Gina Cody School of Engineering and Computer Science
Department of Electrical and Computer Engineering

Project Report on

The Design Analysis of The Magnetic Levitation System using MATLAB

A report prepared to fulfill the requirements of ENGR 6131

Submitted to:
Dr. Kash Khorasani

Prepared by:
Shrawan Kumaar Kannan (Student ID: 40169539)
Pranav Kumar Jha (Student ID: 40081750)

15th December, 2021

Abstract

This project focuses on the design of a suitable control system for the ECP Model 730 magnetic levitation system. We have developed a system model and identified its non-linearities, followed by the creation of a linearized model. Input-output equations, state and output equations, and transfer functions of the open-loop system have been provided. We developed controllable, observable, and Jordan canonical forms for the open loop system and provided impulse response and step response, along with bode plots and root loci. In addition, we designed a PID controller and provided simulation results, demonstrating transient and steady-state characteristics, including responses to step, square wave, and sinusoidal inputs. We also showed the robustness of the controllers. A full state feedback controller was designed to meet project specifications, with step, square wave, and sinusoidal responses provided with arbitrary initial conditions. The design of a full-order observer is also presented, with step, square wave, and sinusoidal responses provided. Finally, we conducted a comparative study between classical control theory design approaches and modern control theory design approaches.

Contents

1 Introduction

The project report is prepared to address and provide in sufficient details to the following design and analysis issues, validation and verification through simulations:

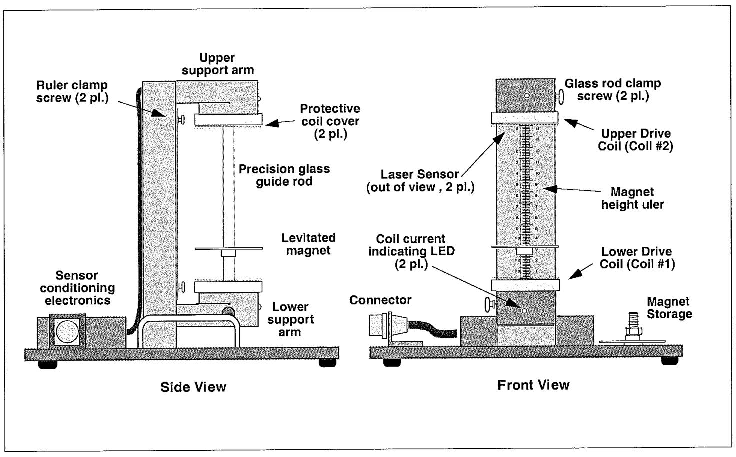
1. Input-output equations of the system, and state and output equations.
2. Transfer function of the open-loop system.
3. Controllable, Observable, and Jordan canonical forms in Step (1).
4. Impulse response and the step response in Step (1) with arbitrary initial conditions.
5. Bode plot of the uncompensated system as well as the root-locus of the open-loop system.
6. Design of a Lead-lag or PID controller to meet certain design specifications with transient as well as steady state characteristics.
7. Step response, square wave and sinusoidal responses.
8. Control input signal for each set point input in Step (7).
9. Robustness of the design by introducing noise and parameter variations or uncertainty in the system.
10. Design of a full state feedback control to meet the design specification indicated in Step (6).
11. Step, square wave and sinusoidal responses with arbitrary initial conditions. and comparison of the results with the results in Step (7).
12. Plot of the control input signal for each set point input in Step (11) and comparison of the result with the result in Step (8).
13. Design of a full-order and a reduced-order observer with step and sinusoidal responses.
14. Transfer function of the observer and controller and the type of controller.
15. Comparative study between the classical control design advantages and disadvantages with that of modern control theory design.
16. Justifications for our comparative study through numerical simulations.

1.1 Experimental Setup

Fig. 1.1(a) [1] shows the experimental set of model 730 Magnetic Levitation apparatus. This setup consists of two coils mounted on top and bottom that can be energized separately to levitate one or two magnets along a glass rod. This can be configured as single-input single-output (SISO), single-input multiple-output (SIMO), multiple-input single-output (MISO) or multiple-input multiple-output (MIMO). In this report, control of magnet position using upper and lower coils is considered which makes this setup a MIMO system.

The Maglev Apparatus plant as shown in Fig. 1.1(b) [2] consists of upper and lower coils which produce a magnetic field when we provide the DC current through the coils. The precision glass guide rods are used by the magnets to move along, vertically. The lower coil is energized to produce a repulsive magnetic force to levitate the magnet at the desired position and the upper coil is using the attractive force to levitate the magnet. The stronger the magnetic field strength, the higher will be the levitating magnet height and this can be controlled by increasing the coil current which, in turn, will increase the strength of the magnetic field [3].

(a) The Model 730 Magnetic Levitation



(b) MagLev Apparatus

Figure 1.1: The experimental setup of Magnetic Levitating System using Model 730

1.2 Overview of Real-Time Control Systems

The overview of a real-time control system is presented in Fig. 1.2 [2] which consists of three sub-systems which are:

1. The Mechanism which includes 'Actuators' and 'Sensors',
2. The Drive Electronics, also called as "Control Box" and,
3. User/System Interface "Executive" Program.

