Control of Magnetic levitation using PID and LQR Controllers - Team 6

Saravanane, Narendhiran **Siva Surya Venkat, Busi**

Pydah, Anirudh **Chandrappa, Goutham**

1 Introduction

Rapid advancements in electromechanical systems have revolutionized engineering and automation, introducing magnetic levitation as a transformative development with applications in transportation and precision manufacturing. This project focuses on employing PID and LQR controllers to address the intricacies of electromechanical coupling in magnetic levitation systems. The objective is precise control over the position of a suspended steel ball in a magnetic field, showcasing the robust capabilities of these controllers for stable levitation. This promises advancements in stability, efficiency, and precision across technological domains.

Acknowledging the challenges in precise control due to complex dynamics influenced by factors like applied voltage and drive current, the project deems traditional control methods inadequate. The adoption of PID and LQR controllers becomes imperative to enhance system performance. The scope involves assessing these controllers for governing the magnetic levitation of a steel ball, focusing on applied voltage, steel ball position, velocity, and drive current. The goal is to achieve precise control of the ball's position within the magnetic field, demonstrating stable levitation and positioning. The project's outcomes hold potential applications in transportation, manufacturing, and robotics, contributing insights into practical controller implementation. Additionally, the project includes modeling the magnetic levitation system as a Single Input Single Output (SISO) system, incorporating dynamic equations and linearization around an operational point to align with control theory principles.

2 Model of Magnetic Levitation System

To characterize a physical system as a Single Input Single Output (SISO) system, the initial step involves identifying the system's input and output. In this context, the input corresponds to the applied current I, and the output is represented by the system's position x. The modeling process commences with the fundamental dynamic equation of motion, wherein gravitational forces Fg and magnetic forces Fm play a role. Subsequently, linearization is applied—a common technique in control theory—aimed at simplifying the equations of motion around an operational point denoted as (xe, Ie).

2.1 Nonlinear Equations

The equation of motion describes the dynamics of the steel ball under the influence of gravity and magnetic force.

$$m\ddot{x} = F_g - F_m = mg - \frac{K_c I^2}{x^2} \tag{1}$$

where respectively m, x, g, F_g, F_m, I, and K_c denote the mass of the steel-ball, its position, gravity constant, gravitational force, magnetic force, applied driving current, and force constant of electromagnet respectively. Based on equation (1), we can define a nonlinear function as

$$f(x, I) = g - \frac{K_c I^2}{mx^2} \tag{2}$$

Meanwhile, the dynamics of applied driving current can be denoted as

$$\dot{I} = \frac{V}{L} - \frac{IR}{L} \tag{3}$$

where consecutively R, L, and V are the resistance, inductance, and operating voltage.

Usually, the nonlinear system is expected to operate at an equilibrium point. In our case, we expect that $\ddot{x} = 0, \dot{x} = 0$, and $\dot{I} = 0$. Since $\ddot{x}$ and $\dot{I}$ are zeros, it indicates that the magnetic and gravitational force are equal. Hence, the steel-ball can be levitated at an equilibrium position.

Take a look back at equation (1), since $F_g = F_m$, we can calculate the applied driving current at the equilibrium as

$$I_e = x_e \sqrt{\left(mgK_c^{-1}\right)} \tag{4}$$

where respectively I_e and x_e are the applied driving current and position of the steel-ball at the equilibrium point. Furthermore, we can obtain the voltage at equilibrium point, V_e, as

$$V_e = I_e R x_e \sqrt{\left(mgK_c^{-1}\right)} R. \tag{5}$$

2.2 Linearization

We can obtain the linearized model of MLS by applying Taylor Series Expansion as

$$\ddot{x} = c_1 \left(x - x_e\right) + c_2 \left(I - I_e\right), \tag{6}$$

where

$$c_1 \left.\frac{\partial f(x, I)}{\partial x}\right|_{x=x_e, I=I_e} = \frac{2K_c I_e^2}{mx_e^3}$$

and

$$c_2 \left.\frac{\partial f(x, I)}{\partial I}\right|_{x=x_e, I=I_e} = \frac{2K_c I_e}{mx_e^2}$$

Let the input u be the operating voltage V_e, then we can define the state space as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u = \begin{bmatrix} 0 & 1 & 0 \\ c_1 & 0 & c_2 \\ 0 & 0 & -\frac{R}{L} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} u, \tag{7}$$

where the state vector be

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^\top.$$

The state variables are x_1, x_2, and x_3 that respectively denote the position of the steel-ball, its velocity, and drive current. Meanwhile, the output can be defined as

$$y = \mathbf{C}\mathbf{x} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x} = x_1. \tag{8}$$

3 Analysis of properties

The system's stability is assessed by analyzing the eigenvalues of matrix A, revealing values of 44.2719, -44.2719, and -250.0000. With at least one eigenvalue having a positive real part (44.2719), the system is deemed unstable. An unstable system implies unbounded output growth over time, posing risks of system failure or unsafe conditions.

