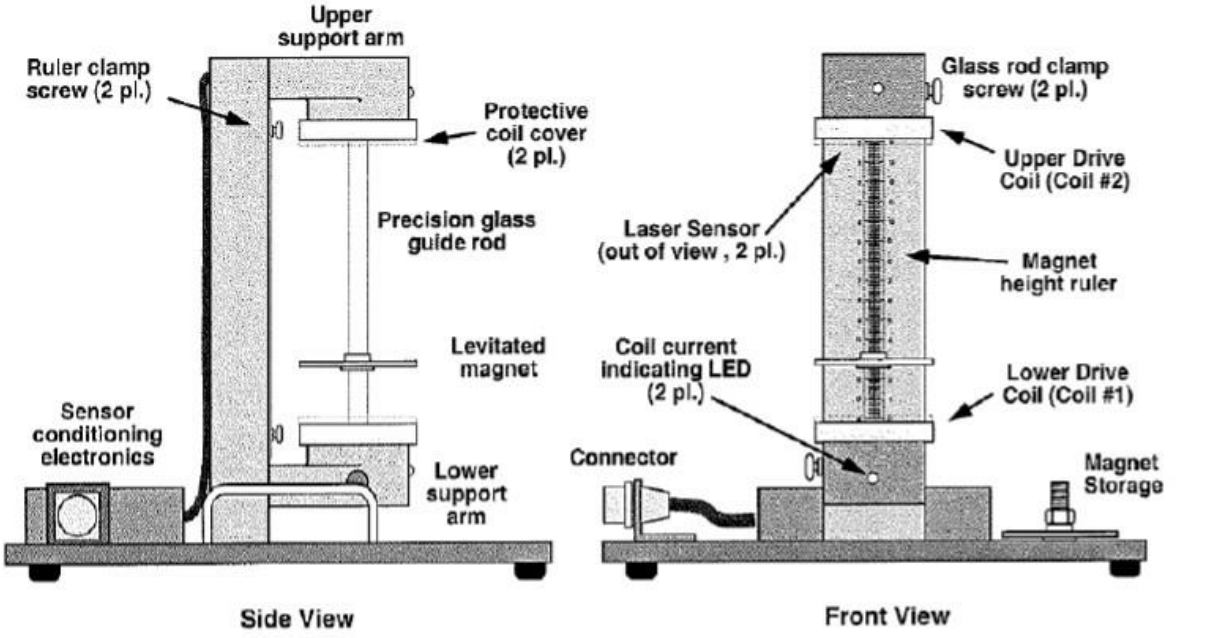

Figure 2- magnetic levitation system

Figure 2 represents that either magnet is acted on by forces from either drive coil, from the other magnet, from gravity, and from friction.

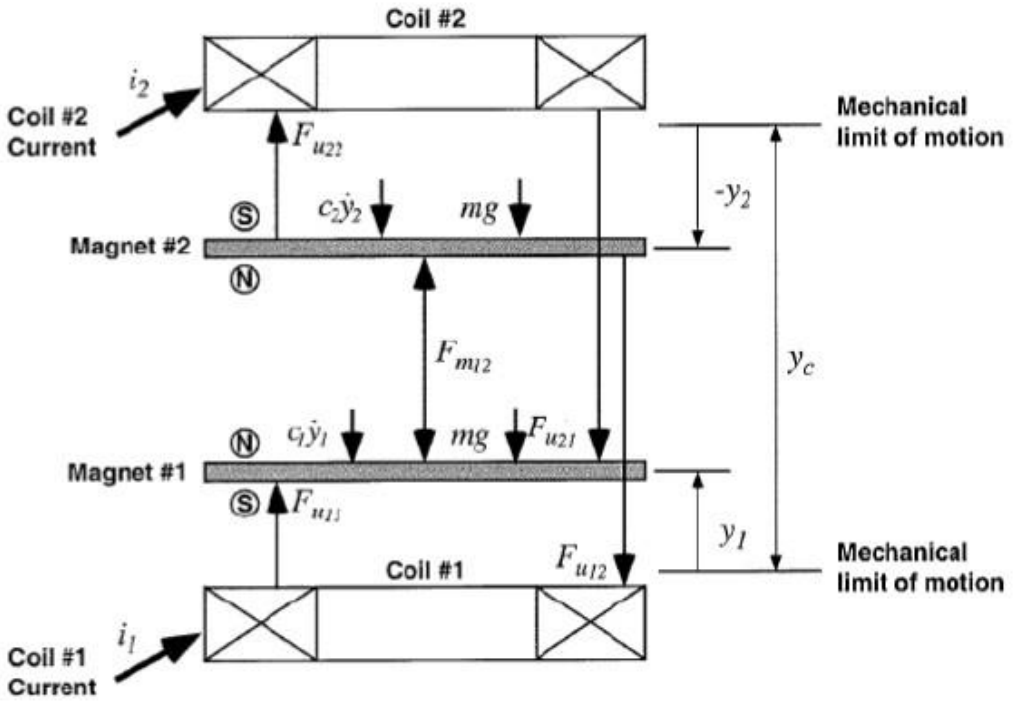

Figure 3- free body diagram and dynamic configuration

As shown in Figure 2 either magnet disk is acted on by forces from either drive coil, from the other magnet, from gravity, and from friction. To construct the mathematical model of the plant we will rely on a basic relation of Newton's second law, in this case for the first magnet we have

$$m\ddot{y}_1 + c_1\dot{y}_1 + F_{m12} = F_{u11} - F_{u21} - mg \quad [1]$$

Similarly, for the second magnet

$$m\ddot{y}_2 + c_2\dot{y}_2 - F_{m12} = F_{u22} - F_{u12} - mg \quad [2]$$

The magnetic force terms are modeled as having the following terms

$$F_{u_{11}} = \frac{i_1}{a(y_1 + b)^N}$$

$$F_{u_{12}} = \frac{i_1}{a(y_{c+}y_2 + b)^N}$$

$$F_{u_{21}} = \frac{i_2}{a(y_{c-}y_1 + b)^N}$$

$$F_{u_{22}} = \frac{i_2}{a(-y_2 + b)^N}$$

$$F_{m_{12}} = \frac{c}{(y_{12} + d)^N} \quad [3]$$

$$y_{12} = y_c + y_2 - y_1 \quad [4]$$

The table below is showed variables used for the modeling magnetic levitation system.

Constant or Variable	**Definition**	**Constant or Variable**	**Definition**
$c1$, $c2$	Friction/wind resistance	m	Mass of the magnet
$Fu22$	Force applied to magnet 2 from coil 2	g	Force of gravity
$Fu11$	Force applied to magnet 1 from coil 1	$y1$	Distance between magnet 1 and coil 1
$Fu21$	Force applied to magnet 1 from coil 2	$y2$	Distance between magnet 2 and coil 2.
$Fu12$	Force applied to magnet 2 from coil 1	yc	Distance between the coils
$i1$, $i2$	Current through the respective coils (input)	a, b, c, d, N	Constant 3 < N < 4.5
$Fm12$	The force between each magnet		

Table 1- magnetic levitation model parameters

The cross magnet/actuator forces $Fu12$ and $Fu21$ are generally small compared to $Fu22$ and $Fu11$ for typical values of coil current and for magnets in their normal operating range. Also, the

friction forces are also typically small regarding $Fu22$ and $Fu11$ the following simplified model is considered for analyzing the system and control design.

$$m\ddot{y}_1 + F_{m12} = F_{u11} - mg$$

$$m\ddot{y}_2 - F_{m12} = F_{u22} - mg \qquad [5]$$

The operating condition for this system considered as $y1$=2.00 cm, $y2$ =-2.00 cm, and yc=12.00 cm. other model parameters are N=4, m=120 (g), a=1.65, b=6.2, c=2.69, and d=4.2.

3. Linearization (Jacobian) and State apace representation

The equations of the project are nonlinear. When we want to use them, first, we must change them to linear equations. The first step, using state equations and then using the Jacobian Method around equilibrium states.

3.1. state space representation

To linearize the project equations, we use state space representation. [2] The internal state variables are the smallest possible subset of system variables that can represent the entire state of the system at any given time. The minimum number of state variables required to represent a given system, n, is usually equal to the order of the system's defining differential equation.[3]

$$\begin{cases} \dot{x}(t) = A(t)x(t) + B(t)u(t) \\ y(t) = C(t)x(t) + D(t)u(t) \end{cases} \quad \text{MIMO system} \qquad [6]$$

$$x_1 = y_1$$

$$x_2 = \dot{y}_1$$

$$x_3 = y_2$$

$$x_4 = \dot{y}_2 \qquad [7]$$

$$\dot{x_1} = x_2 \qquad [8]$$

$$\dot{x_2} = \frac{u_1}{am(x_1+b)^4} - \frac{c}{m(y_c+x_3-x_1+d)^4} \text{-g} \quad [9]$$

$$\dot{x_3} = x_4$$

$$\dot{x_4} = \frac{u_2}{am(-x_3+b)^4} + \frac{c}{m(y_c+x_3-x_1+d)^4} \text{-g} \quad [10]$$

3.2. Linearization of the project equations (Jacobian)

3.2.1. Jacobian

The term "Jacobian" often represents both the Jacobian matrix and determinants, which are defined for the finite number of functions with the same number of variables. Here, each row consists of the first partial derivative of the same function with respect to the variables. The Jacobian matrix can be of any form. It may be a square matrix (the number of rows and columns are equal) or a rectangular matrix (the number of rows and columns are not equal).

Suppose $\mathbf{f}$: $\mathbf{R}^n \to \mathbf{R}^m$ is a function such that each of its first-order partial derivatives exists on $\mathbf{R}^n$. This function takes a point $\mathbf{x} \in \mathbf{R}^n$ as input and produces the vector $\mathbf{f}(\mathbf{x}) \in \mathbf{R}^m$ as output. Then the Jacobian matrix of $\mathbf{f}$ is defined to be an $m \times n$ matrix, denoted by $\mathbf{J}$, whose (i,j) the entry is[1]:

$$\mathbf{J}_{ij} = \frac{\partial f_i}{\partial x_j}, \quad [12]$$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \nabla^{\mathrm{T}} f_1 \\ \vdots \\ \nabla^{\mathrm{T}} f_m \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \quad [13]$$

The Jacobian Matrix of the project equations:

$$\dot{x_1} = f_1(x,u)$$
$$\dot{x_2} = f_2(x,u)$$
$$\dot{x_3} = f_3(x,u)$$
$$\dot{x_4} = f_4(x,u)$$

[14]

Matrix A equal to:

$$\begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_4} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_4}{\partial x_1} & \cdots & \frac{\partial f_4}{\partial x_4} \end{pmatrix}$$

[15]

$$\frac{\partial f_1}{\partial x_1} = \frac{\partial f_3}{\partial x_1} = \frac{\partial f_2}{\partial x_2} = \frac{\partial f_3}{\partial x_2} = \frac{\partial f_4}{\partial x_2} = \frac{\partial f_1}{\partial x_3} = \frac{\partial f_3}{\partial x_3} = \frac{\partial f_1}{\partial x_4} = \frac{\partial f_2}{\partial x_4} = \frac{\partial f_4}{\partial x_4} = 0 \quad [16]$$

$$\frac{\partial f_1}{\partial x_2} = \frac{\partial f_3}{\partial x_4} = 1 \quad [17]$$

$$\frac{\partial f_2}{\partial x_1} = \frac{-4u_1}{ma(x_1+b)^5} - \frac{4c}{m(y_c+x_3-x_1+d)^5} \quad [18]$$

$$\frac{\partial f_2}{\partial x_3} = \frac{-4c}{m(y_c+x_3-x_1+d)^5} \quad [19]$$

$$\frac{\partial f_4}{\partial x_1} = \frac{4c}{m(y_c+x_3-x_1+d)^5} \quad [20]$$

$$\frac{\partial f_4}{\partial x_3} = \frac{4u_2}{ma(-x_3+b)^5} - \frac{4c}{m(y_c+x_3-x_1+d)^5} \quad [21]$$

