AE322A
AIRCRAFT CONTROL SYSTEMS

Term Project
Non-Overshooting PID Controller Design for a Spring-Damper System

---

Submitted By:

Roll No.	Member
210161	Anubhav Bairoliya
210137	Anil Yadav
211093	Syed Adeel Ahmad
210400	Gugulothu Kalyan

Contents

Non-Overshooting PID Controller Design for a Spring-Damper System

Anil Yadav[1], Anubhav Bairoliya[2], Syed Adeel Ahmad[3], and Gugulothu Kalyan[4]

[1]Aerospace Engineering, Indian Institute of Technology Kanpur
[2]Aerospace Engineering, Indian Institute of Technology Kanpur
[3]Aerospace Engineering, Indian Institute of Technology Kanpur
[4]Aerospace Engineering, Indian Institute of Technology Kanpur

Abstract

This report introduces a PID (Proportional-Integral-Derivative) controller that eliminates overshoot in spring-damper systems. Traditional PID controllers often cause overshoot, leading to instability and excessive oscillations. By implementing a non-overshooting PID controller, we achieved stable and precise position control in a spring-damper system. The approach holds promise for applications requiring precise control, like automotive suspension and precision machinery. This development also has application in various engineering fields.

1. Introduction

The use of PID (Proportional-Integral-Derivative) controllers is widespread in various engineering applications due to their simplicity and effectiveness in controlling system behaviour. However, one common issue with traditional PID controllers is the tendency to overshoot the desired setpoint, which can lead to instability and oscillations. In this report, we have tried to presents a novel approach to PID control design that eliminates overshoot, focusing on its implementation on spring-damping systems. A spring damper system is a mechanical arrangement used to absorb and dissipate energy, commonly employed in engineering to control oscillations and vibrations in various systems. It consists of a combination of a spring element, which stores mechanical energy, and a damper element, which dissipates energy through friction or fluid resistance.

Spring-damper system is a second-order stable system. In a spring damper system, overshooting can lead to excessive oscillations and potentially destabilize the system, causing instability or even damage. For applications requiring precise control and stability, such as automotive suspension or precision machinery, a non-overshooting control ensures that the system response remains within desired bounds, promoting safety, efficiency, and reliability.

1.1 Existing Work on Non-Overshooting PID Controllers

Attaining a non-overshooting or minimum overshoot step response is a crucial requirement in various real-world plants and systems. Over the years, several methods have been introduced and developed to achieve this goal, each offering unique insights and approaches. The following is a brief summary of existing work in the field of non-overshooting PID controllers:

1. **Min-Max Optimization Approach**
   The work employed a min-max optimization approach to determine the optimum location of zeros in the system. By strategically placing zeros, the goal is to attain a minimum-overshoot transient response, offering a systematic optimization framework.

2. **Compensator Design for Minimum Phase Systems**
   A compensator is designed specifically for the special case of minimum phase systems. The objective is to achieve a non-overshooting closed-loop system step response, addressing the unique characteristics of these systems.

3. **Rational Two-Parameter Controller**
   A rational two-parameter controller is proposed to eliminate overshoot in closed-loop system step responses. This controller design aimed to provide a simple yet effective solution to the overshoot problem.

4. **Modified PID Controller: I+PD Controller**
   A modified version of the PID controller, known as the I+PD controller, was introduced. This controller design utilizes process step response and damping optimum criteria to achieve a satisfactory transient response without overshoot.

1.2 Shortcomings

Despite the advancements in the field, as noted in the literature, there is still no guarantee of achieving a non-overshooting step response in all cases. Each method offers unique advantages and considerations, catering to specific system requirements and characteristics. These diverse approaches contribute to the rich landscape of non-overshooting PID controller design, providing engineers and researchers with a range of tools to tackle the overshoot problem in control systems.

1.3 Key Idea

This paper aims to create PD and PID controllers that achieve non-overshooting step responses in specific systems. The method involves selecting controller coefficients to ensure the closed-loop frequency response decreases monotonically. This leads to derived inequalities for PID parameters, defining specific design regions. Controllers meeting set conditions, such as gain crossover frequency and isodamping property, are chosen. Isodamping maintains a flat phase around the gain crossover, ensuring system robustness against uncertainties while achieving a non-overshooting step response.

