GEBZE TECHNICAL UNIVERSITY
ELECTRONICS ENGINEERING DEPARTMENT

ELEC 322 CONTROL SYSTEMS

DC Motor Speed Control

Prepared by:

Efe BAYRAKÇEKEN- 210102002026
Fouad ALADHAMI- 210102002808
Sefa DEMİRTÜRK- 1901022254
Beyzanur CAM- 210102002037
Erhan GÖK- 200102002027

Table of Contents

1 Introduction

This project focuses on developing a comprehensive understanding of DC motor speed control through mathematical modeling and control strategies. The key objectives include developing a Linear Time-Invariant (LTI) mathematical model, determining the transfer function and state space representation, and evaluating the transient response of the system. Additionally, the project involves performing stability checks, analyzing steady state responses for standard input signals, and optimizing the system using Proportional (P), Proportional-Integral (PI), and Proportional-Integral-Derivative (PID) controllers.

By employing the Ziegler-Nichols tuning method and plotting the root locus of the system, we aim to identify the best combinations of controller gains to minimize errors and improve system stability and response quality. Furthermore, the project explores the use of lead, lag, or lead-lag compensators for output optimization and discusses the impact of cascade and feedback compensations on achieving the ultimate design goals.

2 System modeling

2.1 Transfer Function Determination with Laplace Transform

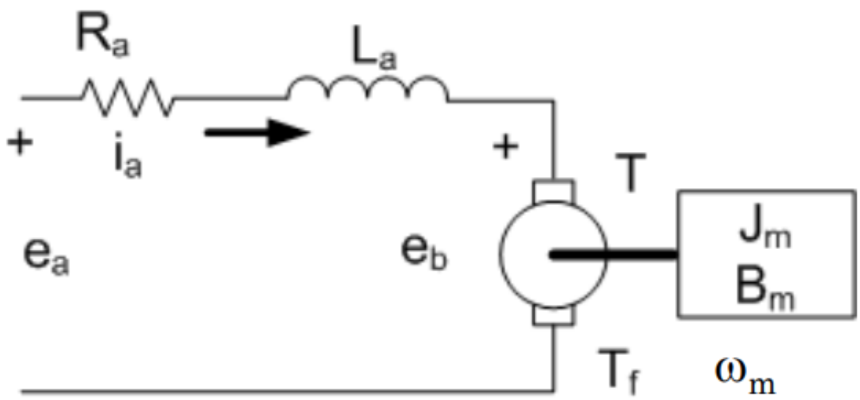

Table 1: DC Motor Circuit Model.

First, the time domain representation of the signals must be found. Since there is a direct correlation between e_b and the output torque using the electromechanical relationships to couple the two equations, a KVL loop around the armature can be written as below.

$$e_a(t) = i_a R_a + L\frac{di_a(t)}{dt} + e_b(t)$$

$$T(t) = K_t \cdot i_a(t)\,,\; e_b = K_e \cdot w_m(t)$$

Using the mechanical dynamics of the system, the formulas can be written.

$$T(t) = \frac{J_M\big(dw_m(t)\big)}{dt} + B_m w_m(t)\; Mechanical\ dynamics$$

To find the transfer function of the system the system, the Laplace transform of the system needs to be calculated.

$$E_a(s) = I_a(s) \cdot R_a + LsI_a(s) + E_b(s)$$
$$E_a(s) = I_a(s)[R_a + sl] + E_b(s)$$
$$E_a(s) - E_b(s) = I_a(s)[R_a + sl]$$
$$I_a(s) = \frac{1}{ls + R_a}[E_a(s) - E_b(s)]$$
$$T(s) = K_E I_a(s)\ E_b(s) = K_e \cdot \Omega_m(s)$$
$$T(s) = [T_m s + B_m]\Omega_m(s) \Rightarrow \Omega_m(s) = \frac{1}{J_m s + B_m} \cdot T(s)$$

Using the feedback formula, the system's closed loop transfer function can be calculated.

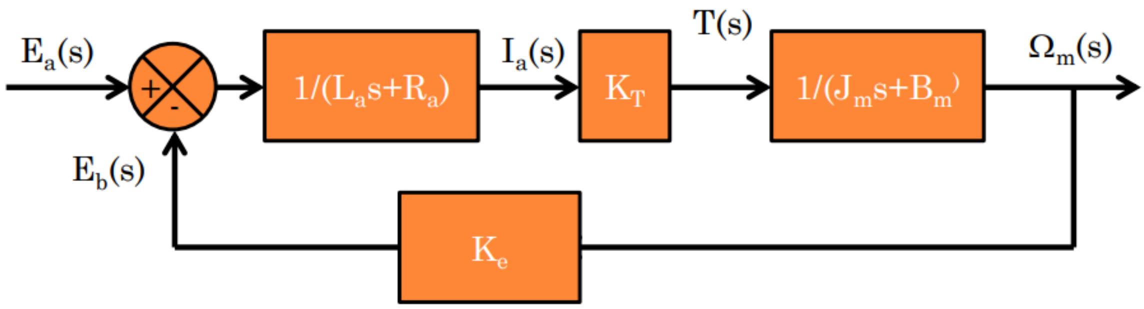

Figure 1: Block Diagram of System.

$$\frac{\Omega_m(s)}{E_a(s)} = \frac{G(s)}{1 + G(s)}$$

$$G(s) = \frac{K_T}{(L_a s + R_a)(I_m s + B_m)}, \qquad H(s) = K_e$$

$$\frac{\Omega_m(s)}{E_a(s)} = \frac{G(s)}{1 + G(s)} = \frac{K_T}{L_a I_m s^2 + (R_a J_m + B_m L_a)_s + (K_T K_E + R_a \cdot B_m)}$$

The problem with this closed loop TF is that the input is a voltage, and the output is angular velocity. This can be solved by adding a feedback loop with unit gain to the output. Then the calculation of the total closed loop TF can be achieved.

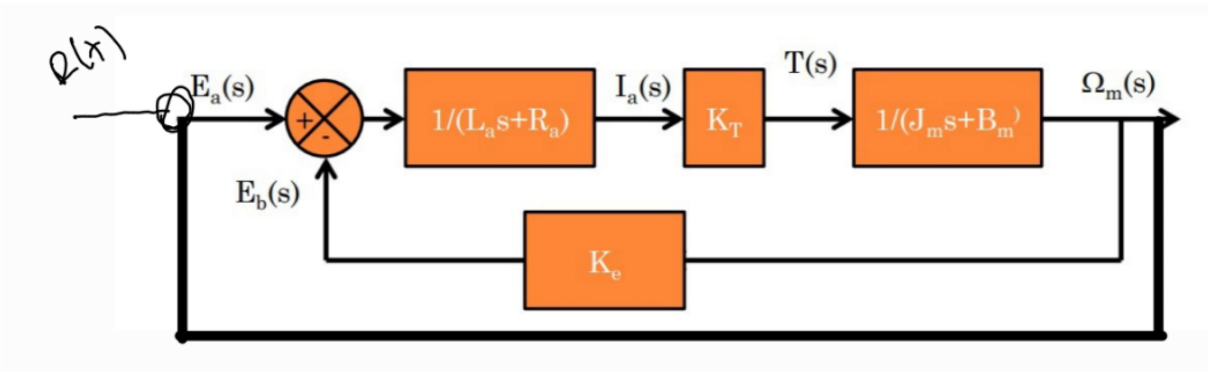

Figure 2: Block Diagram of System with Angular Velocity as Input.

$$G_1(s) = \frac{\Omega_m(s)}{E_a(s)} = \frac{G(s)}{1 + G(s)} = \frac{K_T}{L_a I_m s^2 + (R_a J_m + B_m L_a)_s + (K_T K_E + R_a \cdot B_m)}$$

$$\frac{\Omega_m(s)}{\Omega_r(s)} = \frac{G_1(s)}{1 + G_1(s)} = \frac{K_T}{L_a I_m s^2 + (R_a J_m + B_m L_a)_s + (K_T K_E + R_a \cdot B_m + K_T)}$$

$\frac{L_a J_m s^2}{k_T}$=α $\frac{(R_a J_m + B_m L_a)s}{K_T}$=β $\frac{(k_T K_E + R_a B_m + k_T)}{K_T}$= γ

2.2 State Space Representation of The System

To find the state space representation, the time domain differential equations need to be written.

Since the differential equations are of second order, two state variables must be defined.

The output of the system depends on the state variable x_1 the output matrices is as below.

$$\Omega_r(s) = \Omega_m(s)\alpha s^2 + \Omega_m(s)\,\beta s + \Omega_m(s)\gamma$$
$$wr(t) = \alpha \ddot{w}_m(t) + \beta \dot{w}_m(t) + \gamma w_m(t)$$
$$w_m = x_1 \qquad \dot{x}_1 = \dot{w}_n = x_2$$
$$\dot{w}_m = x_2 \qquad \dot{x}_2 = \ddot{w}_m = \frac{e_a}{\alpha} - \frac{\beta \dot{w}_m}{\alpha} + \frac{\gamma}{\alpha} w_m = \frac{e_q}{\alpha} - \frac{\beta}{\alpha} x_2 + \frac{\gamma}{\alpha} x_1$$
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{\gamma}{\alpha} & \frac{-\beta}{\alpha} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{\alpha} \end{bmatrix} e_q$$
$$w_m = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \dfrac{k_T K_2 + R_a B_m + K_1}{L_q J_m} & -\dfrac{R_a J_m + B_m L_q}{L_q J_m}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ 1\\ \dfrac{}{L_a J_m}\end{bmatrix} e_q$$

$$\frac{K_T}{L_a J_m} X = \frac{(K_T K_E + R_a \cdot B_m + K_r)}{L_a J_m}$$

$$X = \frac{(k_T k_E + R_a B_m + K_T)}{K_t}$$

$$\frac{\Omega_m(s)}{\Omega_r(s)} = \frac{\dfrac{K_T}{L_a J_m}}{s^2 + \dfrac{\left(R_a J_m + B_m L_q\right)_s}{L_a J_m} + \dfrac{(K_T K_E + R_a \cdot B_m + K_T)}{L_a J_m}} = \frac{1}{x}\frac{\omega_n^2}{s^2 + 2\xi W_n S + \omega_n^2}$$

$$w_n = \sqrt{\frac{(K_T K_E + R_a \cdot B_m + K_r)}{L_a J_m}} \qquad \xi = 2\sqrt{\Bigg(\frac{(R_a J_m + B_m L_a)^2}{(L_a J_m)(K_T K_e + R_a K_m + Kt)}}$$

$$w_d = w_n\sqrt{1-\xi^2} = \sqrt{\frac{(K_T K_E + R_a \cdot B_m \cdot K_T)}{L_a J_m}}\sqrt{1 - 4\frac{(R_a J_m + B_m L_m)^2}{(L_m J_m)(K_r k_e + L_m m_m + K_t)}} = \sqrt{\frac{(K_T K_E + R_a \cdot B_m + K_k)}{L_a J_m} - 4\frac{(R_a J_m + B_m L_m)^2}{(L_a J_m)^2}}$$

3 Transient Response Quality of The Original System to Step Excitation.

3.1 Analytical solution

The system is second order, so the simple calculations that are general to the step excitation can be found. The damping frequency and the damping factor can be calculated parametrically.

To find the quality of the response, the time domain representation of the system must be found.

The transient response, maximum overshoot, settling time, and peak time can be calculated.