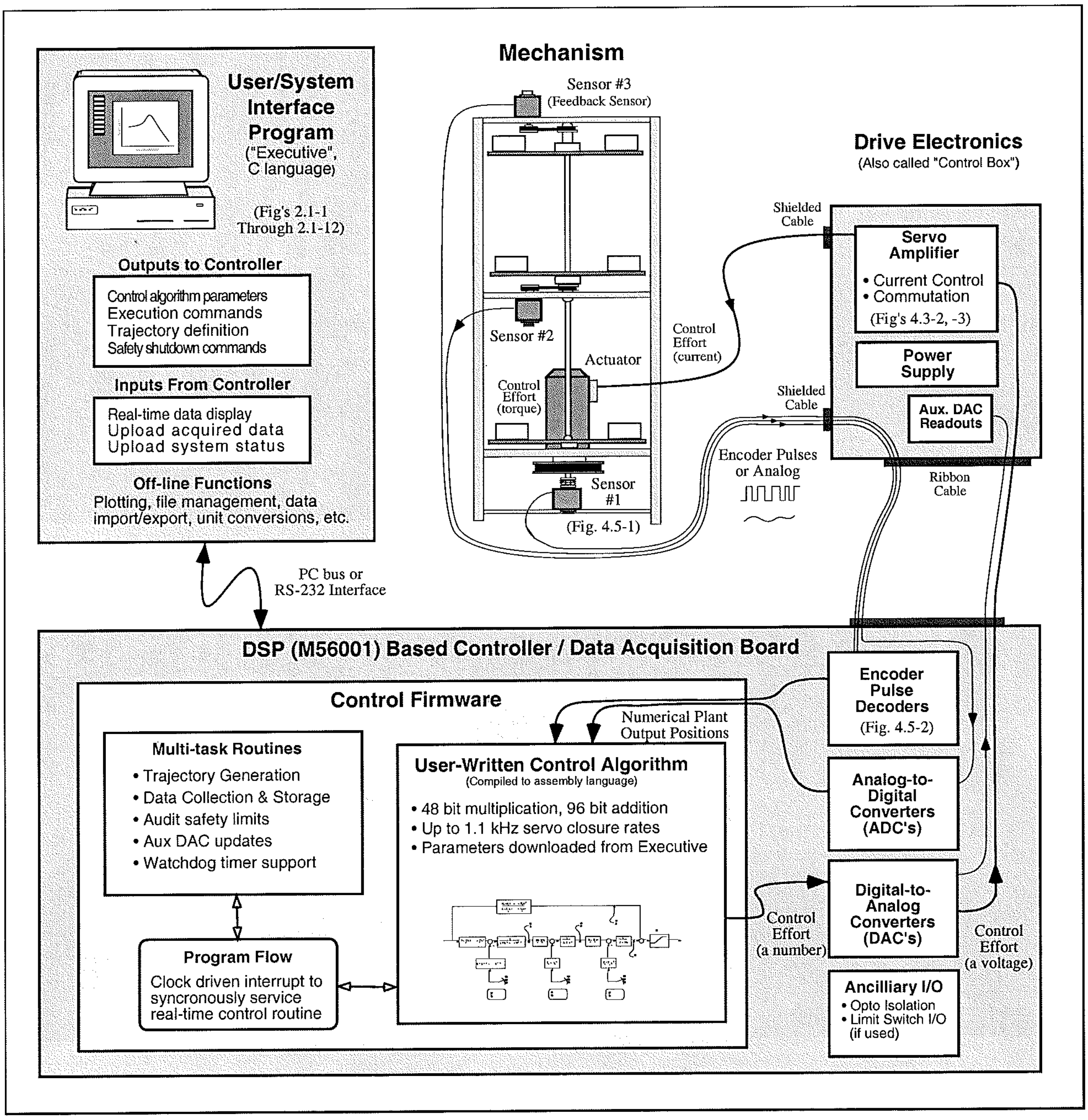


Figure 1.2: Overview of Real-Time Control Systems

2 Problem Statement

2.1 Full Order Nonlinear Model

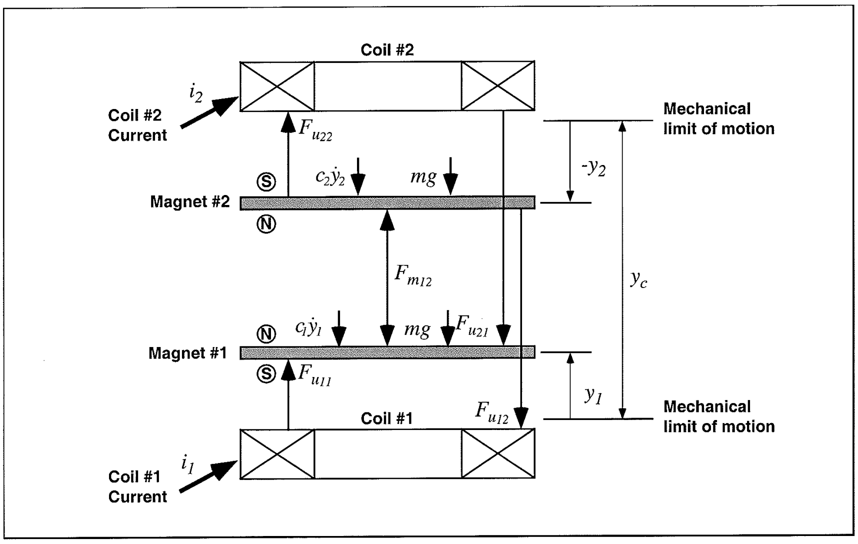


Figure 2.1: Free body diagram and dynamic configuration

Fig. 2.1 shows the free body diagram of two suspended magnets in the Model 730 apparatus. From Fig. 2.1, the resulting force equation for the first magnet is [2]

$$m\ddot{y}_1 + c_1\dot{y}_1 + F_{m_{12}} = F_{u_{11}} - F_{u_{21}} - mg, \tag{2.1.1}$$

and similarly, the force equation for the second magnet is

$$m\ddot{y}_2 + c_1\dot{y}_2 - F_{m_{12}} = F_{u_{22}} - F_{u_{12}} - mg. \tag{2.1.2}$$

The magnetic force terms are modeled as

$$F_{u_{11}} = \frac{i_1}{a(y_1+b)^N}, \tag{2.1.3a}$$

$$F_{u_{12}} = \frac{i_1}{a(y_c+y_2+b)^N}, \tag{2.1.3b}$$

$$F_{u_{21}} = \frac{i_2}{a(y_c-y_1+b)^N}, \tag{2.1.3c}$$

$$F_{u_{22}} = \frac{i_2}{a(-y_2+b)^N}, \tag{2.1.3d}$$

$$F_{m_{12}} = \frac{c}{(y_{12}+d)^N}, \tag{2.1.3e}$$

where,

$$y_{12} = y_c + y_2 - y_1, \tag{2.1.4}$$

and a, b, c, d and N are constants which can be determined by numerical modeling of the magnetic configurations. The value of N typically lies between 3 and 4.5.

2.2 Simplified Equations of Motion

The simplified equations of motion are given as [2]

$$m\ddot{y}_1 + F_{m_{12}} = F_{u_{11}} - mg, \tag{2.2.1}$$

$$m\ddot{y}_2 - F_{m_{12}} = F_{u_{22}} - mg, \tag{2.2.2}$$

where,

$$F_{u_{11}} = \frac{u_1}{a(y_1+b)^4}, \tag{2.2.3a}$$

$$F_{u_{22}} = \frac{u_2}{a(-y_2+b)^4}, \tag{2.2.3b}$$

$$F_{m_{12}} = \frac{c}{(y_{12}+d)^4}. \tag{2.2.3c}$$

The parameters given are mentioned below in Table 2.1, where the values for y_1^o, y_2^o and y_c^o are operating point values [2].

Table 2.1: Parameters used in this experimental setup

Parameters	**Value**
y_1^o	2.00 cm
y_2^o	-2.00 cm
y_c^o	12.00 cm
N	4
m	120 g
a	1.65
b	6.2
c	2.69
d	4.2

2.3 Linearized Equations of Motion

The linearized equations of motion can be obtained by solving the simplified equations of motion in (2.2.1) and (2.2.2) using the Taylor's series expansions at the operating points up to the first order terms only [4, 5]. Considering the equation (2.2.1) as 'f' and defining the operating points as y_1^o, y_2^o, y_{12}^o, u_1^o and u_2^o, the Taylor's series expansion of f is written as

$$f(y_1^o, y_2^o, u_1^o, t) + \left.\frac{\partial f}{\partial y_1}\right|_{\substack{y_1^o \\ y_2^o \\ u_1^o}} * (y_1 - y_1^o) + \left.\frac{\partial f}{\partial y_2}\right|_{\substack{y_1^o \\ y_2^o \\ u_1^o}} * (y_2 - y_2^o) + \left.\frac{\partial f}{\partial u_1}\right|_{\substack{y_1^o \\ y_2^o \\ u_1^o}} * (u_1 - u_1^o). \tag{2.3.1}$$

The chosen operating points are generally considered as equilibrium points which satisfy the equation

$$F_{m_{12}} - F_{u_{11}} + mg\Big|_{\substack{y_1^o \\ y_2^o \\ u_1^o}} = 0. \tag{2.3.2}$$

Using the results from (2.2.1) and (2.3.2) in (2.3.1), the resulting expression will be

$$m\ddot{y}_1 + \left(\frac{4c}{(y_{12}^o + d)^5} + \frac{4u_1^o}{a(y_1^o + b)^5}\right)(y_1 - y_1^o) - \frac{4c}{(y_{12}^o + d)^5}(y_2 - y_2^o) = \frac{1}{a(y_1^o + b)^4}(u_1 - u_1^o), \tag{2.3.3}$$

which can be written as,

$$m\ddot{\hat{y}}_1 + (w_{y_1} + w_{y_{12}})\hat{y}_1 - w_{y_{12}}\hat{y}_2 = w_{u_1}\hat{u}_1. \tag{2.3.4}$$

Similarly, for (2.2.2), the resulting expression will be

$$m\ddot{y}_2 + \left(\frac{4c}{(y_{12}^o + d)^5} - \frac{4u_2^o}{a(-y_2^o + b)^5}\right)(y_2 - y_2^o) - \frac{4c}{(y_{12}^o + d)^5}(y_1 - y_1^o) = \frac{1}{a(-y_2^o + b)^4}(u_2 - u_2^o), \tag{2.3.5}$$

which can be written as,

$$m\ddot{\hat{y}}_2 + (w_{y_{12}} - w_{y_2})\hat{y}_2 - w_{y_{12}}\hat{y}_1 = w_{u_2}\hat{u}_2, \tag{2.3.6}$$

where,

$$\hat{y}_i = y_i - y_i^o, \quad i = 1, 2, \tag{2.3.7a}$$

$$\hat{u}_i = u_i - u_i^o, \quad i = 1, 2, \tag{2.3.7b}$$

$$w_{y_1} = \frac{4u_1^o}{a(y_1^o + b)^5}, \tag{2.3.7c}$$

$$w_{y_2} = \frac{4u_{2_o}}{a(-y_2^o + b)^5}, \tag{2.3.7d}$$

$$w_{y_{12}} = \frac{4c}{(y_{12}^o + d)^5}, \tag{2.3.7e}$$

$$w_{u_1} = \frac{1}{a(y_1^o + b)^4}, \tag{2.3.7f}$$

$$w_{u_2} = \frac{1}{a(-y_2^o + b)^4}, \tag{2.3.7g}$$

The equilibrium point values for the input currents can be defined from (2.3.2) as,

$$u_1^o = a(y_1^o + b)^4\left(\frac{4c}{(y_{12}^o + d)^5} + mg\right), \tag{2.3.8}$$

and

$$u_2^o = a(y_2^o + b)^4\left(\frac{4c}{(y_{12}^o + d)^5} + mg\right). \tag{2.3.9}$$

3 Design Specifications, Methodology, Analysis and Results

3.1 State Space Representation

The state space representation of this 4th order MIMO system by defining four state variables as $x_1 = \hat{y}_1$, $x_2 = \hat{y}_2$, $x_3 = \dot{\hat{y}}_1$, and $x_4 = \dot{\hat{y}}_2$ will be

$$\begin{aligned} \dot{x} &= Ax + BU(t), \qquad \text{(state equations)} \\ Y &= Cx + DU(t), \qquad \text{(output equation)} \end{aligned} \tag{3.1.1}$$

where A, B and C is given as,

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -w_{y_1}/m - w_{y_{12}}/m & w_{y_{12}}/m & 0 & 0 \\ w_{y_{12}}/m & -w_{y_2}/m - w_{y_{12}}/m & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ w_{u_1}/m & 0 \\ 0 & w_{u_2}/m \end{bmatrix}, \tag{3.1.2}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad U(t) = \begin{bmatrix} \hat{u}_1(t) \\ \hat{u}_2(t) \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad y = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix}.$$

After calculating the values of the elements of these matrices based on the parameters given in Table 2.1, we have,

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -6.4113 & 0.0624 & 0 & 0 \\ 0.0624 & -6.1247 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.0034 & 0 \\ 0 & 0.0034 \end{bmatrix}, \tag{3.1.3}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

The space state model is generated using MATLAB and the following command:

```
sys=ss(A,B,C,D);
```

and the complete code for our project is presented in the 'Appendix' section at the end of the report.

