AJMS
Asian Journal of Mathematical Sciences

RESEARCH ARTICLE

Analysis and simulation of Accuracy and Stability for Closed Loop Control Systems

***Musa Adam Abdullah, Saaleh mussa Abdallah Hamballah[1], Subhi Abdalazim Aljily Osman[2].**

**Department of Mathematics, College of Computer Science and Information Technology, University of the Holy Quran and Tassel of Science, Sudan*

[1]Department of Mathematics, Faculty of Education, University of the Holy Quran and Tassel of Science, Sudan

[2]Department of Mathematics, Faculty of Computer Science and Information Technology, University of ALBUTANA, Sudan

Corresponding Email: musa.adam125@gmail.com

Received: 12-04-2023; Revised: 28-04-2023; Accepted: 08-05-2023

ABSTRACT

The design and assessment of closed loop control systems must include accurate and stable accuracy analysis and simulation. An overview of the significance of closed loop control systems and its uses in several engineering disciplines is given in this paper. The paper then explores the different techniques used for evaluating the accuracy and stability of closed loop control systems, including steady-state error analysis, stability analysis, and simulation techniques. It also discusses the design and tuning of control algorithms to optimize the performance of the system, and identifies the different factors that can affect the accuracy and stability of closed loop control systems, such as sensor noise, model inaccuracies, and environmental disturbances.

Keyword: Simulation; Accuracy and Stability; Closed Loop; Control Systems

INTRODUCTION

A closed-loop system of control is one type of system that controls and regulates physical processes. Due to the presence of a feedback loop, the system is able to modify its output in response to feedback from its surroundings [1]. The elements that make up the input loop are a controller, a procedure, and a sensor. The controller is responsible for generating the control signal that is sent to the process. The process is the physical system that is being controlled, and the sensor is used to measure the output of the process. Using the feedback signal the sensor generates, the controller adjusts the control signal it gives to the process. A closed loop control system tries to keep the desired output constant despite interruptions or changes in the input [2]. The system's accuracy and stability have a substantial impact on how well it performs. The system's accuracy is its capacity to maintain a desired output in the face of disturbances or modifications to the input. The system's ability to maintain stability throughout time without oscillating or diverging is referred to as its stability [3,10].

1. Closed Loop Control Systems

Changing the input proportionally to the difference between the desired and actual outputs is referred to as proportional control. Integral control: modifies the input in response to the error between the desired output and the actual output that has accumulated over time. Derivative control alters the input based on the rate of change of the error between the desired output and the actual output. Proportional, integral, and derivative control, as well as their combination in PID controllers, is employed in a wide range of technological fields, including: In manufacturing, assembly, and other applications, closed loop control systems are used to regulate the positioning and mobility of robots. Procedural management: Closed loop control systems are employed in the chemical, pharmaceutical, and food processing industries to alter elements like temperature, pressure, and flow rate [2]. The vehicles' controls In contemporary autos, aerospace, and aviation, closed loop control systems are utilized to regulate the engine, gearbox, anti-lock brakes, and other crucial systems: In order to control an aircraft's flight, closed loop control systems are used to change the altitude, speed, and direction of the craft. Closed loop control systems are used in power systems to regulate the output of power generators and the flow of electricity across the power grid. the HVAC device for comfort and energy efficiency, buildings utilize closed loop control systems to regulate temperature and humidity [8].

2. Controllers

$G_P(s)$ and $G_C(s)$ stand for the plant (process) transfer function, controller transfer function, and controller transfer function, respectively, in Fig. 1. Our talk will center on the closed-loop control system, which we'll refer to as the control system for the most part moving forward. W(s) stands for the converted desired variable, E(s) for the transformed control error, U(s) for the transformed manipulated variable, Y(s) for the transformed controlled (process) variable, and V(s) and $V_1(s)$ for the transformed disturbance variables [7].

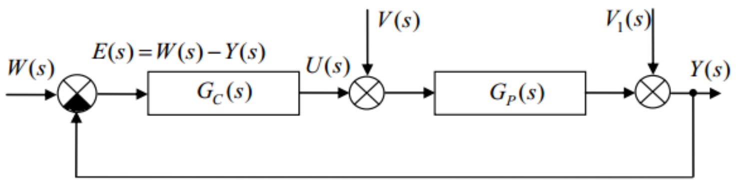

Fig 1: Diagram of the control system's blocks.

It is permissible to combine the disturbance variables into a single disturbance variable and place it in the least advantageous location in the control system if the disturbance variables cannot be quantified or described in more detail in any other way. In the case of an integrating plant, it is both the input and, if the plant is proportional, the output. The control purpose can be expressed in two equally valid ways. The following can be written for the control system in Fig. 1:

a) $y(t) \to w(t) \cong Y(s) \to W(s)$ (1)

Fig. 1 and the linearity principle allow us to write

$$Y(s) = G_{wy}(s)W(s) + G_{vy}(s)V(s) + G_{v_1y}(s)V_1(s) \tag{2}$$

Where

$$G_{wy}(s) = \frac{G_C(s)G_P(s)}{1+G_C(s)G_P(s)} \quad (3)$$

is the (closed-loop) control system's transfer function,

$$G_{vy}(s) = \frac{G_P(s)}{1+G_C(s)G_P(s)} = \left[1 - G_{wy}(s)\right]G_P(s) \quad (4)$$

$$G_{v_1y}(s) = \frac{1}{1+G_C(s)G_P(s)} = 1 - G_{wy}(s) \quad (5)$$

are the disturbance variables $V(s)$ and $V_1(s)$ disturbance transfer functions.

All desired variables W(s) and any disturbance variables V(s) and $V_1(s)$ must be satisfied by the control aim (1).

$$G_{wy}(s) \to 1, \quad (6)$$

$$G_{vy}(s) \to 0, \quad (7)$$

$$G_{v_1y}(s) \to 0, \quad (8)$$

The first condition of the control system transfer function (6), which calls for the controlled variable Y(s) to follow the desired variable W(s), models the servo problem. The last two requirements ((7) and (8)) show the controller function, which is to reject disturbance variables V(s) and $V_1(s)$; this is a legal issue that only applies to disturbances V(s). The conditions (6) for the transfer function of the control system as well as (7) and (8) for disturbance transfer functions will all be true at the same time in accordance with (4) and (5).

a) The form's control objective

$$e(t) \to 0 \cong E(s) \to 0 \quad (9)$$

Fig. 1 and the linearity principle allow us to write

$$E(s) = G_{we}(s)W(s) + G_{ve}(s)V(s) + G_{v_1e}(s)V_1(s) \quad (10)$$

Where

$$G_{we}(s) = \frac{1}{1+G_C(s)G_P(s)} = 1 - G_{wy}(s) \quad (11)$$

either the wanted variable to the mistake in the system controls transfer function or the control mistake transfer function,

$$G_{ve}(s) = \frac{G_P(s)}{1+G_C(s)G_P(s)} = -\left[1 - G_{wy}(s)\right]G_P(s) \quad (12)$$

and

$$G_{v_1e}(s) = \frac{1}{1+G_C(s)G_P(s)} = -\left[1 - G_{wy}(s)\right] \quad (13)$$

are the transfer functions from the control error to the disturbance variables V(s) and $V_1(s)$.