Controllability is evaluated using matrices A and B, and the controllability matrix 'co' is computed. The matrix rank is 3, matching the number of state variables in matrix A, indicating a controllable system. Despite instability, the system can be driven to any desired condition with an appropriate input.

Observability is determined through the observability matrix 'ob' using matrices A and C. The matrix rank is 3, signifying an observable system. Despite instability, the current state can be determined in finite time using only the outputs. While the system is unstable, its full controllability and observability suggest the potential for stabilization through controller design and state estimation with an observer.

4 Results and Discussion

Here we present the results of this research, including the necessary parameters.

4.1 System Parameters

Before stepping any further, it is necessary to define the parameters of MLS that used in this research. Those parameters can be seen in Table 1.

Table 1. Parameters of magnetic levitation system.

Parameters	Values
m	$0.008\ \mathrm{kg}$
K_c	$6.8823 \times 10^{-5}\mathrm{Nm}^2\ \mathrm{A}^{-2}$
R	$2.5\mathrm{ohm}$
L	$0.01\mathrm{H}$

Simultaneously, we establish the steel-ball equilibrium position at $x_e = 0.01\ \mathrm{m}$. Substituting these values into equation(4) yields the driving current equilibrium point as $I_e = 0.3375\ \mathrm{A}$. Additionally, equation(5) provides the operating voltage at equilibrium as $V_e = 0.8538\ \mathrm{V}$.

4.2 PID Controller

The PID continuously adjusts its output based on the information provided by the sensor, thus modifying the electromagnetic field intensity influences the ball's position. The paper presents the basic theory revolving around the maglev system and PID controlling and then presents simulation results of the system step response.

Applying the Laplace transform on linearized expression gives :

$$ms^2X(s) = -\left(\frac{2Ci_0}{x_0^2}\right)I(s) + \left(\frac{2Ci_0^2}{x_0^3}\right)X(s) \tag{9}$$

$$X(s)\left(ms^2 - \frac{2Ci_0^2}{x_0^3}\right) = -\left(\frac{2Ci_0}{x_0^2}\right)I(s) which gives : \tag{10}$$

$$\frac{X(s)}{I(s)} = \frac{\frac{-2Ci_0}{x_0^2}}{ms^2 - \frac{2Ci_0^2}{x_0^3}} \tag{11}$$

that is to say :

$$G(s) = \frac{-\left(\frac{2ci_0}{mx_0^2}\right)}{s^2 - \left(\frac{2ci_0^2}{mx_0^3}\right)} \tag{12}$$

This expression of second order is obtained considering that $\mathrm{X(s)}$, the position of the ball, is the output of the system and that I(s), the current fed to the electromagnet, is the input of it[9]. Considering the equilibrium point $(-1.5\ \mathrm{V}, 0.8\ \mathrm{A}), \mathrm{C} = -0.69$ and $\mathrm{m} = 0.02\ \mathrm{kg}$, the following function is obtained :

$$G(s) = \frac{24,5}{s^2 - 13} = \frac{24,5}{(s - 3,6)(s + 3,6)} \tag{13}$$

This function has two poles and no zeros. One of the poles is in the right half plane which makes the open loop unstable, the root locus is given at the figure below :

then the root locus is shown in the figure as

A PID controller is added to the system to ensure stability, the closed-loop is given at the figure below : The system response is evaluated according to a step reference. The results are displayed on a scope using the simulink application on Matlab. The tuning function is used to determine the PID parameters :

On figure 5, the dotted response corresponds to the initial PID parameters, which are values chosen to ensure stability but not performance. The full line corresponds to the tuned response