Matrix

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -6 \cdot 4 & 0 & 0 \cdot 06 & 0 \\ 0 & 0 & 0 & 1 \\ 0 \cdot 06 & 0 & -6 \cdot 16 & 0 \end{bmatrix} \quad [22]$$

Matrix b is equal to:

$$\begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_2} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_4}{\partial u_1} & \cdots & \frac{\partial f_4}{\partial u_2} \end{pmatrix} \quad [23]$$

$$\frac{\partial f_1}{\partial u_1} = \frac{\partial f_1}{\partial u_2} = \frac{\partial f_2}{\partial u_2} = \frac{\partial f_3}{\partial u_2} = \frac{\partial f_3}{\partial u_1} = \frac{\partial f_4}{\partial u_1} = 0 \quad [24]$$

$$\frac{\partial f_2}{\partial u_1} = \frac{1}{ma(-x_3+b)^4} \quad [25]$$

$$\frac{\partial f_4}{\partial u_2} = \frac{1}{ma(-x_3+b)^4}$$

Matrix

$$B = \begin{bmatrix} 0 & 0 \\ 0 \cdot 003 & 0 \\ 0 & 0 \\ 0 & 0 \cdot 003 \end{bmatrix} \quad [26]$$

$$y_1 = x_1 \quad [27]$$

$$y_2 = x_3 \quad [28]$$

Matrix C = $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ [29]

Matrix D = [0] [30]

4. Transfer Function open loop

This system is MIMO with two input and two output so we have four transfer function. We could neglect two of them in our analysis because effect of input one on output two and effect of input two on output one is insignificant.

$$TF1 = \frac{1.117e^{-6}s^3+1.195e^{-25}s^2-6.087e^{-13}s^1+1.329e^{-31}}{s^4+1.084e^{-18}s^3+2.324e^{-6}s^2+6.204e^{-15}s^1-4.306e^{-31}} \quad [31]$$

$$TF2 = \frac{1.117e^{-6}s^3+3.204e^{-12}s^1-6.087e^{-13}s^1}{s^4+1.084e^{-18}s^3+2.324e^{-6}s^2+6.204e^{-15}s^1-4.306e^{-31}} \quad [32]$$

5. Controllable, observable, Jordan Canonical form:

5.1. Controllable:

First, we check if the system has any poles that can be controlled with making C matrix.

$C_x = (B \;\; AB \;\; A^2B \;\; A^3B)$

$$C_x = \begin{bmatrix} 0 & 0 & 0.1117 & 0 & 0 & 0 & -0.2826 & 0.0371 \\ 1.117\times10^{-6} & 0 & 0 & 0 & -0.2826 & 0.0371 & 0 & 0 \\ 0 & 0 & 0 & 0.1117 & 0 & 0 & -0.0371 & 0.0238 \\ 0 & 1.117\times10^{-6} & 0 & 0 & -0.0371 & 0.0238 & 0 & 0 \end{bmatrix} \quad [33]$$ [34]

Matrix C is controllable because no one columns are dependent on the other.

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 39.46 & -1 \cdot 265 \times 10^{-15} & -12 \cdot 58 & -1 \cdot 685 \times 10^{-16} \end{bmatrix} \quad [35]$$

$$B_c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad [36]$$

Observable:

$$O_x = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \end{bmatrix} \quad [37]$$

$$O_x = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -0.25 & 0 & 0.033 & 0 \\ -0.033 & 0 & 0.0213 & 0 \\ 0 & -0.25 & 0 & 0.033 \\ 0 & -0.033 & 0 & 0.0213 \end{matrix} \quad [38]$$

$$A_o = \begin{bmatrix} 0 & 0 & 0 & 39.47 \\ 1 & 0 & 0 & -1 \cdot 27 \times 10^{-15} \\ 0 & 1 & 0 & -12.53 \\ 0 & 0 & 1 & -1 \cdot 865 \times 10^{-16} \end{bmatrix}$$

[39]

$$C_o = [0 \quad 0 \quad 0 \quad 1]$$

[40]

Jordan:

$$A_{jordan} = \begin{bmatrix} 2.56i & 0 & 0 & 0 \\ 0 & 2.67i & 0 & 0 \\ 0 & 0 & -2.43i & 0 \\ 0 & 0 & 0 & -2.49i \end{bmatrix}$$

6. impulse and step response:

The step and impulse response of the SISO system for each coil are represented below.

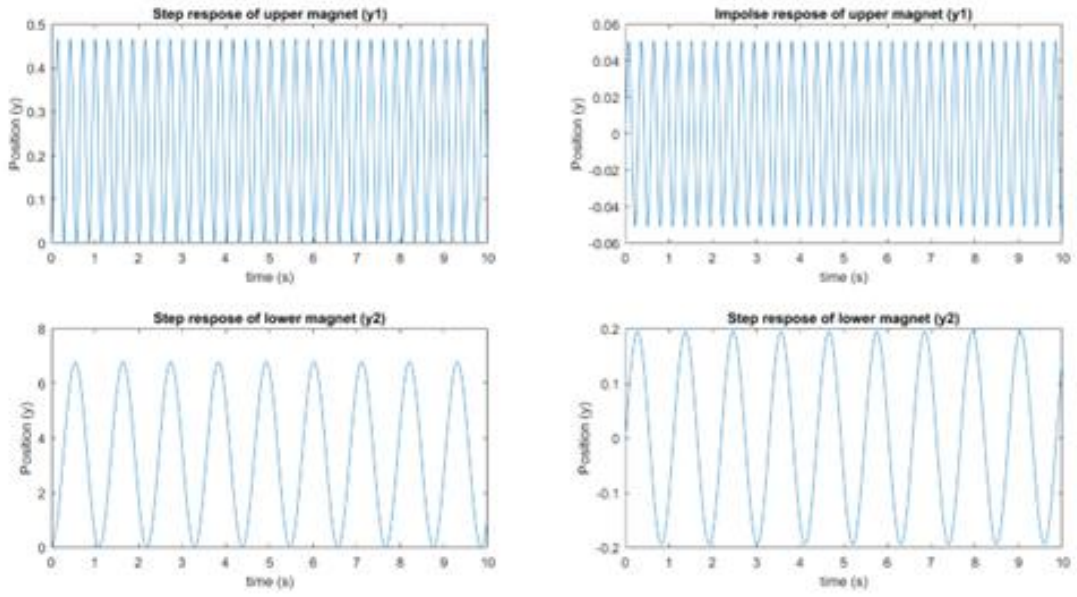

Figure 4&5- Impulse Response, step response

It is obvious that both impulse and step responses are unstable because the overall system is unstable. figures depict how the system displays oscillatory behavior both in step and impulse responses. because no damping is present, system oscillates around its final value forever.

6.1. The Bode plot and root locus of the uncompensated system:

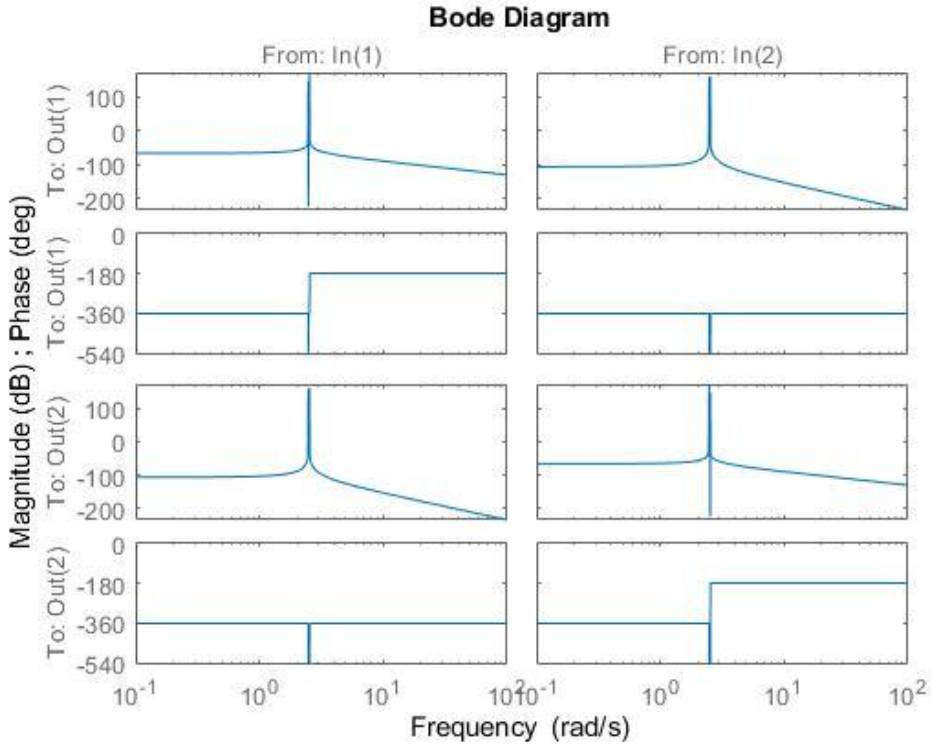

Figure 6- Bode diagram

This system is MIMO with two input and two output so we have four transfer function. We could neglect two of them in our analysis because effect of input one on output two and effect of input two on output one is insignificant.

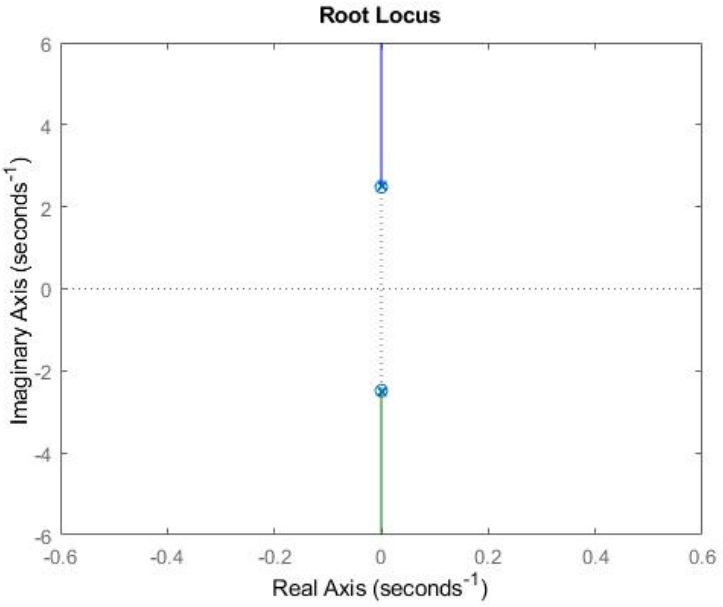

Figure7- Root Locus loop1

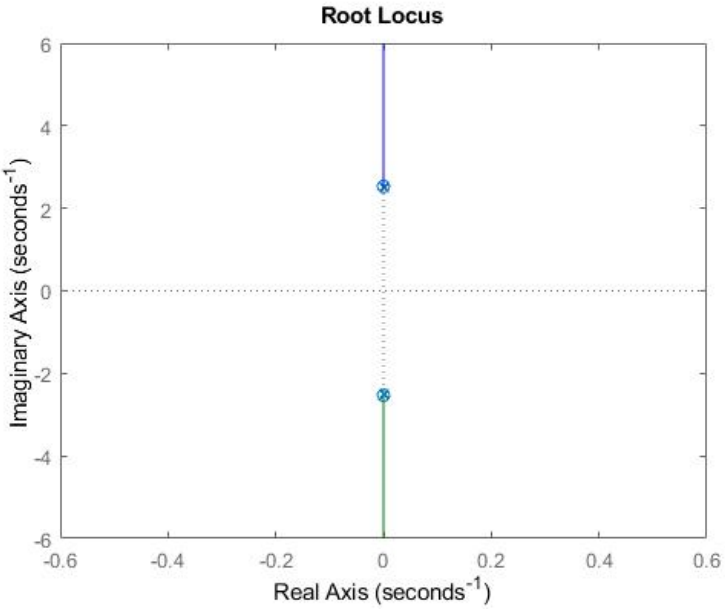

Figure 8- Root Locus loop 2

As could be seen in the above figures, root locus is plotted to show where poles are in the s-plane, and two imaginary poles in S-plane lead to oscillatory responses.