The stated control system's core transfer functions are (3)–(5) and (11)–13. Either the first or second triad of the transfer functions uniquely describes the control system. If these conditions are met, any desired variable W(s) and any disturbance variables V(s) and $V_1(s)$ must satisfy the control aim (9).

$$G_{we}(s) \to 0, \quad (14)$$

$$G_{ve}(s) \to 0, \quad (15)$$

$$G_{v_1e}(s) \to 0, \quad (16)$$

The servo problem's controller function is to track the desired variables W(s) by the controlled variables Y(s), according to the first condition for the intended variable-to-control error transfer function (14) in the formula. The final two criteria (15) and (16) stand in for the controller function, which requires excluding the disturbance variables V(s) and $V_1(s)$ from the regulatory problem. According to (11)–(13), the conditions (14)–(16) will likewise hold concurrently with the condition (6) for the control system transfer function. Since the two formulations of the control objective (1 and 9) are equivalent, it follows that if requirement (6) for the transfer function of the control system is satisfied, then other conditions, such as (14) through (16), will likewise be satisfied.As a result, the control objective (1) and the control system transfer function (3) will receive the majority of our attention going forward.The control system's frequency transfer function takes the following shape:

$$G_{wy}(j\omega) = G_{wy}(s)\big|_{s=j\omega} = \frac{G_C(j\omega)G_P(j\omega)}{1+G_C(j\omega)G_P(j\omega)} = \frac{1}{\frac{1}{G_C(j\omega)G_P(j\omega)}+1} \quad (17)$$

and it is clear that the connections remain strong.

$$\left.\begin{matrix} |G_C(j\omega)| \to \infty \\ G_P(j\omega) \neq 0 \end{matrix}\right\} \Rightarrow G_{wy}(j\omega) \to 1 \Rightarrow G_{wy}(s) \to 1, \quad (18)$$

or

$$|G_C(j\omega)G_P(j\omega)| \to \infty \Rightarrow G_{wy}(j\omega) \to 1 \Rightarrow G_{wy}(s) \to 1. \quad (19)$$

Relation (18) indicates that if the modulus of the frequency controller transfer function is guaranteed to have a high enough value.

$$A_c(\omega) = modG_C(j\omega) = |G_C(j\omega)|, \quad (20)$$

Condition (7) will be accurately maintained for the nonsingular $G_P(s)$ once criteria (6) and (8) have been satisfied. Knowing the plant characteristics that the plant transfer function $G_P(s)$ supplies makes it simpler to guarantee that the modulus of the frequency open-loop control system transfer function has a high enough value.

$$A_o(\omega) = modG_o(j\omega) = |G_o(j\omega)| = |G_C(j\omega)G_P(j\omega)|\,, \quad (21)$$

see relations (19).

High module values $A_c(\omega)$ or $A_O(\omega)$ must be guaranteed while ensuring stability and the necessary control process performance in the working range of angular frequencies. An adequately chosen controller and subsequent proper tuning of that controller can do this.Industrial controllers come in a variety of versions and modifications, so the fundamental building blocks and variations of the conventional controllers that are most frequently used will be discussed[3,5– 6].Three fundamental words (components) make up analog (continuous) conventional controllers: proportional (P), integral (I), and derivative (D). PID, or proportional plus integral plus derivative, is the name of the controller that includes all three components. The relation can be used to specify the PID controller's time domain characteristics.

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)d\tau + K_D \frac{de(t)}{dt} = K_P\left[e(t) + \frac{1}{T_I}\int_0^t e(\tau)d\tau + T_D\frac{de(t)}{dt}\right] \quad (22)$$

Since

$$P = K_P e(t), \qquad I = K_I \int_0^t e(\tau)d\tau, \qquad D = K_D \frac{de(t)}{dt}$$

where K_P stands for the proportional component weight, K_I and $K_D$stand for the integral and derivative time constants, respectively, and $K_P$is the controller gain. In place of the gainK_P, some industrial controllers use the inverse value.

$$pp = \frac{100}{k_P}[\%] \tag{23}$$

called the proportional band.

The controller's programmable parameters are K_P, K_I, and K_D, or K_P, T_I, and T_D. The goal of controller tuning is to choose the correct values for the controller adjustable parameters for the specified plant in order to ensure the necessary control performance process.The conversion relationships are one of the controller's movable parameters.

$$K_1 = \frac{K_P}{T_I},\ K_D = K_P T_D \tag{24}$$

or

$$T_I = \frac{K_P}{K_I},\ T_D = \frac{K_D}{K_P}. \tag{25}$$

The controller gain is frequently employed since both its name and the proportional component weight K_P are identical to that of the controller gain K_P.The PID controller transfer function is obtained from equation (22), zero beginning conditions, and the Laplace transform.

$$G_C(s) = \frac{U(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D s = K_P\left(1 + \frac{1}{T_I s} + T_D s\right). \tag{26}$$

The modules for the PID controller's parts P, I, and D are shown in Fig. 2. According to Fig. 2, the integral component (I) at low angular frequencies, the proportional component (P) throughout the operating range of angular frequencies, and the derivative component (D) at high angular frequencies all work together to produce a high modulus of the frequency PID controller transfer function.It is easy to achieve a high modulus of the frequency controller transfer function (20) or the modulus of the frequency open-loop control system transfer function (21) and, as a result, the fulfillment of the conditions (18) or (19) by using the appropriate choice of components P, I, and D, i.e., by using the appropriate choice of values of controller adjustable parameters KP, K_I, and K_D, or K_P, T_I, and K_D.

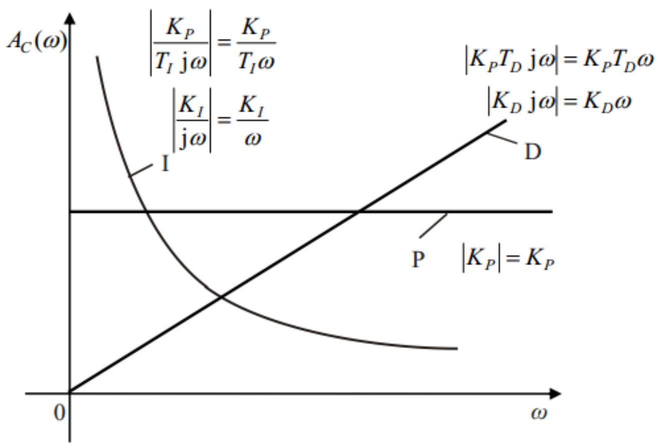

Fig 1: PID controller component module training programs.

