EE4150 : Control Systems Laboratory

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Post Lab Report

Experiment 4

PID tuning and state feedback controller design for a DC motor

Nakul C	**122101024**
Rittika Ghosh	**122101047**

Problem 1

1. **Design a PID controller for the obtained model with respect to the step input to satisfy the following specifications:**
   (a) **Settling time less than 0.3 seconds, curve within 5% of the final value.**
   (b) **Overshoot less than 2%.**
   (c) **Steady-state error less than 1%.**
2. **Study and calculate the effect of how the controller gains K_p, K_i, and K_d affect the time response parameters.**
3. **Plot the step response of the closed-loop system with the tuned PID controller and check whether the design criteria are met or not. And find out the following.**
   (a) **Settling time**
   (b) **% Overshoot**
   (c) **Steady-state error**

Transfer Function Modelling

A DC motor is a common actuator in control systems, providing rotary motion (e.g., robotic arms) or, when coupled with wheels or cables, translational motion (e.g., conveyor belts). Controlling the motor speed is essential for practical applications.

Table 1: Parameter description and values

Parameter	Description	Value
J	Rotor moment of inertia	0.01 kg·m^2
b	Motor viscous friction constant	0.1 N·m·s
K_e	Back emf constant	0.01 V/rad·s
K_t	Motor torque constant	0.01 N·m/Amp
R	Electric resistance	1 Ω
L	Electric inductance	0.5 H

In an armature-controlled DC motor, the speed is regulated by varying the input voltage. The transfer function of the DC motor, with angular speed as the output and the armature voltage as the input, is derived as follows.

$$\frac{\dot{\theta}(s)}{v(s)} = \frac{K}{(Js+b)(Ls+R)+K^2}$$

Substituting all these values, we get the transfer function of the motor as:

$$\frac{\dot{\theta}(s)}{v(s)} = \frac{K}{JLs^2+(JR+bL)s+bR+K^2}$$

Where $K_e = K_t = K$:

$$\frac{\dot{\theta}(s)}{v(s)} = \frac{0.01}{0.005s^2 + (0.06)s + 0.1001}$$

PID Controller

The PID controller comprises proportional, integral, and derivative components. It regulates the system's output by continuously adjusting the controller's output based on the difference between the reference set point and the actual measured process variable. The controller output can be expressed as:

$$C(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}$$

Where:

- $K_p \rightarrow$ Proportional gain
- $K_i \rightarrow$ Integral gain
- $K_d \rightarrow$ Derivative gain

The proportional component generates a control output proportional to the error signal, reducing steady-state error. The proportional gain K_p determines the system's sensitivity, and increasing it can make the controller more aggressive, potentially leading to oscillations if set too high.

The integral component accumulates the error over time, generating a control output to eliminate any steady-state error. The integral gain K_i affects how quickly the controller responds to the accumulated error; a higher K_i results in faster correction but may also cause overshoot and instability if too high.

The derivative component predicts future error by measuring the rate of change of the error over time, helping to reduce overshoot and dampen oscillations. Increasing the derivative gain K_d can enhance damping but may also increase noise amplification if set too high.

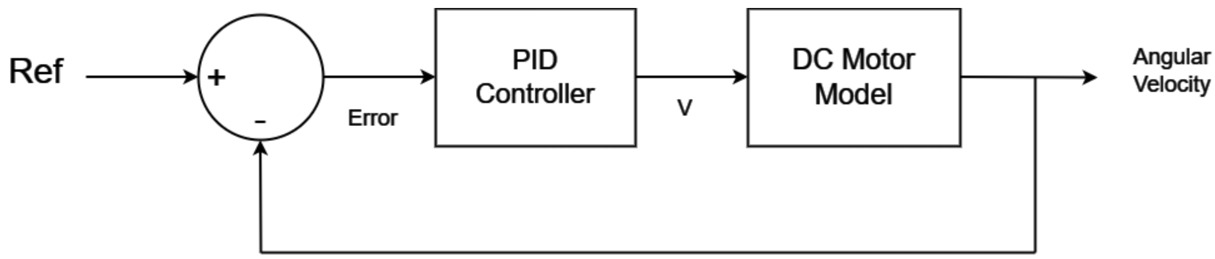

Figure 1: Block diagram of DC motor model with controller

Tuning the PID controller gains using Trial and error method

To achieve an effective control system, we begin by setting the integral gain (K_i) and derivative gain (K_d) to zero while tuning the proportional gain (K_p). This initial step aims to establish a balance between overshoot and steady-state error. Once we determine a suitable K_p, we proceed to adjust K_i until we meet the specified steady-state error requirements. With both

K_p and K_i established, the final tuning step involves adjusting K_d to ensure that the overshoot remains within acceptable limits. This systematic approach allows us to fine-tune the PID controller for optimal performance in our application.

Tuning Proportional Controller - Kp

Table 2: Performance Parameters for PID Controller

Kp	Kd	Ki	Rise Time	Settling Time	Percentage Overshoot	Steady State Error
60.0	0	0	0.1397	0.4473	15.7513	14.3464
80.0	0	0	0.1150	0.3909	20.7890	11.1357
100.0	0	0	0.0993	0.5100	24.9143	9.0258
120.0	0	0	0.0883	0.4832	28.3564	7.7214
140.0	0	0	0.0801	0.4555	31.3364	6.7440
160.0	0	0	0.0737	0.4306	33.8458	5.8752
180.0	0	0	0.0684	0.4088	36.1924	5.1939
200.0	0	0	0.0641	0.5043	38.2111	4.7414
220.0	0	0	0.0606	0.4915	40.0278	4.4086
240.0	0	0	0.0576	0.4766	41.6841	4.0710
260.0	0	0	0.0550	0.4620	43.1507	3.7091
280.0	0	0	0.0526	0.4481	44.4153	3.3877