3.1.1 Time Domain Representation

Reminders: $\omega_d = \omega_n\sqrt{1-\xi^2} \qquad \omega_n = \frac{\omega_d}{\sqrt{1-\xi^2}}$

$$\mathcal{L}[e^{-at}\sin\omega t] = \frac{\omega}{(st-\omega^2+\omega)} \quad \mathcal{L}[e^{-at}\cos\omega t] = \frac{s+a}{(s+a)^2+\omega}$$

$$\frac{\Omega_m(s)}{\Omega(s)} = \frac{\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2}\cdot\frac{1}{X}\; E(s) = \frac{1}{s}$$

$$\Omega(s) = \frac{\omega_n^2}{s(s^2+2\varepsilon\omega_n s+\omega_n^2)}\cdot\frac{1}{X} = \left(\frac{A}{s}+\frac{Bs+c}{s^2+2\xi\omega_n s+\omega_n^2}\right)\cdot\frac{1}{X}$$

$$= \left(\frac{1}{s}-\frac{s+2\xi\omega_n}{s^2+2\xi\omega_n s+w_n^2}\right)\frac{1}{x}$$

$$= \left(\frac{1}{s}-\frac{s+\varepsilon\omega_n}{(s+\xi\omega_n)^2+\omega_d^2}-\frac{\varepsilon\omega_n}{(s+\xi\omega_n)^2+\omega_d^2}\right)\cdot\frac{1}{x}$$

$$= \left(\frac{1}{s}-\frac{s+\frac{\xi\cdot\omega_d}{\sqrt{1-\varepsilon^2}}}{\left(s+\frac{\xi\cdot\omega_d}{\sqrt{1-\varepsilon^2}}\right)^2+\omega_d^2}-\frac{\xi\cdot\frac{\omega_d}{\sqrt{1-\varepsilon^2}}}{\left(s+\frac{\varepsilon\cdot\omega_d}{\sqrt{1-\xi^2}}\right)^2+\omega_d^2}\right)\cdot\frac{1}{x}$$

$$\omega(t) = \mathcal{L}^{-1}[\Omega(s)] = \left(1-e^{-\xi\omega_n t}\cdot\cos\omega_d t-\frac{\varepsilon}{\sqrt{1-\varepsilon^2}}\cdot e^{-\varepsilon\omega_n t}\sin\omega_d t\right)\cdot\frac{1}{x}$$

$$\omega(t) = \frac{1}{x}-\frac{e^{-\varepsilon\omega_n t}}{x}\cdot\cos\omega_d t-\frac{\varepsilon}{x\sqrt{1-\varepsilon^2}}\cdot e^{-\varepsilon\omega_n t}\cdot\sin\omega_d t$$

$$x = \frac{K_T K_E + R_a B_{m+1} K_T}{K_T}$$

3.1.2 Quality of Transient Response

$$t_r = \frac{l}{w_d} \cdot \tan^{-1}\left(-\frac{w_d}{\xi w_n}\right)$$

$$\mu_p = \frac{1}{x}\left(1 - e^{-\frac{11\xi}{\sqrt{1-\xi^2}}}\right)$$

$$t_p = \frac{\pi}{w_d}$$

$$\%2 \Rightarrow t_s = \frac{4}{5w_n}$$

$$G(s) = \frac{\Omega_m(s)}{E_a(s)} = \frac{K_T}{L_a J_m s^2 + (R_a J_M + B_m L_a)s + (K_T K_E + R_a \cdot B_m)}$$

$$e_{ss}=e(\infty) = \lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s) = \lim_{s\to 0} \frac{sR(s)}{1+G(s)}$$

$$E(s) = \frac{R(s)}{1+G(s)}$$

$$\text{Impulse } \rightarrow e_{ss} = \lim_{s\to 0} \frac{5}{1+G(s)} = 0$$

$$\text{Step}\left(R(s) = \frac{1}{s}\right) \rightarrow e_{ss} = \lim_{s\to 0} \frac{1}{1+G(s)} = \frac{K_T K_E + R_a B_m}{K_T}$$

$$\text{Ramp}\left(R(s) = \frac{1}{s^2}\right) \rightarrow e_{ss} = \lim_{s\to 0} \frac{1}{s+sG(s)} = \infty$$

$$H(s) = \frac{\Omega_m(s)}{\Omega_r(s)} = \frac{\frac{K_T}{L_a J_m}}{\frac{s^2 + (R_a J_m + B_m L_a)_s}{L_a J_m} + \frac{(K_T K_E + R_a \cdot B_m + K_T)}{L_a J_m}}$$

$$\lim_{t\to\infty} c(t) = \lim_{s\to 0} H(s) \cdot R(s) \cdot s \quad \Big\} \text{ steady state response}$$

$$\text{Impulse } \rightarrow C(\infty) = \lim_{s\to 0} H(s) \cdot s = 0$$

$$\text{Step}\left(R(s) = \frac{1}{s}\right) \rightarrow C(\infty) = \lim_{s\to 0} H(s) = \frac{K_T}{k_T k_E + R_Q R_m + k_T} = \frac{K_T}{K_T K_E + R_a B_m + K_T}$$

$$\text{Ramp}\left(R(s) = \frac{1}{s^2}\right) \rightarrow C(\infty) = \lim_{s\to 0} \frac{H(s)}{s} = \infty$$

3.2 Simulink Simulation

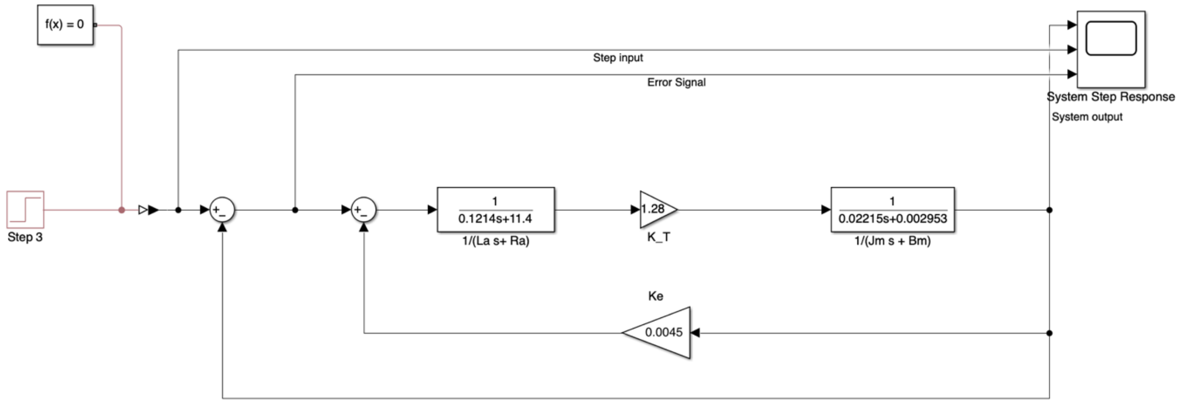

Figure 3: System Block Diagram for step input.

The system was set up in Simulink, as shown in the Figure 3. The Simulink model shown in the first image represents a control system designed to simulate the response of a motor to a step input. It includes blocks for the step input, error signal computation, electrical dynamics (represented by $1/\left(L_a * s + R_a\right)$), and mechanical dynamics (represented by $\frac{1}{J_m * s + B_m}$). Gain blocks for the motor constant (KT) and feedback constant (K_e) are also present. The feedback loop compares the output with the input to adjust the error signal, ensuring stability and performance. The output of the system is shown at Figure 4.

By using the cursor, the %5 error time is found as shown in Figure 4. And then the time constant τ is calculated again.

$$Error\ Time = 3\tau = 0.559s \qquad so \qquad \tau = 0.186\ s.$$

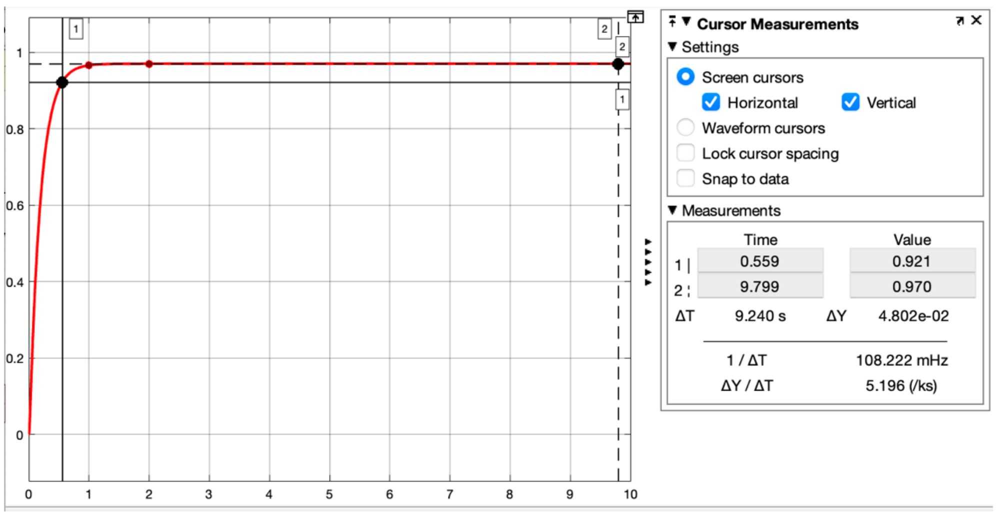

Figure 4: System Step Response graph with cursor at %5 error.

By using the cursor, the %2 error time is found as shown in Figure 5. And then the time constant to calculated again.

$$Error\ Time \ = \ 4\tau \ = \ 0.718s \qquad so \qquad \tau \ = \ 0.180\ s.$$

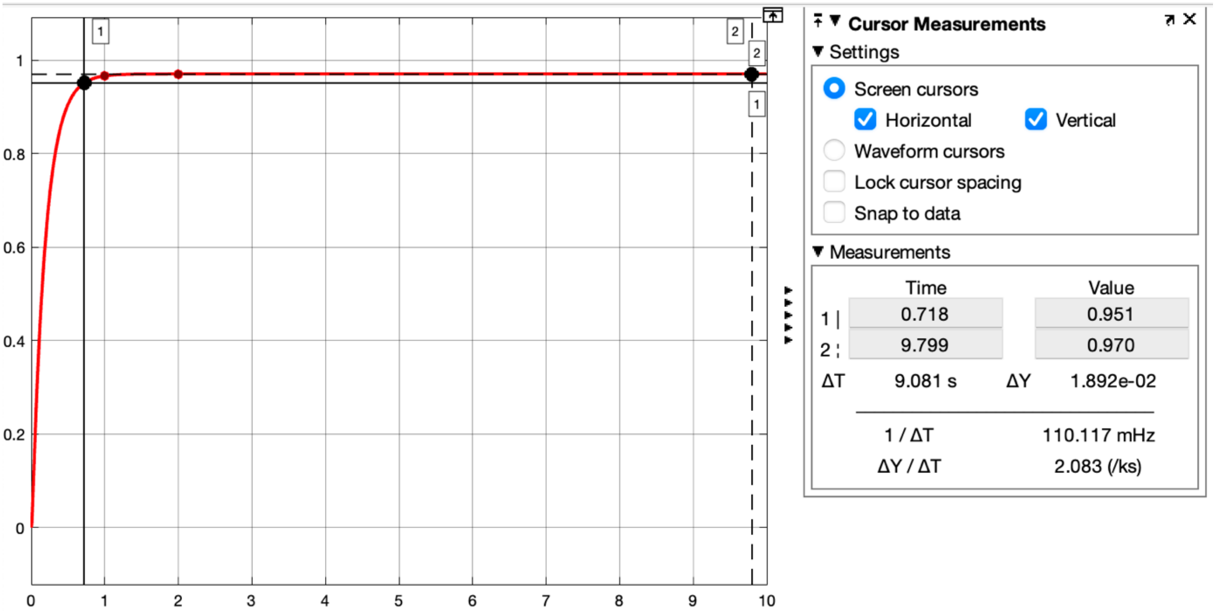

Figure 5: System Step Response graph with cursor at %2 error.

4 Stability Check

A stable system gives finite output for a finite input. For a system to be stable all poles in the transfer function must be in the left-hand side of the imaginary axes.

For this, an example motor with the parameters below was selected. For all the calculations in the report, the parameters in Table 2 are used.

Parameters	**Values**
Armature Resistance (R_a)	11.4 Ω
Armature Inductance (L_a)	0.1214 H
Rotational Inertia (J_m)	0.02215 kg*m^2
Viscous Friction (B_m)	0.002953 Nm/A
Torque Constant (K_t)	1.28 V_s/A
Back Emf Constant (K_e)	0.0045 V_s/rad

Table 2: Example Motor Parameters.

Poles of this system with the parameters in the table was plotted by using MATLAB as shown in Figure 6 below. Here in this system, there are two poles in -5.54 and -88.5 which consequently makes the system stable. And since all the poles are on the real axes and the system is a second order system, it can be said that this systems ζ is greater than one, so this system is overdamped.

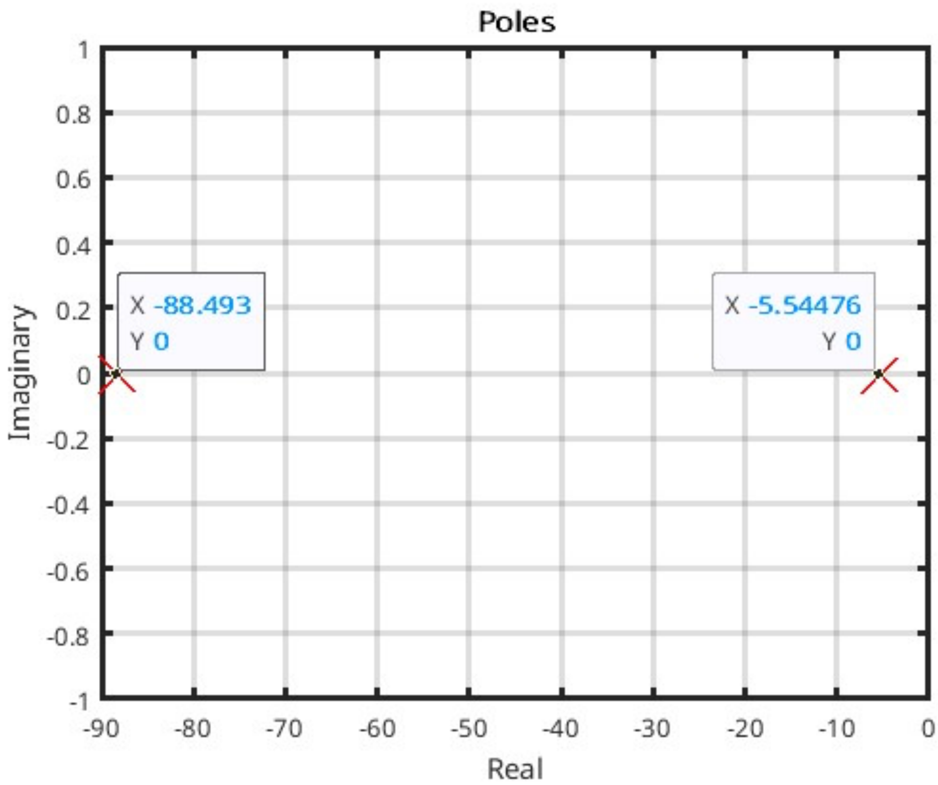

Figure 6: Zero Pole Diagram of System.