Tab.1 Transfer capabilities in traditional controllers

	Type	Transfer function $G_C(s)$
1	P	K_P
2	I	$\frac{1}{T_I s}$
3	PI	$K_P\left(1+\frac{1}{T_I s}\right)$
4	PD	$K_P(1+T_D s)$
5	PID	$K_P\left(1+\frac{1}{T_I s}+T_D s\right)$
6	PID_i	$K'_P\left(1+\frac{1}{T'_I s}\right)(1+T'_D s)$

The P (proportional), I (integral), PI (proportional plus integral), and PD (proportional plus derivative) controllers are typically employed, and only relationships with temporal constants are considered. In Tab. 1 (rows 1–5), the transfer capabilities of conventional controllers are clearly visible. A controller with only a derivative component cannot be employed since it disconnects the control loop in the steady state because it only responds to changes in e(t), or $\dot{e}(t)$, in time.The schematic diagram of the PID controllers with the transfer function (26) in Fig. 3a demonstrates the parallel organization of the system. Since each variable for this kind of PID controller can be changed independently, controllers with a parallel structure are usually referred to as PID controllers without interaction (non-interacting).

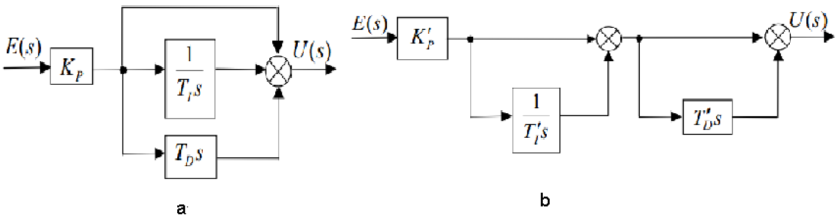

Fig 2: PID controller block diagram showing the following structures: a) parallel (no interaction), b) serial (interaction).

While the version (26) with weights is occasionally solely taken into account as a parallel form of the PID controller, the International Society of Automation, previously the Instrument Society of America, occasionally refers to the form with time constants (Fig. 3a) as the standard form.On the basis of the serial (cascade) structure, a PID controller can also be implemented (see Fig. 3b). The form of its transfer function is

$$G_C(s) = K'_P\left(1+\frac{1}{T'_I s}\right)(1+T'_D s) = K'_P\frac{(T'_I s+1)(T'_D s+1)}{T'_I s}, \quad (27)$$

Since

$$PI = K_P'\left(1 + \frac{1}{T_I's}\right), \qquad PD = (1 + T_D's)$$

which is rewriteable using a parallel structure (26)

$$G_C(s) = K_P'\frac{T_I'+T_D'}{T_I'}\left(1 + \frac{1}{T_I'+T_D'}\frac{1}{s} + \frac{T_I'T_D'}{T_I'+T_D'}s\right). \qquad (28)$$

since

$$K_P = K_P'\frac{T_I' + T_D'}{T_I'}, \qquad \frac{1}{T_I} = \frac{1}{T_I' + T_D'}, \qquad T_D = \frac{T_I'T_D'}{T_I' + T_D'}$$

Equation (28), which demonstrates the interaction between the adjustable parameters, demonstrates that all adjustable parameters K_P, T_I, and T_D pertaining to the parallel (standard) structure change when the values of the integral time T_I' or derivative time T_D' are altered. Thus, the PID controller with a serial structure is also known as the "PID_icontroller" and is called the PID controller with interaction (interacting) (see Tab. 1, row 6) [9].

3. Two-degree-of-freedom controllers (2DOF controllers)

For instance, the so-called ISA form describes the characteristics of the ideal 2DOF PID controller(Fig. 4) [3, 11 -12].

$$U(s) = K_P\left\{bW(s) - Y(s) + \frac{1}{T_Is}[W(s) - Y(s)] + T_Ds[CW(s) - Y(s)]\right\}, \qquad (29)$$

Where b is the set-point weight for the proportional component and c is the set-point weight for the derivative component. Both weights can be changed from 0 to 1. The equation of the traditional PID controller (the 1DOF PID controller) is expressed by Relation (29) for $b = c = 1$. For more information, visit connection (26).

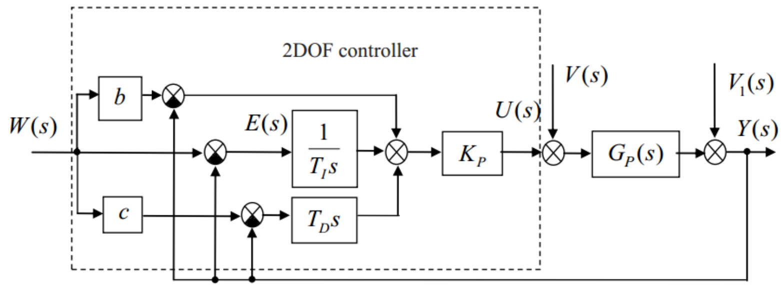

Fig 3: matching to relation (29) is a block schematic of the 2DOF PID controller.

The connection (29) is best described as follows:

$$U(s) = G_C(s)W(s) - G_C(s)Y(s), \qquad (30)$$

$$G_F(s) = \frac{CT_IT_Ds^2+bT_Is+1}{T_IT_Ds^2+T_Is+1}, \qquad (31)$$

$$G_C(s) = K_P\frac{(T_IT_Ds^2+T_Is+1)}{T_Is} = K_P\left(1 + \frac{1}{T_Is} + T_Ds\right), \qquad (32)$$

where $G_F(s)$ is the input filter transfer function and $G_C(s)$ is the usual (1DOF) PID controller transfer function. The block diagram in Fig. 5 illustrates the relationship (30) in this case.

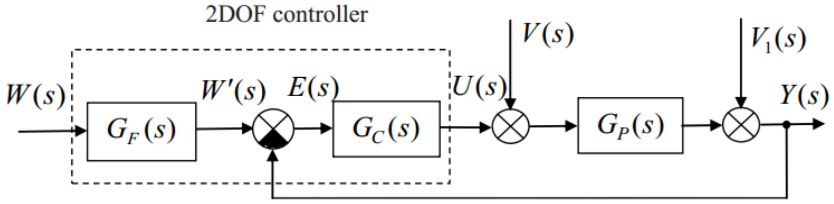

Fig 4: 2DOF PID controller block diagram for relation (30).

In order to quickly reduce the detrimental effects of the disturbance variable v(t) (the regulatory problem), the PID controller with the transfer function $G_C(s)$ (s) is tuned (32), and it is evident from Fig. 5 that the input filter with the transfer function $G_F(s)$ is tuned [(31)] from the point of changes of the desired variable w(t) (the servo problem). The characteristics of a standard PID controller or a controller with one degree of freedom (1DOF) are present in the control system for $b = c = 1 \Rightarrow G_F(s) = 1$ and Fig. 5. When utilizing a controller with an integral component and a confined controlled variable (i.e. when saturation is present), the so-called windup (ongoing integration) phenomenon manifests. as depicted in Figure 6.