2. Problem Formulation

The paper focuses on the implementation of the non-overshooting PID control strategy on a spring damping system. The goal is to control the position of a mass attached to a spring while minimizing overshoot and maintaining stability. The system is subject to external disturbances and requires a control algorithm that can respond quickly without oscillations.

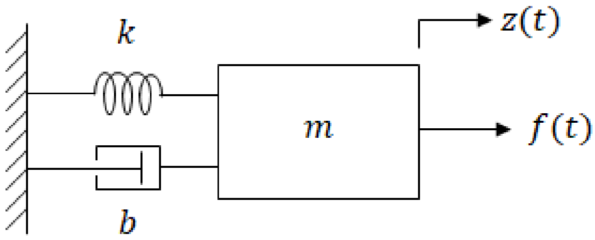

Figure 1: Spring-Damper System

- $f(t)$ = external force applied on the mass.
- $z(t)$ = position(unknown).
- m = mass of the system.
- k = spring constant.
- b = damping coefficient.

2.1 Equation of Motion

Using Newton's 2^{nd} Law of Motion:-

$$f(t) - b\frac{dz}{dt} - kz(t) = m\frac{d^2z}{dt^2} \tag{1}$$

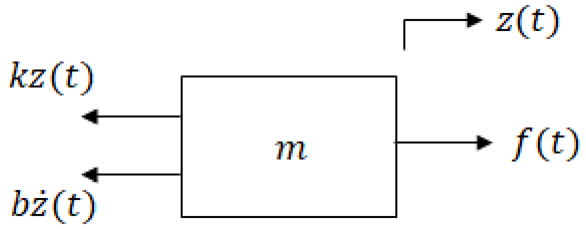

Figure 2: Free Body Diagram of Spring-Damper System

2.2 Transfer Function

Applying Laplace Trabsform on the Equation 1:-

$$F(s) - bsZ(s) - kZ(s) = ms^2Z(s)$$

Solving for Z(s)/F(s):-

$$\frac{Z(s)}{F(s)} = \frac{1}{ms^2 + bs + k}$$

2.3 State Space Model

When there is a mass in a system, its position and velocity are commonly chosen as state variables. Also, position, velocity, and force (input) are sufficient to determine this system's future position (output). For these reasons, position($z(t)$) and velocity($\dot{z}(t)$) are chosen as state variables.

State vector:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} z(t) \\ \dot{z}(t) \end{bmatrix}$$

Input vector:

$$\begin{bmatrix} u(t) \end{bmatrix} = \begin{bmatrix} f(t) \end{bmatrix}$$

Output vector:

$$\begin{bmatrix} y(t) \end{bmatrix} = \begin{bmatrix} z(t) \end{bmatrix}$$

Rewriting eq. 1 in these new notations:

$$u(t) - bx_2(t) - kx_1(t) = m\dot{x}_2(t)$$

Rearranging equations to express $\dot{x}(t)$ and $y(t)$ in terms of $x(t)$ and $u(t)$:

$$\begin{aligned}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_1(t) &= -\frac{k}{m}x_1(t) - \frac{b}{m}x_2(t) + \frac{1}{m}u(t) \\
y(t) &= x_1(t)
\end{aligned}$$

Organising into matrix format:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \begin{bmatrix} u(t) \end{bmatrix}$$

$$\begin{bmatrix} y(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

3. Control Law

From the above analysis we have obtained the following transfer function of a spring-mass damper system:-

$$G_P(s) = \frac{1}{ms^2 + bs + k}$$
$$\text{where, } w_n = \sqrt{\frac{k}{m}},\ \zeta = \frac{b}{2\sqrt{mk}}$$

We have taken the following general parameters to design the transfer function and control law for non-overshooting transient response:

- **Mass** (m): Let's consider a block with a mass of approximately 1 kg.
- **Spring Constant** (k): Let's assume a spring constant of around 4 N/m.
- **Damping Coefficient** (b): Let's assume a damping coefficient of around 5 Ns/m.

With these values, we can describe the spring-mass-damper system of a railway suspension system using the following transfer function:

$$G_P(s) = \frac{1}{s^2 + 4s + 5}$$

Comparing it with the form

$$G_P(s) = \frac{K}{s^2 + As + B}$$

we get $K = 1$, $A = 4$ and $B = 5$. Also, we obtain $w_n = 2.236$ and $\zeta = 0.894$. So, the system is an under-damped system which will show oscillatory behaviour to reach the transient response. Hence, we will apply Proportional and Proportional-Integral-Derivative (PID) control respectively to analyze the cases and observe which control helps us to achieve the objective.