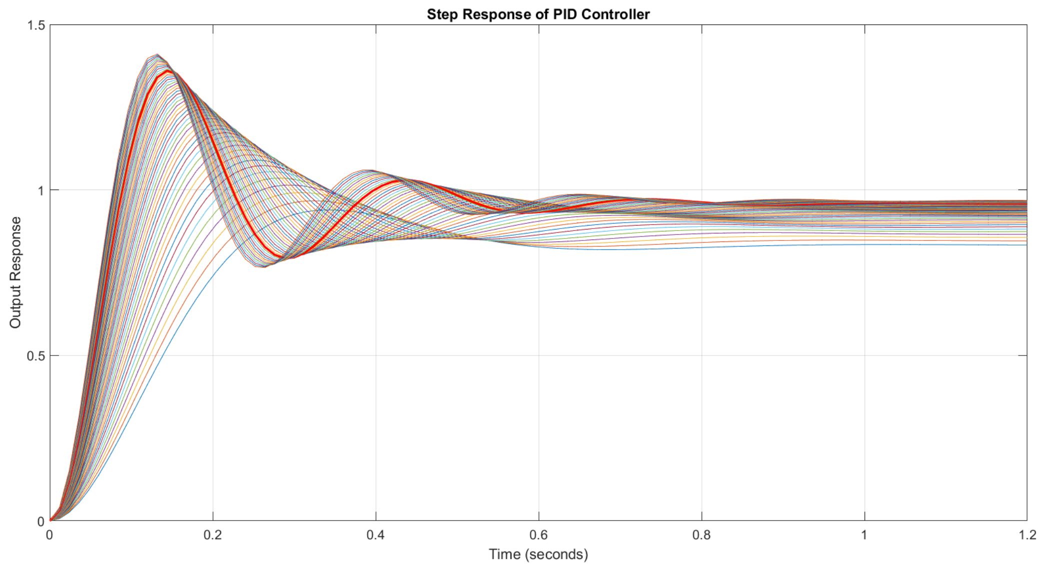

Figure 2: Step Response of PID Controller by Varying K_p and Keeping the Rest Constant

From the table and graph, it is evident that increasing the value of K_p leads to a decrease in steady-state error while simultaneously increasing the percentage overshoot. This reveals a trade-off between steady-state error and percentage overshoot. For K_p values less than 100, both the steady-state error and rise time are considerably large. Conversely, for K_p values above 100, the percentage overshoot becomes significantly larger. Therefore, after careful consideration of these factors, we are choosing $K_p = 100$ as our optimum value. This choice results in an overshoot of **41.6841%** and a steady-state error of **4.0710%**. This value is clearly marked in **RED** in Figure 2.

Tuning Integral Controller - Ki

Table 3: Performance Parameters for PID Controller

Kp	**Kd**	**Ki**	**Rise Time**	**Settling Time**	**Percentage Overshoot**	**Steady State Error**
240	200	0	0.0585	0.5949	41.0365	0.9663
240	210	0	0.0584	0.5918	41.2652	0.8977
240	220	0	0.0584	0.5892	41.4874	0.8342
240	230	0	0.0583	0.4762	41.7031	0.7754
240	240	0	0.0582	0.4776	41.9119	0.7212
240	250	0	0.0582	0.4788	42.1136	0.6711
240	260	0	0.0581	0.4800	42.4037	0.6251
240	270	0	0.0581	0.4811	42.6885	0.5827
240	280	0	0.0580	0.4822	42.9674	0.5439
240	290	0	0.0579	0.4832	43.2401	0.5083
240	300	0	0.0578	0.4842	43.5065	0.4758

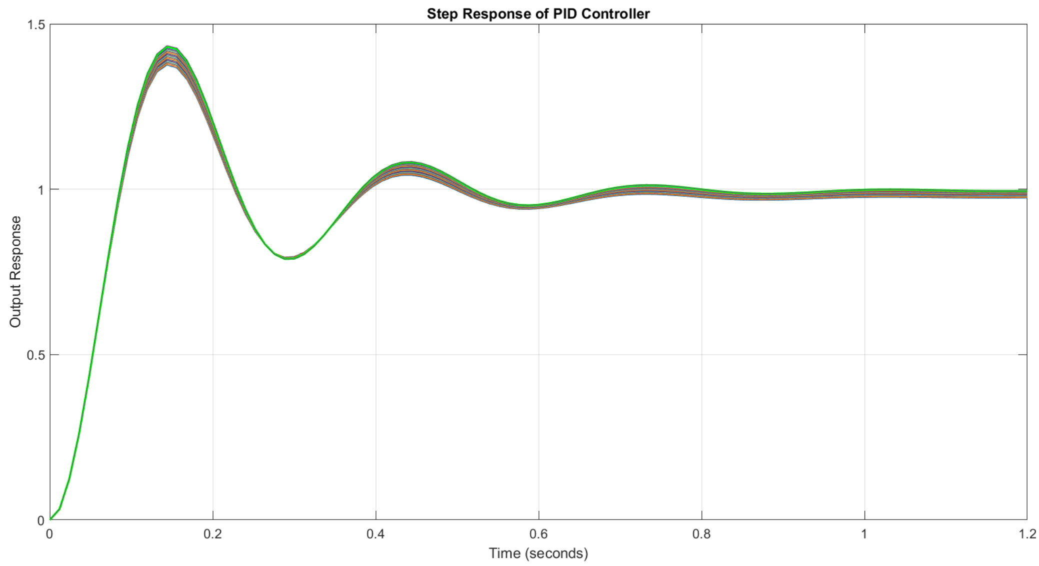

Figure 3: Step Response of PID Controller by Varying K_i and Keeping the Rest Constant

As we increase the value of $\mathbf{K_i}$, we see that the percentage overshoot rises, which means the system responds faster but may also become less stable. On the other hand, the steady-state error and rise time both decrease, indicating that the system reaches its desired output more quickly and accurately. The settling time also decreases slightly, suggesting an overall improvement in performance. From our analysis, we found a range of values for $\mathbf{K_i}$ between **200** and **295**. After evaluating these options, we chose $\mathbf{K_i} = \mathbf{290}$ as our final selection, resulting in a steady-state error of **0.5083**%. This choice effectively meets our performance criteria. By selecting this value, we can maintain an acceptable level of overshoot while ensuring the system settles accurately at the desired output. The chosen value of K_i is highlighted in **GREEN** in Figure 3, illustrating its role in achieving our control goals.