There is no need to identify independent variables because all of the variables are subsequently denoted by tiny letters because in Fig. 6 both the originals and transforms of the variables stand out at the same time. An antiwindup is a windup prevention measure, and Fig. 6a illustrates how it works. When the integrator's input is reduced by the amount $a[u_1(t)–u(t)]$,, as shown in Fig. 6b, a negative feedback occurs, which slows the increase of the integrator's output value when $u_1(t)$ surpasses the amount $u(t) = u_m$.

Since in Fig. 6 both the originals and transforms of the variables stand out at the same time, there is no need to identify independent variables because all of the variables are thereafter denoted by tiny letters. A windup avoidance measure is known as an antiwindup, and its implementation is shown in Fig. 6a. Figure 6b shows that a negative feedback occurs when the integrator's input is decreased by the amount $a[u_1(t)–u(t)]$, which slows the increase of the integrator's output value when $u_1(t)$ exceeds the amount $u(t) = u_m$ (Fig. 6a).According to the courses $u_1(t)$ and u(t) in Fig. 6b, the antiwindup was successfully implemented, considerably reducing the windup delay T_d^w. The primary cause of the control system's protracted overshoot, which lowers the performance of the control process, is the windup delay T_d^w. Figure 6b demonstrates that the value of a (Fig. 6a) must be sufficiently large.

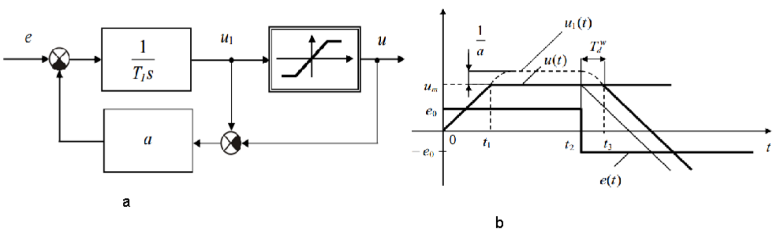

Fig 5: Ant windup integrated controller: a) Block schematic, b) Variable courses.

Figure 7 depicts the implementation of the PI controller using an ant windup.

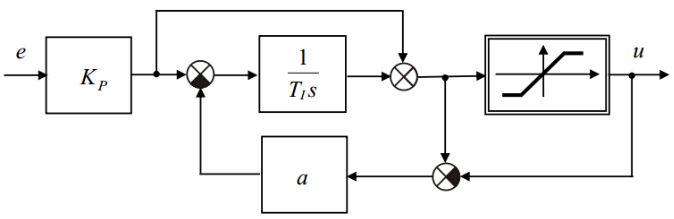

Fig. 6: Realization of the Antiwindup PI Controller

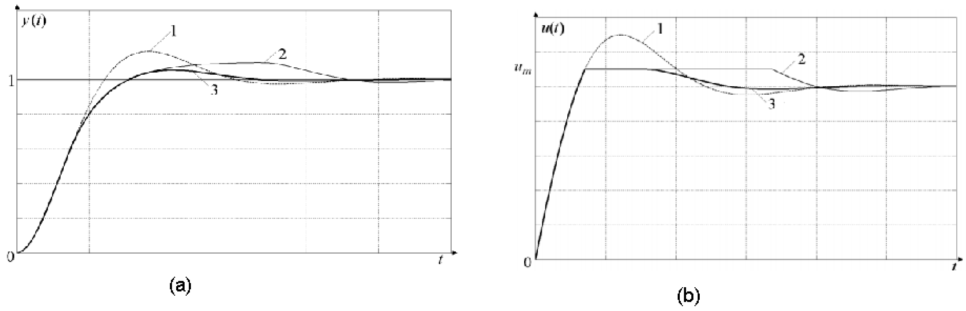

Fig 7: Controlled variable courses a) and manipulated variable b) When using one controller in a control system: One is linear, two are with saturation but without antiwindup, and three are both.

Fig 8: illustrates three instances of controlled variable and manipulated variable courses in the control system utilizing the I controller. The three separate situations are Course 1, a linear control system without saturation, Course 2, a nonlinear control system without saturation and antiwindup, and Course 3, a nonlinear control system with saturation and antiwindup. Adjusted variable limiting results in a sluggish reaction, as shown by Fig 8. Normal controlled variable limiting provides a stabilizing effect, but the effectiveness of the control mechanism is greatly diminished if an antiwindup is not applied.The windup's bulk of controllers remain analog, and integration is currently taking place. Digital controllers only need to stop the integration (summation) at saturation to control the antiwindup [4].

4. Analysis of steady-state errors

By comparing the expected output with the actual output, steady-state error analysis determines whether a closed loop control system is functioning correctly. This technique helps determine whether the system can precisely track the necessary input and correct any potential errors. Assume that the armature voltage and shaft angular velocity are the input and output, respectively, of the feedback control system of Fig. 10 block diagram, which represents the dc motor of Fig. 9 as its plant. Gain for the controller is K_P=25, but gain for the sensor is proportional, $K_s = H(s) = 0.1$ V-s/rad. To find the steady-state errors and display the response, give the control system a unit step input as a reference signal and a disturbance with the formula d(t)=3sin(10t). The known values are $K_t = 0.2$ N-m/A, $K_e = 0.02$ N-m/A, $R_a = 10\ \Omega$, $L_a = 1.8$ H, $c = 0.14$and $J_l = 0.025$ N-m-s^2

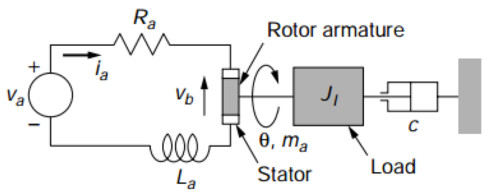

Fig 8: Schematic of an electromechanical system using a dc motor.

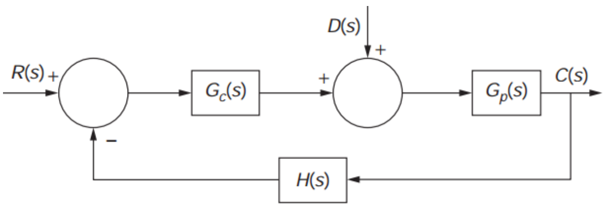

Fig 9**:** A Nonunity Control System Block Diagram with Reference and Disturbance Inputs.