5 Impulse, Step and Ramp responses

To find the steady state error in the system, the final value theorem can be used to find the error at t=∞. For all inputs, the formula for the steady state error is shown below.

The steady state response is the limit of the input and the input at t=∞.

5.1 Steady state response and error of Impulse Input

For the impulse input, R(s)=1, the error was calculated to be 0. This is very much expected since for the Dirac input, the input at t=∞, output must be 0 for an LTI system.
The steady state response was calculated below. For the impulse response, the final value is zero.

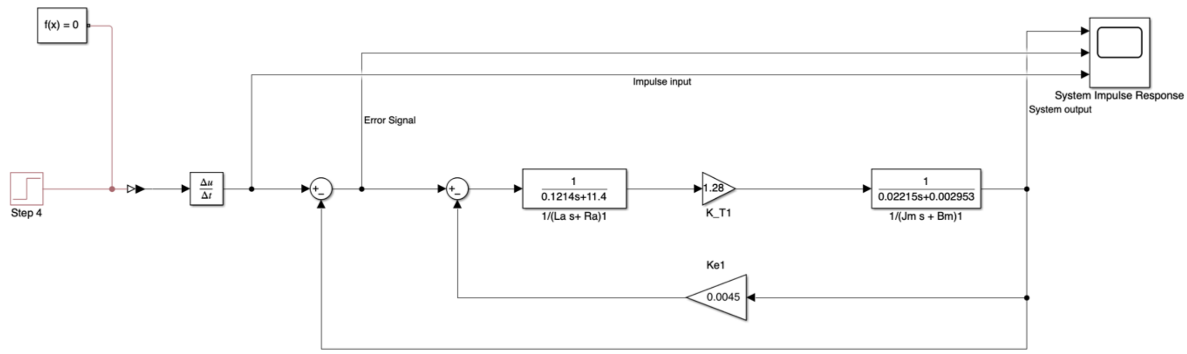

Figure 7: System block diagram in Simulink for impulse input.

The system block diagram in Simulink for impulse input is shown in Figure 7. To generate an impulse input the following steps is used:

- Step Input:

The system begins with a step input block. A step input changes from zero to a specified value at a particular time, typically providing a sudden change in input.

- Differentiation (Δu/Δt):

The step input is then passed through a differentiation block. Differentiation mathematically represents the rate of change of a signal.
When a step input is differentiated, the result is an impulse. This is because the derivative of a step function is a Dirac delta function (impulse), which is a spike of infinite height and infinitesimal width, representing an instantaneous change.

The system output with error signal and the input is plotted as shown in Figure 8. From this simulation it can clearly be seen that the steady state error is zero.

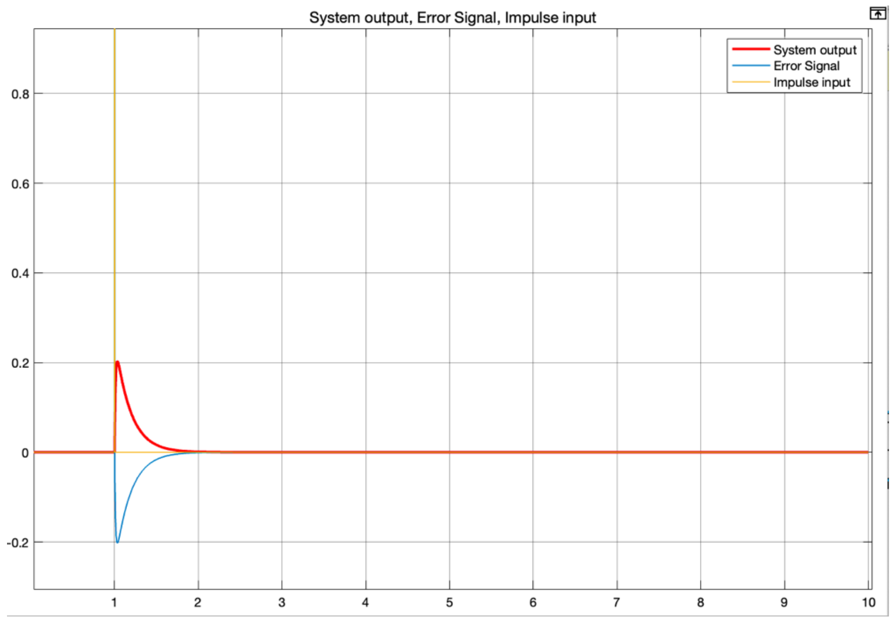

Figure 8: Impulse Response of the system

5.2 Steady state response and error of Step Input

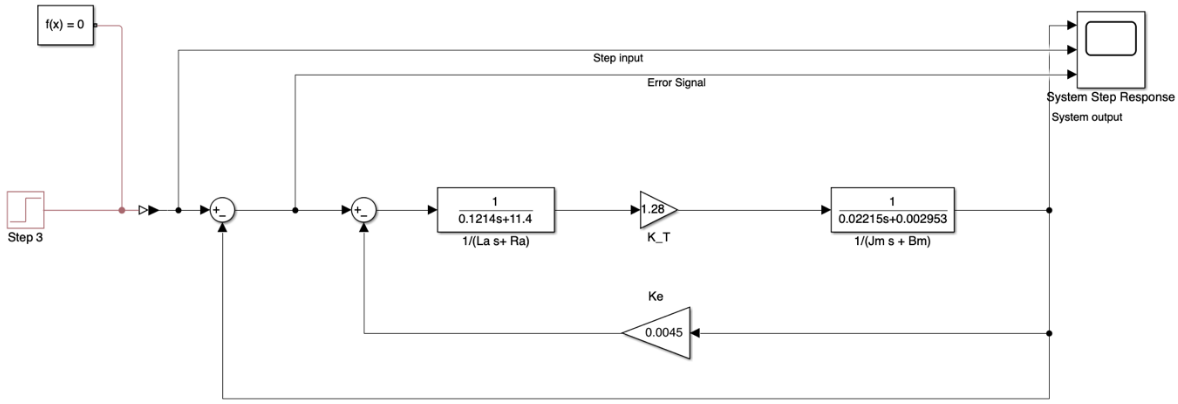

Figure 9: System block diagram in Simulink for step input.

The system block diagram in Simulink for impulse input is shown in Figure 9. The graph effectively illustrates how the system responds to a step input, reaching a steady state with minimal error. The steady state error for a system is the difference between the desired input (the step input in this case) and the actual system output as time approaches infinity (or when the system output stabilizes).

In Figure 8, the step input is one, and the system output stabilizes at approximately 0.970. The steady state error can be calculated as follows:

$$Steady\ State\ Error = Desired\ Input - Steady\ State\ Output$$
$$Steady\ State\ Error = 1 - 0.970 = 0.030$$

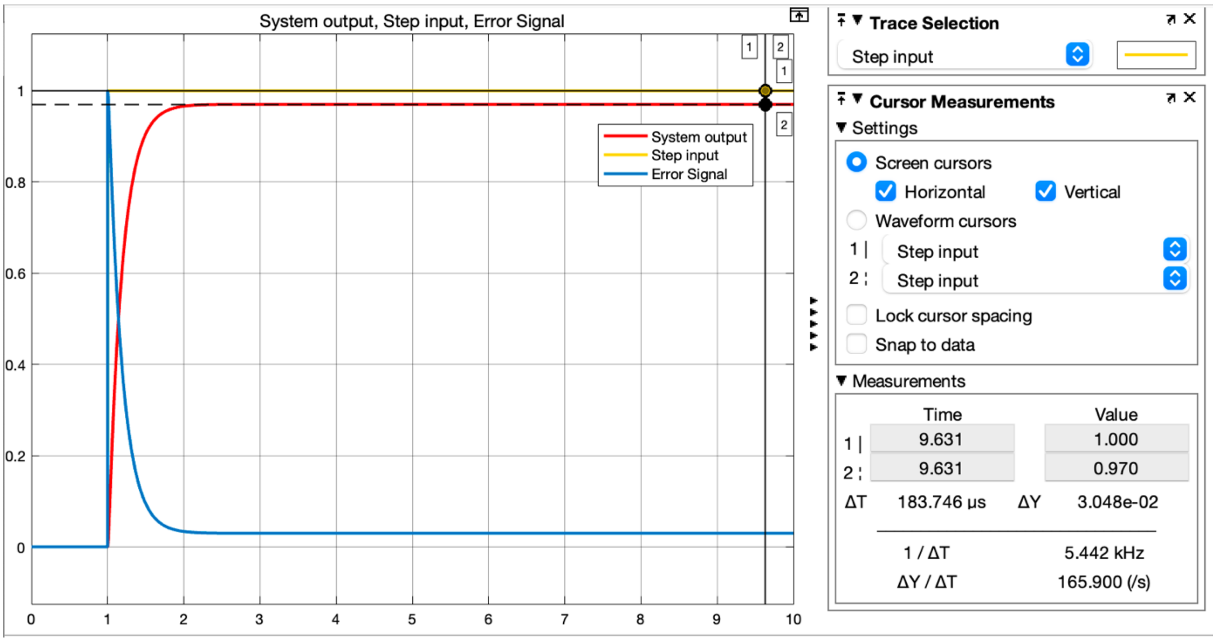

Figure 10: Step response of the system and the error signal.

5.3 Steady state response and error of Ramp Input

For the step input, R(s)=1/s, the error was calculated to be ∞. This means that the system's errors do not decrease. The steady state response was calculated below.

For the step response, the final value is ∞ since the input goes to infinity.

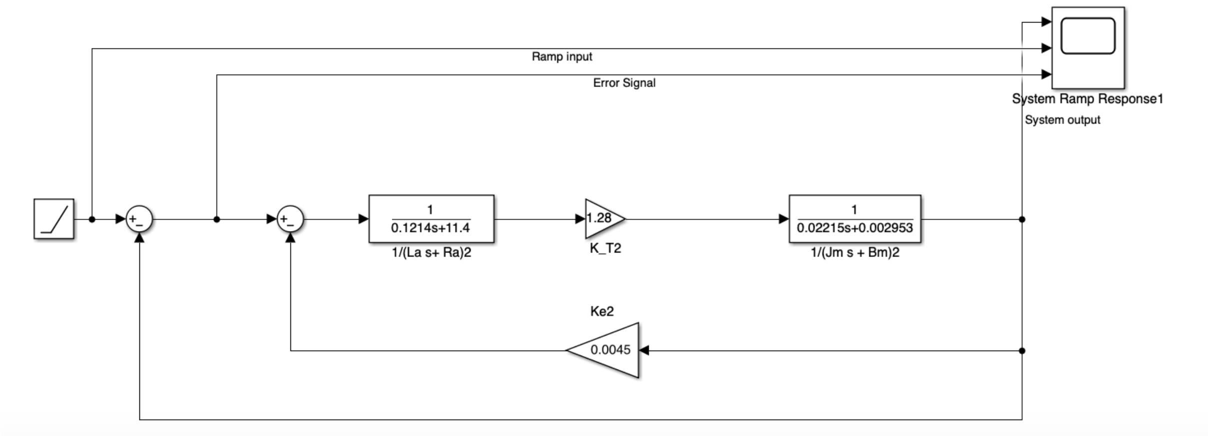

Figure 11: System block diagram in Simulink for ramp input.

The system block diagram in Simulink for impulse input is shown in Figure 11. When a ramp input is given to the system, error goes to infinity too even though it is not clear from Figure 12, it can be observed from Figure 13.