Controller Parameters		
	Tuned	Block
p	11.2992	10
I	20.3355	10
D	1.5416	100
N	209.6701	100
Performance and Robustness		
	Tuned	Block
Rise time	0.0335 seconds	0.00228 seconds
Settling time	0.605 seconds	0.0772 seconds
Overshoot	15.9%	72.7%
Peak	1.16	1.73
Gain margin	-20.6 dB@$3.63\mathrm{rad/s}$	-25.5 dB@$0.316\mathrm{rad/s}$
Phase margin	$69\mathrm{deg}@38.4\mathrm{rad/s}$	$11.5\mathrm{deg}@490\mathrm{rad/s}$
Closed-loop stability	Stable	Stable

The above table represents the Parameters of the PID

4.3 LQR Controller Design and Precompensator

In this section, we delve into the design of the control law and precompensator for the magnetic levitation system. The challenge involves parameterization to attain a stable controller, and LQR is utilized to address this issue. We offer a brief explanation of LQR and further elaborate on the precompensator.

4.3.1 Linear-Quadratic Regulator

The Linear-Quadratic Regulator (LQR) is an optimal control strategy for SISO and MIMO systems, ensuring robust stability. It excels in optimizing energy consumption, surpassing PID and fuzzy controllers. Figure 4 depicts the block diagram of a full-state feedback controller using LQR.

In general, LQR has an objective function such as

$$J(\mathbf{u}) = \int_0^\infty \left(\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \mathbf{u}^\top \mathbf{R}\mathbf{u}\right) \mathrm{d}t \tag{14}$$

that must be minimized. In MLS research, with three states and one input, $\mathbf{Q} \in \mathbb{R}^{3\times3}$ is a 3x3 matrix, and $\mathbf{R} \in \mathbb{R}$ is a scalar. Both $\mathbf{Q}$ and $\mathbf{R}$ impact gain feedback $\mathbf{K}$ differently; increasing $\mathbf{Q}$ leads to a larger $\mathbf{K}$, while a larger $\mathbf{R}$ results in a smaller $\mathbf{K}$.Next, we need to calculate the auxiliary matrix $\mathbf{S}$ by solving this Algebraic Ricatti Equation (ARE) where

$$\mathbf{A}^\top \mathbf{S} + \mathbf{S}\mathbf{A} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{S} + \mathbf{Q} = 0 \tag{15}$$

It is necessary to determine the gain feedback $\mathbf{K}$ such as

$$\mathbf{K} = \mathbf{R}^{-1}\left(\mathbf{B}^\top \mathbf{S}\right) \tag{16}$$

4.3.2 Precompensator

Incorporating precompensator $\bar{N}$ is crucial for zero steady-state errors. In practice, reference signal r undergoes $\bar{N}$ multiplication for an updated reference. Figure 4 illustrates the block diagram of the state feedback controller with the precompensator.

There are some steps in order to calculate the precompensator. First, we need to find $\mathbf{N}$ as

$$\mathbf{N} = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & 0 \end{array}\right]^{-1} \mathbf{Z} \tag{17}$$

where

$$\mathbf{Z} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^\top \tag{18}$$

After obtaining $\mathbf{N}$, we can partitioned it as

$$\mathbf{N} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{19}$$

where $\alpha \in \mathbb{R}^{3\times 1}$ is a column vector, meanwhile $\beta \in \mathbb{R}$ is a scalar. Eventually, the precompensator can be calculated as

$$\bar{N} = \mathbf{K}\alpha + \beta \tag{20}$$

4.3.3 Analysis and properties of LQR

After parameter definition, we examine open-loop system stability. Substituting parameters from Table 1, including gravity constant $g = 9.8\ \mathrm{m\ s^{-2}}$, into state matrix $\mathbf{A}$ and input vector $\mathbf{B}$ yields:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 1960 & 0 & -58.0717 \\ 0 & 0 & -250 \end{bmatrix}, \tag{21}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 100 \end{bmatrix}. \tag{22}$$

To determine the open-loop system's stability, we calculate the eigenvalues of $\mathbf{A}$, resulting in the eigenvector $\mathrm{eig}(\mathbf{A}) = [44.2719, -44.2719, -250]^\top$. The presence of a real positive eigenvalue indicates system instability.

To assess controllability, we calculate the rank of the controllability matrix. The controllability matrix, denoted as $\mathbf{Co}$, is defined as:

$$\mathbf{Co} = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} \end{bmatrix} \tag{23}$$

Utilizing MATLAB, the rank of the controllability matrix $\mathbf{Co}$, a 3-by-3 matrix corresponding to the state variables, is found to be 3. The equality of its rank to the number of state variables confirms system controllability.