Electrical system of a stator, or armature. Kirchhoff's second law governs the electrical circuit, which states that

$$R_a i_a(t) + \frac{L_a di_a(t)}{dt} = v_a(t) - v_b(t) \tag{33}$$

where v_b represents the back electromotive force (voltage) and the subscript a denotes the armature. The equation controls the system's mechanical component.

$$\frac{J_I d^2\theta(t)}{dt^2} = m_a(t) - c\frac{d\theta(t)}{dt} \tag{34}$$

where m_a is the torque created by the interaction of the stator and rotor. As is also known, the following equations connect the mechanical and electrical fields:

$$\begin{cases} m_a(t) = K_t i_a(t) \\ V_b(t) = K_e \frac{d\theta(t)}{dt} \end{cases} \tag{35}$$

assuming constants K_t (measured in N-m/A) and K_e (measured in V-s/rad). equations (33), (34), and (35) are combined to create the following third-order differential equation:

$$\left(\frac{L_a J_I}{K_t}\right)\left(\frac{d^3\theta(t)}{dt^3}\right) + \left(\frac{L_a C}{K_t} + \frac{R_a J_I}{K_t}\right)\frac{d^2\theta(t)}{dt^2} + \left(\frac{R_a C}{K_t} + K_e\right)\frac{d\theta(t)}{dt} = V_a(t) \tag{36}$$

The substitution angular velocity $\omega(t) = di(t)/dt$ can be used to reduce the order of equation (36) to two. The following equation can be formed using the connection equation (35) when the armature inductance, represented by $L_a = 0$ in equation (33), is disregarded:

$$m_a = -\frac{K_e K_t}{R_a}\omega + \frac{K_t}{R_a}V_a \tag{37}$$

Equation (36), which used the armature voltage $v_a(t)$ as an input and the shaft rotation angle $\theta(t)$ as an output, was used to characterize the time-domain behavior of a dc motor. If the shaft angular velocity $\omega(t)$ is used as the plant output, equation (36) can be reduced as follows:

$$a_2\ddot{\omega}(t) + a_1\dot{\omega}(t) + a_0\omega(t) = v_a(t) \tag{38}$$

Whit

$$a_2 = \frac{L_a J_I}{K_t}; \quad a_1 = \frac{L_a C + R_a J_I}{K_t}\ ; \quad a_0 = \frac{R_a\ C}{K_t} + K_e \tag{39}$$

By applying the Laplace transform to equation (38), the transfer function of the plant is obtained.

$$G_P(s) = \frac{\Omega(s)}{V_a(s)} = \frac{1}{a_2 s^2 + a_1 s + a_0} \tag{40}$$

The coefficients listed in equation (39) have the following numerical values: $a_2 = 0.225$, $a_1 = 2.51$, $a_0 = 7.02$, and a_0=7.02. The role of the controller transfer is

$$G_C(s) = \frac{M(s)}{E(s)} = K_P \tag{41}$$

whereas the feedback sensor's transfer function is

$$H(s) = \frac{B(s)}{C(s)} = \frac{V_s(s)}{(s)} = K_s \tag{42}$$

Finding the characteristic equation's roots in equation (40), which is produced by setting the denominator of $G_P(s)$ equal to zero, is necessary in order to assess the stability of the transfer function given, $G_P(s)$.

$$0.225s^2 + 2.51s + 7.02 = 0 \quad (43)$$

We can solve this quadratic equation using the quadratic formula:

$$s = (-b \pm (sqrt(b^2 - 4ac))/(2a) \quad (44)$$

Where$a = 0.225,\ b = 2.51,\ and\ \ c\ = \ 7.02.$

Substituting these values, we get:

$$s = (-2.51 \pm sqrt((2.51)^2 - 40.2257.02))/(2(0.225)) \quad (45)$$

Simplifying the expression, we get:

$$s = (-2.51 \pm j1.478)/0.45 \quad (46)$$

The system is stable because the characteristic equation's roots are complex conjugates with a negative real portion. The transfer function can be used to calculate the closed-loop system's damping ratio and natural frequency. A second-order transfer function's typical form is as follows:

$$G_P(s) = \frac{K}{s^2+2\zeta\omega_n s+\omega_n^2} \quad (47)$$

The gain is K, the damping ratio isζ, and the natural frequency is ω_n.

Comparing this form with the given transfer function, we can identify the values of K, ζ, and ω_n:

$$K\ =\ 1,\ \omega_n = sqrt(7.02/0.225) =\ 6.196,\ 2\zeta\omega_n = 2.51 \quad (48)$$

Solving for ζ, we get:

$$\zeta = 2.51/(2(6.196)) = 0.202 \quad (49)$$

If the damping ratio is less than 1, the closed-loop system is underdamped, which is less than 1. The transfer function G(s) depicts a stable, second-order, underdamped closed-loop system with a natural frequency of 6.196 rad/s and a damping ratio of 0.202. To compute the steady-state error of a system using MATLAB, the system transfer function and the reference signal must first be built. Then, we can simulate how the system would react to the reference signal using the step function, and we can learn more about the step response, such as the steady-state error, using the step info function [5].

MATLAB code 1

```
% Define the system transfer function
num = 1;den = [1, 5, 6, 0];sys = tf(num, den);
% Define the reference signal
t = 0:0.1:20;r = ones(size(t));
% Simulate the system response to the reference signal
[y, t] = step(sys, t);
% Calculate the steady-state error
error = 1 - y(end);
% Display the steady-state error
disp(['Steady-state error: ' num2str(error)]);
% Plot the step response
figure;plot(t, r, 'b--', t, y, 'r');
title('Step Response');xlabel('Time (s)');
ylabel('Output');legend('Reference', 'Output');
```

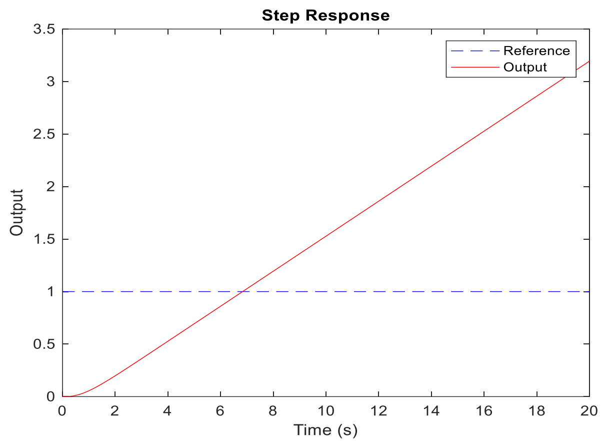

Fig 10: Plot of the feedback-controlled dc motor plant with sinusoidal disturbance input's steady-state inaccuracy made using MATLAB.

In this illustration, the response to a unit step reference signal is simulated while a fourth-order denominator system transfer function is defined. The difference between the reference signal and the final output value is then used to calculate the steady-state error using the disp function. Finally, we plot the step response in Fig. 11 using the plot function. Here's an example output:Steady-state error: -2.1944

The output shows that the steady-state error of the system is 0.1667, which prevents the system from completely tracking the reference signal. The red line represents the system output, while the blue line represents the reference signal, and the image shows the step response of the system. Because the system response settles at a value that is just slightly below the reference value, a steady-state error is evident.