3.2 Transfer Function of The Open Loop System

Based on the state space representation, the transfer function G of the open loop system is given as

$$G(s) = \frac{Y(s)}{X(s)} = C(sI - A)^{-1}B + D, \tag{3.2.1}$$

which is further calculated based on the parameters given in Table 2.1, and given as,

$$G(s) = \begin{bmatrix} \frac{0.003374s^2+0.02067}{s^4+12.54s^2+39.26} & \frac{0.0002107}{s^4+12.54s^2+39.26} \\ G\frac{0.0002107}{s^4+12.54s^2+39.26} & \frac{0.003374s^2+0.02163}{s^4+12.54s^2+39.26} \end{bmatrix}. \tag{3.2.2}$$

As discussed earlier in Section 2, our system is a MIMO system which has two inputs u_1 and u_2 and two outputs y_1 and y_2. Therefore, for each input u_i we will have the corresponding transfer function $G^i(s) = \frac{U_i(s)}{Y_i(s)},\ i = 1,2$ for the open loop system with different zeros but with the same poles[1]. The transfer function can be obtained from the following MATLAB command:

```
G = tf(sys);
```

$$G^1(s) = \frac{0.003374s^2 + 0.02067}{s^4 + 12.54s^2 + 39.26}, \tag{3.2.3a}$$

$$G^2(s) = \frac{0.003374s^2 + 0.02163}{s^4 + 12.54s^2 + 39.26}. \tag{3.2.3b}$$

[1]The superscript notations $i = 1,2$ used for $G^i(s)$ in this section of the report should not be mistaken as 'Exponents'. We have considered this in order to identify the associated terms without any confusion which may arise if multiple symbols will be used as subscripts.

3.3 Controllable, Observable and Jordan Canonical Forms

3.3.1 Controllable and Observable Canonical Forms

The open loop transfer functions in (3.2.3a) and (3.2.3b) will provide C_x and O_x for their corresponding open loop SISO systems which constitute the complete MIMO system in our project.

For open loop system with input u_1 and output y_1, the state space matrices are given as,

$$A^1 = \begin{bmatrix} 0 & -12.5359 & 0 & -39.2629 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad B^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{3.3.1}$$

$$C^1 = \begin{bmatrix} 0 & 0.0034 & 0 & 0.0207 \end{bmatrix}, \quad D^1 = \begin{bmatrix} 0 \end{bmatrix}.$$

The controllability of open loop system in (3.3.1) can be checked using the following MATLAB command:

```
C1x=ctrb(A1,B1);
```

$$C_x^1 = \begin{bmatrix} 1 & 0 & -12.5359 & 0 \\ 0 & 1 & 0 & -12.5359 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.3.2}$$

which is a full rank matrix which makes the system controllable for input u_1 and output y_1. The state space matrices for our controllable canonical form is given as,

$$A_c^1 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ -39.2629 & 0 & -12.5359 & 0 \end{bmatrix}, \quad B_c^1 = \begin{bmatrix} 0 \\ 0.0034 \\ 0 \\ -0.0216 \end{bmatrix}, \tag{3.3.3}$$

$$C_c^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, \quad D_c^1 = \begin{bmatrix} 0 \end{bmatrix}.$$

where C_c^1 is calculated using $T_c^1 = C_x^1 {C_{x_c}^1}^{-1}$ and $C_c^1 = C^1 T_c^1$

$$C_{x_c}^1 = \begin{bmatrix} & 0.0034 & 1 & -0.0216 \\ 0.0034 & 0 & -0.0216 & 0 \\ 0 & -0.0216 & 1 & 0.1387 \\ -0.0216 & 0 & 0.1387 & 0 \end{bmatrix} \tag{3.3.4}$$

For observability of the open loop system in (3.3.1) can be checked using the following MATLAB command:

```
O1x=obsv(A1,C1);
```

$$O_x^1 = \begin{bmatrix} 0 & 0.0034 & 0 & 0.0207 \\ 0.0034 & 0 & 0.0207 & 0 \\ 0 & -0.0216 & 0 & -0.1325 \\ -0.0216 & 0 & -0.1325 & 0 \end{bmatrix}, \tag{3.3.5}$$

which is a full rank matrix which makes the system observable for input u_1 and output y_1. The state space matrices for our observable canonical form is given as,

$$A_o^1 = \begin{bmatrix} 0 & 0 & 0 & -39.2629 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -12.5359 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad B_o^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{3.3.6}$$

$$C_o^1 = \begin{bmatrix} 0 & 0.0034 & 0 & -0.0216 \end{bmatrix}, \quad D_o^1 = \begin{bmatrix} 0 \end{bmatrix}.$$

where C_o^1 is calculated using ${T_o^1}^{-1} = {O_{x_o}^1}^{-1} O_x^1$ and $C_o^1 = C^1 T_o^1$

$$O_{x_o}^1 = \begin{bmatrix} 0 & 0.0034 & 1 & -0.0216 \\ 0.0034 & 0 & -0.0216 & 1 \\ 0 & -0.0216 & 0 & 0.1387 \\ -0.0216 & & 0.1387 & 0 \end{bmatrix} \tag{3.3.7}$$

Based on the 'controllability' and 'observability' results, it is evident that the open loop system in (3.3.1) is controllable as well as observable.

[2]For open loop system with input u_2 and output y_2, the state space matrices are given as,

$$A^2 = \begin{bmatrix} 0 & -12.5359 & 0 & -39.2629 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad B^2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{3.3.8}$$

$$C^2 = \begin{bmatrix} 0 & 0.0034 & 0 & 0.0216 \end{bmatrix}, \quad D^2 = \begin{bmatrix} 0 \end{bmatrix}.$$

[2]The superscript notations $i = 1, 2$ used for A^i, B^i, C^i, D^i in this section of the report should not be mistaken as 'Exponents'. We have considered this in order to identify the associated terms without any confusion which may arise if multiple symbols will be used as subscripts.

[3]The controllability of the open loop system in (3.3.8) can be checked using the following MATLAB command:

```
C2x=ctrb(A2,B2);
```

$$C_x^2 = \begin{bmatrix} 1 & 0 & -12.5359 & 0 \\ 0 & 1 & 0 & -12.5359 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.3.9}$$

which is a full rank matrix which makes the system controllable for input u_2 and output y_2. The state space matrices for our controllable canonical form is given as,

(3.3.10)

where C_c^2 is calculated using $T_c^2 = C_x^2 {C_{x_c}^2}^{-1}$ and $C_c^2 = C^2 T_c^1$

$$C_{x_c}^2 = \begin{bmatrix} 0 & 0.0034 & 0 & -0.0207 \\ 0.0034 & 0 & -0.0207 & 1 \\ 0 & -0.0207 & 0 & 0.1266 \\ -0.0207 & 0 & 0.1266 & 0 \end{bmatrix} \tag{3.3.11}$$

[3]The superscript notations $i = 1,2$ used for $A_c^i, B_c^i, C^i, C_c^i, C_x^i, C_{x_c}^i, D_c^i, T_c^i$ in this section of the report should not be mistaken as ‘Exponents’. We have considered this in order to identify the associated terms without any confusion which may arise if multiple symbols will be used as subscripts.

For observability of the open loop system in (3.3.8) can be checked using the following MATLAB command:

```
O2x=obsv(A2,C2);
```

$$O_x^2 = \begin{bmatrix} 0 & 0.0034 & 1 & 0.0216 \\ 0.0034 & 0 & 0.0216 & 1 \\ 0 & -0.0207 & 0 & -0.1325 \\ -0.0207 & 0 & -0.1325 & 0 \end{bmatrix}, \tag{3.3.12}$$

which is a full rank matrix which makes the system observable for input u_2 and output y_2. [4]The state space matrices for our observable canonical form is given as,

$$A_o^2 = \begin{bmatrix} 0 & 0 & 0 & -39.2629 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -12.5359 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad B_o^2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{3.3.13}$$

$$C_o^2 = \begin{bmatrix} 0 & 0.0034 & 0 & -0.0207 \end{bmatrix}, \quad D_o^2 = \begin{bmatrix} 0 \end{bmatrix}.$$

where C_o^2 is calculated using ${T_o^2}^{-1} = {O_{x_o}^2}^{-1} O_x^2$ and $C_o^2 = C^2 T_o^2$

$$O_{x_o}^2 = \begin{bmatrix} 0 & 0.0034 & 1 & -0.0207 \\ 0.0034 & 0 & -0.0207 & 1 \\ 0 & -0.0207 & 0 & 0.1266 \\ -0.0207 & 0 & 0.1266 & 0 \end{bmatrix} \tag{3.3.14}$$

Based on the ‘controllability’ and ‘observability’ results, it is evident that the SISO system in (3.3.8) is controllable as well as observable.