Tuning Differential Controller - Kd

Table 4: Performance Parameters for PID Controller

Kp	Kd	Ki	Rise Time	Settling Time	Percentage Overshoot	Steady State Error
240	290	15	0.0594	0.0754	1.973	0.3269
240	290	20	0.0555	0.0766	0	0.3118
240	290	25	0.0510	0.0808	0	0.2937
240	290	30	0.0462	0.0899	0	0.2739
240	290	35	0.0415	0.1045	0	0.2556
240	290	40	0.0370	0.1179	0	0.2427
240	290	45	0.0330	0.1265	0	0.2386
240	290	50	0.0295	0.1306	0	0.2453
240	290	55	0.0265	0.1305	0	0.2635
240	290	60	0.0240	0.1266	0	0.2928
240	290	65	0.0218	0.1187	0	0.3322
240	290	70	0.0200	0.1066	0	0.3801
240	290	75	0.0184	0.0895	0	0.4352

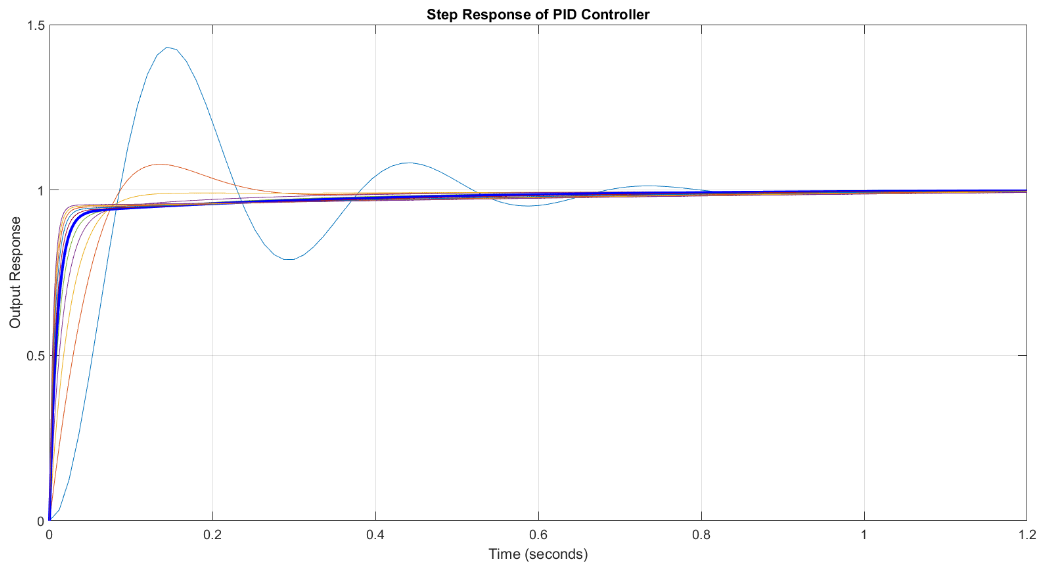

Figure 4: Step Response of PID Controller by Varying K_d and Keeping the Rest Constant

As we increase the value of $\mathbf{K_d}$, the rise time tends to increase, indicating a slower response to changes in the system. Conversely, both the percentage overshoot and settling time exhibit a decrease, suggesting that the system becomes more stable and is better at maintaining its desired output without significant fluctuations. After analyzing the range of values for $\mathbf{K_d}$ from **0** to **250**, we have decided to select $\mathbf{K_d = 50}$ as our final choice. This selection results in a maximum overshoot of **0**, effectively satisfying the overshoot requirement for our control system. This balance between responsiveness and stability is crucial for achieving optimal performance in our application. The chosen value of K_d is highlighted in **BLUE** in Figure 4.

Closed-loop System with the tuned PID controller

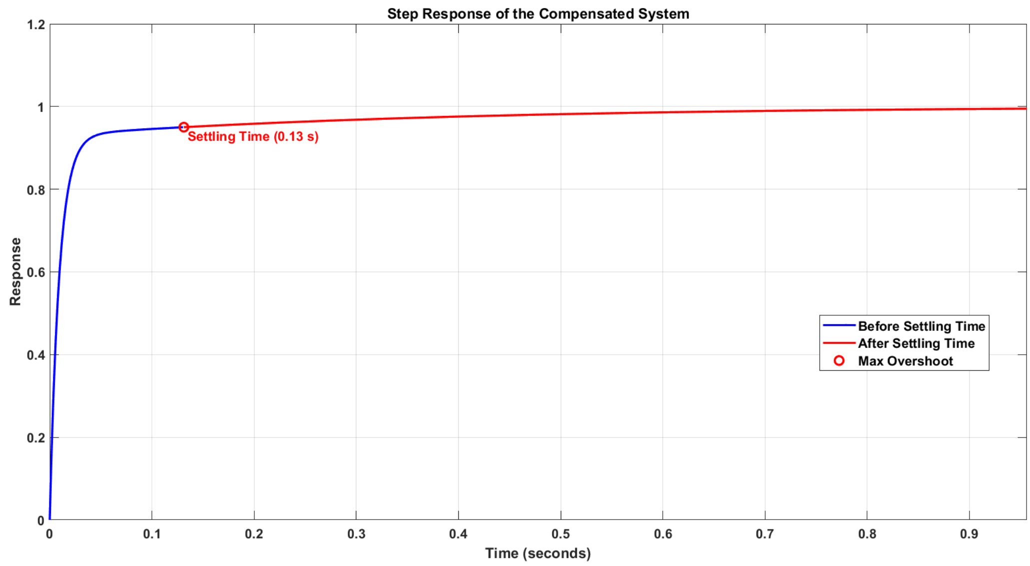

Figure 5: Step response of the closed-loop system with the tuned PID controller

The Tuned PID Parameters are

Parameter	K_p	K_i	K_d
Value	240	290	50

The system performance metrics are as follows: the **rise time** is **0.0295 s**, indicating a rapid response to input changes, and the **transient time** is **0.1306 s**, reflecting the time taken for the system to stabilize. The **settling time** is also **0.1306 s**, with a **settling minimum** of **0.9012** and a **settling maximum** of **0.9968**, demonstrating the range within which the output stabilizes after a disturbance. The system exhibits an **overshoot** of **0%** and an **undershoot** of **0%**, indicating it achieves the desired output without exceeding or falling short. Additionally, the **peak** value reached is **0.9968**, occurring at a **peak time** of **1.1140 s**, illustrating the system's efficiency in reaching near-target performance.

The settling time is less than **0.3 s**, the percentage overshoot is less than **2%**, and the steady-state error is less than **1%**. Therefore, the tuned system satisfies the required conditions.