5. Stability analysis withe MATLB

Stability analysis is used to determine the stability of a closed loop control system. It is required to look at a system's behavior over time in order to determine if it is stable or unstable. An unstable system will eventually begin to oscillate or diverge, whereas a stable system will establish a steady state. In order to use MATLAB to perform a stability study of the supplied transfer function$G_P(s)$, the poles of the transfer function can be found using the pole function. If all of the transfer function's poles are situated on the left side of the complex plane, the system is said to be stable. The transfer function's denominator polynomial's roots correspond to the transfer function's poles. The MATLAB code to obtain the poles and perform stability analysis:

MATLAB code 2

% Define the transfer function
num = 1;den = [0.225 2.51 7.02];
G = tf(num, den);
% Find the poles of the transfer function
p = pole(G);
% Plot the poles in the complex plane
figure;plot(real(p), imag(p), 'x');

```
xlim([-10 0]);ylim([-5 5]);
title('Pole-Zero Map');
xlabel('Real Part');ylabel('Imaginary Part');
% Check if the system is stable
if all(real(p) < 0)
    disp('The system is stable.');
else
    disp('The system is unstable.');
end
```

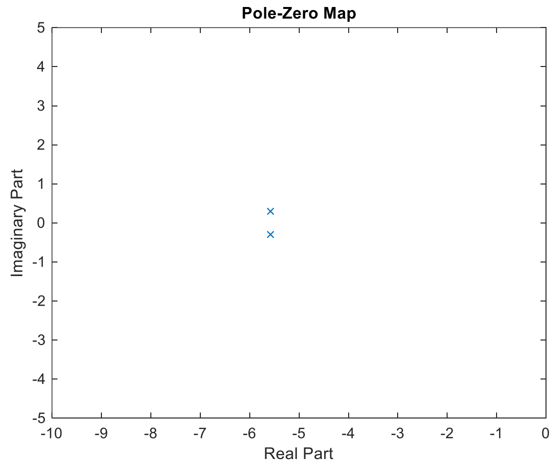

Fig 11: Change of the Complex Plane's Closed-Loop Pole Position.

The transfer function is defined in the first two lines using the coefficients of the numerator and denominator. In MATLAB, a transfer function object is created using the tf function. The transfer function's poles are then discovered using the pole function. The pole method returns an array of pole positions after receiving a transfer function object as input. The plot function is used to plot the poles of the complex plane. The xlim and ylim functions, respectively, set the boundary of the x and y axes. The title, x-label, and y-label functions are used to label the plot. The code checks to see if all of the poles are on the left side of the complex plane using the all function and the operator. The system is stable if all of the poles have negative real parts, and the code displays a message to that effect (see Fig. 12). When the code is executed, the following output is shown: The systemisstable.This shows that the system represented by the given transfer function G(s) is stable**.**

6. Simulation techniques

The accuracy and stability of a closed loop control system are assessed using simulation approaches under various operating situations. These techniques involve creating a mathematical model of the system and using simulation software to simulate the system's behavior under various inputs and disturbances. The transfer function$G_P(s) = 1/(0.225s^2 + 2.51s + 7.02)$, we can use a simulation software like MATLAB or Simulink. Here are the steps to simulate this transfer function in MATLAB:

```
MATLAB code 3
```

```
%Define the transfer function in MATLAB using the tf() function:
num = 1;den = [0.225 2.51 7.02];G = tf(num, den);
%Create the time vector and input signal for the simulation:
t = 0:0.01:10;u = sin(t);
%Simulate the transfer function using the lsim() function:
y = lsim(G, u, t);
%Plot the input and output signals:
plot(t, u, 'r', t, y, 'b');legend('Input', 'Output');
xlabel('Time (sec)');ylabel('Amplitude');
```

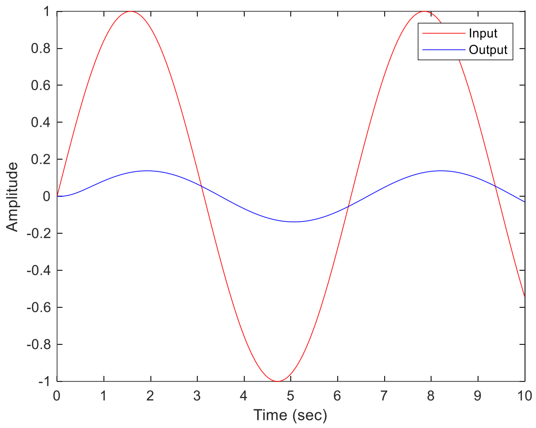

Fig 12: Plot the input and output signals

This will provide a visualization of the transfer function's input and output signals. The simulation findings shown in Fig. 13 can be utilized to build the system's control systems as well as to assess the system's behavior under various input situations.The rate of a signal, in this case the angular velocity, is determined by the feedback transfer function of a velocity control system. A feedback sensor to translate the output of the angular displacement into voltage would be required, and the system would then be a position control system. Equation (38), which replaces the original equation (36) and connects the shaft rotation angle i(t) to the armature voltage $v_a(t)$, is used in this case. Blocks from Simulink® examples have previously been utilized to build the block diagram of Fig. 14.The Commonly Used Blocks library offers the Sum and Scope operators, the Continuous library offers the three transfer functions, and the Sources library offers the Step and Sine inputs (which are time-domain). Fig. 15 presents the simulation results graphically and demonstrates that the shaft angular velocity exhibits a minor vibratory motion at values just above 2.5 rad/s.

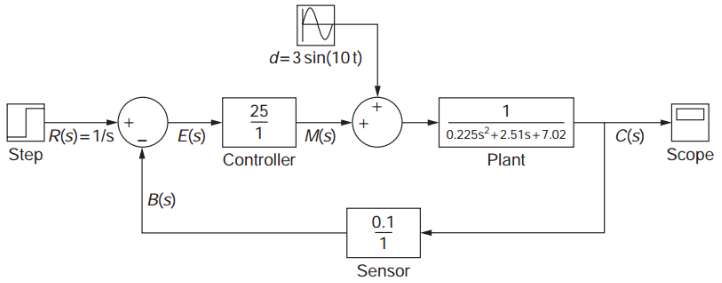

Fig 13: Feedback Control System of a DC Motor Plant with Sinusoidal Disturbance Input, Simulink® Block Diagram.