[4]The superscript notations $i = 1,2$ used for $A_o^i, B_o^i, C^i, C_o^i, O_x^i, O_{x_o}^i, D_o^i, T_o^i$ in this section of the report should not be mistaken as ‘Exponents’. We have considered this in order to identify the associated terms without any confusion which may arise if multiple symbols will be used as subscripts.

3.3.2 Jordan Canonical Form

The Jordan canonical form gives us a diagonal matrix A_j with eigenvalues at it's diagonal. This can be generated using the following MATLAB command:

```
[M,R] = eig(A);
Aj = jordan(A);
Bj = inv(M)*B;
Cj = C*M;
Dj = D;
```

This gives us the jordan matrices as,

$$A_j = \begin{bmatrix} -2.5346i & 0 & 0 & 0 \\ 0 & -2.4722i & 0 & 0 \\ 0 & 0 & 2.5346i & 0 \\ 0 & 0 & 0 & 2.4722i \end{bmatrix}, \quad B_j = \begin{bmatrix} 0.0018 & -0.004 \\ 0.0018 & -0.004 \\ -0.004 & -0.0018 \\ -0.004 & -0.0018 \end{bmatrix}, \tag{3.3.15}$$

$$C_j = \begin{bmatrix} -0.3593i & 0.3593i & 0.0765i & -0.0765i \\ 0.0749i & -0.0749i & 0.3671i & -0.3671i \end{bmatrix}, \quad D_j = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

3.4 Impulse Response and Step Response

The impulse response can be obtained taking the inverse of the Laplace transformation of the transfer function as given in (3.4.1), where initial conditions are assumed to be zero.

$$g(t) = L^{-1}\{G(s)\} \tag{3.4.1}$$

This can be done using the following MATLAB command:

```
impulse(sys);
step(sys);
```

The impulse and step responses of open loop systems in (3.3.1) and (3.3.8) are presented as,

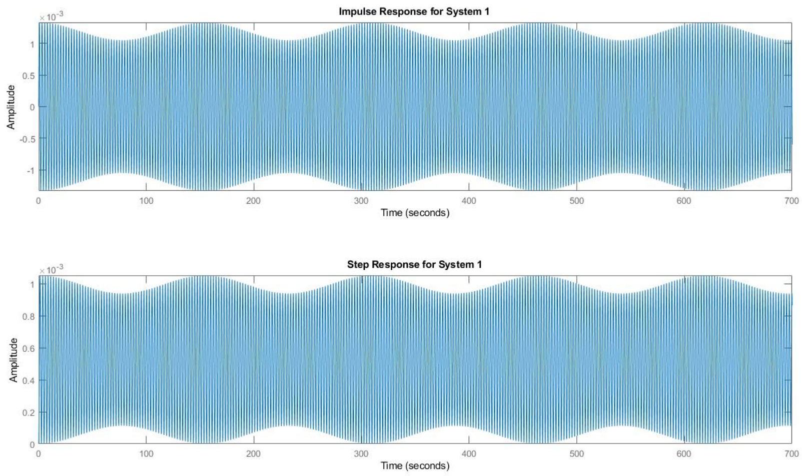


Figure 3.1: Impulse and Step Response for The First System

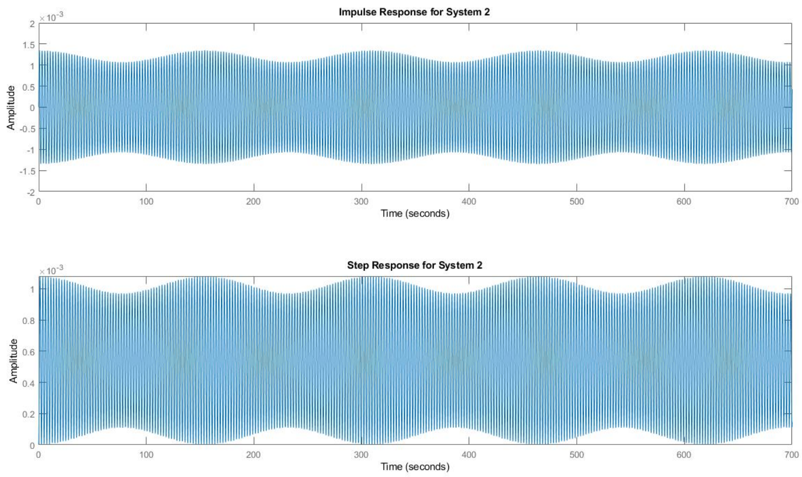


Figure 3.2: Impulse and Step Response for The Second System

Based on the impulse and step responses, we need a controller for the system to make it stable.

3.5 Bode Plot and Root Locus of the Uncompensated System

The Bode plot and the Root locus of the open loop transfer functions $G^1(S)$ and $G^2(S)$ in (3.2.3a) and (3.2.3b), respectively, are presented as,

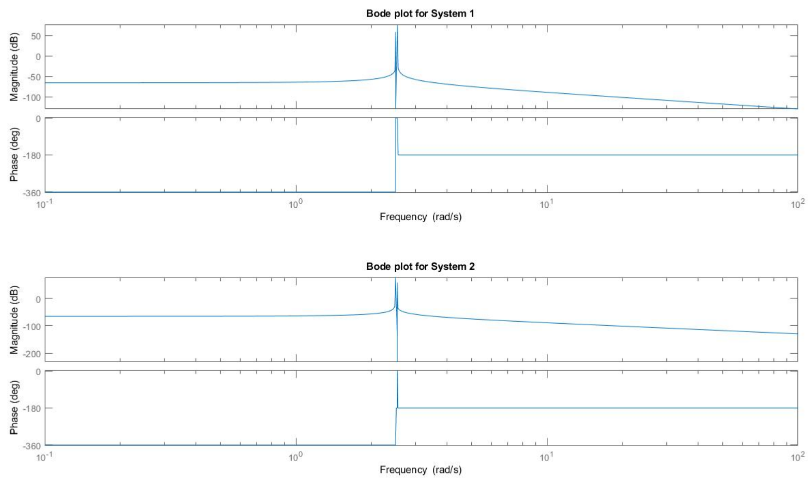


Figure 3.3: Bode Plot of the open loop systems

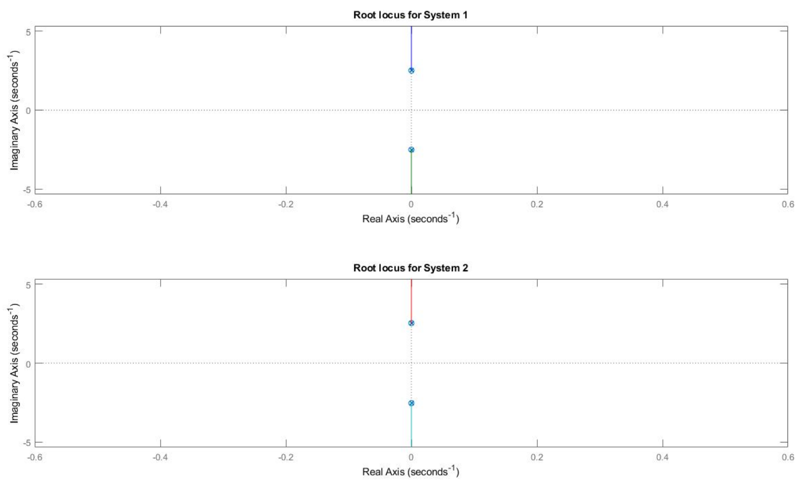


Figure 3.4: Root Locus of the open loop systems

3.6 PID Controller Design

The controller parameters selected for our experiment to meet with the requirements of the output of the transfer functions $G^1(s)$ and $G^2(s)$ in (3.2.3a) and (3.2.3b), respectively, are given as,

- Damping Ratio ξ: 0.707
- Time Constant τ: 1s

The obtained response provides the desired settling time, which is 2s, for step input in both cases. The most suitable parameters found for our experiment is based on the following design specifications:

3.6.1 Controller for $G^1(s)$

To find the suitable controller for $G^1(s)$ in (3.2.3a) for our design purposes, we have used the step input for tuning and considered the gains $K_d = 2000$, $K_p = 6000$ and $K_i = 100003$. The open loop transfer function of the controller is given as,

$$G^1(s) = \frac{2000s^2 + 6000s + 10000}{s}, \tag{3.6.1}$$

and the closed loop transfer function is given as,

$$G^1(s) = \frac{6.748s^4 + 20.25s^3 + 75.07s^2 + 124s + 206.7}{s^5 + 6.748s^4 + 32.78s^3 + 75.07s^2 + 163.3s + 206.7}, \tag{3.6.2}$$

3.6.2 Controller for $G^2(s)$

[5]To design the suitable controller for $G^2(s)$ in (3.2.3b), we have used the step input for tuning and considered the gains $K_d = 1500$, $K_p = 4000$ and $K_i = 8000$. The open loop transfer function of the controller is given as,

$$G^2(s) = \frac{1500s^2 + 4000s + 8000}{s}, \tag{3.6.3}$$

and the closed loop transfer function is given as,

$$G^2(s) = \frac{5.061s^4 + 13.5s^3 + 59.44s^2 + 86.53s + 173.1}{s^5 + 5.061s^4 + 26.03s^3 + 59.44s^2 + 125.8s + 173.1}, \tag{3.6.4}$$

[5]The superscript notations $i = 1,2$ used for $G^i(s)$ in this section of the report should not be mistaken as 'Exponents'. We have considered this in order to identify the associated terms without any confusion which may arise if multiple symbols will be used as subscripts.