Effects of Increasing Parameters

Parameter	**Rise Time**	**Settling Time**	**Percentage Overshoot**	**Steady State Error**
K_p	Decreases	Decreases	Increases	Decreases
K_i	Decreases	Decreases then increases	Increases	Decreases
K_d	Increases	Decreases	Decreases	Decreases

Problem 2

1. **Simulate the system using the `ode` function in MATLAB editor.**
2. **Analyze the controllability and observability of the given system manually and check whether the state feedback controller can be designed or not. Verify your results from MATLAB.**
3. **Find the eigenvalues of the system both manually and through MATLAB. Also comment on the stability of the system based on it.**
4. **Using an arbitrary initial condition, design a state feedback controller $u = -K_1x_1 - K_2x_2$. Make a proper choice of feedback gains K_1 and K_2 to asymptotically stabilize the system states to the origin i.e., $(x_1, x_2) = (0, 0)$.**
5. **Design a state feedback controller using the pole placement technique described in the page below (using method 2) to satisfy the following specifications**
   - **Settling time $t_s < 2$ seconds and percentage overshoot $M_p < 5\%$.**
   - **Steady state error $e_{ss} < 1\%$ for unit step reference of angular velocity (1 rad/s).**

State space Modelling

From the working mechanism of the DC motor, the torque generated by a DC motor is directly proportional to the armature current and strength of the magnetic field. Assuming that the magnetic field remains constant, as a consequence, torque is directly proportional to the armature current.

$$T = K_t i$$

Also, back emf (e) is directly proportional to the angular velocity of the shaft.

$$e = K_e \dot{\theta}$$

$$\text{Torque constant} K_t = \text{back emf constant} K_e = K$$

$$K_t = K_e = K$$

On applying Newton's second law and KVL

$$J\ddot{\theta} + b\dot{\theta} = Ki$$

$$\frac{1}{L} \cdot \frac{di}{dt} + Ri = v - K\dot{\theta}$$

State variables are chosen as

$$x_1 = \omega = \dot{\theta}$$

$$x_2 = i$$

$$\text{input } u = v$$

$$\text{output } y = \omega = x_1$$

The state equation is derived as

$$\frac{dx_1}{dt} = -\frac{b}{J}x_1 + \frac{K}{J}x_2$$

$$\frac{dx_2}{dt} = -\frac{K}{L}x_1 - \frac{R}{L}x_2 + \frac{1}{L}v$$

$$y = \frac{d\theta}{dt} = \omega = x_1$$

From the equations mentioned previously, the state space model of a DC motor is obtained as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} u$$

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

Controller Designing using state feedback Pole Placement method

State feedback/pole placement technique, is used to place the closed-loop poles of a plant at pre fixed positions in the s-plane. In this technique, all the state variables are known to the controller. Reference pre-filter Kr is designed to get rid of the steady state offset. Kr is taken as the reciprocal of the DC gain of the system in a closed loop. In this experiment the input v is designed using state feedback method to stabilise the system.

ODE Simulation

The state-space model is solved utilizing the built-in ODE45 solver in MATLAB. A function is defined with specified initial conditions and a predetermined time duration. The initial condition is set to $(x_1, x_2) = (0, 1)$. The function outputs $\omega = \dot{\theta}(x_1)$ and the current x_2 as functions of time.

The input current remains constant over the specified time interval, as indicated by the initial current value of **1 A**. The angular velocity, denoted as $\omega(x_1)$, increases from its initial state and eventually stabilizes. In the absence of input to the system $(u = 0)$, both ω and the current i decrease from their initial values to zero.

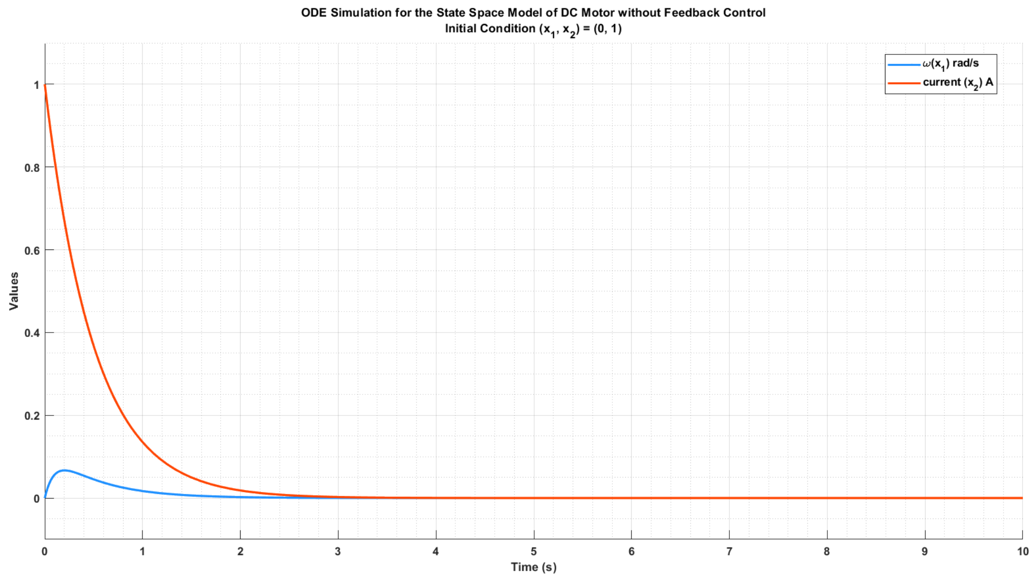

Figure 6: Resultant Plot of Omega (x1) ,Current (x2) vs time for u = 0 and initial condition (0,1)

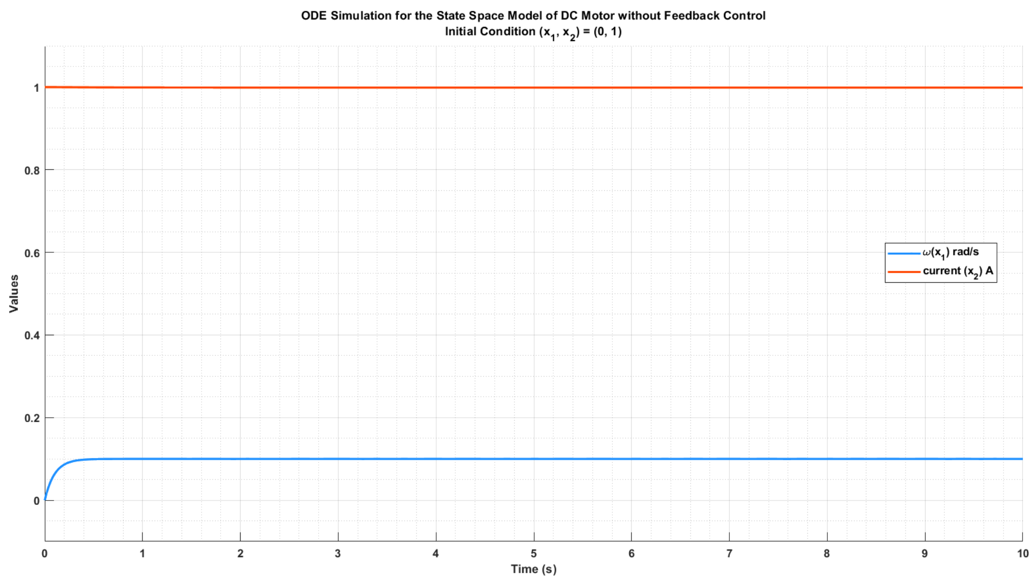

Figure 7: Resultant Plot of Omega (x1) ,Current (x2) vs time for u = 1 and initial condition (0,1)

Controlablity and Observablity

The analysis of the DC motor system's controllability and observability is crucial for understanding its ability to be controlled and monitored effectively. These properties are assessed using the built-in MATLAB functions `ctrb()` for controllability and `obsv()` for observability.