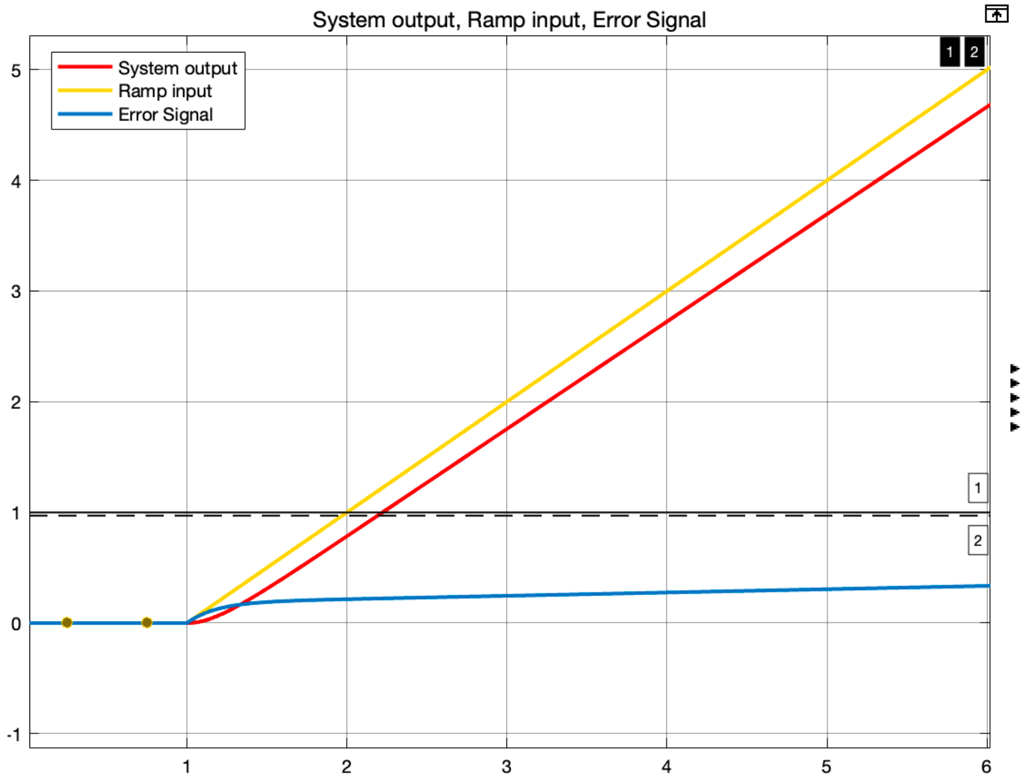

Figure 12: Ramp response of the system and the error signal.

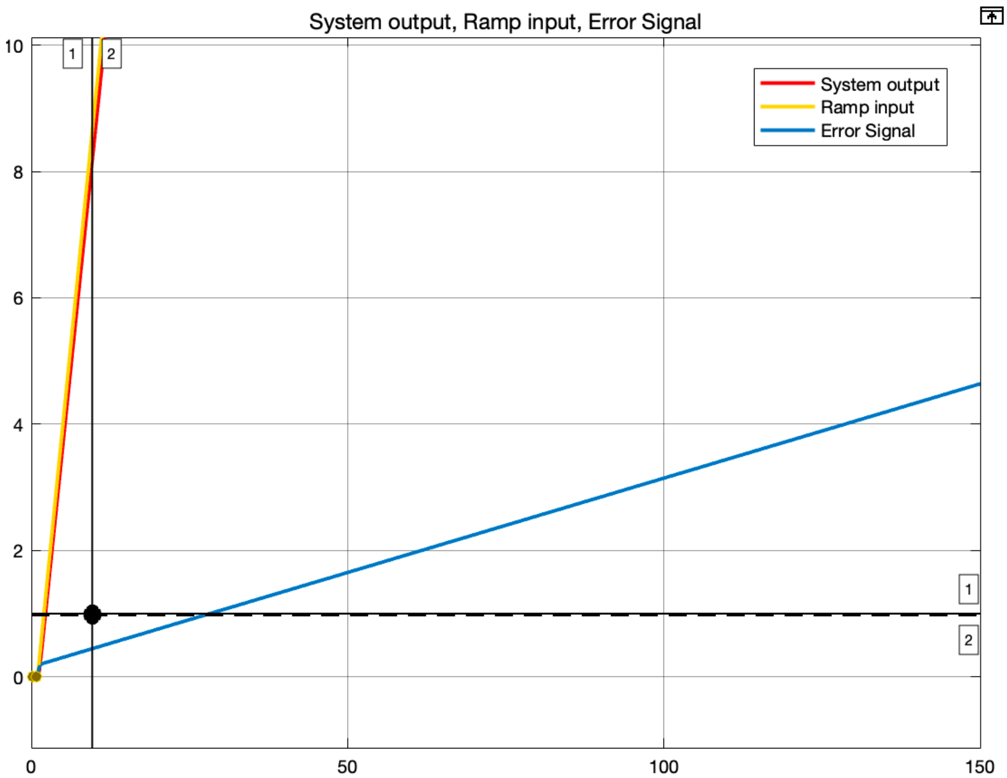

Figure 13: Ramp response of the system and the error signal zoomed in.

6 PID

When kp value is increased by ten times an overshoot is observed from the graphs seen in Figure 14 and Figure 15.

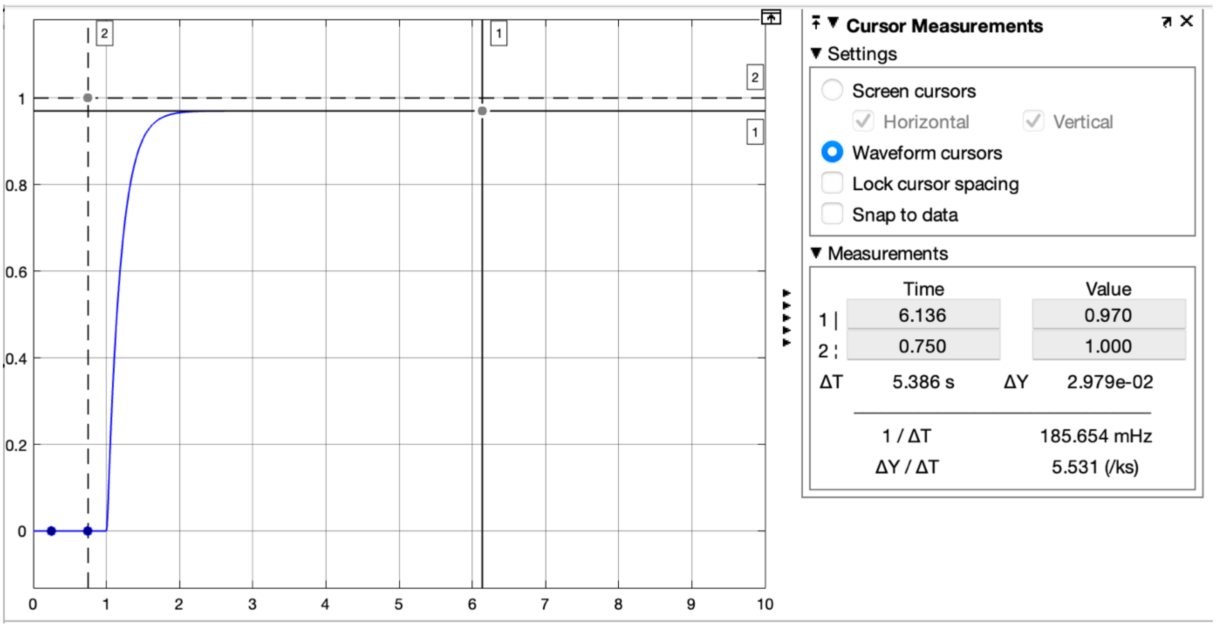

Figure 14: Step Response for kp=1, ki=0, kd=0.

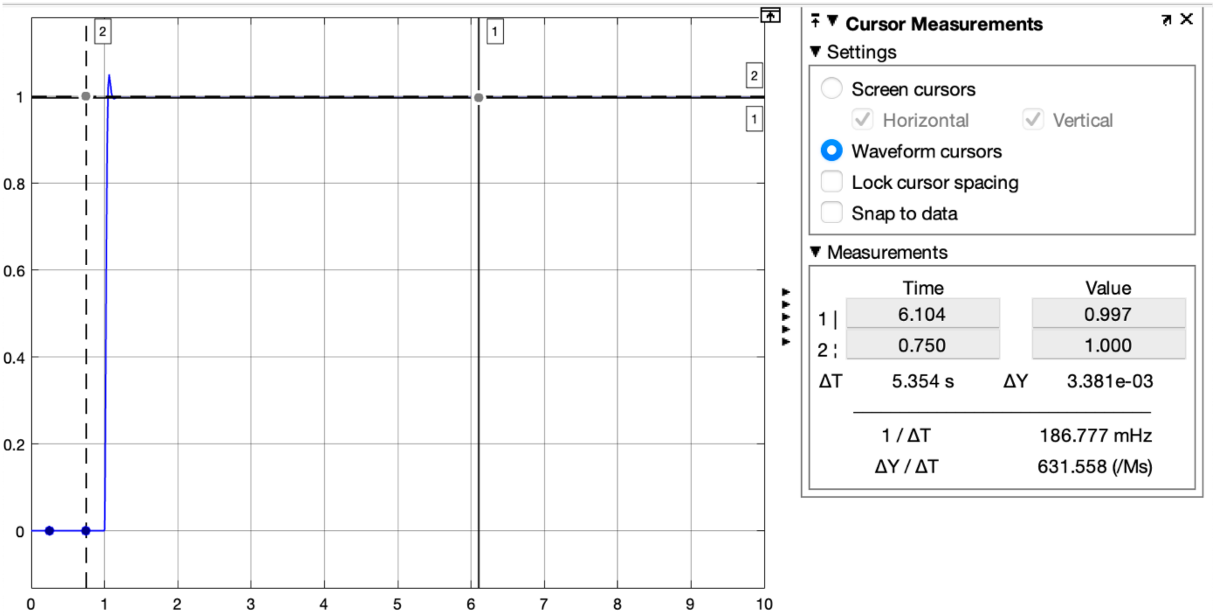

Figure 15: Step Response for kp=10, ki=0, kd=0.

When kp value is hold stable and ki is increased by 10 an overshoot and a weak oscillation is observed from the graphs seen in Figure 16 and Figure 17.

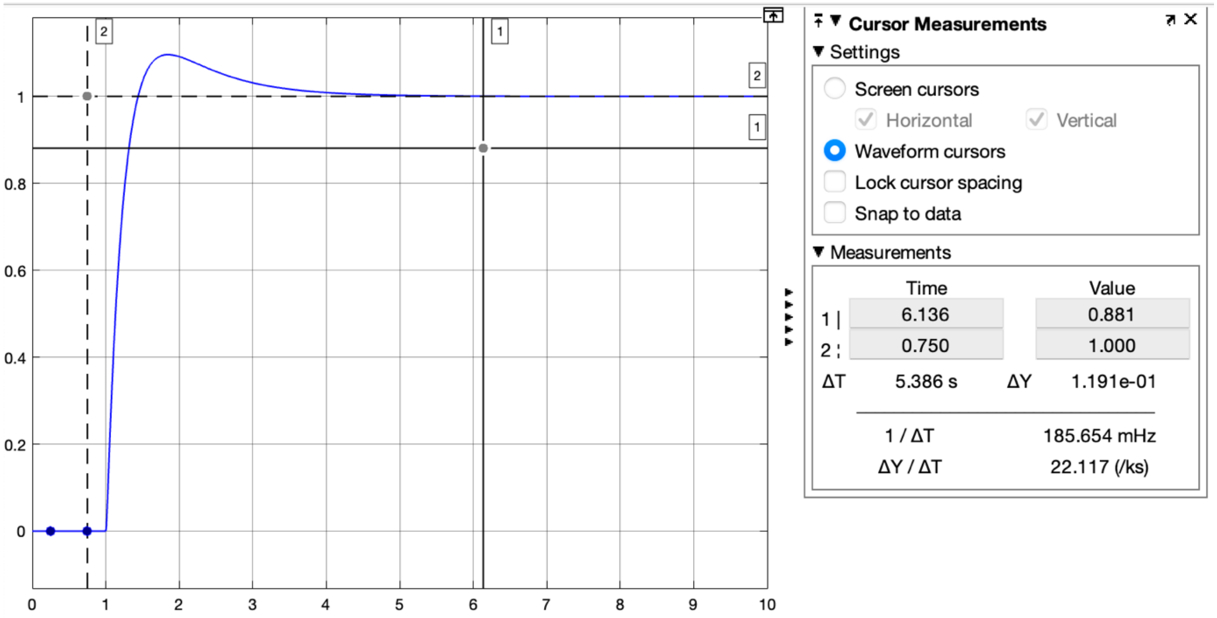

Figure 16: Step Response for kp=1, ki=1, kd=0.