Moving forward, controller design using LQR necessitates parameter determination, including the compression.

$$\mathbf{Q} = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix} \tag{24}$$

$$\mathbf{R} = 1 \tag{25}$$

With the parameters for $\mathbf{C}$, $\mathbf{B}$, $\mathbf{Q}$, and $\mathbf{R}$ in place, the solution to the Algebraic Ricatti Equation in equation(15) is obtained using MATLAB. This results in the determination of the auxiliary matrix as:

$$\mathbf{S} = \begin{bmatrix} 6964 & 136.25 & -7.3089 \\ 136.25 & 3.644 & -0.193 \\ -7.3089 & -0.193 & 0.0884 \end{bmatrix} \tag{26}$$

Eventually, equation(16) can be solved and the feedback gain by:

$$\mathbf{K} = \begin{bmatrix} -730.8898 & -19.3003 & 8.8431 \end{bmatrix} \tag{27}$$

With the gain feedback obtained, closed-loop system stability is assessed similarly to the open-loop system. Eigenvalues for the closed-loop system are $\mathrm{eig}(\mathbf{A} - \mathbf{BK}) = [-1029.2, -24.3122, -80.7729]^\top$, confirming stability. Computation of the precompensator, $\bar{N} = -348.0447$, is achieved by solving equations (17), (18), (19) and (20) and substituting parameters.

Proceeding to simulate the closed-loop system's response, the initial states are set to $\mathbf{x}_0 = [0, 0, 0]^\top$. Initially, the steel ball is at rest with no driving current. The desired states, $\mathbf{x}_d = [0.01, 0, 0.3375]^\top$, represent the desired equilibrium. The MLS response with the state feedback controller and precompensator is shown in Figure 3.

The response rapidly converges, evidenced by a rise time of 0.097 seconds and a settling time of 0.1766 seconds, with no steady-state errors. Figure 8 illustrates the dynamics of states x_1, x_2, and x_3 corresponding to x_e, $\dot{x}_e$, and I_e. Initially stationary, the steel ball begins levitating with the applied operating voltage. x_2 increases until $x_1 \approx x_e$, and to maintain $x_1 = x_e$, x_2 must be zero. Simultaneously, the driving current x_3 rises until reaching equilibrium I_e, indicating achieved equilibrium for x_e as the driving current correlates with the steel ball's position.

5 Conclusion

This project centered on optimizing the performance of a magnetic levitation system through modeling and control. Key findings encompassed successful system modeling, linearization at equilibrium points, and the design of a Linear Quadratic Regulator (LQR) state feedback controller. A notable enhancement was the introduction of a precompensator to eliminate steady-state errors, ensuring precise and stable control. Matlab simulations validated the design with impressive metrics, including a swift 0.097 seconds rise time, a settling time of 0.1766 seconds, and no overshoot. The precompensator played a pivotal role in achieving zero steady-state errors.

Critical design decisions favored the adoption of an LQR state feedback controller for optimal performance, and the precompensator addressed steady-state errors. Challenges arose from inherent nonlinearities in the magnetic levitation system, prompting the ongoing refinement of the model to capture real-world complexities.

The project emphasized the importance of linearization for simplifying complex systems in controller design. Ongoing efforts involve refining the model to accommodate nonlinear elements, conducting robustness analysis, and transitioning to experimental validation. This work aims to enhance the accuracy, robustness, and real-world applicability of the magnetic levitation control system.

References

[1] H.-W. Lee, K.-C. Kim, and J. Lee. Review of maglev train technologies. IEEE Transactions on Magnetics, 42(7):1917-1925, 2006.

[2] J. Asama, Y. Hamasaki, T. Oiwa, and A. Chiba. Proposal and analysis of a novel single-drive bearingless motor. IEEE Transactions on Industrial Electronics, 60(1):129-138, 2013.

[3] M. Hagiwara, T. Kawahara, T. Iijima, and F. Arai. High-speed magnetic microrobot actuation in a microfluidic chip by a fine v-groove surface. IEEE Transactions on Robotics, 29(2):363-372, 2013.

[4] C. V. Aravind, R. Rajparthiban, R. Rajprasad, and Y. V. Wong. A novel magnetic levitation assisted vertical axis wind turbine design procedure and analysis. In 2012 IEEE 8th International Colloquium on Signal Processing and its Applications, pages 93-98, 2012.