Controllability Matrix

The controllability matrix $\mathcal{C}$ is constructed from the state-space representation of the system. It is defined as follows:

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix}$$

For the given DC motor system, the controllability matrix as obtained after using the in-build `ctrb()` is :

$$\mathcal{C} = \begin{bmatrix} 0 & 2 \\ 2 & -4 \end{bmatrix}$$

Observability Matrix

Similarly, the observability matrix $\mathcal{O}$ is derived from the system's output equation and is defined as:

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

In our case, the observability matrix as obtained after using the in-build `obsv()` is:

$$\mathcal{O} = \begin{bmatrix} 1 & 0 \\ -10 & 1 \end{bmatrix}$$

Controllability

A system is controllable if it can be directed from any initial state to any desired final state within a finite interval of time. To check controllability, we need to construct the controllability matrix $\mathcal{C} = [B\, AB]$, where B is the input matrix and A is the state matrix.

Given the matrices for the DC motor system:

$$A = \begin{bmatrix} -10.0000 & 1.0000 \\ -0.0200 & -2.0000 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

We can compute AB:

$$AB = A \cdot B = \begin{bmatrix} -10.0000 & 1.0000 \\ -0.0200 & -2.0000 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ -4 \end{bmatrix}$$

Thus, the controllability matrix becomes:

$$\mathcal{C} = [B\ AB] = \begin{bmatrix} 0 & 2 \\ 2 & -4 \end{bmatrix}$$

Observability

A system is observable if it is possible to determine the values of all of its internal states from the measurements of its outputs. To check observability, we require the observability matrix $\mathcal{O} = \begin{bmatrix} C \\ CA \end{bmatrix}$, where C is the output matrix and A is the state matrix.

Given the output matrix:

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

We can compute CA:

$$CA = C \cdot A = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} -10.0000 & 1.0000 \\ -0.0200 & -2.0000 \end{bmatrix} = \begin{bmatrix} -10 & 1 \end{bmatrix}$$

Thus, the observability matrix becomes:

$$\mathcal{O} = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -10 & 1 \end{bmatrix}$$

Rank and System Properties

The rank of both matrices is 2, indicating that they are full rank. Specifically, a rank equal to the number of states in the system (which is 2 in this case) implies that the system is both controllable and observable.

- **Controllability** ensures that it is possible to drive the state of the system to any desired state within a finite time using suitable control inputs.
- **Observability** ensures that it is possible to determine the state of the system from the output over time.

Thus, the findings confirm that the DC motor system can be effectively controlled and monitored.

Eigen Values

Eigenvalues are scalars associated with a square matrix A that indicate the factor by which an eigenvector v is scaled during the linear transformation $Av = \lambda v$. They are obtained by solving the characteristic equation $\det(A - \lambda I) = 0$, where I is the identity matrix. In control systems, negative eigenvalues signify stability, meaning the system will return to equilibrium, while positive eigenvalues indicate instability. Thus, eigenvalues are essential for understanding the behavior of linear systems.

Simulation

The eigenvalues of the DC motor system are obtained using the built-in function `eig()` in MATLAB. The eigenvalues obtained through the MATLAB simulation are **-9.9975** and **-2.0025**. Eigenvalues denote the system's poles. Since we have obtained negative eigenvalues, this indicates that the system's poles are located in the left half-plane. Thus, we can conclude that the system is stable.

Calculation

Eigenvalues are found using the equation:

$$\det(A - \lambda I) = 0$$

Where

$$A - \lambda I = \begin{bmatrix} -10.0000 & 1.0000 \\ -0.0200 & -2.0000 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

This simplifies to:

$$\begin{bmatrix} -10.0000 - \lambda & 1.0000 \\ -0.0200 & -2.0000 - \lambda \end{bmatrix}$$

Calculating the determinant:

$$\det(A - \lambda I) = (-10.0000 - \lambda)(-2.0000 - \lambda) + 0.02$$

On simplifying the determinant and equating it to 0, we have:

$$\lambda^2 + 12\lambda + 20.02 = 0$$

Solving this quadratic equation yields the eigenvalues $\lambda = -9.997$ and $\lambda = -2.002$. Hence, the manual calculation matches the MATLAB result.

Again, eigenvalues denote the system's poles, and since we have obtained negative eigenvalues, which indicates that the system's poles are located in the left half-plane, we can conclude that the system is stable.

Designing State Feedback Controller

By setting the initial condition to $(1, 1)$, we indicate that the system starts with an angular velocity of 1 rad/s and an initial current of 1 A. Initially, we will fix $K_2 = 0$ and vary K_1 to find its optimal value. After determining the appropriate K_1, we will then set K_1 to zero and select a suitable value for K_2.

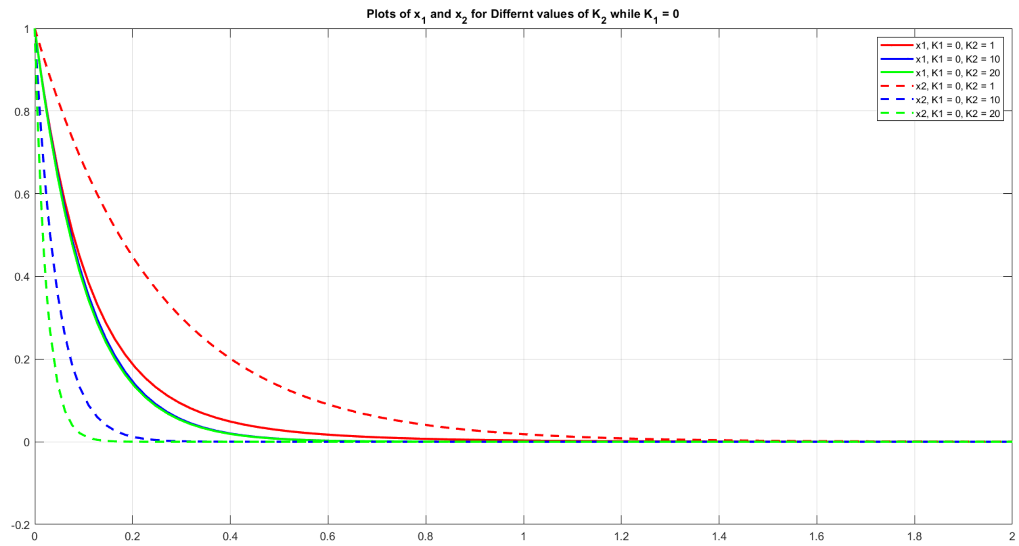

Figure 8: Varying K1 while keeping K2 = 0. x1 is plotted using solid line while x2 using dashed lines.