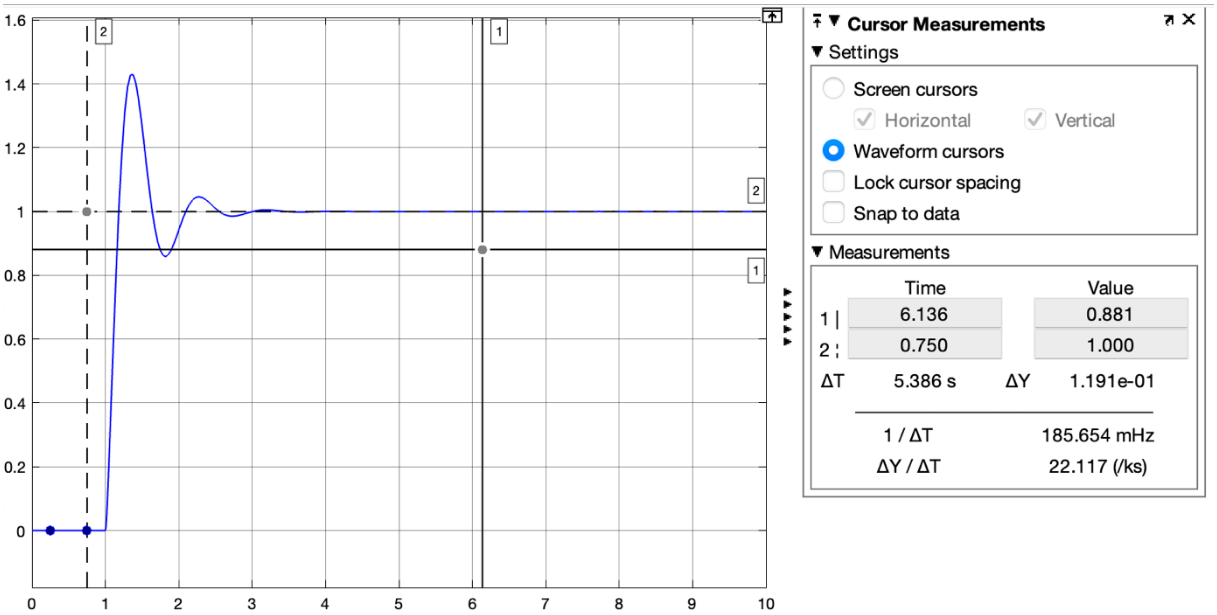

Figure 17: Step Response for kp=1, ki=10, kd=0.

When kp and ki value hold stable and kd is increased by 10 an overshoot and oscillation is observed from the graphs seen in Figure 18 and Figure 19.

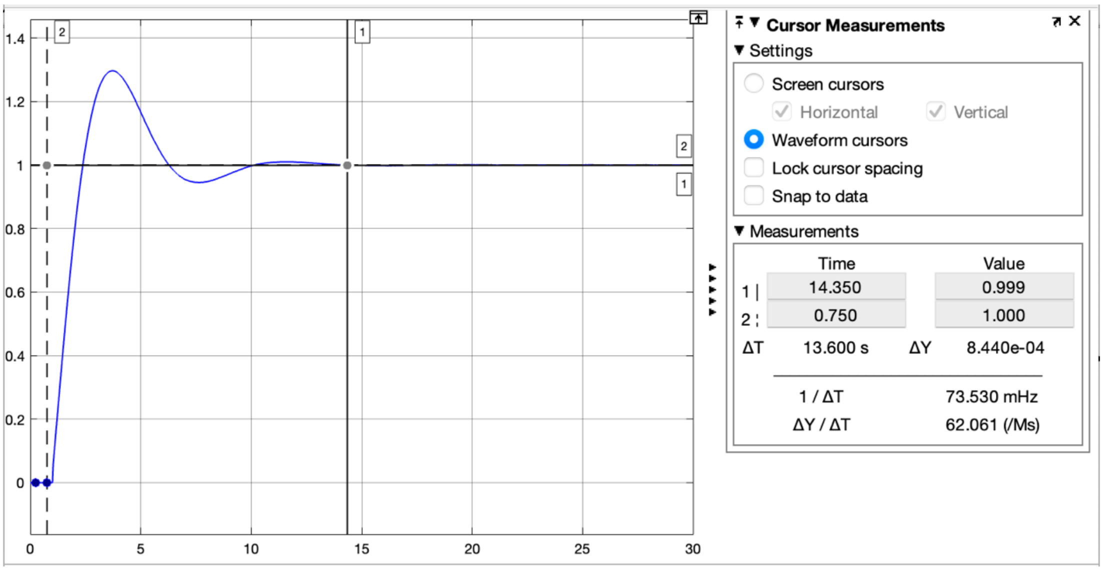

Figure 18: Step Response for kp=1, ki=1, kd=1.

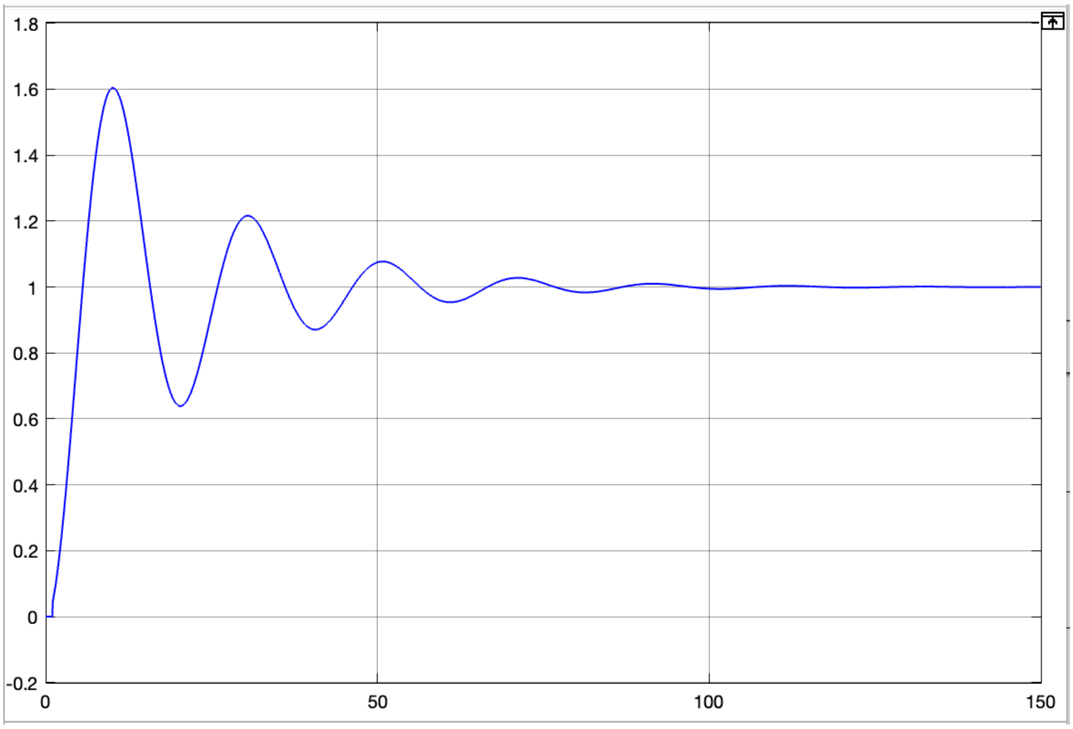

Figure 19: Step Response for kp=1, ki=1, kd=10.

7 Root Locus

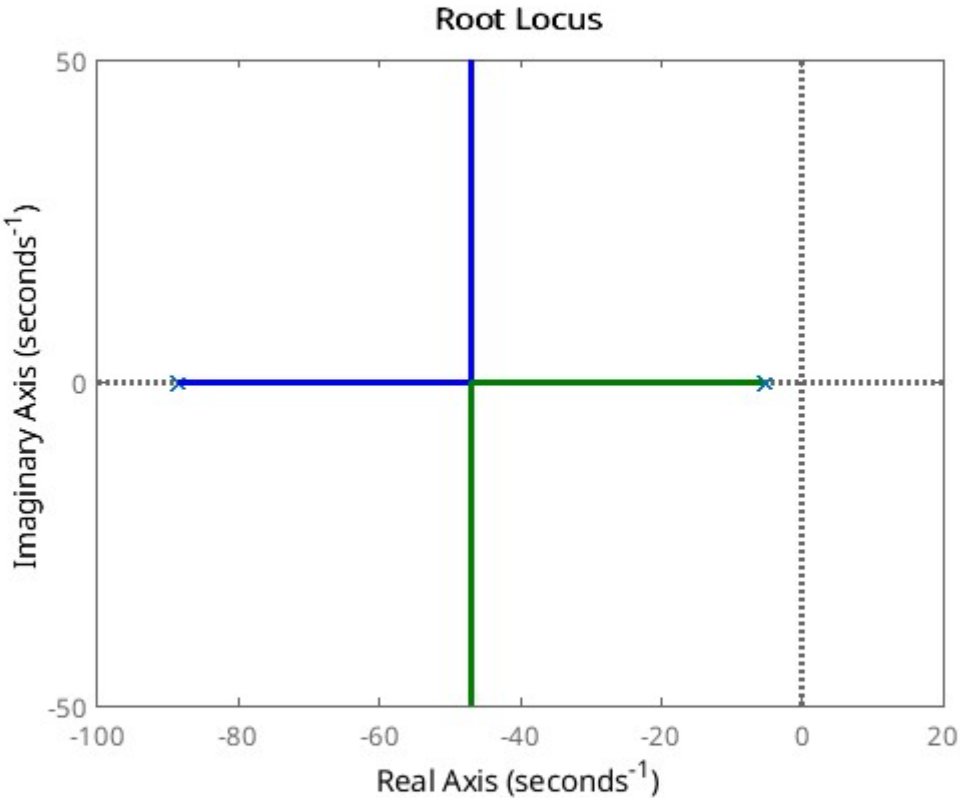

Figure 20: Zero Pole Diagram of System.

Since the number of the poles are two, there must be 2 branches present in the root locus. The root locus begins at the open-lop poles and ends at the open loop zeros. This means that there are zeros at infinity as shown in Figure 20. The angle and the real-axis intercept of these asymptotes can be calculated using the formulas below.

$$n = 2$$
$$m = 0$$
$$S' = \frac{\sum \text{real part of poles} - \sum \text{real part of zeros}}{n - m}$$

$$\Phi = \frac{(2i+1)180^\circ}{n-m} \Rightarrow \Phi = 90^\circ,\ 270^\circ$$

$$S' = \frac{\frac{-b+\sqrt{b^2-4ac}}{2a} + \frac{-b-\sqrt{b^2-4ac}}{2a}}{n-m} = -\frac{(R_a J_m + B_m L_a)s}{2 \cdot L_a J_m}$$

As can be seen, the root locust has 2 asymptotes going to ±j∞.

8 Compensator design

Compensation in control systems is used to enhance performance and stability by adding compensators, such as lead, lag, or lead-lag networks. The main objectives include improving stability, transient response, and steady-state accuracy. Lead compensators increase phase margin and speed up response, lag compensators reduce steady-state error, and lead-lag compensators provide a balanced improvement in both areas. Compensation ensures the system meets desired specifications, responds effectively to disturbances, and maintains robust performance despite parameter variations.

8.1 Lead Compensator

Designing a lead compensator involves placing a zero and a pole in the Laplace domain to improve a system's transient response without sacrificing stability. Typically, the zero is positioned closer to the origin to introduce phase lead, while the pole is located at a frequency higher than the dominant poles of the system to boost high-frequency gain. By calculating the compensator transfer function based on these placements and implementing it in the control system, the system can meet the desired specifications.