3.7 Step Response, Square Response and Sine Wave Response for Closed Loop System

3.7.1 Step Response

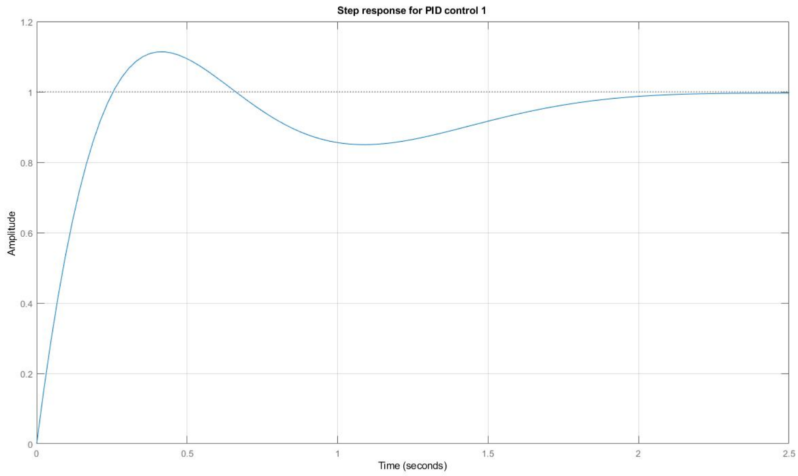


Figure 3.5: Step Response of The First Controller

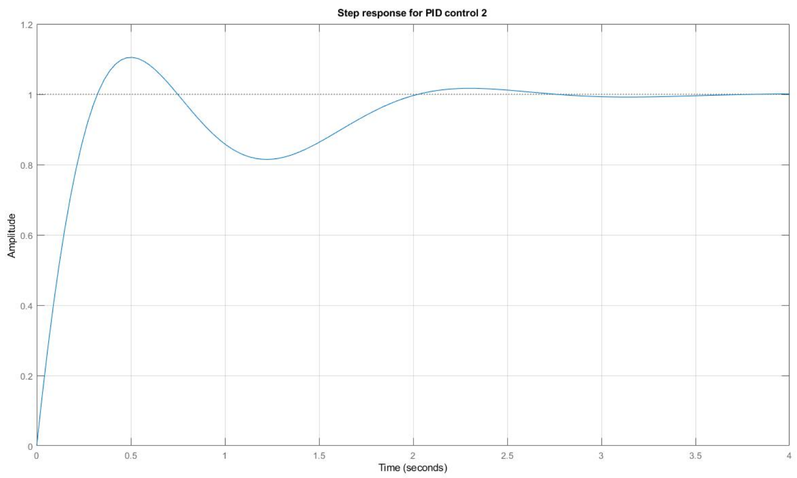


Figure 3.6: Step Response of The Second Controller

Settling time for $G^1(s) \cong 1.9026s$,

Overshoot for $G^1(s) = 11.3591$

Settling time for $G^2(s) \cong 1.9131s$.

Overshoot for $G^2(s) = 10.8883$

3.7.2 Square Wave Response

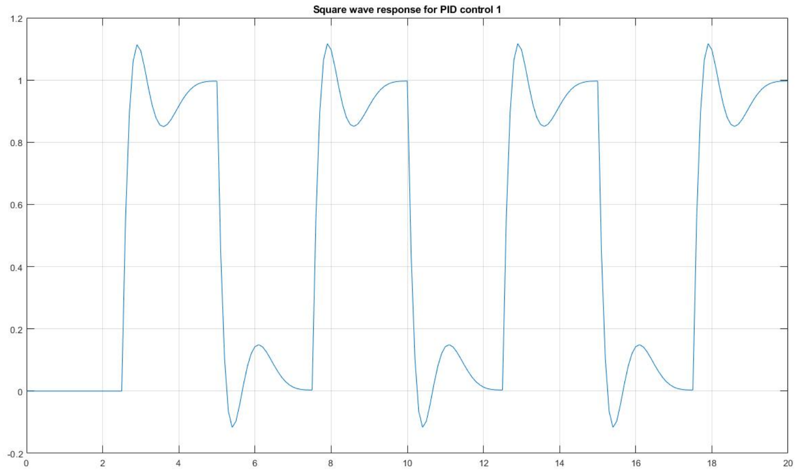


Figure 3.7: Square Wave Response of The First Controller

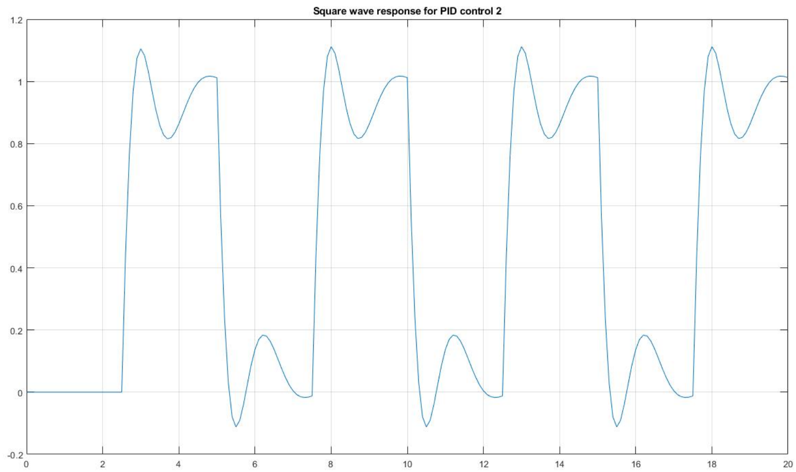


Figure 3.8: Square Wave Response of The Second Controller

3.7.3 Sinusoidal Response

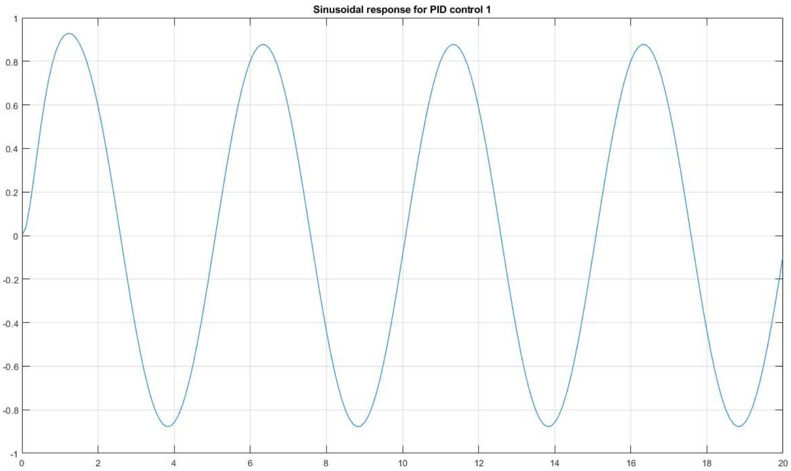


Figure 3.9: Sinusoidal Response of The First Controller

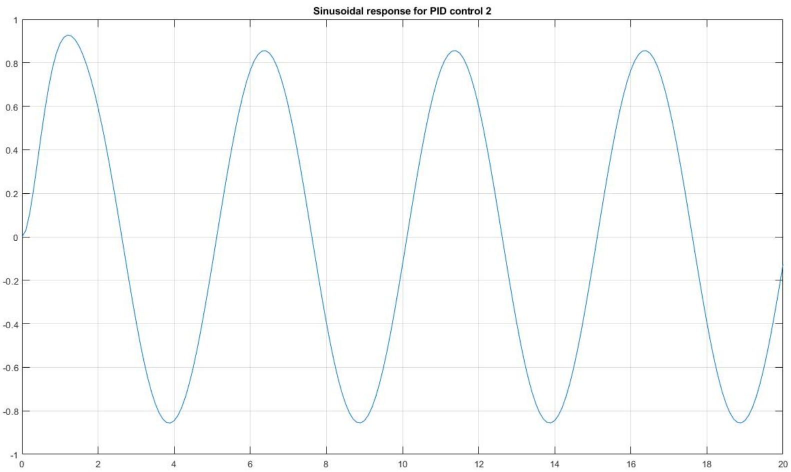


Figure 3.10: Sinusoidal Response of The Second Controller

3.8 Robustness

The robustness of the system can be examined by observing the bode plot for the closed loop transfer functions of the system. This can also be determined by plotting the response of the closed loop system when a step disturbance input enters into it.