3.1 Proportional Control

We will try to design a Proportional Control for the above system and try to analyze if it is possible to design a non-overshooting case for the same. Figure 3 shows the block diagram for the same.

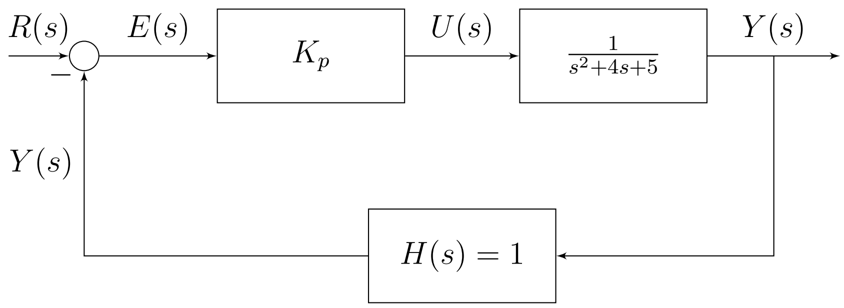

Figure 3: Proportional Control Block Diagram for the System

The Open-Loop Transfer Function for the above control system is:

$$G_0(s) = \frac{K_p}{s^2 + 4s + 5}$$

The Closed-Loop Tranfer Function is the following:

$$H(s) = \frac{G_0(s)}{1 + G_0(s)} = \frac{K_p}{s^2 + 4s + 5 + K_p} \tag{2}$$

For the sake of analysis we will assume $K_p = 250$, we obtain the following transient response:

$$H(s) = \frac{250}{s^2 + 4s + 255} \tag{3}$$

The Maximum Overshoot, $M_P(\%)$ for this case is:

$$M_P = e^{\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \times 100\% = 67.2609\%$$

Since, we have no such constrain on Peak Time, Rise Time and Settling Time, so we will check for Maximum Overshoot for a range of values of K_p. By plotting a curve between $M_P(\%)$ vs K_p, we will try to analyze the general trend of the overshoot with respect to K_p and deduce if we can design a non-overshooting proportional controller.
The following is the obtained trend:
We can see that for small values of K_p the steady state does not reach the required output value. So, we have to keep K_p considerably high($K_p \geq 250$). As calculated

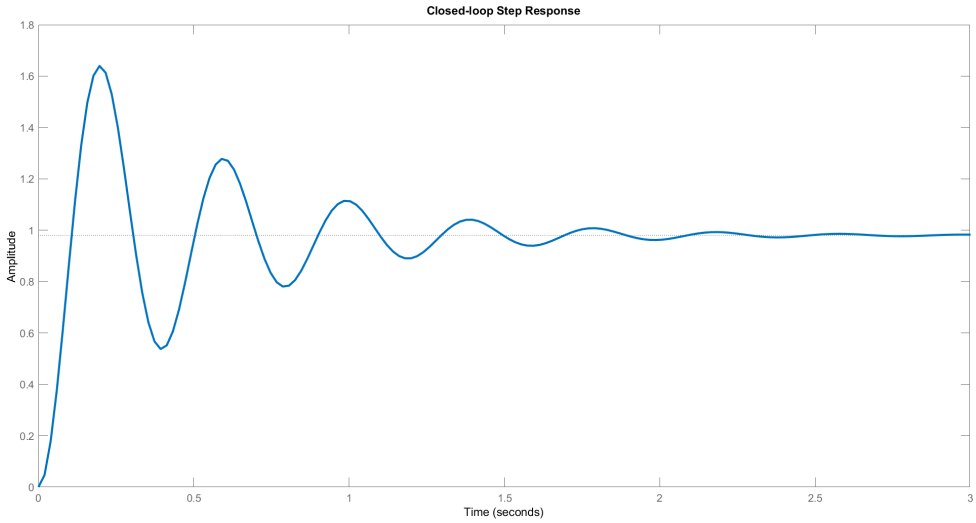

Figure 4: Transient response for the Closed Loop Transfer Function given by Equation 3