By setting $K_1 = 0$ and varying K_2, we observe that for values of both K_1 and K_2 greater than 10, the states x_1 and x_2 asymptotically converge to (0, 0) but initially move towards negative values. This behavior is not desirable; therefore, we will select values for K_1 and K_2 that are less than 10.

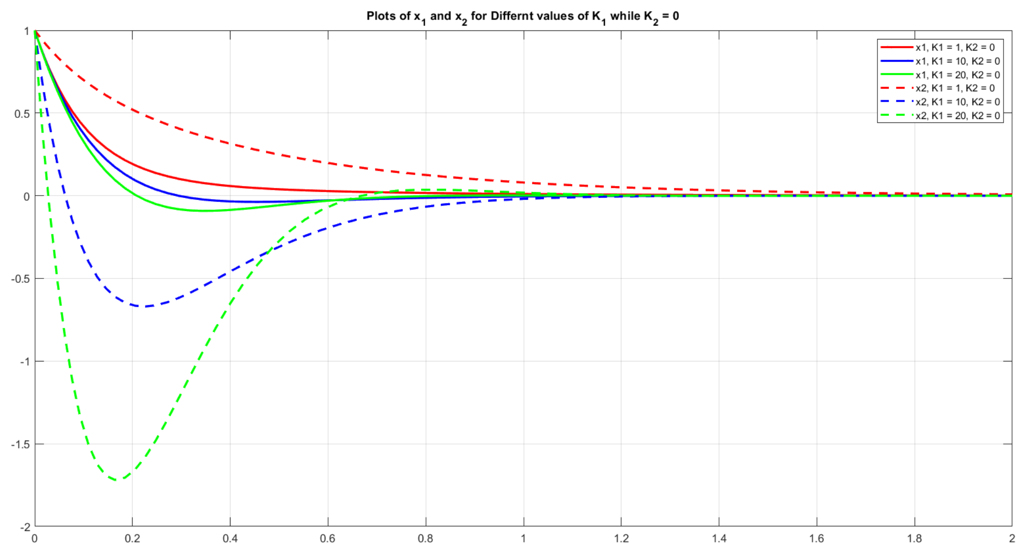

Figure 9: Varying K2 while keeping K1 = 0. x1 is plotted using solid line while x2 using dashed lines.

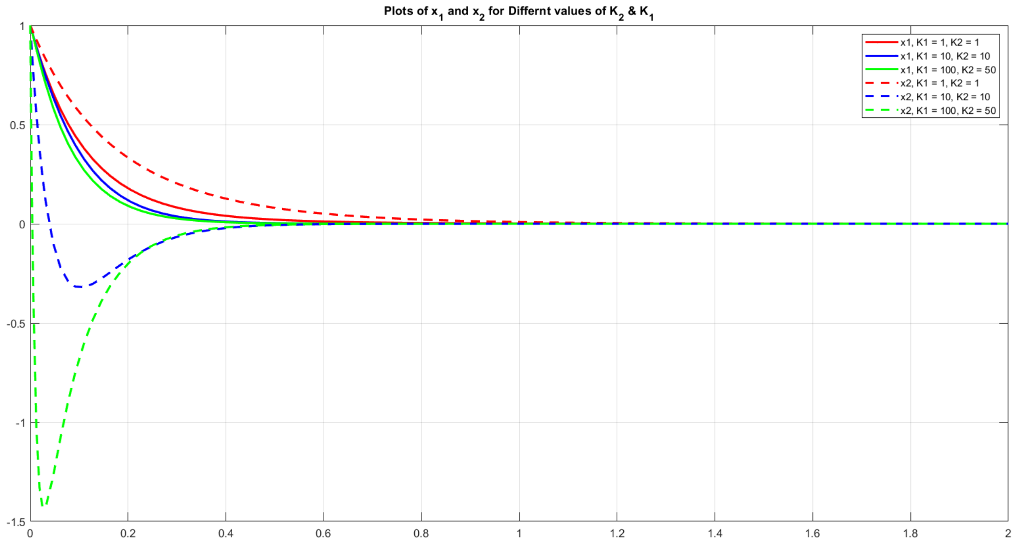

Figure 10: x1 and x2 plots for different values of K1 and K2

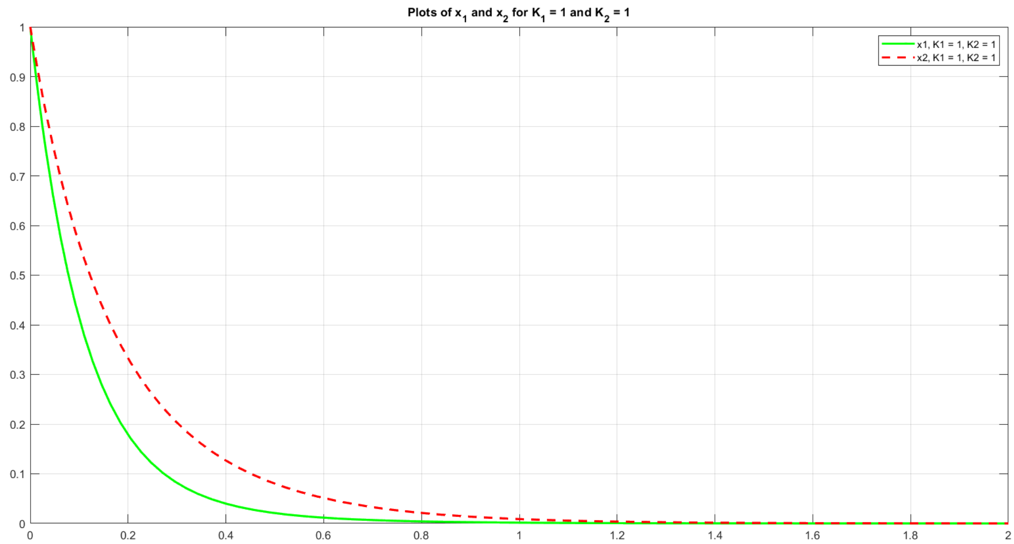

Figure 11: Choosing K1 = K2 = 1. The plots of x1 and x2 are asymptotically stabilizing at (0,0).