[5] R. Uswarman, A. I. Cahyadi, and O. Wahyunggoro. Control of a magnetic levitation system using feedback linearization. In 2013 International Conference on Computer, Control, Informatics and Its Applications (IC3INA), pages 95-98, 2013.

[6] S. Hlangnamthip, C. Thammarat, and D. Puangdownreong. Cs-based pid controller design for maglev vehicle suspension system. In 2019 7th International Electrical Engineering Congress (iEECON), pages 1-4, 2019.

[7] N. Sahoo, A. Tripathy, and P. Sharma. Single axis control of ball position in magnetic levitation system using fuzzy logic control. In IOP Conference Series: Materials Science and Engineering, volume 323, 2018 .

[8] R. Uswarman, A. I. Cahyadi, and O. Wahyunggoro. Modified sliding mode control with uncertainties behavior of a magnetic levitation system. In 2013 International Conference on Robotics, Biomimetics, Intelligent Computational Systems, pages 194-199, 2013.

[9] T. Sakanushi, K. Yamada, Y. Ando, T. M. Nguyen, and S. Matsuura. A design method for simple multiperiod repetitive controllers for multiple-input/multiple-output plants. ECTI Transactions on Electrical Engineering, Electronics, and Communications, 9(1):202-211, 2011.

[10] T. Sakanushi, K. Yamada, I. Murakami, J. Hu, and S. Matsuura. A design method for robust stabilizing simple multi-period repetitive controllersfor multiple-input/multiple-output plants. ECTI Transactions on Electrical Engineering, Electronics, and Communications, 10(1):1-13, 2012.

[11] T. Sakanushi, K. Yamada, Y. Ando, S. Matsuura, and J. Hu. A design method for simple repetitive controllers for multiple-input/multiple-output time-delay plants. ECTI Transactions on Electrical Engineering, Electronics, and Communications, 10(1):24-34, 2012.

[12] Priyatmadi, A. P. Sandiwan, H. Wijaya, and A. Cahyadi. Application of spsa lqr tuning on quadrotor. In 2016 6th International Annual Engineering Seminar (InAES), pages 32-36, 2016.

[13] H. Maghfiroh, A. Ataka, O. Wahyunggoro, and A. I. Cahyadi. Optimal energy control of dc motor speed control: Comparative study. In 2013 International Conference on Computer, Control, Informatics and Its Applications (IC3INA), pages 89-93, 2013.

[14] O. A. Dhewa, A. Dharmawan, and T. K. Priyambodo. Model of linear quadratic regulator (lqr) control method in hovering state of quadrotor. Journal of Telecommunication, Electronic and Computer Engineering, 9(3):135-143, 2017.