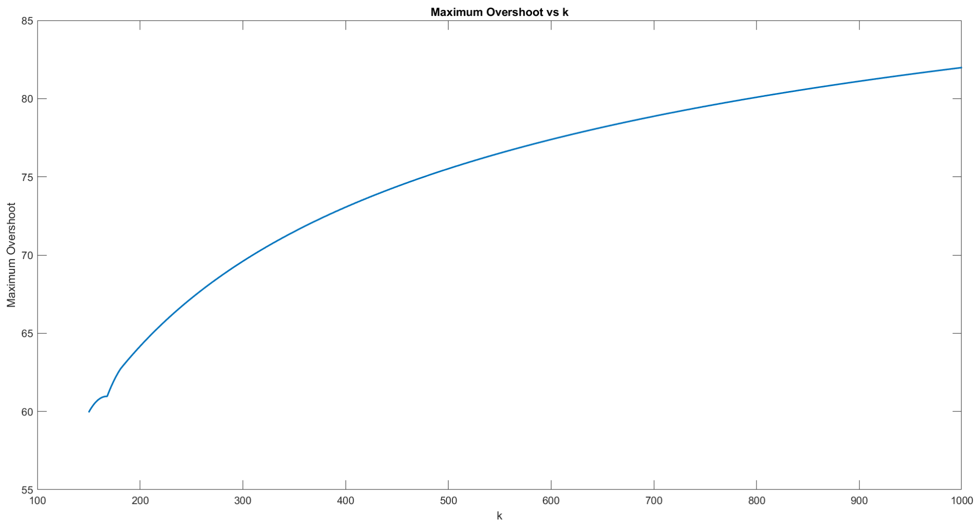

Figure 5: $M_P(\%)$ vs K_p

above and seeeing the trend observed, we can see that the $M_P(\%)$ is increasing with increasing K_p. Hence, $M_P(\%)$ is always greater than 67% approximately and flattens at 80% approximately.

3.2 Proportional-Integral-Derivative (PID) Control

Seeing the above case of Proportional Control, we will try to design a PID control for the system and apply relevant algorithm to make it non-overshooting.
We have a stable second order plant system with:

$$G_P(s) = \frac{K}{s^2 + As + B} \tag{4}$$

We have the following open-loop transfer function for a PID Control:-

$$G_0(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s\right) \tag{5}$$

Figure 6 shows the block diagram for the above assumed system and PID control added to control it.

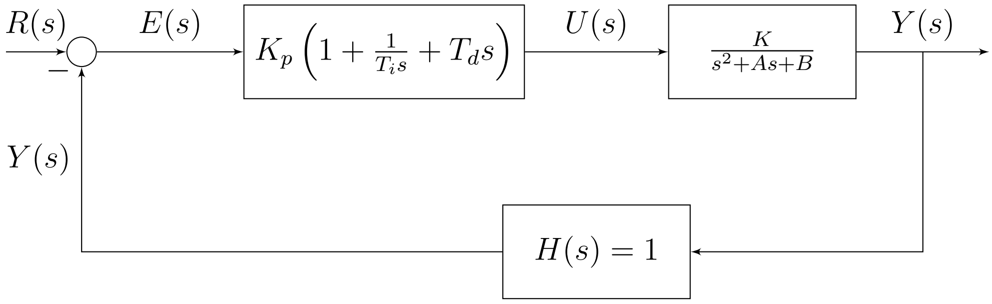

Figure 6: PID Control Block Diagram for the System

Applying PID Control, we will get the following Closed-Loop Tranfer Function:

$$H(s) = \frac{KK_p(T_iT_ds^2 + T_is + 1)}{T_is^3 + T_i(A + KK_pT_d)s^2} + T_i(B + KK_p)s + KK_p \tag{6}$$

We will use the following algorithm and conditions to find a non-overshooting response for the above system. The algorithms and conditions can be applied only on a stable system and are taken from the results derived from the research paper listed in references and hence their derivation are beyond the scope of this research.

Algorithm:- We have a stable second-order system whose poles lie on the left s-plane and have the following additional conditions for implementing the algorithm:

$$A > 0, B > 0 \tag{7}$$

$$A^2 > 2B \tag{8}$$

$$K_p > 0, T_i > 0, T_d > 0 \tag{9}$$

Step 1:- Choose a positive value for K_p as per the Equations 9 and 10.

$$K_p \leq \frac{A^2 - 2B}{2K} \tag{10}$$

Step 2:- Apply the following inequality to decide T_d.

$$T_d < \frac{AKK_p}{B^2 + 2BKK_p} + \frac{\sqrt{A^2K^2K_p^2 + B(B + 2KK_p)(A^2 - 2B - 2KK_p)}}{B^2 + 2BKK_p} \tag{11}$$