Values of placed poles and zeros can be found using phase condition. By using geometry in the zero-pole diagram, the phase can be calculated for the desired point that root locus should contain.

$$90 + 90 + \tan^{-1}\frac{P_2 - P_1}{\omega_d} \neq 180$$

$$\tan^{-1}\frac{P_2 - P_1}{\omega_d} + \theta_1 - \theta_2 = 0$$

$$\tan^{-1}\frac{P_2 - P_1}{\omega_d} = \tan^{-1}\frac{\omega_d}{P_2 - P_1} - \tan^{-1}\frac{\omega_d}{P_2 - z} = 0$$

$$P_1 - P = \frac{\omega_d}{\tan\left(-\tan^{-1}\left(\frac{P_2 - P_1}{\omega_d}\right) + \tan^{-1}\left(\frac{\omega_d}{P_1 - z}\right)\right)}$$

$$P = P_1 - \frac{\omega_d}{\tan\left(-\tan^{-1}\left(\frac{P_2 - P_1}{\omega_d}\right) + \tan^{-1}\left(\frac{\omega_d}{P_1 - z}\right)\right)}$$

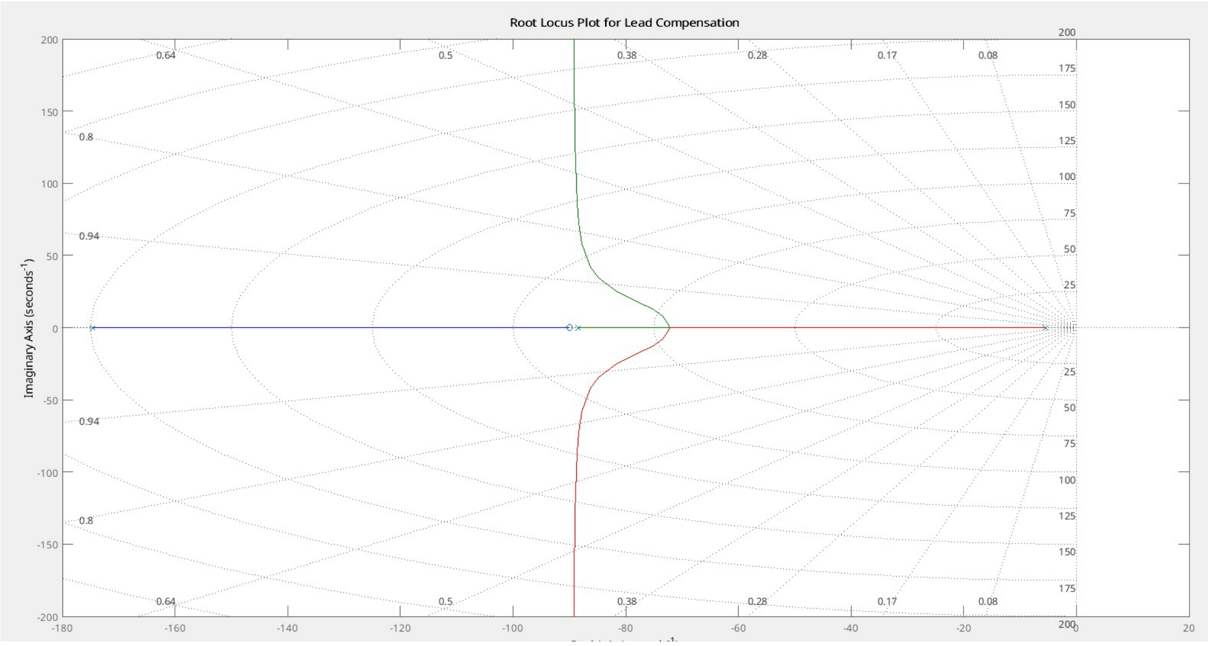

Figure 21: The Root Locus plot after Lead Compensator

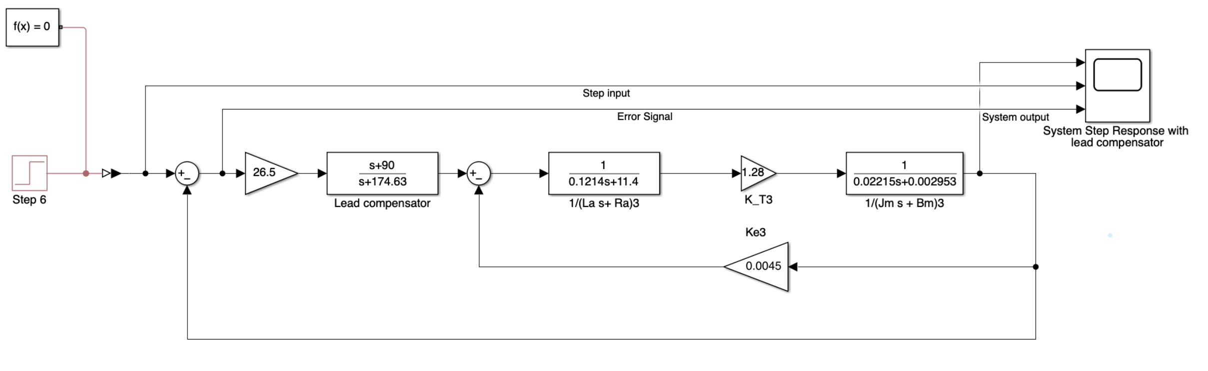

Figure 22: System with Lead Compensator

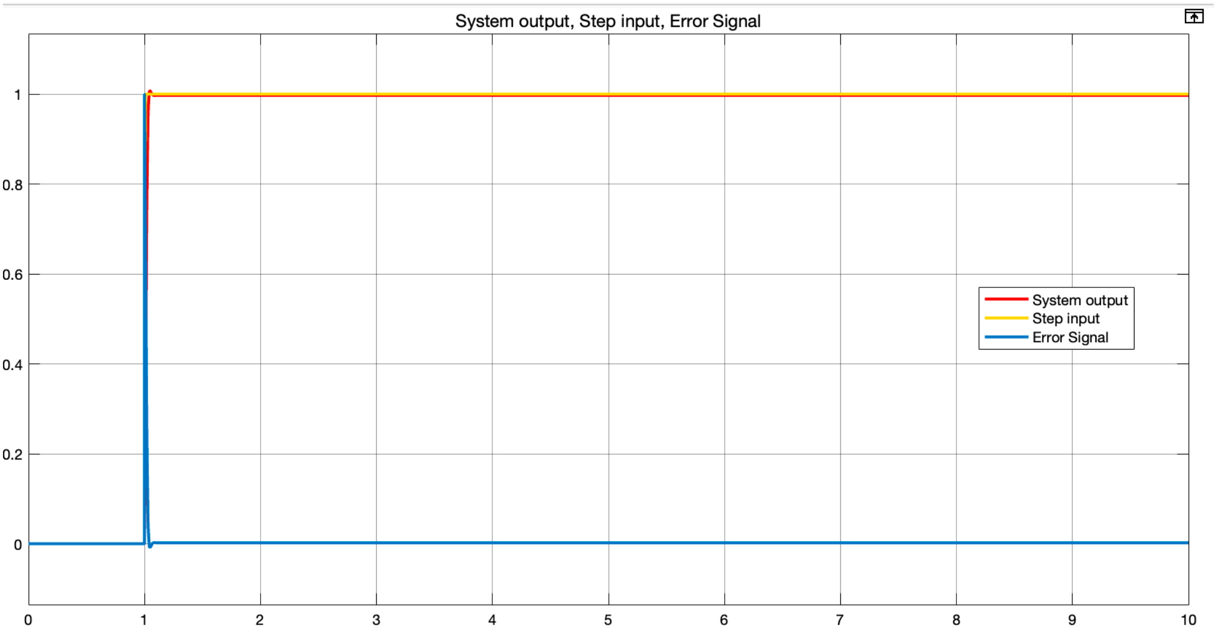

Figure 23: Unit Response of the System with Lead compensator.

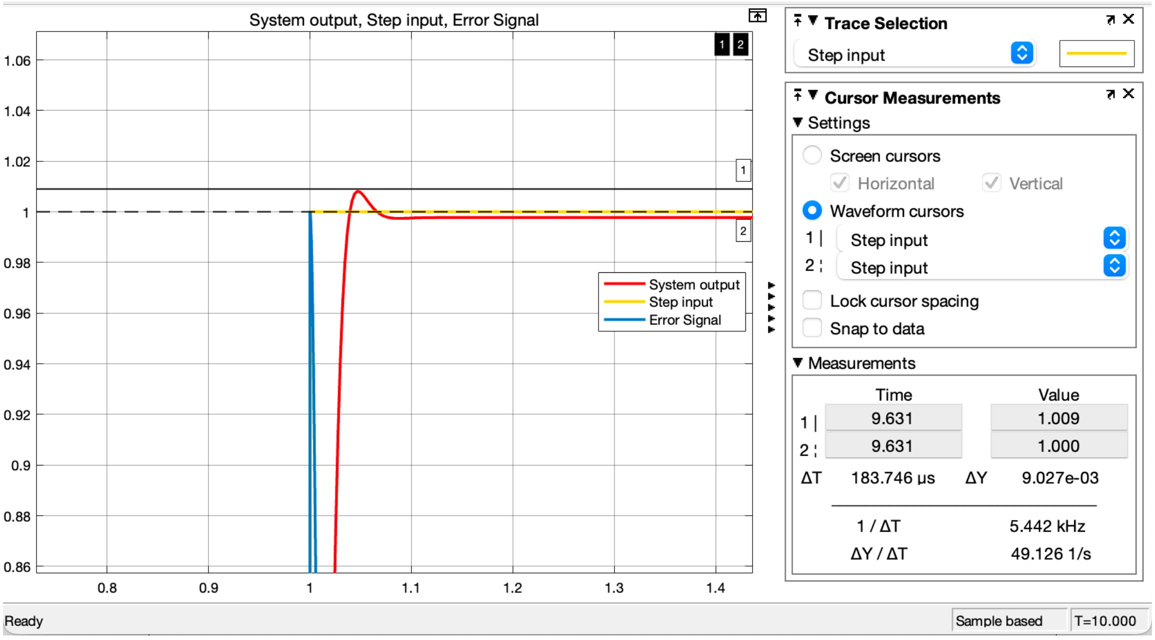

Figure 24: Max overshoot of the system is %0.9.

The lead compensator is designed and the root locus of the system with lead compensator is plotted as shown in Figure 21 and Figure 22. The designed lead compensator achieved a maximum overshoot of 0.9% and a steady-state error of 0.003 as shown in, Figure 23 and Figure 24. The compensator was configured with a zero at z = 90, a pole at p = -174.6307 and the system gain K = 26.5 was adjusted to meet the steady-state error requirement. This design successfully met the specified performance criteria.

8.2 Lag Compensator

Designing a lag compensator involves positioning a pole and a zero in the Laplace domain to enhance a system's steady-state accuracy while maintaining its transient response. By carefully selecting the locations of the pole and zero, compensator's transfer function can be found to meet desired specifications. Adjustments may be needed to ensure stability and optimal performance, followed by validation through simulations or experiments.

By placing the asymptote to the dominant pole and calculating its centroid and phase the pole and zero values can be calculated.

$$\text{centroid} = \frac{\sum Real\ part\ of poles - \sum Real\ part\ of zeros}{n-m} = \frac{P_1 + P_2 + P - Z}{2} = p_2$$

$$P_2 = P_1 + P - Z$$

$$Z = P_1 + P - P_2$$

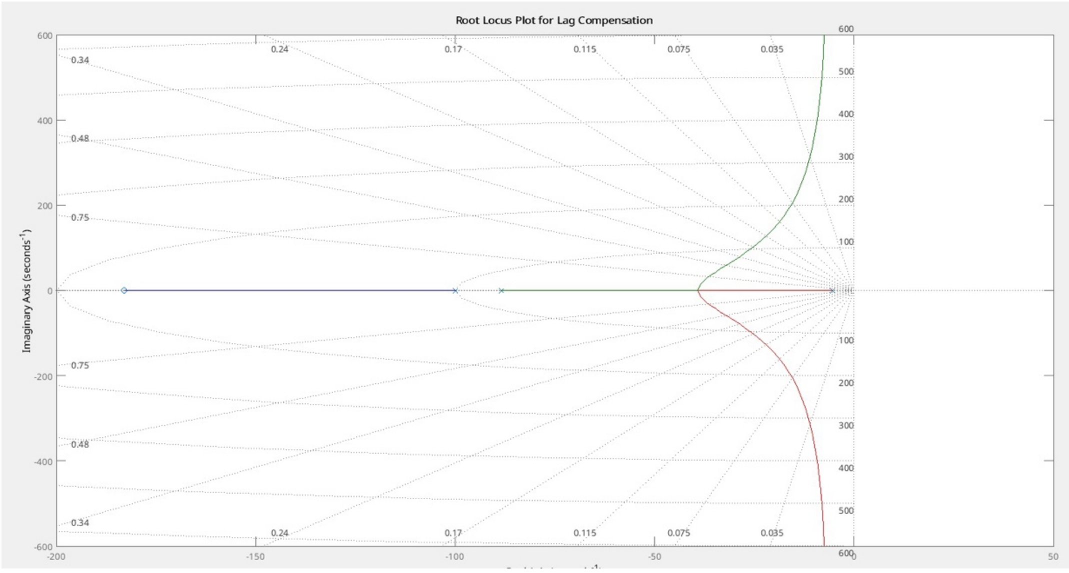

Figure 25: The Root Locus plot after Lag compenzation

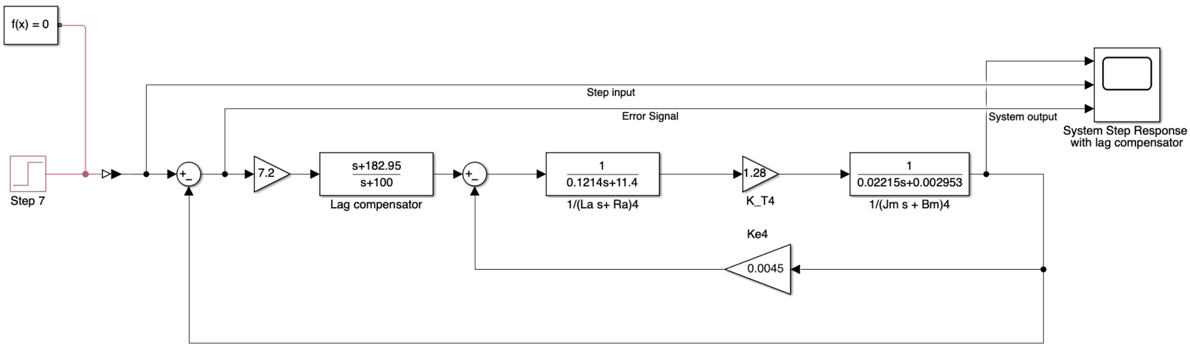

Figure 26: System with Lag compensator.