3.8.1 Robustness of The Controllers for $G^1(s)$ and $G^2(s)$

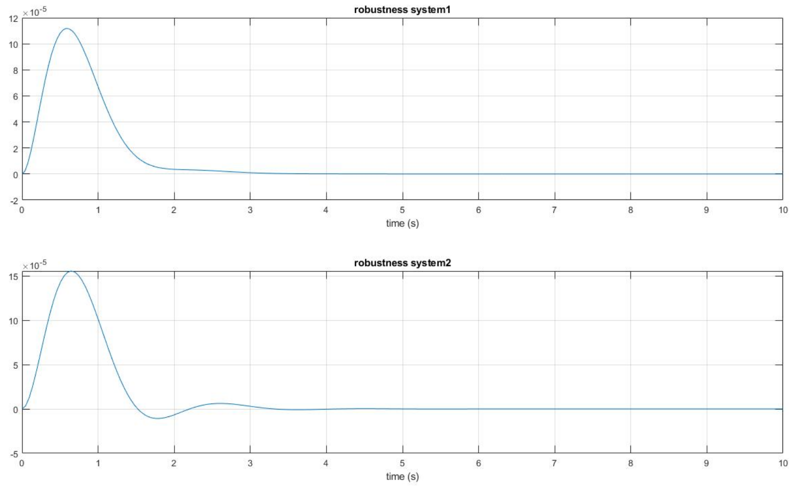


Figure 3.11: The disturbance output of the controllers

3.9 Full State Feedback Control

According to root locus diagram for open loop transfer function, two of the zeros are cancelled by poles, so the system acts as a second order system. We are going to place two poles at desired locations as follows: The design of this state feedback controller is done with the below transient specifications:

- Damping ratio ξ: 0.707
- Settling Time τ_s: 4s

In order to comply with the specifications, the system will be approximated to a second order system with the following characteristic polynomial:

$$\tau_s = 4\tau = 4s; \;\; \xi = 0.707; \;\; \xi\omega_n = 1/\tau; \;\; \omega_n = 1.414$$

The characteristic polynomial with poles at $-1 \pm 1i$ will be approximated as,

$$s^2 + 2\xi\omega_n s + \omega_n^2 = s^2 + 4s + 7.95. \tag{3.9.1a}$$

To find the values of the State feedback matrix K, the following is used as a MATLAB command:

```
%Pole placement
% system 1
Pole_1 = [2.4956i, -2.4956i, -1+1i, -1-1i];
K1 = place(A1,B1,Pole_1);
%System 2
Pole_2 = [2.5362i, -2.5362i, -1+1i, -1-1i];
K2 = place (A2,B2,Pole_2);
```

The obtained feedback matrices K_1 and K_2 are given as

$$K_1 = \begin{bmatrix} 2 & -4.4328 & 12.456 & -27.6074 \end{bmatrix}, K_2 = \begin{bmatrix} 2 & -4.2285 & 12.8646 & -27.1988 \end{bmatrix}. \tag{3.9.2}$$

[6]Given this State feedback matrix, the A matrix of the system will be changed as $A_f = A - BK$. This gives is matrices A_f^1 and A_f^2 given as,

$$A_f^1 = \begin{bmatrix} -2 & -8.228 & -12.456 & -12.456 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad A_f^2 = \begin{bmatrix} -2 & -8.4323 & -12.8646 & -12.8646 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{3.9.3}$$

3.10 Step Response, Square Wave Response, Sine Wave Response for Closed Loop System with State Feedback Control

The closed loop transfer functions obtained are given as,

$$G^1(s) = \frac{1.993s^2 + 12.46}{s^4 + 2s^3 + 8.228s^2 + 12.46s + 12.46}, \tag{3.10.1a}$$

$$G^2(s) = \frac{2.007s^2 + 12.86}{s^4 + 2s^3 + 8.432s^2 + 12.86s + 12.86}. \tag{3.10.1b}$$

[6]The superscript notations $i = 1,2$ used for A_f^i in this section of the report should not be mistaken as 'Exponents'. We have considered this in order to identify the associated terms without any confusion which may arise if multiple symbols will be used as subscripts.

3.10.1 Step Response

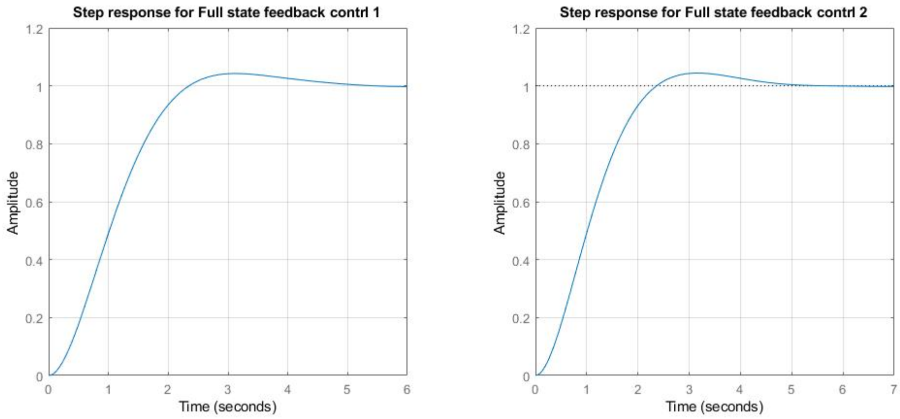


Figure 3.12: Step responses for the closed loop systems with state feedback control

3.10.2 Square Wave Response

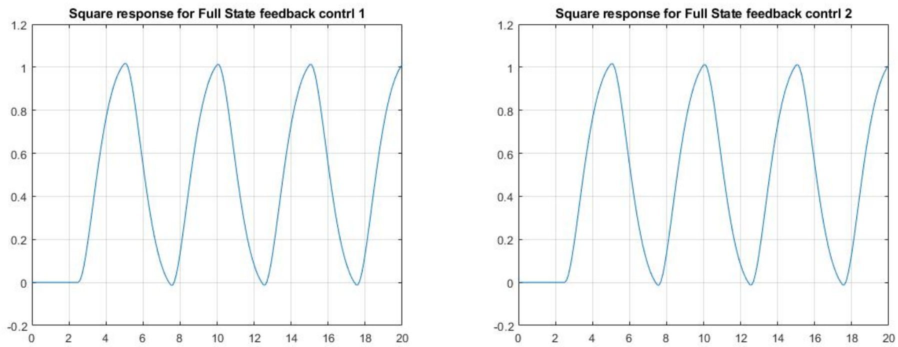


Figure 3.13: Square wave responses for the closed loop systems with state feedback control

3.10.3 Sinusoidal Response

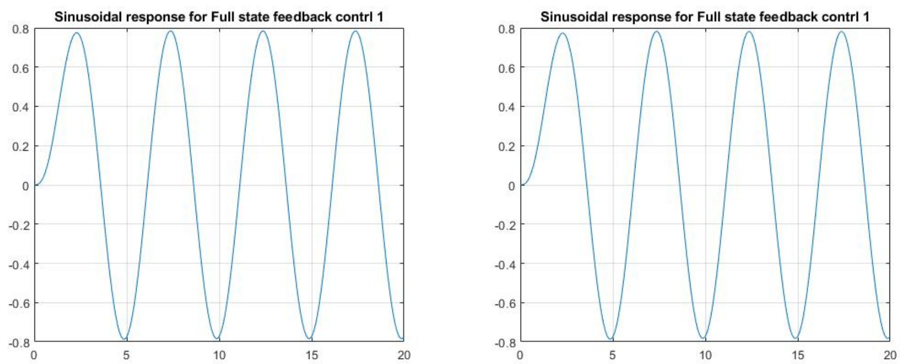


Figure 3.14: Sinusoidal responses for the closed loop systems with state feedback control

3.11 Full Order Observer Design

The time constant of the controlled system is calculated and set at $\tau = 1s$. To estimate the system based on our requirement, the required time constant of the observer needs to be 2 to 3 times faster. By choosing it 4 times of the original time constant, the parameters of the observer will be,

$$\tau = 1/3s; \;\; \xi = 0.979; \;\; ;\omega_n = 3.09$$

Based on the parameters mentioned above, the desired second order characteristic polynomial at poles $-3 \pm 0.5i$ will be given as,

$$s^2 + 6s + 9.55$$

The estimator matrices G_1 and G_2 are chosen as

$$\begin{aligned} G_1 &= \begin{bmatrix} 2.4956i & -2.4956i & -3+0.5i & -3-0.5i \end{bmatrix}, \\ G_2 &= \begin{bmatrix} 2.5362i & -2.5362i & -3+0.5i & -3-0.5i. \end{bmatrix}. \end{aligned} \tag{3.11.1}$$

In order to find the coefficients of G, the $\det(sI–A+GC) = \alpha_c(s)$, where $\alpha_c(s)$ is the characteristic polynomial formed from the desired poles locations. The full order observer state space equations can be obtained from $A_x^e = A - GC - BK$.