6 Appendix

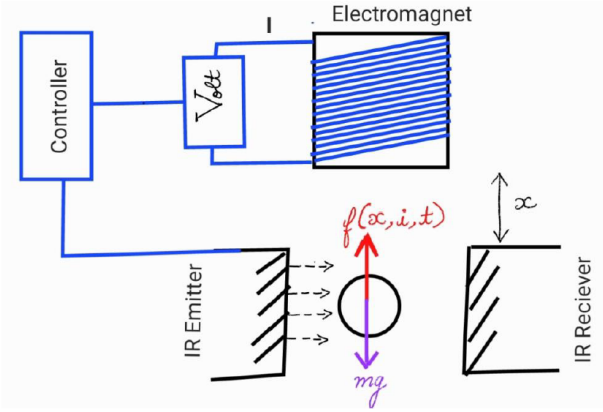

***Figure 1**: Schematics of Magnetic Levitation System with Free Body Diagram*

Figure 1: Schematics of Magnetic Levitation System with Free Body Diagram

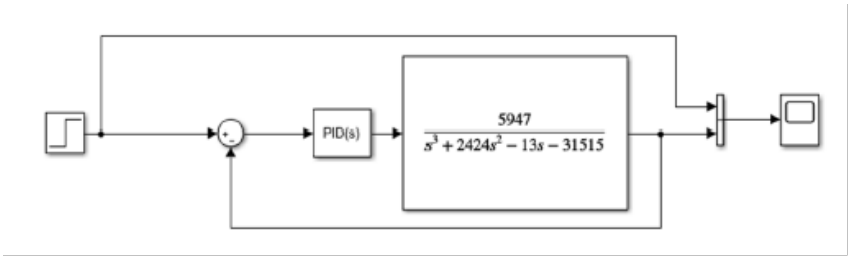

Figure 2: Schematic diagram of the closed-loop system with PID

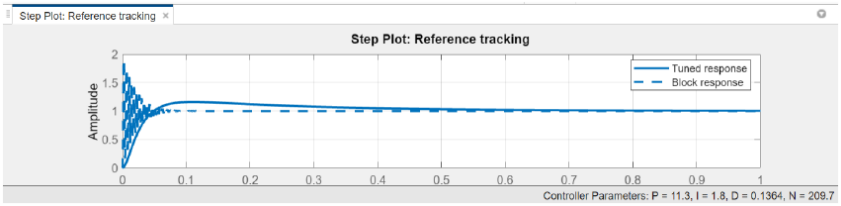

Figure 3: Step response of the closed-loop system

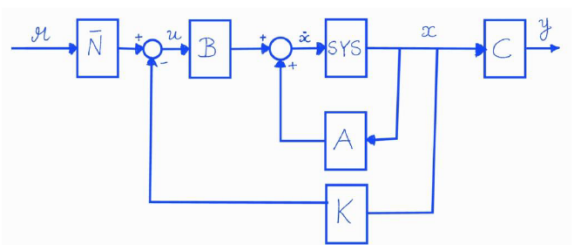

Figure 4: Block diagram of a state feedback controller with precompensator

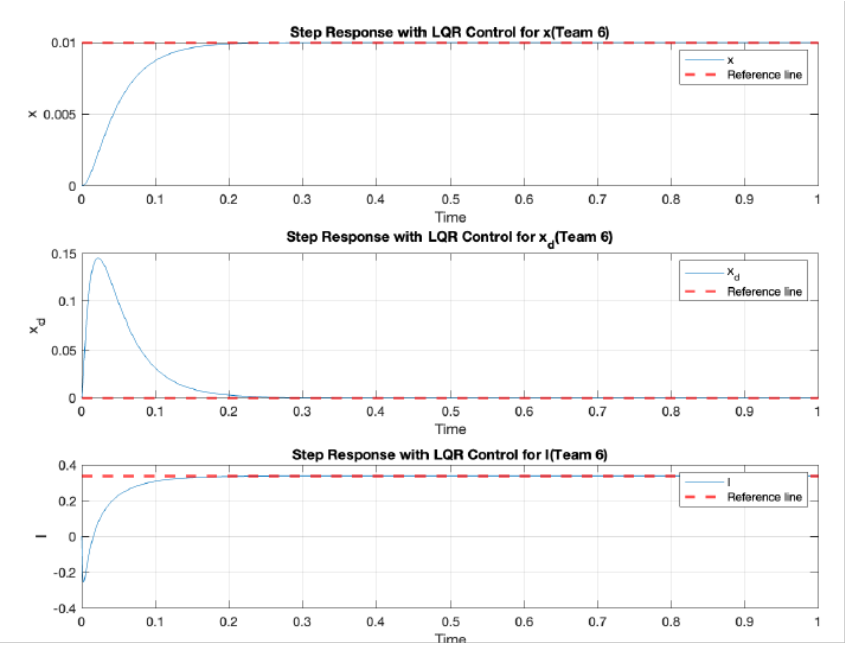

Figure 5: Step response with LQR and pre-compensator (for each states)