Step 3:- Choose an appropriate value of T_d as per the above analysis.
Step 4:- Apply the following inequality to decide T_i.

$$T_i > \frac{KK_p}{(A + KK_pT_d)(B + KK_p)} \tag{12}$$

$$T_i \geq \frac{2AKK_p}{B^2 + 2BKK_p} \tag{13}$$

$$T_i \geq 2T_d \tag{14}$$

$$T_i \geq \frac{2T_d(A^2 + AKK_pT_d - 2B - 2KK_p)}{A^2 + 2KK_pT_d(A - BT_d) - 2B - 2KK_p - B^2T_d^2} \tag{15}$$

Step 5:- Choose an appropriate value of T_i as per the above analysis.

4. Results and Observations

Algorithm Implementation:-
As per the example equation of the stable second-order system given by equation 4, we have $K = 1, A = 4, B = 5$. These values satisfy the conditions to apply the algorithm given by the equations 7 and 8.
Step 1:- Applying equations 9 and 10, we get the valid range of $K_p \leq 3$. So, we choose $K_p = 2$ for the sake of analysis
Step 2 & 3:- Applying equations 9 and 11, we get the valid range of $T_d < 0.45$ for $K_p = 2$. So, we choose $T_d = 0.35$ for the sake of analysis
Step 4 & 5:- Applying equations 9, 12, 13, 14 and 15, we get the valid range of $T_i > 0.7$ for $K_p = 2$ and $T_d = 0.35$. So, we choose $T_i = 0.8$ for the sake of analysis.
Plotting the transient response of the controller with $K_p = 2$, $T_d = 0.35$ and $T_i = 0.8$, we get $M_P(\%) = 0.0064309\%$, which is fairly close to zero and the following plot:-

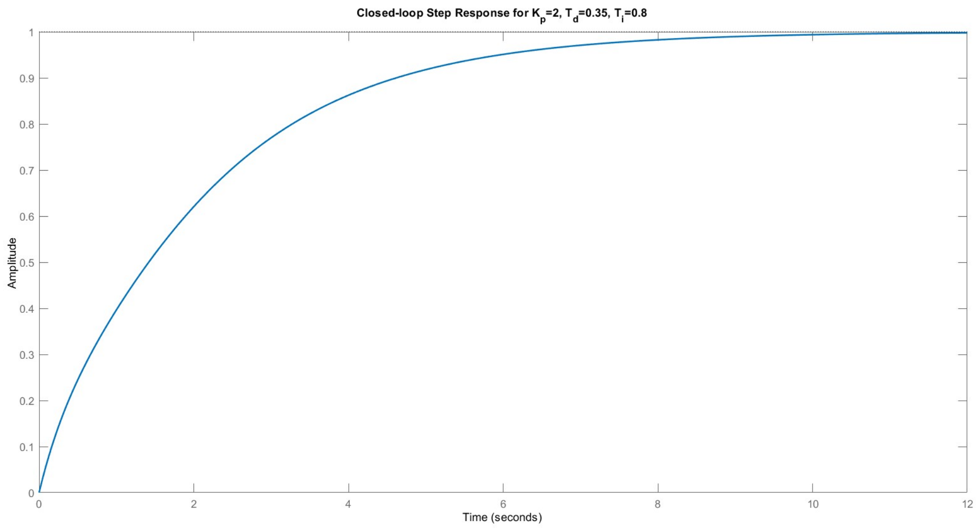

Figure 7: Closed-loop Step Response for $K_p = 2$, $T_d = 0.35$, $T_i = 0.8$

We have used the above algorithm for many other values of K_p and listed the results in the following table:-

	K_p	T_d	T_i	$M_P(\%)$
Case 1	2.5	0.4	0.85	0.00019656%
Case 2	2	0.35	0.8	0.0064309%
Case 3	1	0.45	1	0.0000049949%

Table 1: Maximum Overshoot for different values of controller parameters

So, we can see that after solving the inequalities, we get various region of values for the controller parameters. Choosing some random values of these parameters satisfying the range, we can see that the Maximum Overshoot will always be very small and can be fairly ignored. Hence, we achieve a no-overshooting PID Controller for the system.

5. Applications

The spring-damper system can be used in the following industrial applications:-

- **Machinery Vibration Isolation:** Industrial machinery, such as compressors, pumps, and generators, can generate vibrations during operation. Spring-mass-damper systems are used to isolate these vibrations, preventing them from transferring to surrounding structures and minimizing noise levels.