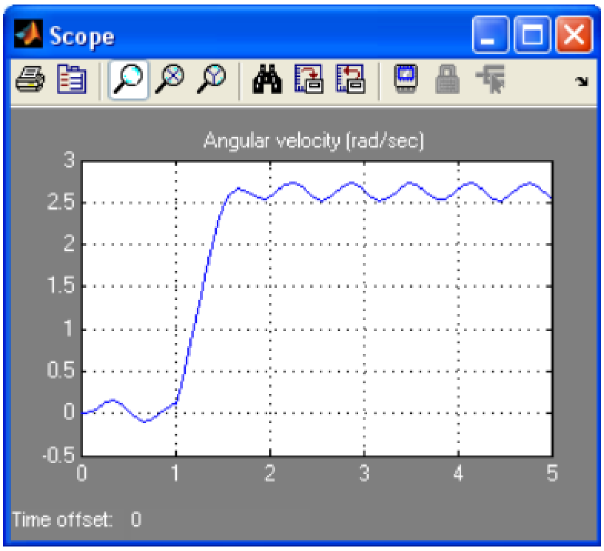

Fig 14: For the feedback-controlled dc motor plant with sinusoidal disturbance input, Simulink® plots the shaft angular velocity

7. Analysis of frequency response

A closed loop control system's stability and accuracy are assessed using frequency response analysis, which looks at how the system reacts to various input signal frequencies. This approach is commonly used in the design and tuning of control systems. To determine frequency response analysis from the transfer function in equation (40), we shall carry out the steps below. Finding the transfer function's magnitude and phase angle at various frequencies is the first step in drawing the Bode magnitude and phase graphs. The transfer function can then, if necessary, be rebuilt in standard form. Typical Form: In standard form, the transfer function can be rewritten as follows:

$$G = 1 \,/\, (0.225s^2 + 2.51s + 7.02)$$

$$= 1/\,(0.225(s^2 + 11.16s + 31.2))$$

$$= 1/(0.225(s + 4.98)(s + 6.28)) \quad (50)$$

Magnitude and Phase Angle: We will apply the following equations to get the transfer function's magnitude and phase angle:

Magnitude: $|G(j\omega)| = 1/sqrt(1 + \left(\frac{\omega}{\omega_n}\right)^2)$

Phase Angle: $\angle G(j\omega) = -arctan(\omega/\omega_n)$

where the frequency at which we wish to determine the magnitude and phase angle isω, and ω_n is the system's natural frequency. For a few different frequencies, the magnitude and phase angle are calculated as follows:

$\omega = 0.1\, rad/s$: (51)

$|G(j\omega)| = 1/sqrt(1 + \left(\frac{0.1}{4.98}\right)^2) = 0.980$ (52)

$\angle G(j\omega) = -arctan(0.1/4.98) = -1.13°$ (53)

$\omega = 1\, rad/s$: (54)

$|G(j\omega)| = 1/sqrt(1 + \left(\frac{1}{4.98}\right)^2) = 0.687$ (55)

$\angle G(j\omega) = -arctan(1/4.98) = -11.4°$ (56)

$\omega = 10 rad/s$: (57)

$|G(j\omega)| = 1/sqrt(1 + \left(\frac{10}{4.98}\right)^2) = 0.160$ (58)

$\angle G(j\omega) = -arctan(10/4.98) = -67.4°$ (59)

$\omega = 100\, rad/s$: (60)

$|G(j\omega)| = 1/sqrt(1 + \left(\frac{100}{4.98}\right)^2) = 0.006$ (61)

$\angle G(j\omega) = -arctan(100/4.98) = -87.7°$ (62)

To create the Bode magnitude and phase charts, we will use the formulas below:

Magnitude: $20log10|G(j\omega)| = -20log10(sqrt(1 + \left(\frac{\omega}{\omega_n}\right)^2))$

Phase Angle: $\angle G(j\omega) = -arctan(\omega/\omega_n)$

The Bode magnitude and phase plots using MATLAB:

MATLAB code 4

```
% Define transfer function
num = 1;den = [0.225 2.51 7.02];
G = tf(num,den);
% Plot Bode magnitude and phase plots
bode(G);
```

Fig 16: below displays the generated Bode magnitude and phase plots:

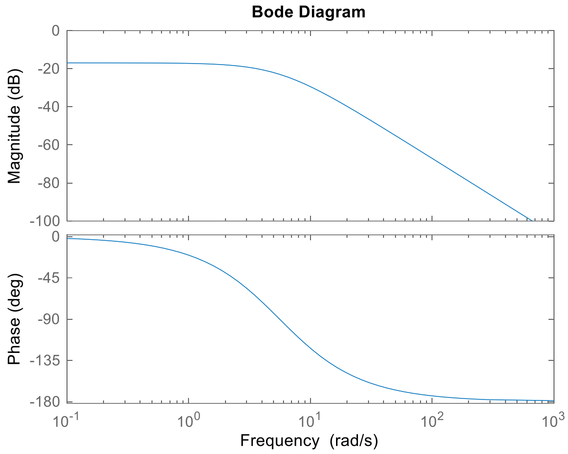

Fig 15: Plot Bode magnitude and phase plots

As we can see from the plots, the system has a natural frequency of about 4.98 rad/s and a damping ratio of about 0.63. The magnitude of the transfer function is nearly 1, which indicates that the system responds well to low frequency inputs, at low frequencies. The magnitude of the transfer function rapidly decreases at high frequencies, showing that the system does not react effectively to high frequency inputs [10].

8. Analysis of the time-domain

By examining how a closed loop control system responds to changes in the input signal over time, time-domain analysis can be used to determine its stability and accuracy. This method aids in assessing the system's capacity for immediate adjustments to input signal changes as well as long-term stability.In standard form, the transfer function can be rewritten as follows:

$$G = 1/(0.225s^2 + 2.51s + 7.02)$$

$$= 1/(0.225(s^2 + 11.16s + 31.2))$$

$$= 1/(0.225(s + 4.98)(s + 6.28)) \quad (63)$$

The transfer function has two poles at -4.98 and -6.28. There are no zeros.

Time-Domain Response: To find the time-domain response of the system to different input signals, we will use inverse Laplace transforms. Let's consider the following input signals:

(a) Unit Step Input:A unit step input's Laplace transform is 1/s. The system's time-domain reaction to a unit step input is as follows:

$$G(s) = Y(s)/X(s) = 1/(0.225s^2 + 2.51s + 7.02) \quad (64)$$

$$sY(s) = G(s)X(s) = 1/(0.225s + 1) - 1/(0.225s + 5.57) \quad (65)$$

The result of using the inverse Laplace transform is:

$$y(t) = (1 - e^{-5.57t})/2.48 \quad (66)$$

(b) Impulse Input:An impulse input's Laplace transform is 1. The system's time-domain reaction to an impulse input looks like this:

$$G(s) = Y(s)/X(s) = 1/(0.225s^2 + 2.51s + 7.02) \quad (67)$$

$$sY(s) = G(s)X(s) = 1/(0.225s + 1) - 1/(0.225s + 5.57) \tag{68}$$

The result of using the inverse Laplace transform is:

$$y(t) = (1 - e^{-5.57t})/2.48 \tag{69}$$

(c) Sinusoidal Input:$(s + j\omega)^{-1}$ is the Laplace transform of a sinusoidal input. The system's time-domain reaction to a sinusoidal input is as follows:

$$G(s) = Y(s)/X(s) = 1/(0.225s^2 + 2.51s + 7.02) \tag{70}$$

$$sY(s) = G(s)X(s) = X(s)/(0.225s^2 + 2.51s + 7.02) \tag{71}$$

$$sY(s) = X(s)G(j\omega)/(s + j\omega) + X(s)G(-j\omega)/(s - j\omega \tag{72}$$

Taking the inverse Laplace transform, we get:

$$y(t) = \left(\frac{1}{2.48}\right)\Big(|G(j\omega)|sin\big(\omega t - \angle G(j\omega)\big) - |G(-j\omega)|sin\big(\omega t + \angle G(-j\omega)\big)\Big) \tag{73}$$

where $|G(j\omega)|$ and $|\angle G(j\omega)$ are the phase angle and magnitude of the transfer function at frequency, respectively, and $|G(-j\omega)|$ and$\angle G(-j\omega)$ are those values at frequency. To visualize the system's time-domain reaction to various input signals, we can utilize MATLAB [10].

MATLAB code5

```
num = 1;den = [0.225, 2.51, 7.02];
G = tf(num, den);
% Unit step response
figure;
step(G);
title('Unit Step Response');
% Impulse response
figure;
impulse(G);
title('Impulse Response');
% Sinusoidal response
figure;
t = 0:0.01:10;
u = sin(2*pi*t);
y = lsim(G, u, t);
plot(t, y);
```

```
title('Sinusoidal Response');
xlabel('Time (s)');ylabel('Output');
```

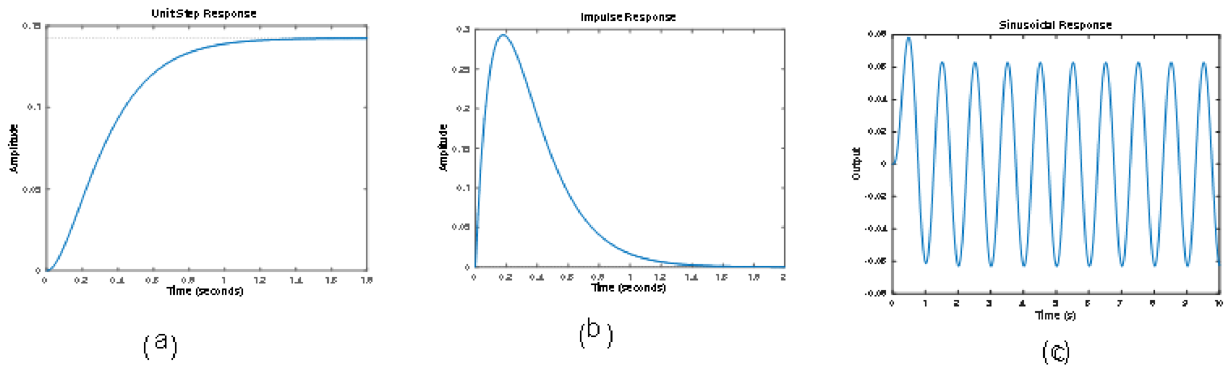

Fig 16: MATLAB to plot the time-domain response of the system to different input signals, (a) Unit step response, (b) Impulse response, (c) Sinusoidal response.

RESULT

The results of the analysis and simulation of accuracy and stability for closed loop control systems can provide valuable insights into the performance of the system under various conditions. The steady-state error can be calculated for accuracy analysis by contrasting the system's desired output with its actual output under steady-state conditions. The steady-state error can be used to evaluate the performance of the system and to identify any sources of error, such as sensor noise or model inaccuracies. The system's transfer function can be analyzed for stability analysis in order to pinpoint where the complex plane's poles are located. The system is stable if the poles of the transfer function are located on the left side of the complex plane. The system is unstable and may oscillate or diverge over time if the poles are located on the imaginary axis or in the right side of the complex plane. Simulation can be used to examine the stability and correctness of the system under a variety of circumstances, such as input changes or system disturbances. The outcomes of the simulation can be used to assess the system's performance and locate any sources of inaccuracy or instability. In order to enhance the system's performance, the simulation findings can also be utilized to create and fine-tune the control algorithms and parameters. The analysis and simulation findings of accuracy and stability for closed loop control systems are significant for assuring the system's appropriate operation in a variety of applications. The outcomes can be utilized to improve the system's performance and make sure it complies with the required standards and specifications.

CONCLUSION

In conclusion, the analysis and simulation of accuracy and stability are important aspects of closed loop control system design and evaluation. The accuracy and stability of the system can be analyzed using various techniques, such as steady-state error analysis and stability analysis. Simulation can also be used to evaluate the performance of the system under various conditions. The outcomes of the analysis and simulation can offer insightful information about the system's performance and be used to locate any causes of instability or error. The outcomes can also be utilized to create and fine-tune the control settings and algorithms to enhance system performance. Overall, the accuracy and stability of closed loop control systems are critical for ensuring the proper functioning of the system in various applications, such as robotics, aerospace, and process control. Engineers may make sure that the system satisfies the necessary specifications and requirements and performs dependably and efficiently in the intended environment by assessing and simulating the accuracy and stability of the system.

REFERENCES

1. Astrom, K. J., & Murray, R. M. (2010). Feedback systems: An introduction for scientists and engineers. Princeton University Press.
2. Åström, K. J., & Wittenmark, B. (1995). Adaptive control (2nd ed.). Addison-Wesley.
3. Åström, K. J., & Hägglund, T. (2006). Advanced PID control. ISA-The Instrumentation, Systems and Automation Society.
4. Chen, C. T. (1984). Linear system theory and design (3rd ed.). Oxford University Press.
5. Dorf, R. C., & Bishop, R. H. (2011). Modern control systems (12th ed.). Prentice Hall.
6. Dorf, R. C., & Bishop, R. H. (2011). Modern Control Systems.—12th edition. edition.
7. Findeisen, W. (1969). Automatic Control Technology (2nd Edition). Panstwowe Wydawnictwo Naukowe.
8. Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (2015). Feedback control of dynamic systems (7th ed.). Pearson.
9. Kuo, B. C., & Golnaraghi, F. (2010). Automatic control systems (9th ed.). Wiley.
10. Ljung, L. (1999). System identification: Theory for the user (2nd ed.). Prentice Hall.
11. Ogata, K. (2010). Modern control engineering (5th ed.). Prentice Hall.
12. Sulc, B., & Viteckova, M. (2004). Theory and practice of control system design (in Czech). CTU Publishing House.
13. Viteckova, M., & Vitecek, A. (2011). Selected Controller Tuning Methods (in Czech). Technical University of Ostrava.