Inferences

- The constant coefficients of the control input u are selected as $K_1 = K_2 = 1$.
- Regardless of the values chosen for K_1 and K_2, the system will asymptotically stabilize at $(x_1, x_2) = (0, 0)$.
- Choosing values of K_1 and K_2 greater than 10 will lead to negative values for x_1 and x_2, which is undesirable.

Pole Placement

Design Requirements

1. Settling time $t_s \leq 2$ seconds
2. Percentage Overshoot, $M_p \leq 5\%$
3. Steady state error, $e_{ss} \leq 1\%$ for a unit step reference of angular velocity (1 rad/s).
4. Control law $u = -Kx + K_r r$, where $K = [K_1, K_2]$ and K_r is the reciprocal of the DC gain of the closed-loop system.

Designing the State Feedback Controller

We have the A, B, C, D matrices as follows:

$$A = \begin{bmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

The system is not in phase variable form.

Checking the controllability:

$$e = \begin{bmatrix} 0 & 2 \\ 2 & -4 \end{bmatrix}$$

The rank of the matrix is 2, hence the system is controllable.

Forming the Characteristic Equation for A

The characteristic equation has the form:

$$|SI - A| = s^n + a_1 s^{n-1} + a_2 s^{n-2} + \ldots + a_{n-1}s + a_n$$

From the above equation, it can be seen that a_i's are the coefficients in the characteristic polynomial. The values of a_i are later used to form the W matrix, which is required to create the transformation matrix T.

$$SI - A = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} -10.0000 & 1.0000 \\ -0.0200 & -2.0000 \end{bmatrix}$$

$$SI - A = \begin{bmatrix} s + 10.0000 & -1.0000 \\ 0.0200 & s + 2.0000 \end{bmatrix}$$

$$|sI - A| = s^2 + 12s + 20.0200$$

Therefore,

$$a_1 = 12$$

$$a_2 = 20.02$$

Forming the W matrix:

$$W = \begin{bmatrix} a_1 & 1 \\ 1 & 0 \end{bmatrix}$$

$$W = \begin{bmatrix} 12 & 1 \\ 1 & 0 \end{bmatrix}$$

$$T = eW$$

$$T = \begin{bmatrix} 0 & 2 \\ 2 & -4 \end{bmatrix} \begin{bmatrix} 12 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 20 & 2 \end{bmatrix}$$

Desired Characteristic Equations

Considering the borderline requirements, that is

$$T_s = 2s$$

and

$$M_p = 0.05$$

We can find the desired values of ζ and ω_n:

$$\zeta = \sqrt{\frac{\log(M_p)^2}{\pi^2 + \log(M_p)^2}}$$

$$\zeta = 0.6901$$

$$\omega_n = \frac{4}{\zeta T_s}$$

$$\omega_n = 2.8981\,\mathrm{rad/s}$$

We will select slightly larger values for ω_n as we do not need to lie beyond or exactly on the borderline conditions. Therefore,

$$\omega_n = 5\,\mathrm{rad/s}$$

Once we have obtained the values of ζ and ω_n, we can find the dominant poles, which will be the desired pole locations:

$$\mu_1 = -\zeta\omega_n + j\omega_n\sqrt{1-\zeta^2}$$

$$\mu_1 = -3.4505 + j3.6185$$

$$\mu_2 = -\zeta\omega_n - j\omega_n\sqrt{1-\zeta^2}$$

$$\mu_2 = -3.4505 - j3.6185$$

Therefore, the characteristic equation will have the form:

$$(s-\mu_1)(s-\mu_2) = s^2 + \alpha_1 s + \alpha_2$$

$$(s-\mu_1)(s-\mu_2) = (s + 3.4505 - j3.6185)(s + -3.4505 + j3.6185) = s^2 + 4s + 8.3991$$

$$\alpha_1 = 4$$

$$\alpha_2 = 8.3991$$

Desired K Matrix

$$K = \begin{pmatrix}(\alpha_2 - a_2) & (\alpha_1 - a_1)\end{pmatrix} T^{-1}$$

$$K = \begin{pmatrix}34.1895 & -4.0000\end{pmatrix}$$

Desired Control Law u

K_r is the reciprocal of the DC gain of the final closed-loop system. The transfer of the closed-loop system can be obtained as:

$$T(s) = C[sI - A_{CL}]B$$

where

$$A_{CL} = A - BK$$

$$T(s) = \frac{2}{s^2 + 4s + 8.399}$$

DC gain:

$$DCgain = 0.2381$$

$$K_r = 4.1995$$

Thus, the final control law u is:

$$u = -Kx + K_r r$$

$$u = -K_1 x_1 - K_2 x_2 + K_r$$

where r is the reference signal, kept as 1, corresponding to $\omega = 1\,\text{rad/s}$ (unit step input).

$$u = 34.1895x_1 - 4x_2 + 4.1995$$

By passing the values of K_1, K_2, and K_r to the ODE function built in the first problem and using the ODE45 solver, we can find the output ($x_1 = \omega$) of the final system with a feedback controller.