- **Precision Manufacturing Equipment**: High-precision manufacturing equipment, such as CNC machines, lithography systems, and semiconductor manufacturing tools, require stable and vibration-free environments for precise operation. Spring-mass-damper systems are used to dampen vibrations and ensure accurate machining and fabrication processes.

- **Aerospace and Aviation**: Aircraft and spacecraft employ spring-mass-damper systems in their landing gears, suspension systems, and vibration isolation mounts to absorb landing shocks, dampen vibrations, and ensure passenger comfort. These systems also provide structural protection during takeoff, landing, and in-flight operations.

- **Railway and Transportation Systems**: Railway vehicles, such as trains and trams, utilize spring-mass-damper systems in their suspension systems to provide a comfortable ride, improve stability, and reduce wear and tear on tracks and wheels. These systems absorb shocks and vibrations from uneven tracks and external disturbances.

- **Energy Absorption and Safety Devices**: Spring-mass-damper systems are used in safety devices, such as impact attenuators, crash barriers, and shock absorbers, to absorb and dissipate kinetic energy during collisions and impacts. These systems protect personnel, equipment, and infrastructure from damage and minimize injury risks.

6. Conclusion

This report highlights the implementation of the non-overshooting PID controller on a spring-damper system, which is a second-order stable system. By implementing this controller on a spring damping system, we have demonstrated its effectiveness in maintaining stability and achieving precise control over the system's position. This has promising applications in various fields where precise control without overshoot is crucial for system performance and safety. The advancement in control theory presented by the paper opens doors to new possibilities for achieving stable and precise control in engineering systems.

7. Matlab code for simulation results

Listing 1: Closed Loop Transient Response Code for Proportional Control - Figure 4

```
% Define transfer functions G(s) and C(s)
num_G = 1;
den_G = [1 4 5]; % Numerator and denominator coefficients of G(s)
G = tf(num_G, den_G); % Transfer function G(s)

num_C = 250; % Numerator coefficients of C(s)
den_C = 1;
C = tf(num_C, den_C); % Transfer function C(s)

% Calculate the closed-loop transfer function T(s)
T = feedback(C*G, 1);

% Plot the step response of the closed-loop system
figure;
step(T);
title('Closed-loop Step Response');
xlabel('Time');
ylabel('Amplitude');
```

Listing 2: Code for $M_p(\%)$ vs K_p - Figure 5

```
k_values = 150:1:1000;
max_overshoots = zeros(size(k_values));

for i = 1:length(k_values)
    % Define transfer functions G(s) and C(s)
    num_G = 1;
    den_G = [1 4 5]; % Numerator and denominator coefficients
        of G(s)
    G = tf(num_G, den_G); % Transfer function G(s)

    k = k_values(i);
    num_C = k; % Numerator coefficients of C(s)
    den_C = 1;
    C = tf(num_C, den_C); % Transfer function C(s)

    % Calculate the closed-loop transfer function T(s)
    T = feedback(C*G, 1);

    % Simulate step response and get overshoot
    step_info = stepinfo(T);
    max_overshoots(i) = step_info.Overshoot;
end

% Plot maximum overshoot vs k with a bold line
figure;
plot(k_values, max_overshoots, 'LineWidth', 1.5); % Set line
   width to 2
xlabel('k');
ylabel('Maximum Overshoot');
title('Maximum Overshoot vs k');
```

Listing 3: Closed Loop Transient Response Code for PID Control - Figure 7

```
% Define transfer functions G(s) and C(s)
num_G = 1;
den_G = [1 4 5]; % Numerator and denominator coefficients of
   G(s)
G = tf(num_G, den_G); % Transfer function G(s)

num_C = [0.56 1.6 2]; % Numerator coefficients of C(s)
```

```
den_C = [0.8 0];
C = tf(num_C, den_C); % Transfer function C(s)

% Calculate the closed-loop transfer function T(s)
T = feedback(C*G, 1);

% Plot the step response of the closed-loop system
figure;
step(T);
title('Closed-loop Step Response for K_p=2, T_d=0.35, T_i=0.8
    ');
xlabel('Time');
ylabel('Amplitude');
```

8. References

1. Mohammad Tabatabaei & Reza Barati-Boldaji, Non-overshooting PD and PID controllers design, Automatika, https://doi.org/10.1080/00051144.2018.1471824