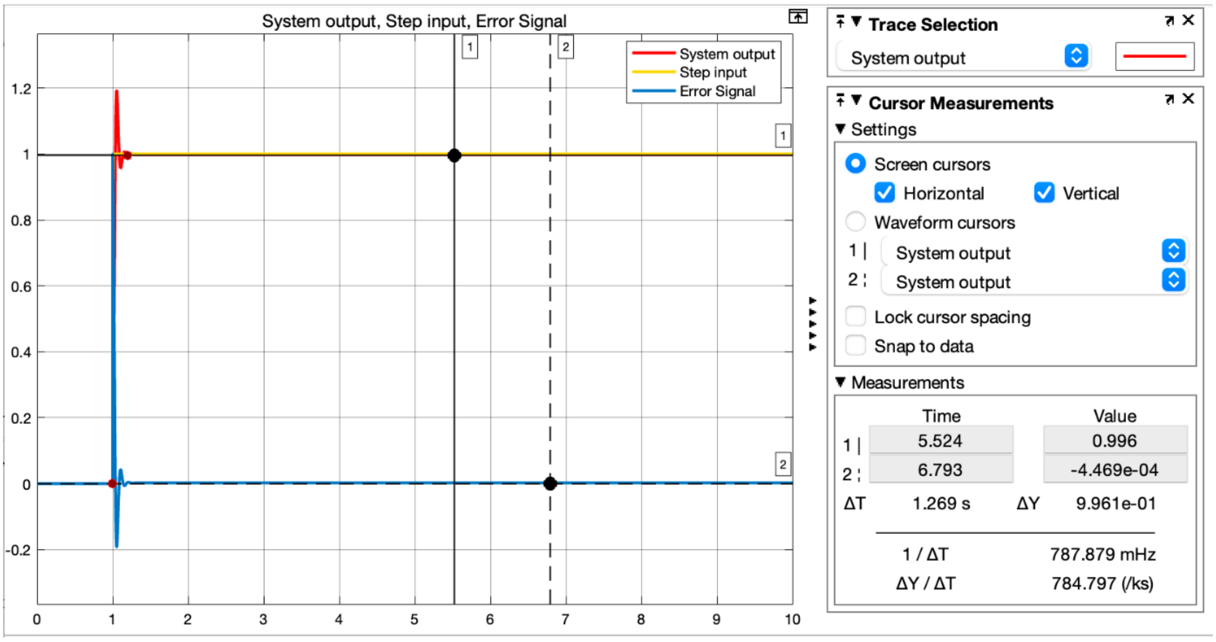

Figure 27: Unit Response of the System with Lag compensator.

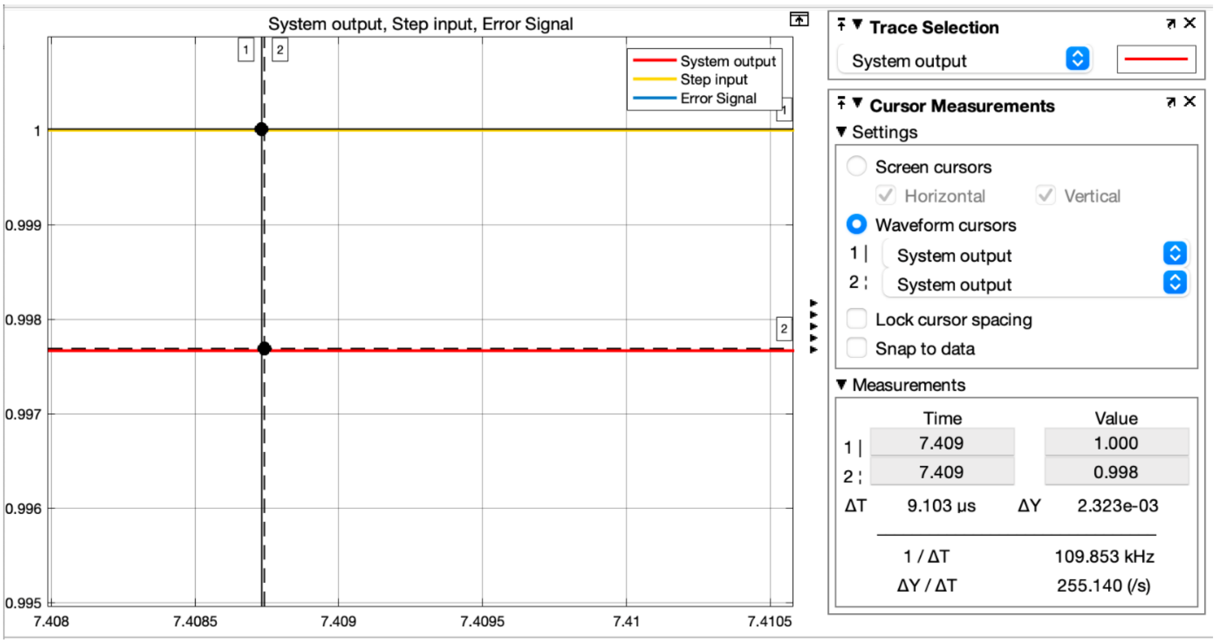

Figure 28: Steady State Error Unit Response of the System with Lag compensator.

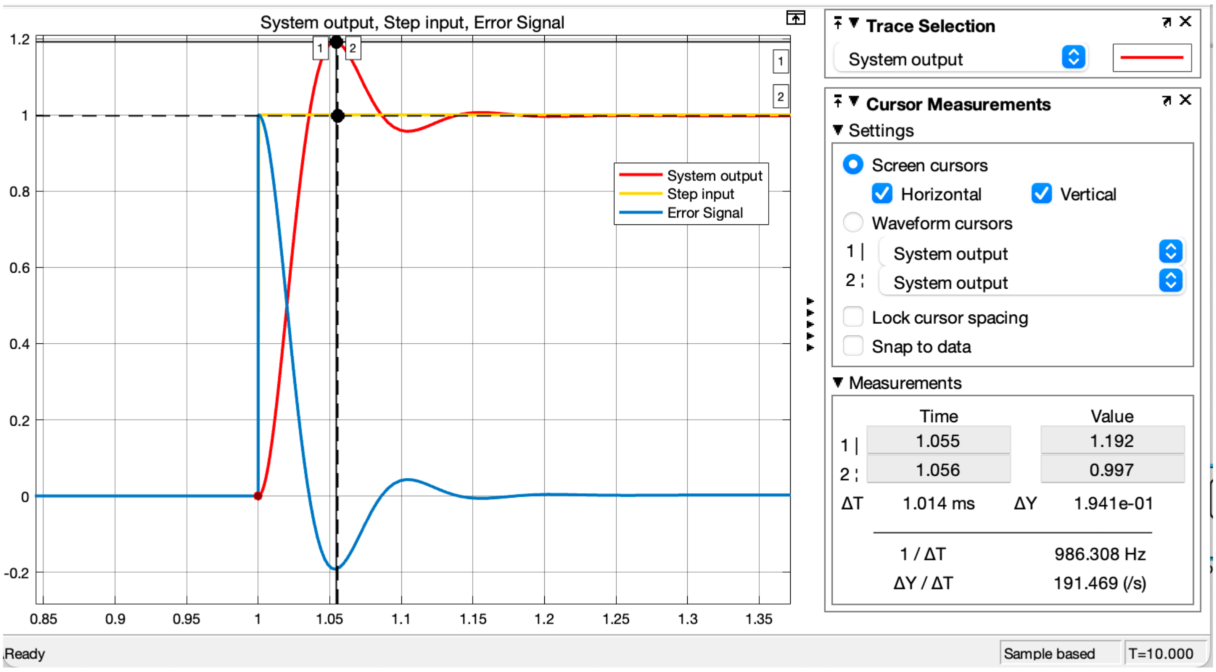

Figure 29: Unit Response of the System with Lag compensator max overshoot.

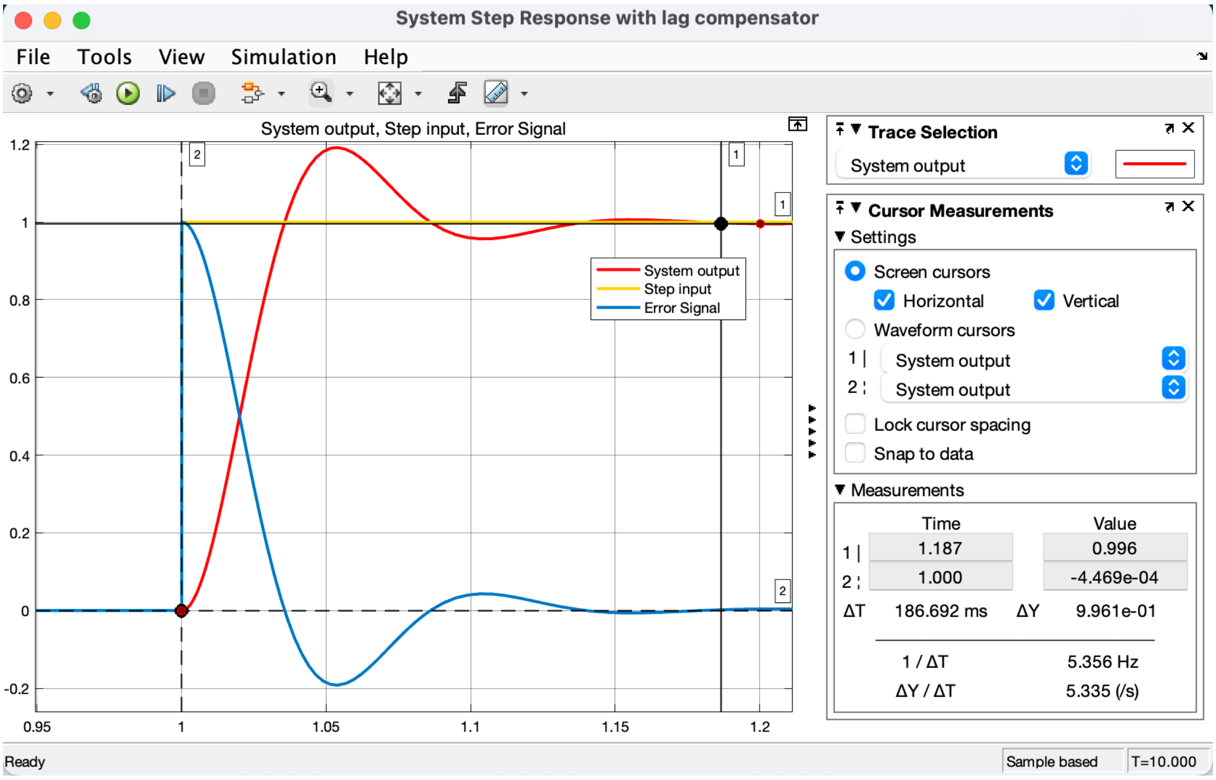

Figure 30: Unit Response of the System with Lag compensator settling time.

The Lag compensator for the system is designed and the root locus of the system is plotted as shown in Figure 25 and Figure 26.
As shown in Figure 26,Figure 27,Figure 28 and Figure 29, the designed lag compensator achieved the maximum overshoot of 19% and a settling time of 0.187 seconds. The compensator was configured with a zero at z = -182.9482, a pole at p = −100, and the

system gain K = 7.2 was adjusted to meet these performance criteria. This design successfully met the specified performance targets.

8.3 Lead Lag Compensator

Lead-lag compensator was designed using root locus plot.