7. PID controller:

The PID controller diagram showed as below, A PID controller is a simple three-term controller. The letters P, I and D stand for:

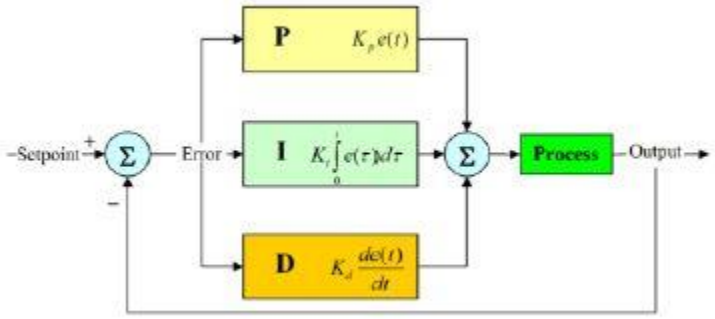

Figure 9-overall PID

There are four major characteristics of the closed-loop step response for a control system, they are:

1. **Rise Time**: the time it takes for the plant output y to rise beyond 90% of the desired level for the first time.

2. **Overshoot**: how much the peak level is higher than the steady state, normalized against the Steady-State.

3. **Settling Time**: the time it takes for the system to converge to its steady state.

4. **Steady-state Error**: the difference between the steady-state output and the desired output.

The increase of each controller parameter will change the characteristics of the closed loop. Table 3-1 summarized all the effects due to the increment of controller parameters.

Response	Rise Time	Overshoot	Settling Time	S-S Error	Stability
K_p	Decrease	Increase	NT	Decrease	Decrease
K_i	Decrease	Increase	Increase	Eliminate	Decrease
K_d	NT	Decrease	Decrease	NT	Improve

Table 2-charachterestics

In this project the most important characteristic is overshoot (must under 10%) and settling time. In order to design the PID, the auto tuner of MATLAB is used for finding PID coefficients and use them to control our system.

Controller Parameters

	Tuned
Kp	712812.1808
Ti	n/a
Td	n/a
N	n/a

Performance and Robustness

	Tuned
Rise time	2.73 seconds
Settling time	4.86 seconds
Overshoot	0.181 %
Peak	1
Gain margin	255 dB @ 0 rad/s
Phase margin	-90 deg @ 9.93e-07 rad/s
Closed-loop stability	Stable

Figure 10- PID Transfer Function 1

In this system we could control the system and achieve our goal with only proportional part of PID. System is now stable, and all characteristics meet the project demands.

Controller Parameters

	Tuned
Kp	4747139.0268
Ti	1.3119
Td	n/a
N	n/a

Performance and Robustness

	Tuned
Rise time	0.312 seconds
Settling time	2.81 seconds
Overshoot	9.17 %
Peak	1.09
Gain margin	-147 dB @ 0.00157 rad/s
Phase margin	0.127 deg @ 0.00169 rad/s
Closed-loop stability	Stable

Figure 11- PID Transfer Function 2

For this transfer function we design PI to satisfy the project demands.

The overall close loop system with PID would be:

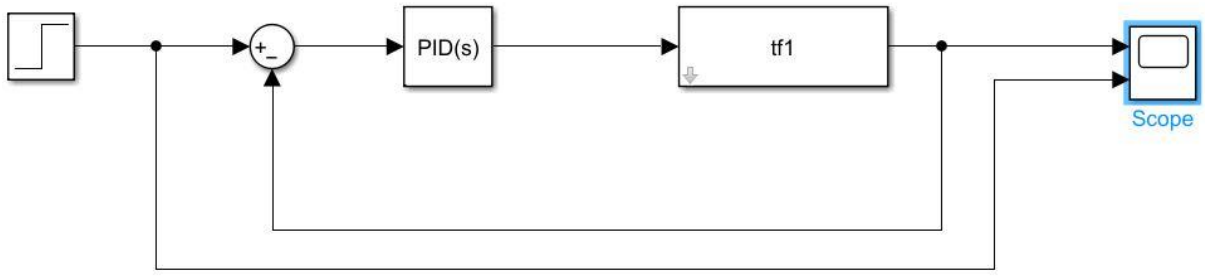

Figure 12- overall close loop

8. Results of different inputs on close loop system with PID:

For transfer function one we have:

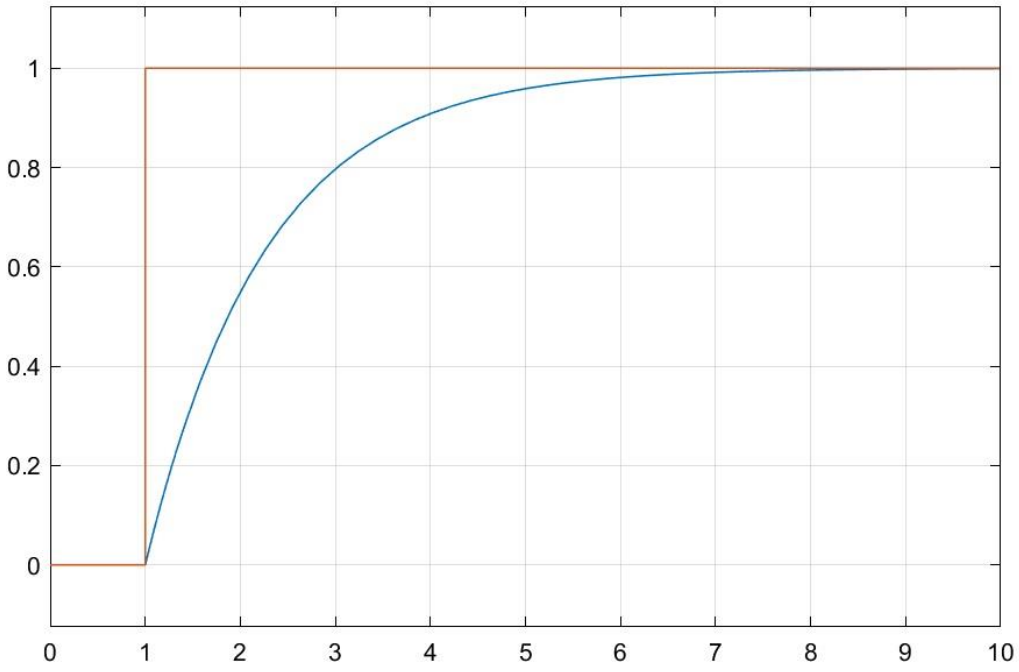

Figure 13- PID step output 1

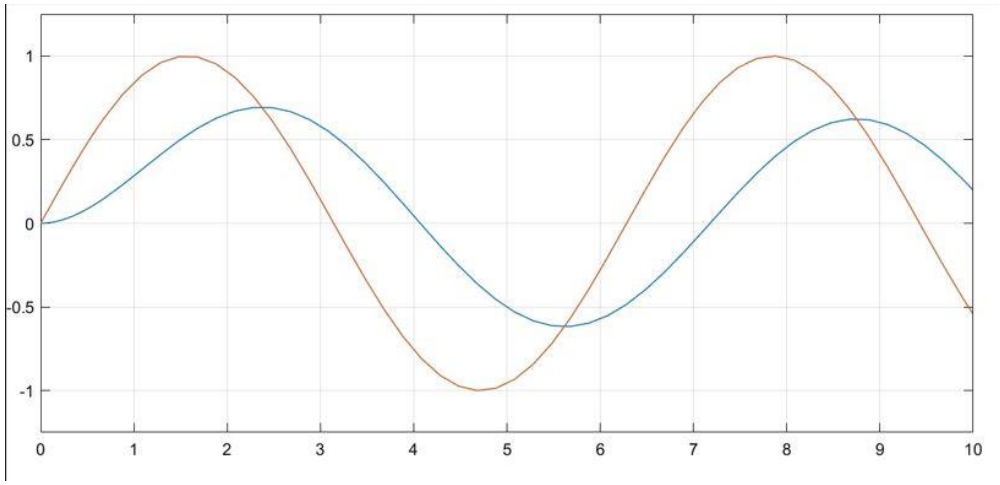

Figure 14- PID sinusoidal output 1

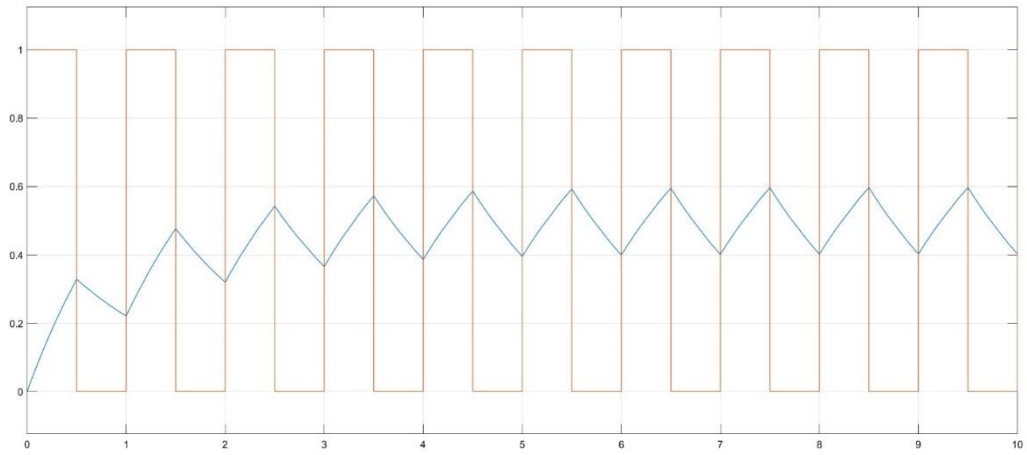

Figure 15- PID square output 1

For transfer function 2 we have:

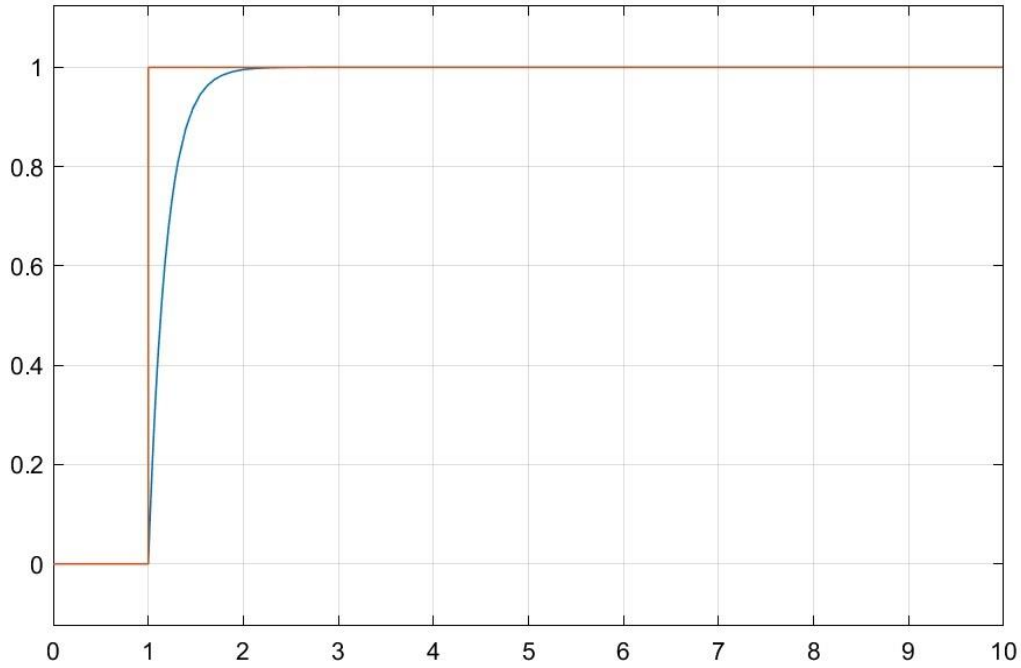

Figure 16- PID step output 2

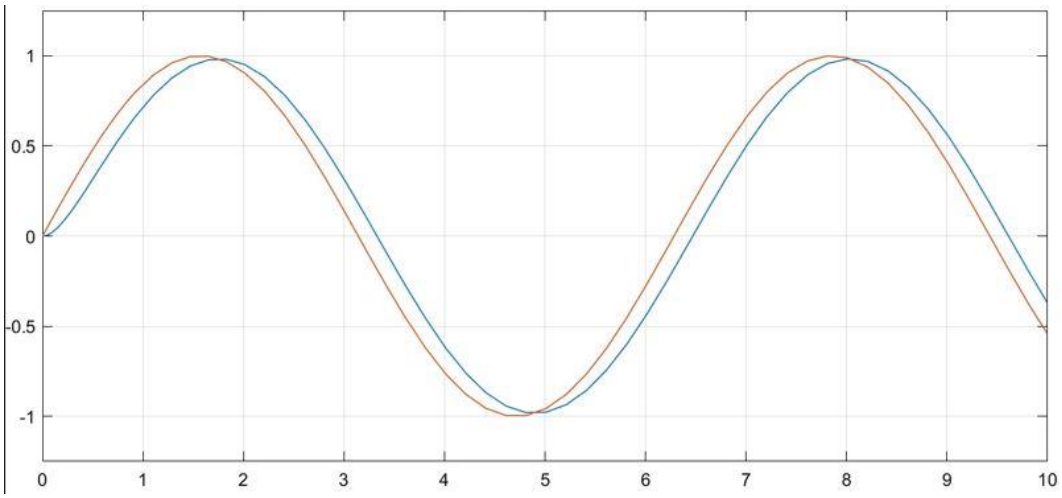

Figure 17- PID sinusoidal output 2

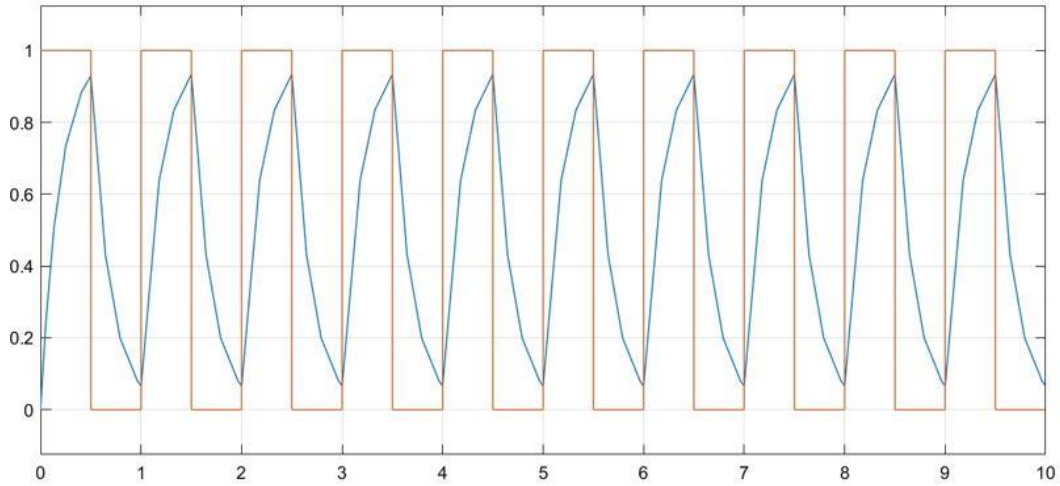

Figure 18- PID square output 2

For control input of different input of transfer function one and two we have:

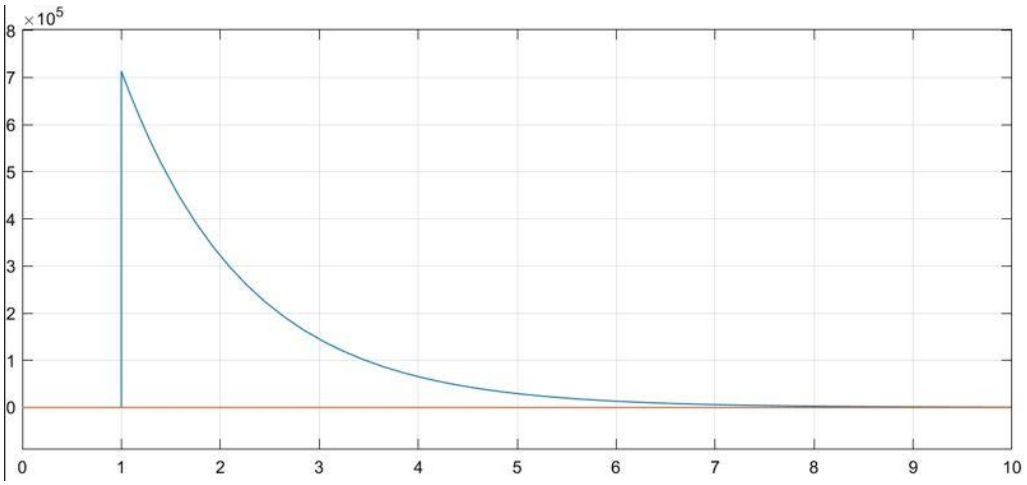

Figure 19- PID step control input 1

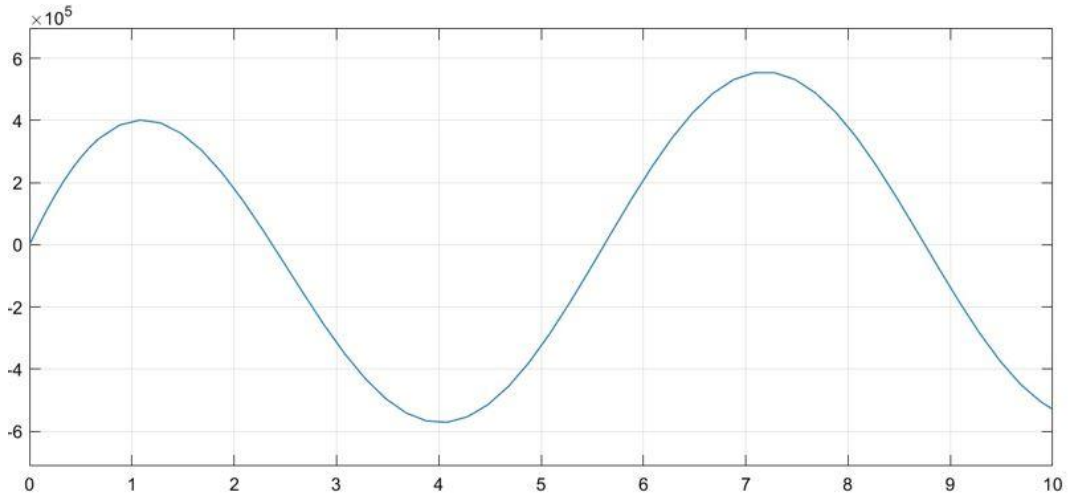

Figure 20- PID sinusoidal control input 1

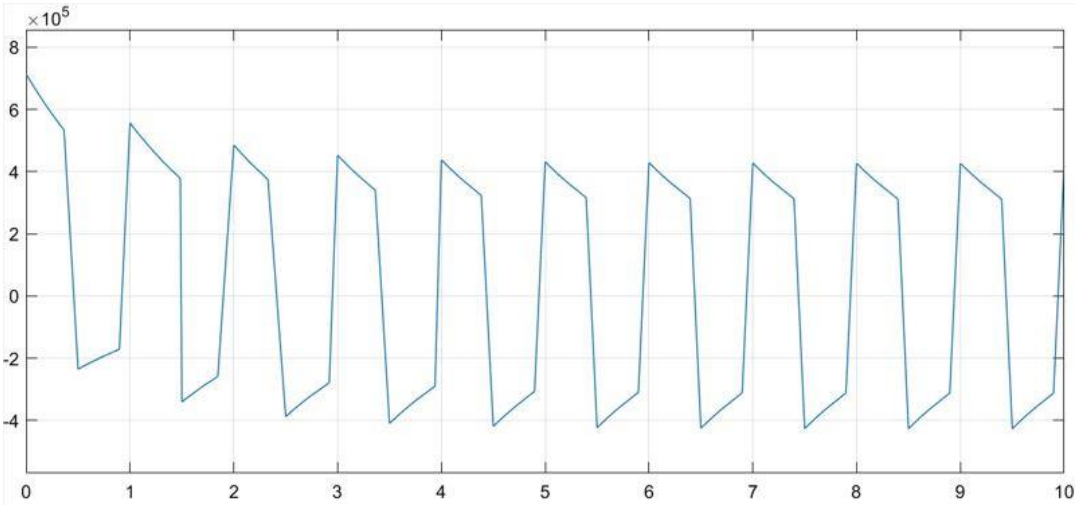

Figure 21- PID square control input 1

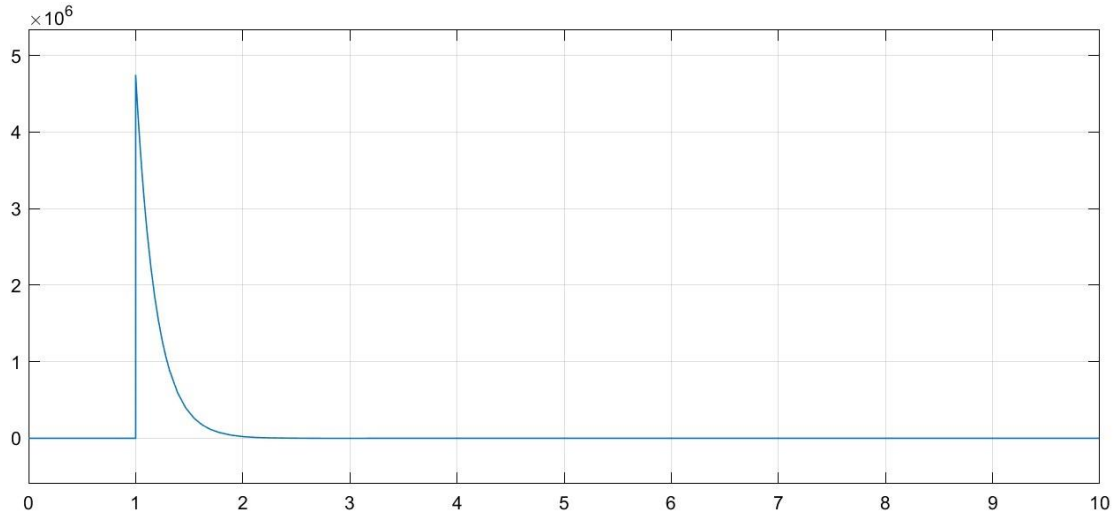

Figure 22- PID step control input 2

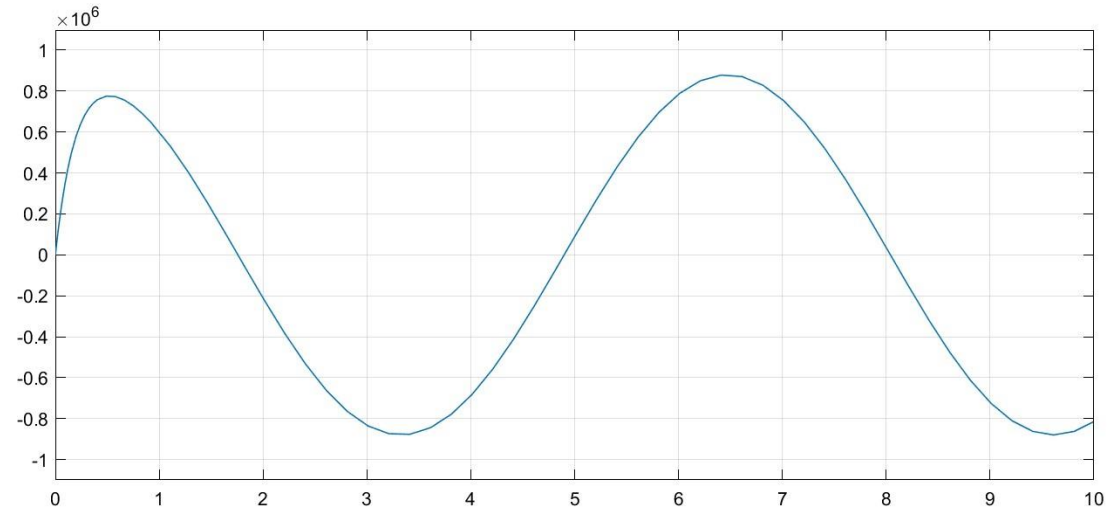

Figure 23- PID sinusoidal control input 2

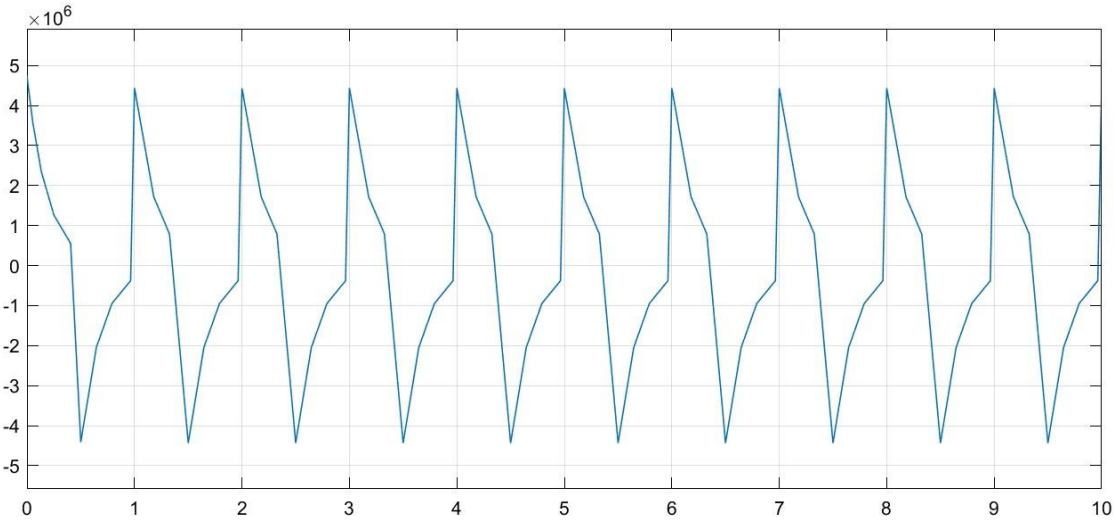

Figure 24- PID square control input 2

9. Test Robustness of system:

To test robustness of this system with PID configuration we put random error generator before the controller. The maximum amount of error was limited to 1% of input signal. For study the robustness we use square wave input for two systems.

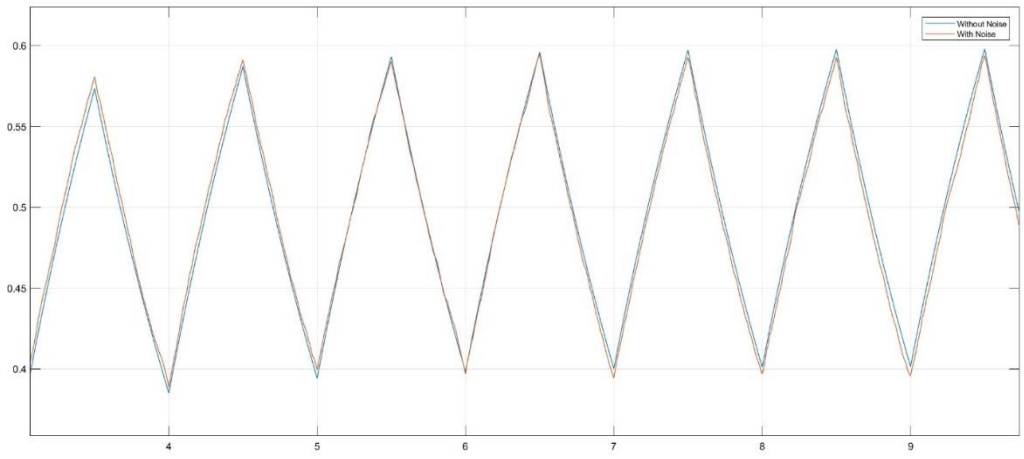

Figure 25- PID square input 1 with noise

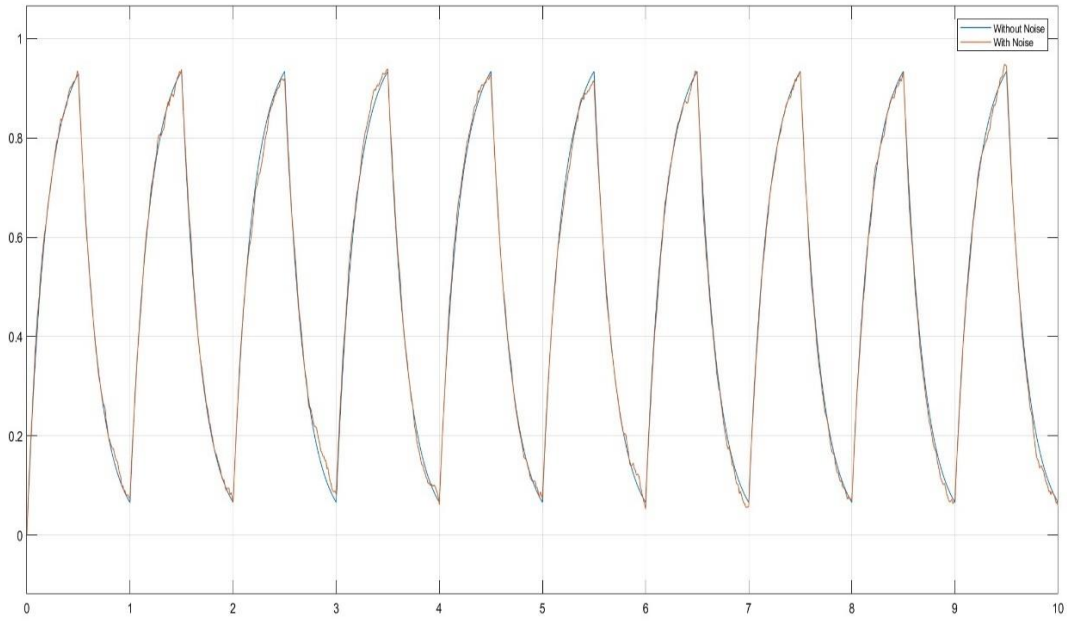

Figure 26- PID square input 2 with noise

As could be seen, the overall performance with noise in closed loop PID is acceptable and noise could not make the system unstable.

10. full-order and reduced-order observer:

In this chapter, the Full state feedback control for stabilizing and controlling the magnetic levitation system is used. Regarding the separation principle in this chapter full state feedback control is designed and in the following chapter, the full order estimator and reduced order estimator are also presented.

$$\dot{x} = Ax + Bu \quad [41]$$

$$u = -Kx \qquad K = [K1\ K2 \dots Kn] \quad [42]$$

The closed loop state representation will be

$$\dot{x} = (A - BK)x \quad [43]$$

we specify a maximum of 10% overshoot and settle time of less than 5 second as our desire design specification which should produce better result compared to PID controller and acceptable practical performance.

$M_p = 1.1$
$T_s = 5$

The new closed loop system step, square wave and sinusoidal responses represented below. In order to achieve these requirements, we could use equations bellow and obtain new poles:

$$M_p = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \quad [44]$$

$$\frac{4}{\zeta\omega_n} = 5 \quad [45]$$

From characteristic equation we could calculate the desire poles:

$$s^2 + 2\zeta\omega_n + \omega_n^2 = 0 \quad [46]$$

Our new poles are:

```
-1+0.5i -1-0.5i -0.8+1.1i -0.8-1.1i
```

Now we could simulate state feedback:

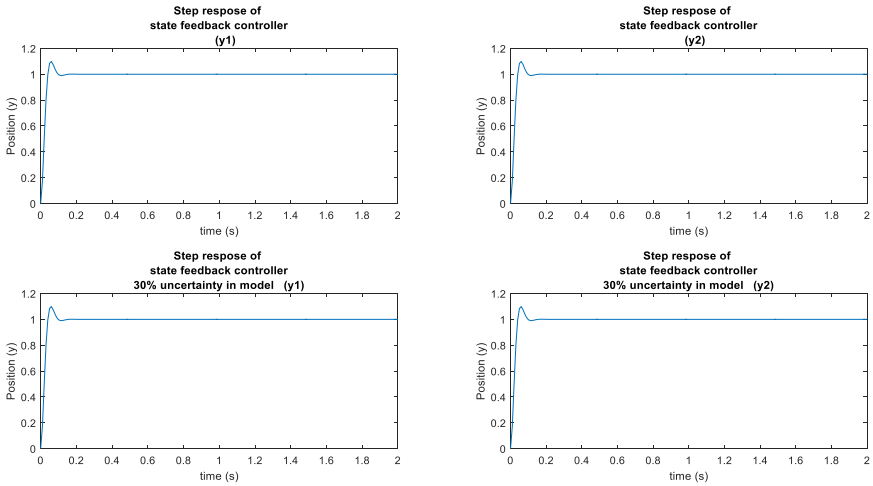

Figure 27- Step response of state feedback controller with and without uncertainty in model

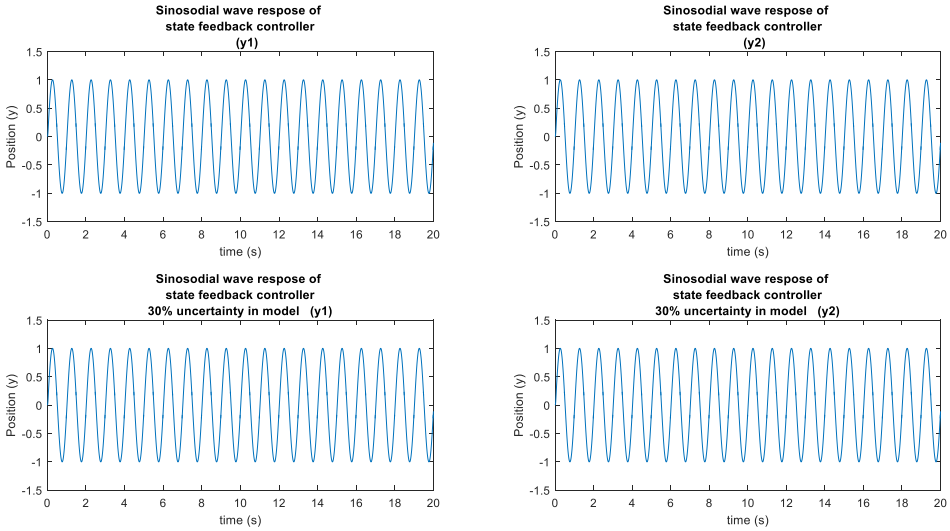

Figure 28- Sinusoidal response of state feedback controller with and without uncertainty in model