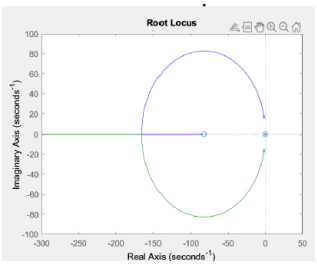

Figure 6: Root locus of system with PID controller

7 Code

```
1
2  % Clearing Workspace
3  clc; clear all;
4
5  % System Parameters
6  % Define physical parameters of the system, such as mass, constants,
       and gravity.
7  m = 0.008; % Mass (kg)
8  kc = 6.8828 * 10^-5; % Motor constant (Nm^2 A^-2)
9  Resis = 2.5; % Resistance (ohm)
10 L = 0.01; % Inductance (H)
11 xe = 0.01; % Displacement (m)
12 g = 9.8; % Acceleration due to gravity (m/s^2)
13
14 % Required Parameters for state matrices
15 Ie = xe * (m*g*kc^-1)^(1/2); % Motor current
16 Ve = Ie*Resis; % Motor voltage
17
18 c1 = 2*kc*Ie*Ie/(m*xe^3); % Coefficient 1
19 c2 = -2*kc*Ie/(m*xe^2); % Coefficient 2
```

```
20
21 % State Matrix
22 A = [0 1 0; c1 0 c2; 0 0 -1*Resis/L];
23 B = [0; 0; 1/L];
24 C = [1 0 0];
25 D = 0;
26
27 states = {'x' 'x_dot' 'I'};
28 inputs = {'Ve'};
29 outputs = {'x'};
30
31 sys_ss = ss(A, B, C, D, 'statename', states, 'inputname', inputs, '
       outputname', outputs)
32
33 sys_tf = tf(sys_ss);
34
35 % LQR Controller Design
36 Q = [100 0 0; 0 100 0; 0 0 100]
37 p = 50
38 R = 1
39
40 [K, S, P] = lqr(sys_ss, Q, R)
41 Nbar = rscale(A, B, C, D, K)
42
43 sys1 = ss(A-B*K, B*Nbar, C, D);
44
45 C2 = [0 1 0];
46 C3 = [0 0 1];
47 sys_cl = ss(A-B*K, B*Nbar, C, D);
48 sys_cl2 = ss(A-B*K, B*Nbar, C2, D);
49 sys_cl3 = ss(A-B*K, B*Nbar, C3, D);
50
51 % System Response Analysis
52 poles = eig(A)
53
54 co = ctrb(sys_ss);
55 controllability = rank(co)
56
57 ob = obsv(A, C);
58 observability = rank(ob)
59
60 % Step Response Plotting
61 t = 0:0.001:1;
62 r = xe * ones(size(t));
63 subplot(3,1,1);
64 [y, t, x] = lsim(sys_cl, r, t);
65 plot(t, y, '-');
66 yline(0.01, 'r--', 'LineWidth', 2);
67 grid on;
68 legend('x', 'Reference line');
69 title('Step Response with LQR Control for x (Team 6)');
70 xlabel('Time');
71 ylabel('x');
72
73 t = 0:0.001:1;
74 r = xe * ones(size(t));
75 subplot(3,1,2);
76 [y, t, x] = lsim(sys_cl2, r, t);
77 plot(t, y, '-');
78 yline(0, 'r--', 'LineWidth', 2);
79 grid on;
80 legend('x_d', 'Reference line');
81 title('Step Response with LQR Control for x_d (Team 6)');
82 xlabel('Time');
83 ylabel('x_d');
```

```
84
85 t = 0:0.001:1;
86 r = xe * ones(size(t));
87 subplot(3,1,3);
88 [y, t, x] = lsim(sys_cl3, r, t);
89 plot(t, y, '-');
90 yline(0.3375, 'r--', 'LineWidth', 2);
91 grid on;
92 legend('I', 'Reference line');
93 title('Step Response with LQR Control for I (Team 6)');
94 xlabel('Time');
95 ylabel('I');
96
97 % Functions
98
99 function [Nbar] = rscale(a, b, c, d, k)
100     % rscale: Compute the scale factor Nbar for eliminating steady-
          state error.
101     % For a continuous-time, single-input system with full-state
          feedback:
102     %    dx/dt = Ax + Bu
103     %    y = Cx + Du
104     % and feedback matrix K, the function computes Nbar to eliminate
105     % steady-state error for a step reference input.
106
107     error(nargchk(2, 5, nargin));
108
109     % Determine which syntax is being used
110     nargin1 = nargin;
111     if (nargin1 == 2) % System form
112         [A, B, C, D] = ssdata(a);
113         K = b;
114     elseif (nargin1 == 5) % A, B, C, D matrices
115         A = a; B = b; C = c; D = d; K = k;
116     else
117         error('Input must be of the form (sys, K) or (A, B, C, D, K)')
118     end
119
120     % Compute Nbar
121     s = size(A, 1);
122     Z = [zeros([1, s]) 1];
123     N = inv([A, B; C, D]) * Z';
124     Nx = N(1:s);
125     Nu = N(1 + s);
126     Nbar = Nu + K * Nx;
127 end
```

Listing 1: Your MATLAB code

```
sys_ss =

  A =
              x   x_dot       I
   x          0       1       0
   x_dot   1960       0  -58.07
   I          0       0    -250

  B =
          Ve
   x       0
   x_dot   0
   I     100

  C =
         x  x_dot      I
   x     1      0      0

  D =
      Ve
   x   0

Continuous-time state-space model.
Model Properties
Q = 3×3
     100     0     0
       0   100     0
       0     0   100

p = 50
R = 1
K = 1×3
   -730.8636  -19.2998    8.8431

S = 3×3
10^3 ×
      6.9636    0.1362   -0.0073
      0.1362    0.0036   -0.0002
     -0.0073   -0.0002    0.0001

P = 3×1
10^3 ×
     -0.0243
     -0.0808
     -1.0292

Nbar = -348.0321
```