The full order observer transfer functions for open loop systems are given as,

$$G^1(s) = \frac{0.08659s^3 + 0.3705s^2 + 0.5393s + 2.307}{s^4 + 8s^3 + 23.05s^2 + 52.31s + 115.4}, \tag{3.11.2a}$$

$$G^2(s) = \frac{0.082s^3 + 0.3713s^2 + 0.5275s + 2.389}{s^4 + 8s^3 + 23.45s^2 + 53.65s + 119.4}. \tag{3.11.2b}$$

The full order observer transfer functions for closed loop systems are given as,

$$G^1(s) = \frac{6949s^3 + 2.973e04s^2 + 4.328e04s + 1.852e05}{s^4 + 8s^3 + 23.05s^2 + 52.31s + 115.4}, \tag{3.11.3a}$$

$$G^2(s) = \frac{6378s^3 + 2.888e04s^2 + 4.103e04s + 1.858e05}{s^4 + 8s^3 + 23.45s^2 + 53.65s + 119.4}. \tag{3.11.3b}$$

3.11.1 Step Response

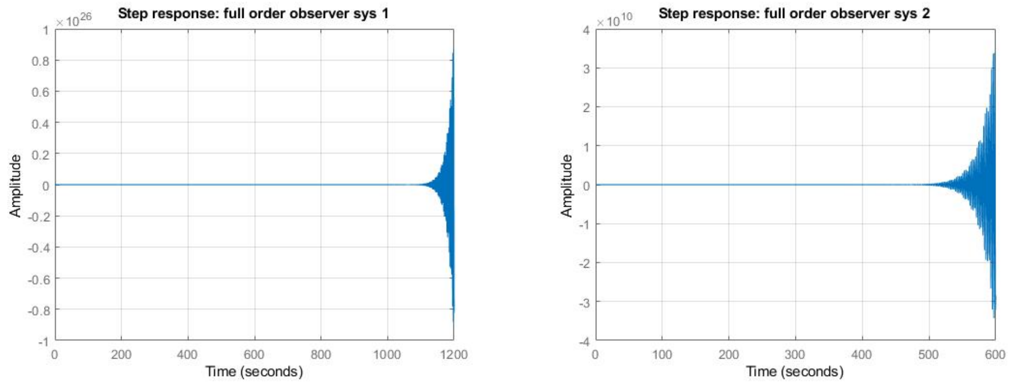


Figure 3.15: Full order observer step responses the systems

3.11.2 Square Wave Response

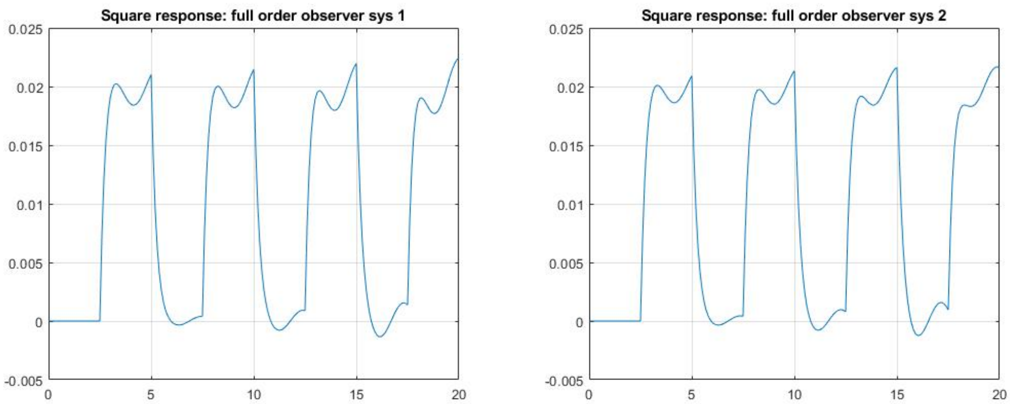


Figure 3.16: Full order observer square wave responses the systems

3.11.3 Sinusoidal Response

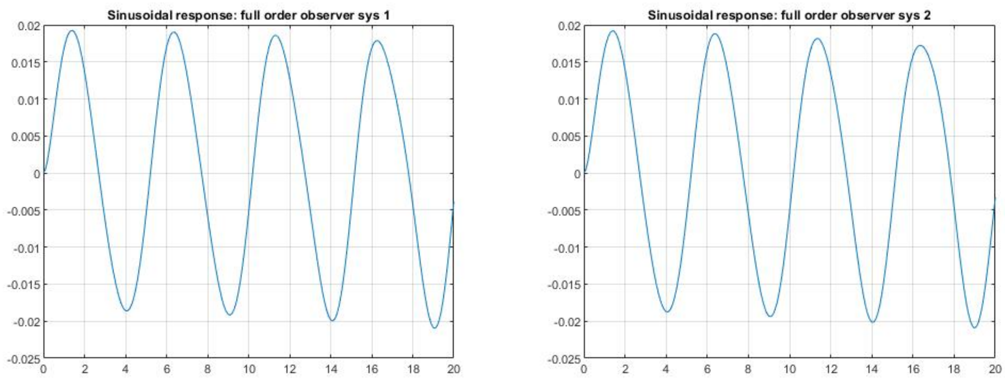


Figure 3.17: Full order observer sinusoidal responses the systems

4 Conclusion

In our report, we have considered the classical as well as modern approach to design a controller for our experimental setup of the magnetic levitation system model 750. The approaches to solve the controller design problem presented in the project description consist of classical control theory approach as well as modern control theory approach. The classical control theory approach requires much more time dealing with the trial and error method to find the suitable poles for the required system and could be beneficial in some cases, only if one has experience in hand. On the other hand, in the modern control theory approach, one can place the desired poles at a specific location, in order to make a system stable, based on certain mathematical calculations, matrix manipulations and a few assumption, which is time and energy efficient if were to compare with that of the classical theory approach. Modern control theory approach gives us mathematical insights about the controllability and observability of the system, based on which, it can be predicted that whether or not , a certain system could be made stable based on the design requirement and specifications and so, a lot of time and effort could be saved. To conclude, the modern theory approach is far more convenient and efficient technique for unstable systems as compared to classical theory approach.

References

[1] ECP. “Model 730: Magnetic levitation,” *Educational Control Products, 5725 Ostin Avenue, Woodland Hills, CA 91367*. [Online]. Available: http://www.ecpsystems.com/controls_maglevit.htm

[2] K. Khorasani. “Project description - magnetic levitation system” *The Department of Electrical and Computer Engineering, Concordia University*. [Online]. Available: https://moodle.concordia.ca/moodle/pluginfile.php/5051766/mod_resource/content/2/ENGR%206131_Project_Mag_Lev_2021.pdf

[3] D. Khimani, S. Karnik, and M. Patil, “Implementation of high performance nonlinear feedback control on magnetic levitation system,” *IFAC-PapersOnLine*, vol. 51, no. 1, pp. 13–18, 2018.

[4] L. Zhu and C. R. Knospe, “Modeling of nonlaminated electromagnetic suspension systems,” *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 1, pp. 59–69, 2009.

[5] D. Wang, F. Meng, and S. Meng, “Linearization method of nonlinear magnetic levitation system,” *Mathematical Problems in Engineering*, vol. 2020, 2020.