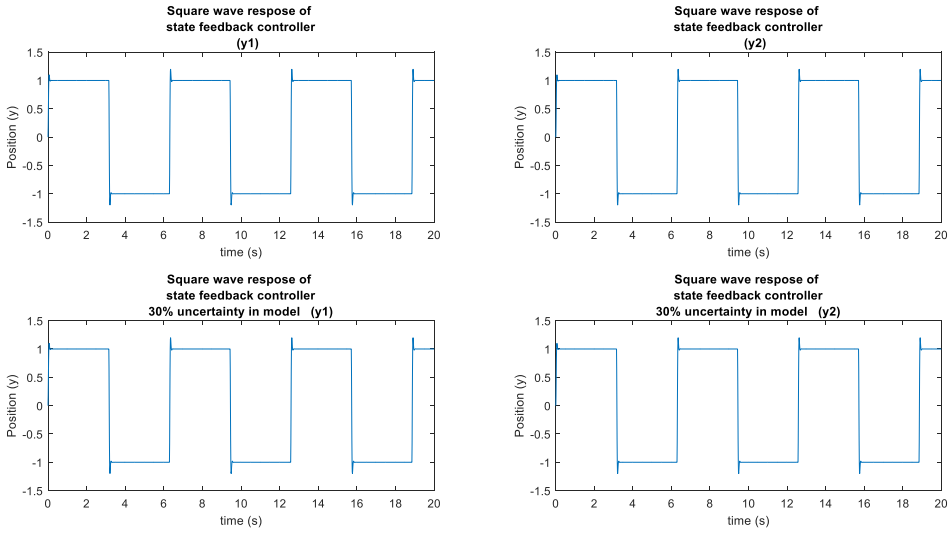

Figure 29- Square wave response of state feedback controller with and without uncertainty in model

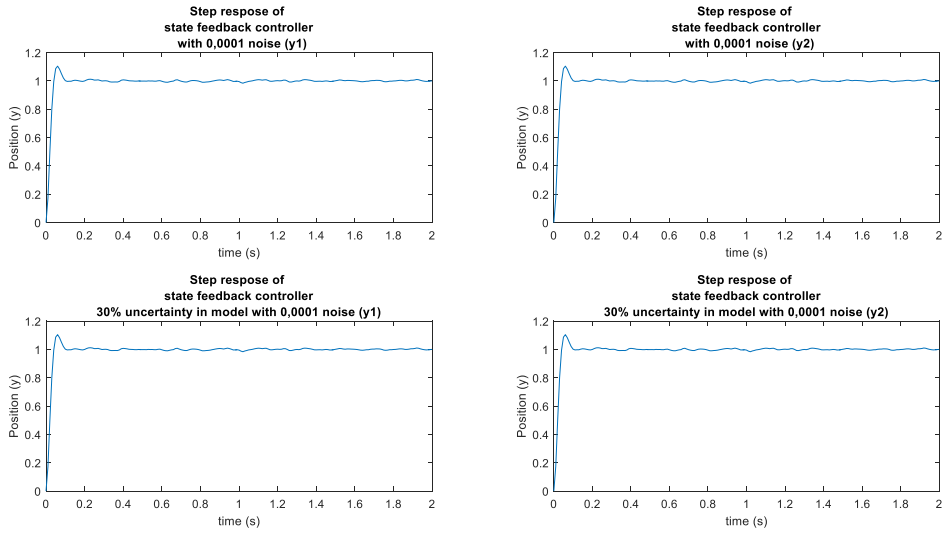

Figure 30- Step response of state feedback controller with 0,0001 noise

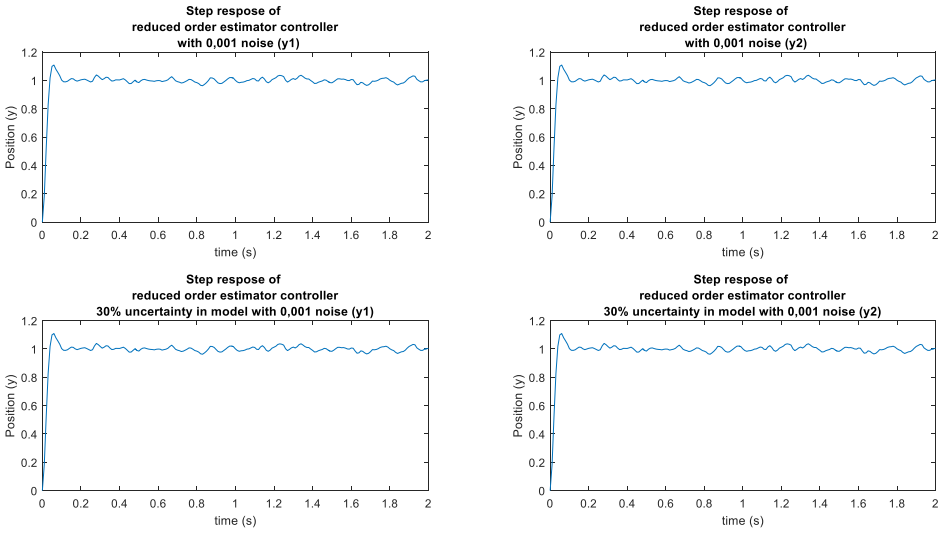

Figure 31- Step response of state feedback controller with 0,0001 noise

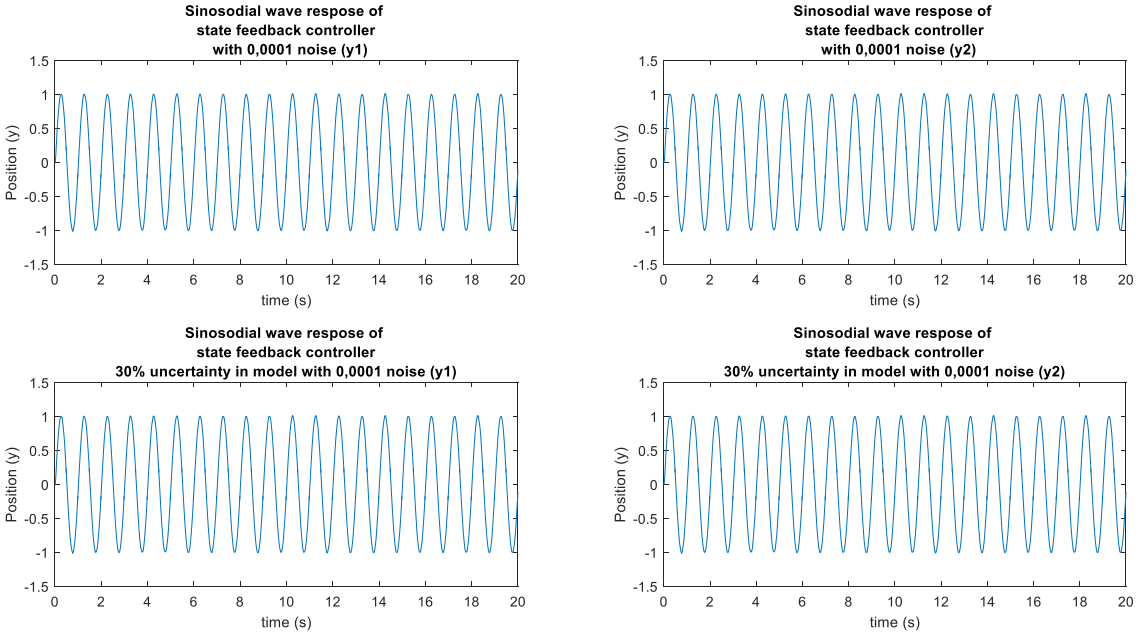

Figure 32- Sinusoidal wave response of state feedback controller with 0,0001 noise

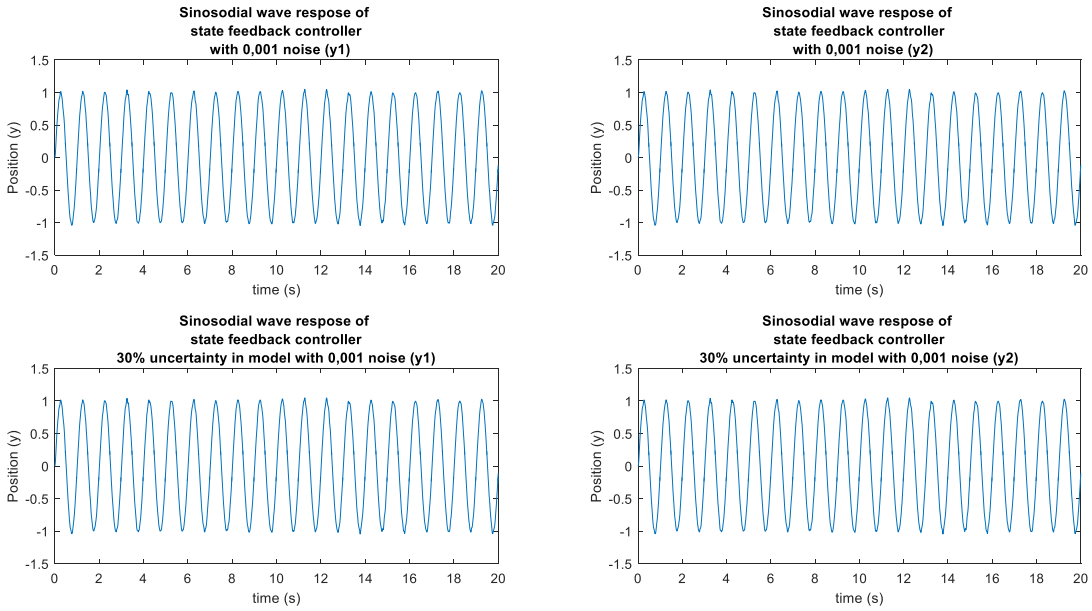

Figure 33- Sinusoidal wave response of state feedback controller with 0,001 noise

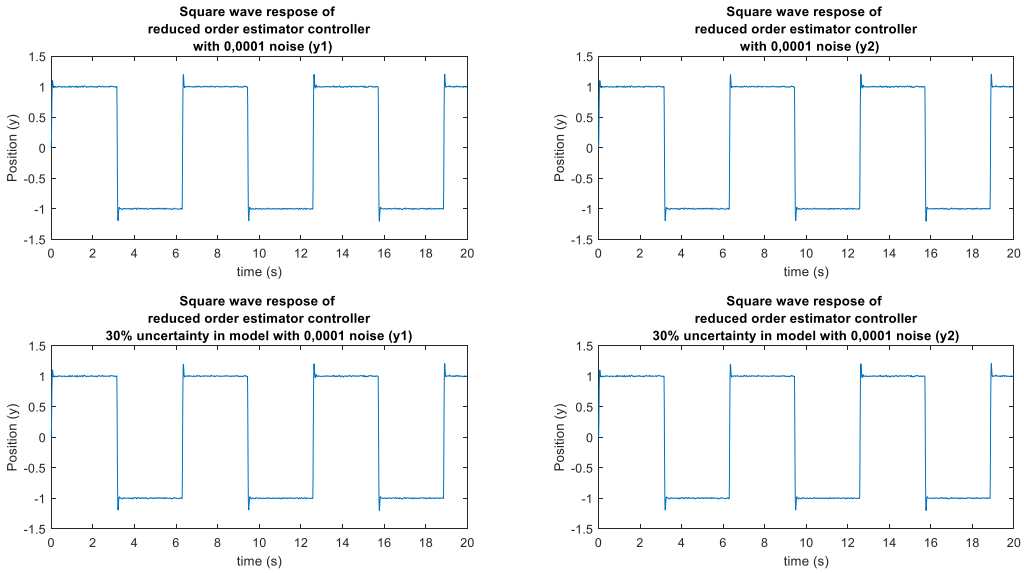

Figure 34- Square wave response of reduced order estimator controller with 0,0001 noise