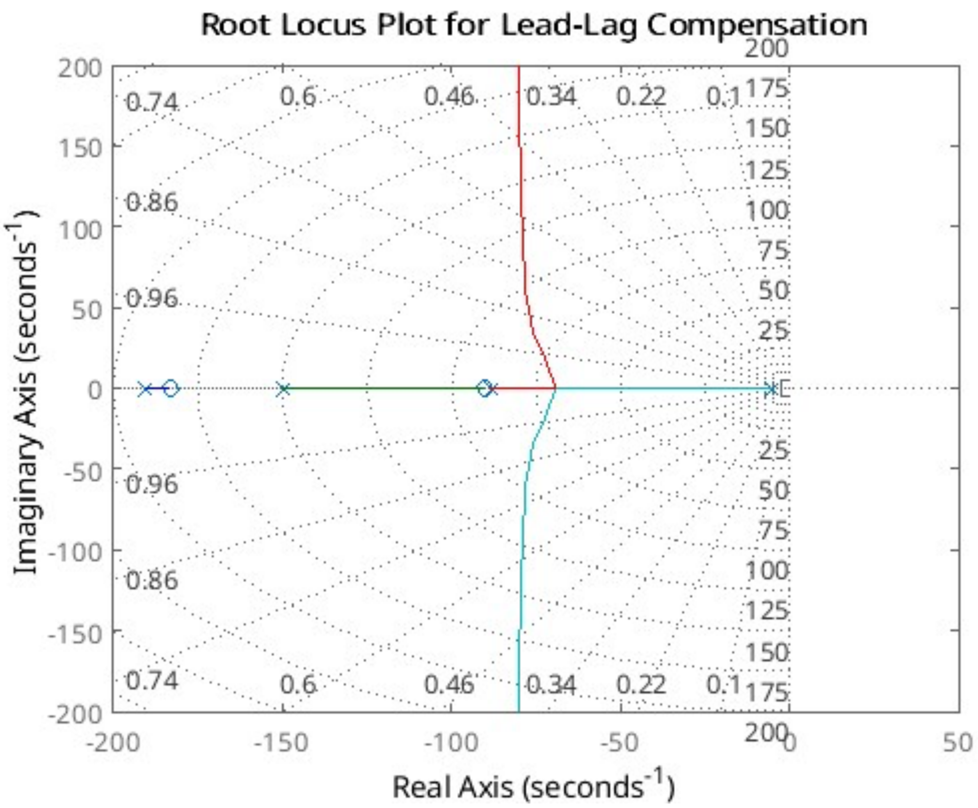

Figure 31: Root locus Plot for System with Lead and Lag compensator.

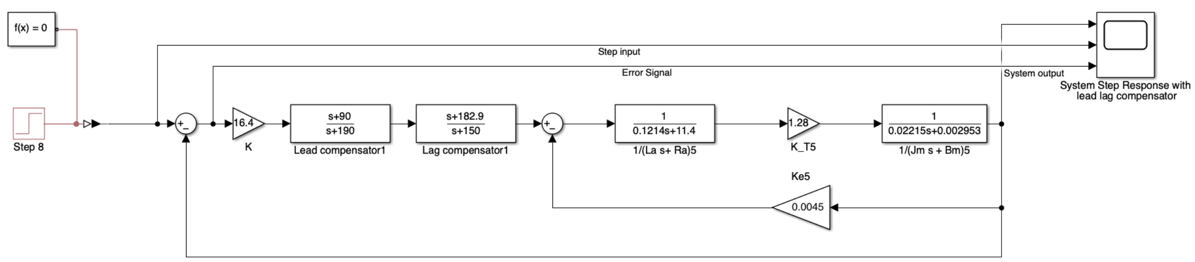

Figure 32: System with Lead and Lag compensator.

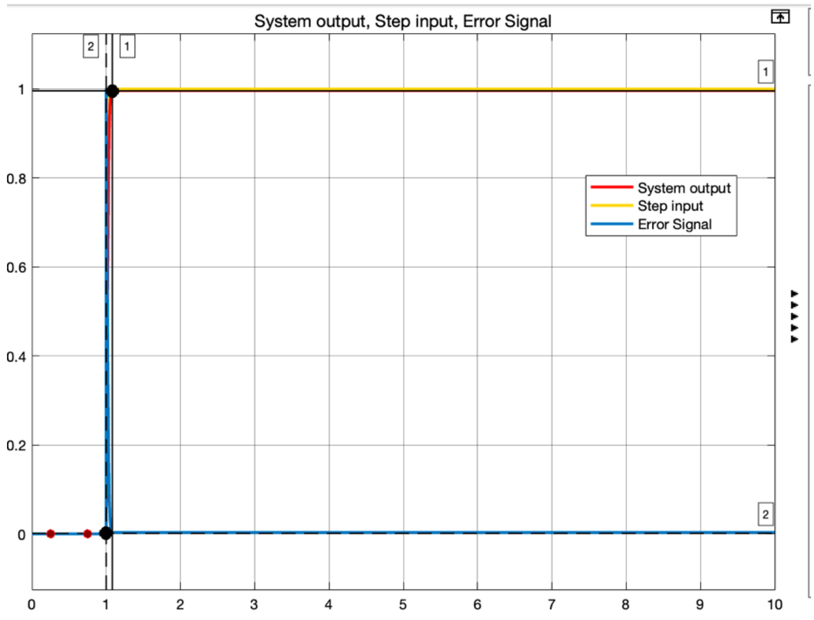

Figure 33: Unit Response of the System with Lead and Lag compensator.

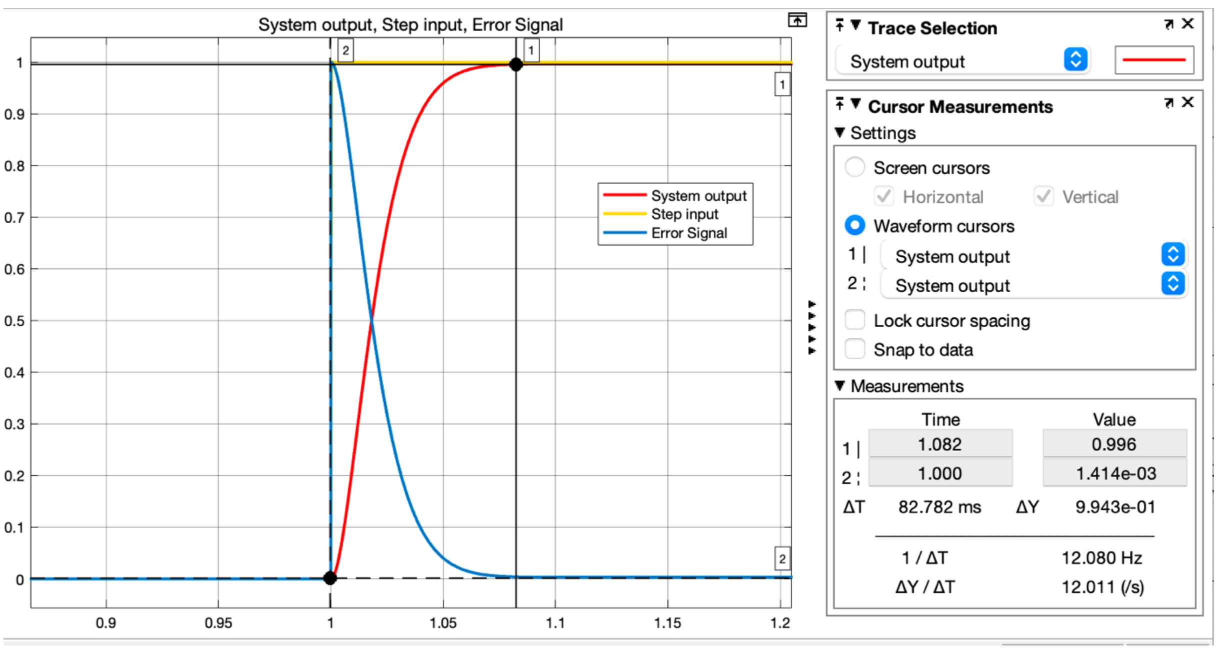

Figure 34: Unit Response of the System with Lead and Lag compensator.

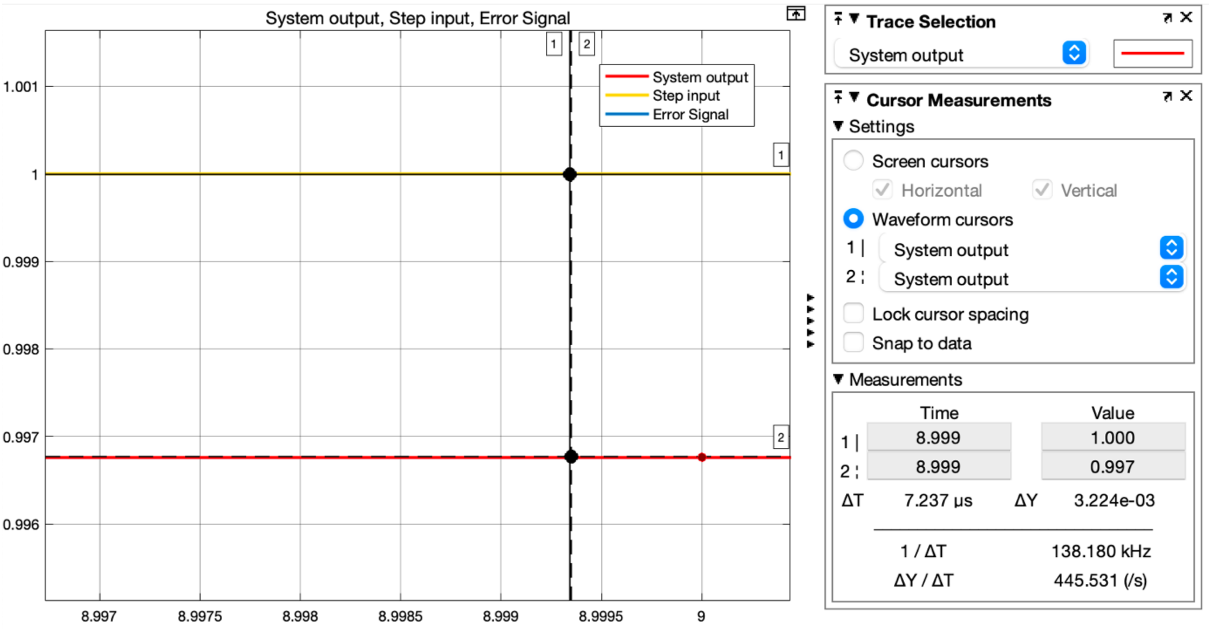

Figure 35: Unit Response of the System with Lead and Lag compensator Steady State Error

The lead lag compensator is designed and the root locus of the system with lead lag compensator is plotted as show in Figure 31 and Figure 32.
The designed lead-lag compensator achieved a settling time of 0.082 seconds and a steady-state error of 0.003 as shown in Figure 33, Figure 34 and Figure 35. The compensator was configured with zeros at z_0 = -90 and z_1 = -182.9482, poles at p_0 = -190 and p_1 = -150, and the system gain K = 16.94 was adjusted to meet the performance criteria. This design successfully met the specified performance targets.

9 Conclusion

In this project, we have thoroughly explored the control of DC motor speed, focusing on the development and analysis of a Linear Time-Invariant (LTI) mathematical model. Through the completion of various tasks, including the determination of the transfer function using Laplace transforms, obtaining the state space representation, and evaluating the transient response, we have gained a deep understanding of the system dynamics.

Our analysis included performing stability checks to ensure the system meets absolute stability criteria and deciding the steady-state responses and errors for different standard input signals. The application of Proportional (P), Proportional-Integral (PI), and Proportional-Integral-Derivative (PID) controllers has allowed us to minimize errors and enhance the system's stability and response quality.

The root locus plots provided insights into the system's behavior and helped in fine-tuning the controllers for better performance. Additionally, the exploration of lead, lag, and lead-lag compensators proved their effectiveness in optimizing the output and achieving the desired control objectives.

Overall, this project has successfully proved the fundamental principles and advanced techniques involved in DC motor speed control.

10 References

[1] M. a. S. W. Gubara, "Comparative study on the speed of DC motor using PID and FLC," 2016.

[2] V. K. K. P. S. R. a. P. M. V. Vishal, "Comparative study of some optimization techniques applied to DC motor control," 2014.

[3] M. Y. a. Z. Saleem, "Simulink model of the PID controller," researchgate.net, 2021.

[4] W. N.S. Nise, Control Systems Engineering, 7th ed., 2015.

[5] D. Spezia, "ECE 438A class notes," Soythern Illinois University.

[6] MathWorks, "Example: Modeling a DC Motor," [Online]. Available: https://www.mathworks.com/help/sps/ug/example-modeling-a-dc-motor.html.

[7] "DC Motor Speed: System Modeling," University of Michigan, [Online]. Available: https://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed&section=SystemModeling.

[8] Z. Haishui, W. Dahu, Z. Tong and H. Keming, "Desing on a DC Motor Speed Control," 2010. [Online]. Available: https://ieeexplore.ieee.org/document/5522813.