Appendix

MATLAB Code

The MATLAB code used to generate results in our project is given below:

```
% initial values
a = 1.65;
b = 6.2;
c = 2.69;
d = 4.2;
y1 = 0.02;
y2 = -0.02;
yc = 0.12;
m = 0.120;
g = 9.81;

% output(given in problem statement
y12 = yc + y2 - y1;

% inputs(currents)
u1 = (a * (y1 + b)^4) * ((4*c/(y12 + d)^5)+ m*g);
u2 = (a * (y2 + b)^4) * ((4*c/(y12 + d)^5)+ m*g);
% assigning variables to sub equations
w_u1 = 1/(a * (y1 + b)^4);
w_u2 = 1/(a * (-y2 + b)^4);
w_y1 = (4*u1)/(a * (y1 + b)^5);
w_y2 = (4*u2)/(a * (-y2 + b)^5);
w_y12 = (4*c)/((y12 + d)^5);

% 1.state space equation
A=[0 0 1 0;0 0 0 1; -(w_y1+w_y12)/m w_y12/m 0 0; w_y12/m -(w_y2+w_y12)/m 0 0];
B=[0 0; 0 0; w_u1/m 0; 0 w_u2/m] ;
C=[1 0 0 0;0 1 0 0] ;
D=[0,0;0,0] ;
sys=ss(A,B,C,D);

% transfer function of the open loop system
G = tf(sys);
G_1 = G(1,1);
G_2 = G(2,2);

% system 1
% observable Canonical form
```

```
csys_1 = canon(G_1,'Companion');
Ao_1 = csys_1.A;
Bo_1 = csys_1.B;
Co_1 = csys_1.C;
Do_1 = csys_1.D;

% controllable canonical form
Ac_1 = transpose(Ao_1);
Bc_1 = transpose(Co_1);
Cc_1 = transpose(Bo_1);
Dc_1 = Do_1;

Cx_1 = ctrb(Ac_1,Bc_1);
Ox_1 = obsv(Ao_1,Co_1);

% system 2
% observable Canonical form
csys_2 = canon(G_2,'Companion');
Ao_2 = csys_2.A;
Bo_2 = csys_2.B;
Co_2 = csys_2.C;
Do_2 = csys_2.D;

% controllable canonical form
Ac_2 = transpose(Ao_2);
Bc_2 = transpose(Co_2);
Cc_2 = transpose(Bo_2);
Dc_2 = Do_2;

Cx_2 = ctrb(Ac_2,Bc_2);
Ox_2 = obsv(Ao_2,Co_2);

% Jordan Canonical Form
[M,R] = eig(A);
Aj = jordan(A);
Bj = inv(M)*B;
Cj = C*M;
Dj = D;

% response system1
% impulse response
subplot(2,1,1);
impulse(G_1);
title('Impulse Response for System 1');
```

```
% step response
subplot(2,1,2);
step(G_1);
title('Step Response for System 1');

% response system 2
% impulse response
figure
subplot(2,1,1);
impulse(G_2);
title('Impulse Response for System 2');
% step response
subplot(2,1,2);
step(G_2);
title('Step Response for System 2');

% Bode Plot
figure
subplot(2,1,1);
bode(G_1);
title('Bode plot for System 1');
subplot(2,1,2);
bode(G_2);
title('Bode plot for System 2');

% Root Locus
figure
subplot(2,1,1);
rlocus(G_1);
title('Root locus for System 1');
subplot(2,1,2);
rlocus(G_2);
title('Root locus for System 2');

% PID controllers
% G_1
[sq,T_sq] = gensig('square',5,20,0.1);
[sin,T_sin] = gensig('sin',5,20,0.1);

% G_1 PID control
Kd=2000; Kp=6000; Ki=10000;
controller_1=tf([Kd Kp Ki], [1 0]);
plant_1 = G_1 * controller_1;
sys_pid_G_1 = feedback(plant_1,1);
```

```
% G_1 - Step response
figure
step(sys_pid_G_1);
grid
S_1 = stepinfo(sys_pid_G_1);
title('Step response for PID control 1');

% G_1 - Square wave input
[y,t]=lsim(sys_pid_G_1, sq, T_sq);
figure
plot(t,y);
grid
title('Square wave response for PID control 1');

% G_1 - Sine wave input
[y,t]=lsim(sys_pid_G_1, sin, T_sin);
figure
plot(t,y);
grid
title('Sinusoidal response for PID control 1');

% G_2 PID control
Kd=1500; Kp=4000; Ki=8000;
controller_2=tf([Kd Kp Ki], [1 0]);
plant_2 = G_2 * controller_2;
sys_pid_G_2 = feedback(plant_2,1);

% G_2 - Step response
figure
step(sys_pid_G_2);
S_2 = stepinfo(sys_pid_G_2);
grid
title('Step response for PID control 2');

% G_2 - Square wave input
[y,t]=lsim(sys_pid_G_2, sq, T_sq);
figure
plot(t,y);
grid
title('Square wave response for PID control 2');

%G_2 - Sine wave input
[y,t]=lsim(sys_pid_G_2, sin, T_sin);
```

```
figure
plot(t,y);
grid
title('Sinusoidal response for PID control 2');

% obtaining the state space equation from transfer function
[num1,den1]= tfdata(G_1,'v');
[A1,B1,C1,D1] = tf2ss(num1,den1);

[num2,den2]= tfdata(G_2,'v');
[A2,B2,C2,D2] = tf2ss(num2,den2);

sys1=ss(A1,B1,C1,D1);
sys2=ss(A2,B2,C2,D2);

%System 1
% Pole placement
Pole_1 = [2.4956i, -2.4956i, -1+1i, -1-1i];
K1 = place(A1,B1,Pole_1);

% closed loop matrix system 1
Af_1 = A1 - B1*K1;
% closed loop updated system 1
syscl_1 = ss(Af_1,B1,C1,D1);
dc_gain1 = dcgain(syscl_1);
updated_cl_1 = ss(Af_1, (B1.*(1/dc_gain1)),C1,D1);
tf_sys1_fdbk =tf(updated_cl_1);

%System 2
%Pole placement
Pole_2 = [2.5362i, -2.5362i, -1+1i, -1-1i];
K2 = place (A2,B2,Pole_2);
%closed loop matrix system 2
Af_2 = A2 - B2*K2;
% closed loop updated system 2
syscl_2 = ss(Af_2,B2,C2,D2);
dc_gain2 = dcgain(syscl_2);
updated_cl_2 = ss(Af_2, (B2.*(1/dc_gain2)),C2,D2);
tf_sys2_fdbk = tf(updated_cl_2);


% step response for system 1 and system 2 for full state feedback
figure
subplot(1,2,1)
```

```
step(updated_cl_1);
grid
title('Step response for Full state feedback contrl 1');

subplot(1,2,2)
step(updated_cl_2);
grid
title('Step response for Full state feedback contrl 2');

% square input response for system 1 and system 2 for full state feedback
[y,t]=lsim(updated_cl_1,sq, T_sq);
figure
subplot(1,2,1)
plot(t,y);
grid
title('Square response for Full State feedback contrl 1')

[y,t]=lsim(updated_cl_2,sq, T_sq);
subplot(1,2,2)
plot(t,y);
grid
title('Square response for Full State feedback contrl 2')

% Sinusoidal response for full state feedback
[y,t]=lsim(updated_cl_1,sin,T_sin);
figure
subplot(1,2,1)
plot(t,y);
grid
title('Sinusoidal response for Full state feedback contrl 1')

[y,t]=lsim(updated_cl_2,sin,T_sin);
subplot(1,2,2)
plot(t,y);
grid
title('Sinusoidal response for Full state feedback contrl 1')

% Full Order Observer
% System 1
% Pole placement
PoleFO_1 = [2.4956i, -2.4956i, -3+0.5i, -3-0.5i];
L1 = place(A1', C1', PoleFO_1);

% Full order observer state space equation OPEN loop
```

```
Afo_1= A1-L1'*C1-B1*K1;
Bfo_1= L1';
Cfo_1 = -K1;
Dfo_1 = [0];
FO_obs_1 = ss(Afo_1,Bfo_1,Cfo_1,Dfo_1);
tf_sys1_obs = tf(FO_obs_1);

% Full order observer state space equation CLOSED loop
dcg_fo_1 = dcgain(FO_obs_1);
k1 = 0.02 * 1/dcg_fo_1;
FO_cl_1 = ss(Afo_1, Bfo_1*k1,Cfo_1,Dfo_1);
tf_sys1_obscl = tf(FO_cl_1);

% System 2
% Pole placement
PoleFO_2 = [2.5362i , -2.5362i, -3+0.5i, -3-0.5i];
L2 = place(A2', C2', PoleFO_2);

% Full order observer state space equation OPEN loop
Afo_2= A2-L2'*C2-B2*K2;
Bfo_2= L2';
Cfo_2 = -K2;
Dfo_2 = [0];
FO_obs_2 = ss(Afo_2,Bfo_2,Cfo_2,Dfo_2);
tf_sys2_obs = tf(FO_obs_2);


% Full order observer state space equation CLOSED loop
dcg_fo_2 = dcgain(FO_obs_2);
k_2 = 0.02 * 1/dcg_fo_2;
FO_cl_2 = ss(Afo_2, Bfo_2*k_2,Cfo_2,Dfo_2);
tf_sys2_obscl = tf(FO_cl_2);

% step response for full order observer
figure
subplot(1,2,1)
step(FO_cl_1);grid
title('Step response: full order observer sys 1');
subplot(1,2,2)
step(FO_cl_2);grid
title('Step response: full order observer sys 2');

% square wave response for full order observer
[y,t]=lsim(FO_cl_1,sq, T_sq);
```

```
figure
subplot(1,2,1)
plot(t,y);
grid
title('Square response: full order observer sys 1')
[y,t]=lsim(FO_cl_2,sq, T_sq);
subplot(1,2,2)
plot(t,y);
grid
title('Square response: full order observer sys 2')

% sinusoidal response for full order observer
[y,t]=lsim(FO_cl_1,sin,T_sin);
figure
subplot(1,2,1)
plot(t,y);
grid
title('Sinusoidal response: full order observer sys 1')
[y,t]=lsim(FO_cl_2,sin,T_sin);
subplot(1,2,2)
plot(t,y);
grid
title('Sinusoidal response: full order observer sys 2')
```