Figure 7: Result1

```
poles = 3×1
     44.2719
    -44.2719
   -250.0000

controllability = 3

observability = 3
```

Figure 8: Result2

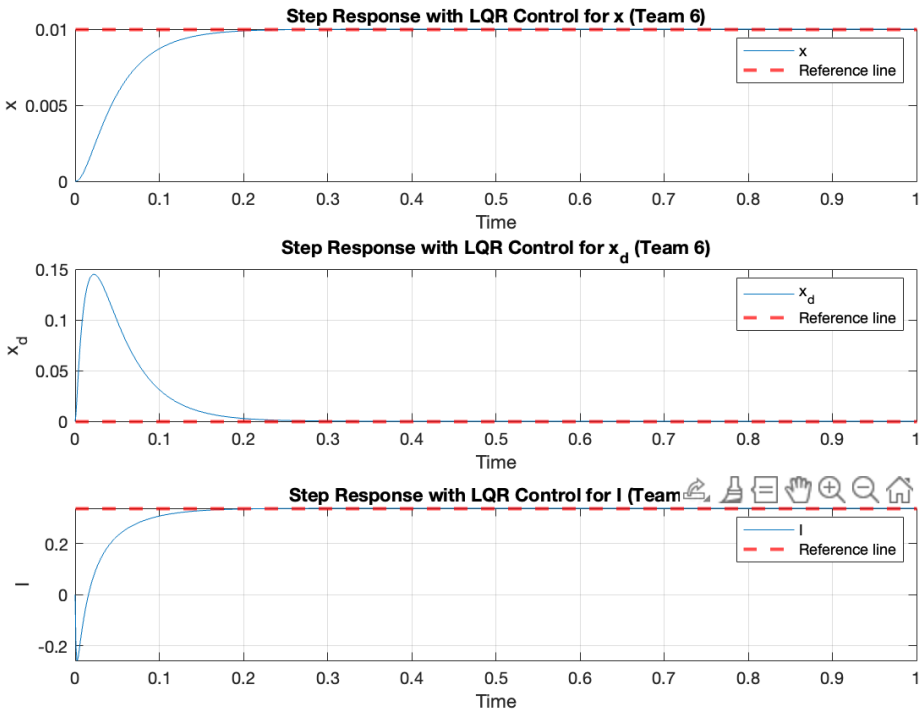

Figure 9: Result3

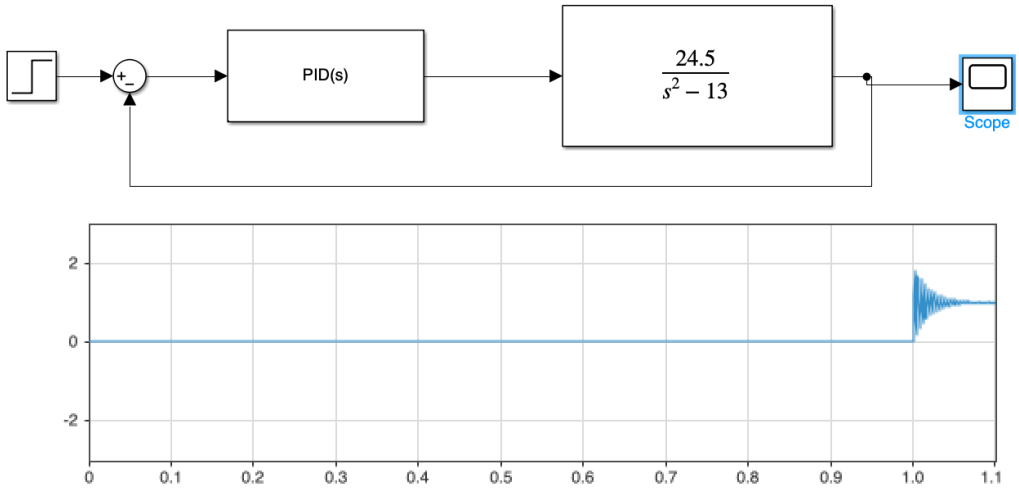

Figure 10: PID Simulink