Results

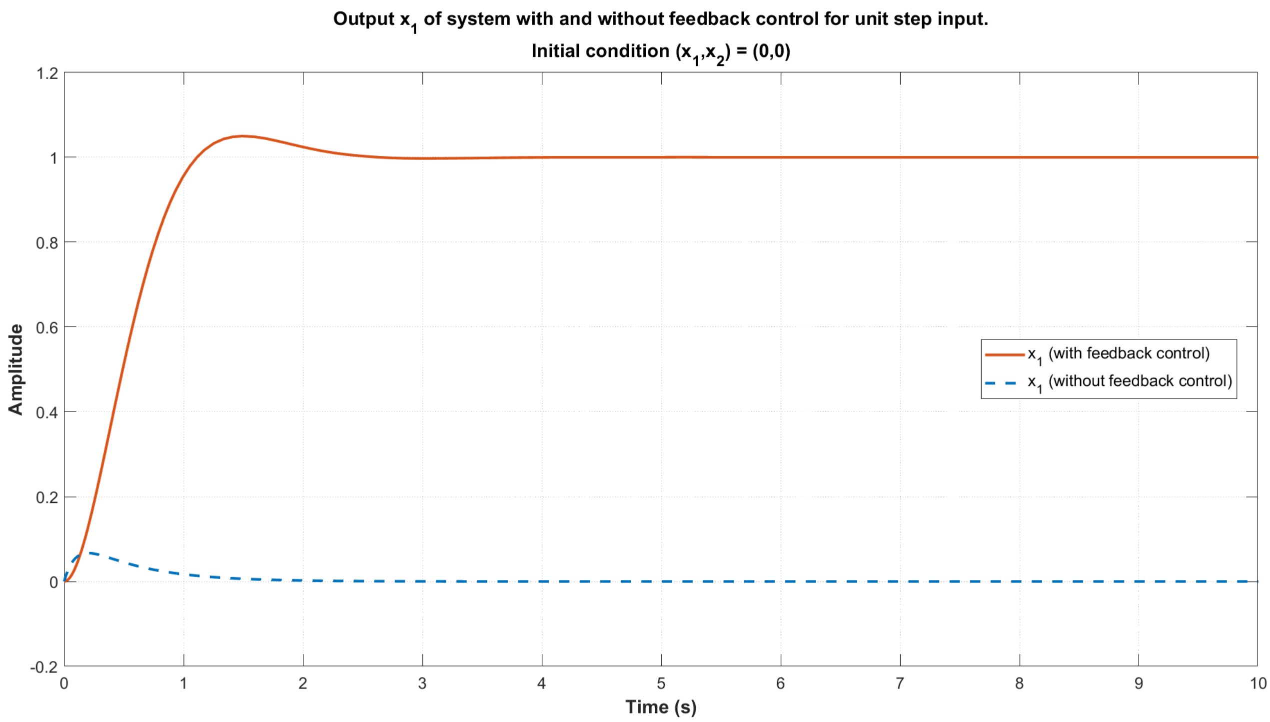

Figure 12: Plot of x1 for a system with and without state feedback controller.

The system, when subjected to a unit step input and initialized at $(0, 0)$, demonstrates the following steady-state behavior:

- Without feedback control, the steady-state value of the output x_1 is approximately 0.1, leading to a steady-state error (SSE) of 90.0084%.
- With state feedback control applied, for the same initial conditions, the steady-state value of the output x_1 is around 1. The SSE is reduced to 0.00042%, significantly lower than the desired specification of 1%.

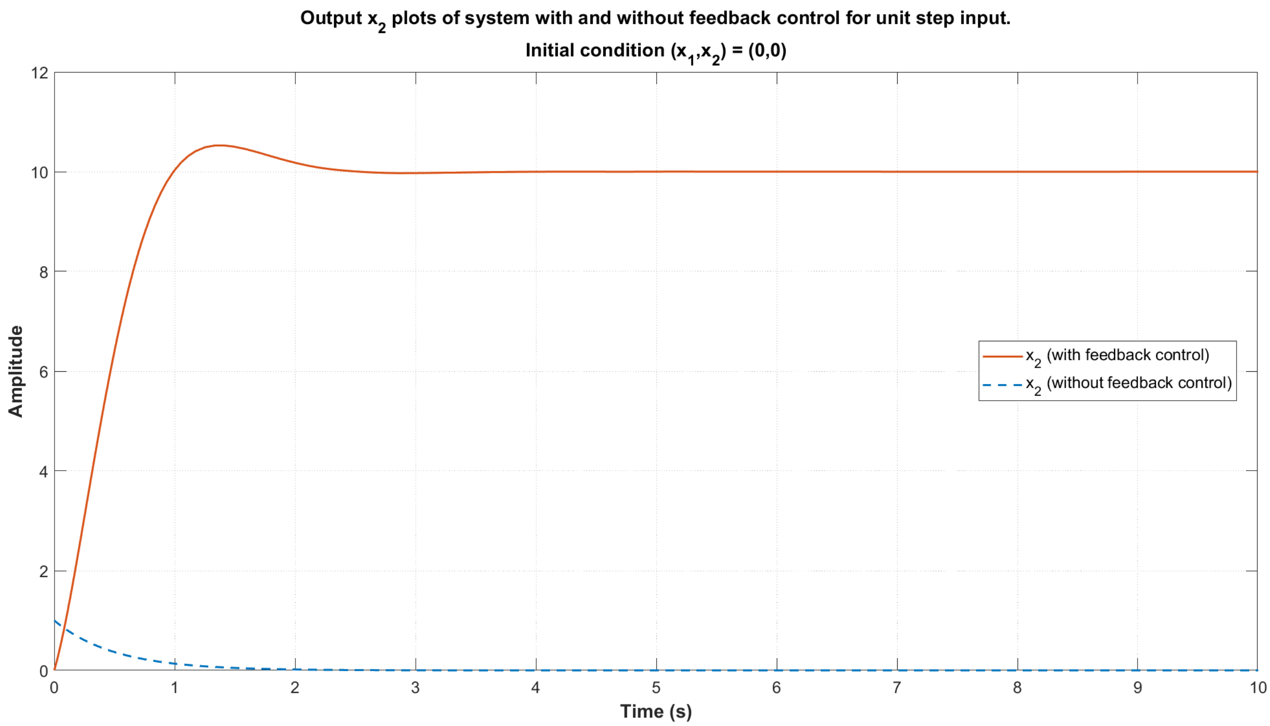

Figure 13: Plot of x2 for a system with and without state feedback controller.

Inferences

- The pole placement method is a widely used technique for designing feedback control systems in the context of state-space representations. It allows for the placement of the system's closed-loop poles at specific desired locations, thus giving us control over the system's closed-loop characteristics, such as eigenvalues, poles, and modes. This control helps achieve the required performance and stability properties.
- The transformation matrix T has the following key properties:
  - $T^{-1}AT = A_{CL}$ brings the system to its phase-variable form.
  - $T^{-1}B = B$
- The values of K_1 and K_2 for the control law u are obtained as 34.1895 and -4, respectively, using the pole placement method.
- The value of K_r, which is the reciprocal of the DC gain of the closed-loop system, is calculated to be 4.1995.
- The feedback-controlled system meets all the design requirements:
  1. The settling time of the system with feedback control is 2 seconds.

2. The percentage overshoot of the closed-loop system is $M_p = 4.9987\%$.
3. The steady-state error of the closed-loop system is $\text{sserror} = 0.00042\%$.