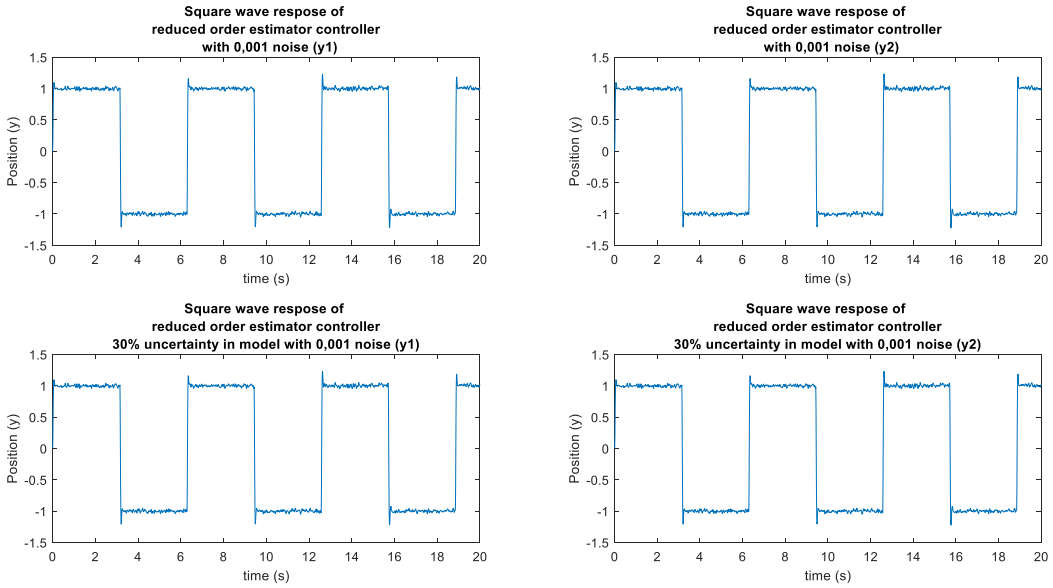

Figure 35- Square wave response of reduced order estimator controller with 0,001 noise

With the full state feedback controller, we get better result in compared to PID configuration. With noise and uncertainty system shows more robustness and track output better.

11. Control input:

In this section control input of step, square wave and sinusoidal responses with full state feedback controller is presented:

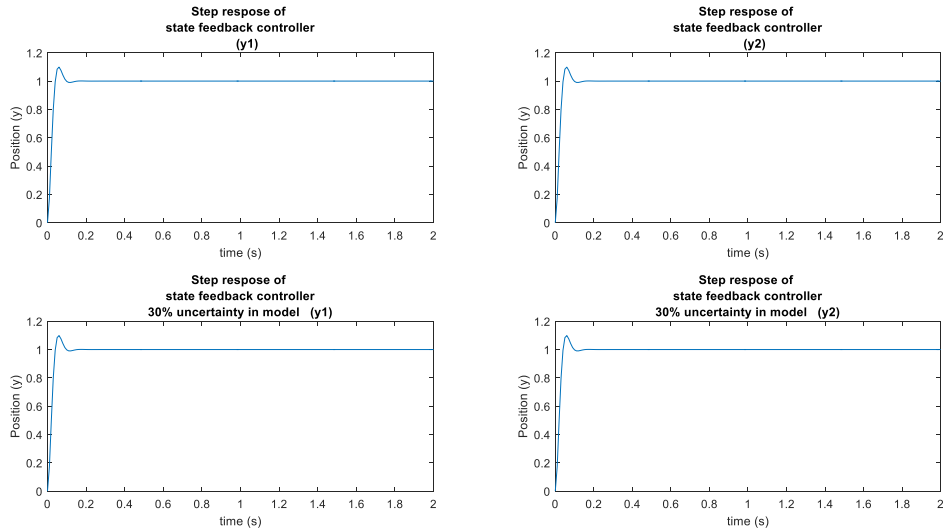

Figure 36- Step response of state feedback controller

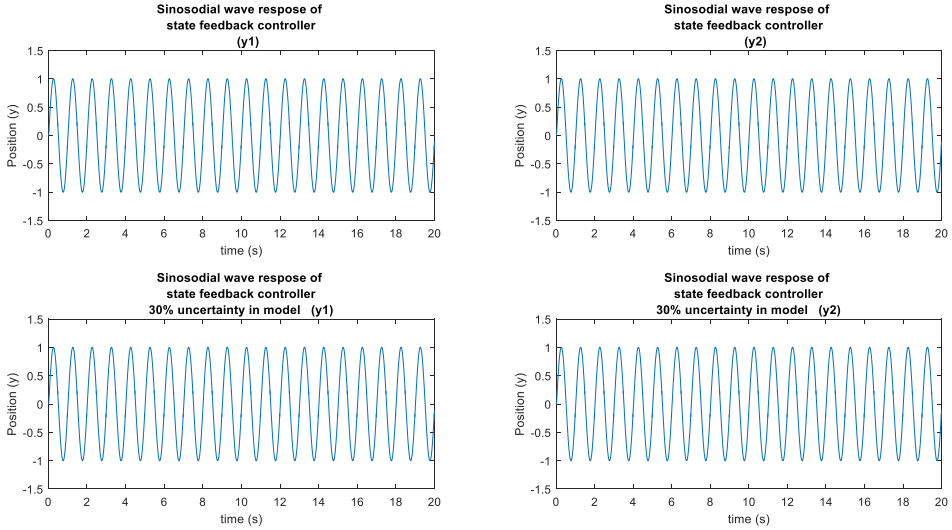

Figure 1 Sinusoidal wave response of state feedback controller

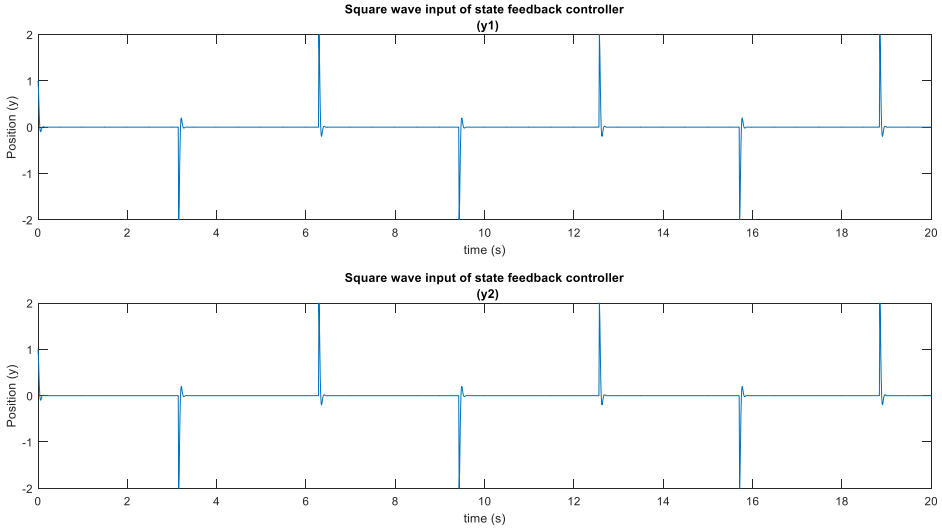

Figure 37- Square wave controller output of state feedback controller

The comparison between PID and state feedback, state feedback has lower settling time, and the maximum overshoot is also less than system with PID controller.

11.1. Full order observer:

We need the observer to be 3 to 5 times faster than real plant and our design specification and in this control design observer response is 3 times faster. The step and sinusoidal responses are represented in the following:

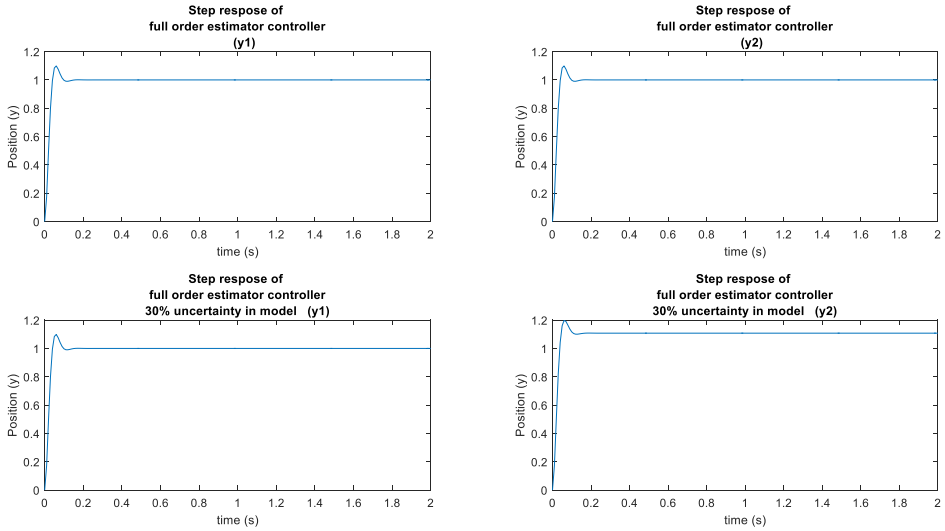

Figure 38-Step response of full order estimator controller

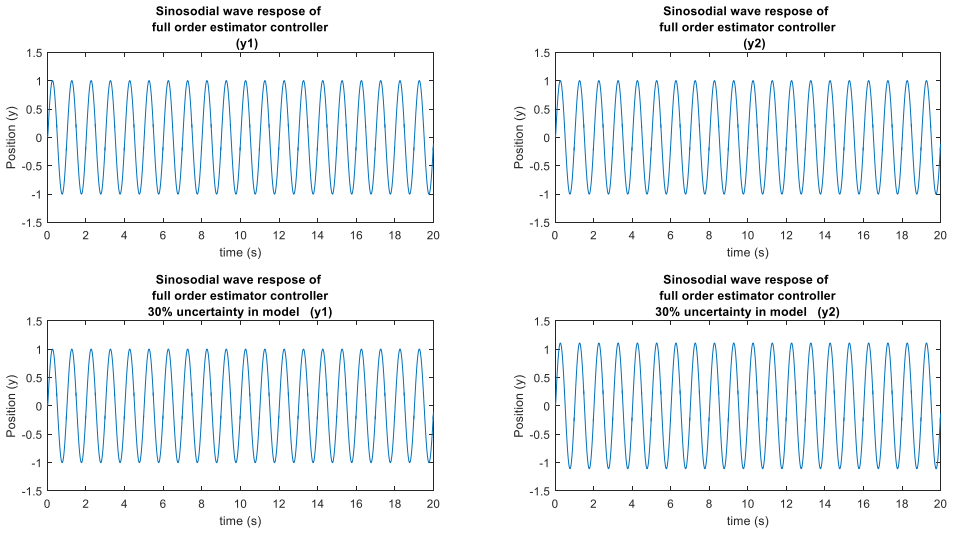

Figure 39-Sinusoidal wave response of full order estimator controller

11.2. reduced order observer:

Estimators developed so far are called full-order estimators since all states of the plant are estimated. However, usually from output measurement some states are directly available. Hence it is not logical to estimate states that are directly measured. In this section with the assumption that the position of disks are the available states a reduced order observer is designed.
The reduced order estimator is designed, and step and sinusoidal responses are in the following:

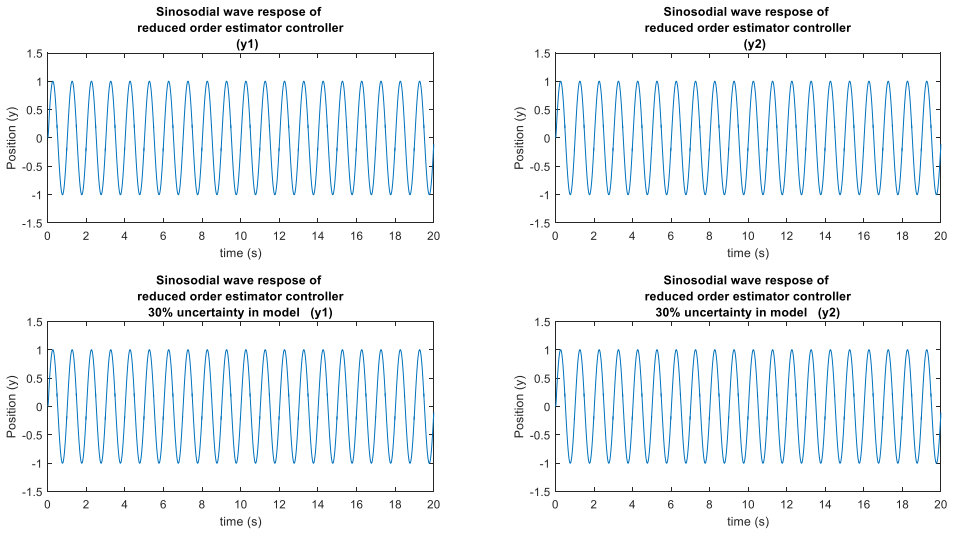

Figure 40-Sinusoidal wave response of reduced order estimator controller

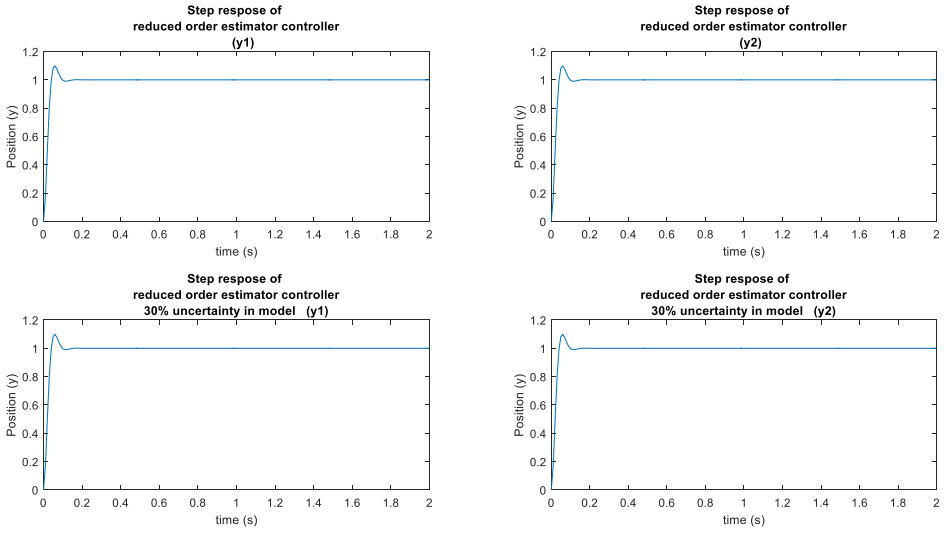

Figure 2 Step response of reduced order estimator controller

It is obvious that using this configuration not recommended. First, because the original system was nonlinear that we converted to linear and then second, we neglect many terms for simplification.

The nature of reduced order is with some uncertainty, and it brings more error to system, because of that it probably could not implemented to the real world.

12.The transfer function of the observer and controller:

Here the transfer function is achieved. The equation 46 is transfer function of the controller-estimator

$$G_{ec}(s) = K(sI - A + BK + GC)^{-1}G \qquad 46$$

So, after using this equation, it leads to these transfer functions:

1.021e12 s^7 + 2.645e14 s^6 - 8.05e15 s^5 - 4.861e18 s^4 + 5.117e19 s^3 + 2.939e22 s^2 - 7.393e23 s - 1.911e25

---

s^8 + 324 s^7 - 2634 s^6 - 8.08e06 s^5 - 1.221e08 s^4 + 8.484e10 s^3 - 2.903e11 s^2 - 3.751e14 s + 1.215e16

1.021e12 s^7 + 2.645e14 s^6 - 8.05e15 s^5 - 4.861e18 s^4 + 5.117e19 s^3 + 2.939e22 s^2 - 7.393e23 s - 1.911e25

---

s^8 + 324 s^7 - 2634 s^6 - 8.08e06 s^5 - 1.221e08 s^4 + 8.484e10 s^3 - 2.903e11 s^2 - 3.751e14 s + 1.215e16

13.Conclusion:

In this chapter, we compare all the system results; controllers and full-state feedback controllers. At first glance, understanding and controlling plant processes using PID controllers is straightforward, because the system models are independent, and the trail-and-error tuning method is simple. However, this trial and error is time consuming and does not guarantee optimal control. Also, noise degrades the performance of the PID controller system, especially at high gains. One of the most important advantages of full-state feedback controllers is that design characteristics (maximum overshoot, settling time, rise time, etc.) can be achieved in a systematic process without the need to measure the entire state of the system. No trial and error. However, it requires knowledge of the system model. In general, feedback controllers can have smaller gains and are more robust to noise and interference. Model uncertainty can be handled well by a full-state feedback controller. Figures show that the state-feedback controller outperforms his MATLAB automatic PID tuner. Note that in most cases it is possible to achieve comparable performance of a PID controller using a full-state feedback controller, but it requires both process knowledge and a lot of trial-and-error effort.

14.References

1- Khimani, D., Karnik, S., & Patil, M. (2018). Implementation of high-performance nonlinear feedback control on magnetic levitation system. *IFAC-PapersOnLine*, *51*(1), 13–18. https://doi.org/10.1016/j.ifacol.2018.05.003

2- Khorasani, K. (2022). *Linear systems* [slides]. Moodle. https://moodle.concordia.ca/moodle/login/index.php

3- Nise, N. S. (2011). *Control Systems Engineering*. Wiley.

4- Automation, W. S. E. A. S. I. C. on S. P. R. and. (2009). *Recent advances in signal processing, Robotics and Automation: Proceedings of the 9th Wseas International Conference on Signal Processing, Robotics and automation (Ispra'10): University of Cambridge, Uk, Februray 20-22, 2010*. WSEAS.

15.Appendix (MATLAB codes)

```
clc
clear
%sourena morteza ghasemi (40171622)
syms x1 x2 x3 x4 u1 u2 a m b c yc g d

y1=0.02;
y2=-0.02;
yc=0.12;
y12=yc-y1+y2;
m=0.12;
a=1.65;
b=6.2;
c=2.69;
d=4.2;
g=9.8;

%linearization(jacobian method)

f1=x2;
f2=-g+(u1/a*m*(x1+b)^4)-(c/m*(yc+x3-x1+d)^4);
f3=x4;
f4=-g+(u2/a*m*(-x3+b)^4)+(c/m*(yc+x3-x1+d)^4);
f=[f1;f2;f3;f4];
x=[x1;x2;x3;x4];
u=[u1;u2];

A=jacobian(f,x);

B=jacobian(f,u);

subs(A, {sym('x1'), sym('x3'), sym('u1')}, {2, -2, 8772960});
subs(B, {sym('x1'), sym('x3')}, {2, -2});

A=[0 1 0 0; -6.4 0 0.06 0; 0 0 0 1; 0.06 0 -6.16 0];
B=[0 0; 0.003 0; 0 0; 0 0.003];
C=[1 0 0 0;0 0 1 0];
D=[0 0;0 0];

sys = ss(A, B, C, D);
%
%
transferFunction = tf(sys);
tf1 = transferFunction(1,1);
tf2 = transferFunction(2,2);
%
%
csys = canon(sys, 'companion');
j = jordan(A);
impulse(sys)
step(sys)
impulseRes = impulse(tf1)
stepRes = step(sys)
```

```
% %
bode(transferFunction)
rlocus(transferFunction(1,1))
rlocus(transferFunction(2,2))
% %
% %
% % %%%%%%%%%%%%
% % %%%fullstate feed back controller
% %
a1 = [0.0032 0 0.01];
b1 = [1 0 12.59 0 39.53];
% %
[A1 B1 C1 D1] = tf2ss(a1,b1);
P1 = [-1+0.5i -1-0.5i -0.8+1.1i -0.8-1.1i];
% %
k1 = place(A1,B1,P1);
% %
Anew1 = A1-B1*k1;
Bnew1 = B1;
Cnew1 = C1;
Dnew1 = D1;
% %
sysnew1 = ss(Anew1, Bnew1, Cnew1, Dnew1);
tfnew1 = tf(sysnew1);
impulse(tfnew1)
step(tfnew1)
% %
% %
% %
% %
% %
a2 = [0.002 0 0.023 ]
b2 = [1 0 12.58 0 39.52]
% %
[A2 B2 C2 D2] = tf2ss(a2,b2)
P2 = [-1+0.5i -1-0.5i -0.8+1.1i -0.8-1.1i]
% %
k2 = place(A2,B2,P2)
% %
Anew2 = A2-B2*k2
Bnew2 = B2
Cnew2 = C2
Dnew2 = D2
% %
sysnew2 = ss(Anew2, Bnew2, Cnew2, Dnew2)
tfnew2 = tf(sysnew2)

step(tfnew2)
impulse(tfnew2)
% %
% % Observer
% %
l = 3*P1
s = tf('s')
```

```
char = (Anew1+( -3 ))*(Anew1+( -3 ))*(Anew1+( -2.4 ))*(Anew1+(-2.4 ))
G = [Cnew1;Cnew1*Anew1;Cnew1*Anew1*Anew1;Cnew1*Anew1*Anew1*Anew1]
g = char*inv(G)*[0;0;0;1]
% % %
Ao = Anew1-g*Cnew1-Bnew1*k1
Bo = g
Co =-k1
Do = Dnew1
% % %
syso = ss(Ao, Bo, Co, Do)
tfo = tf(syso)
% % %
% % %
[k1,l1,sysf1,sysf1f]=FullstateFeedback(tfnew1,P1);
step(sysf1);
step(sysf1f)
[k2,l2,sysf2,sysf2f]=FullstateFeedback(tfnew2,P2);
step(sysf2);
step(sysf2f)
%% full-state feedback
[num,den]=tfdata(sys,'v');

[A,b,c,d]=tf2ss(num,den);

k=place(A,b,eigs);

syms s
I=eye(4);
G0fs=simplify(eval(c*inv(s*I-A+b*k)*b));

[n,d]=numden(G0fs);
G0fs=tf(sym2poly(n),sym2poly(d));
precomp=0.05/0.514;

G0fs=precomp*G0fs;
%% estimator
eigs2=3*eigs;
l=place(A',c',eigs2);
l=l';
Gest=k*inv(s*I-A+b*k+l*c)*l;

[n,d]=numden(Gest);
Gest=tf(sym2poly(n),sym2poly(d));
precomp2=0.05/11.9;
G0est=precomp2*feedback(Gest*sys,1);
```