The Neotia University

Subject: Control System Lab **Subject Code: PC-RE/P/502**

Experiment No: 01

Name of the experiment: Familiarization with MATLAB tools, MATLAB Simulink and PSPICE.

Objective: To be familiar with different relevant MATLAB tools, MATLAB Simulink and PSPICE so that software-based experiments can be performed smoothly in future.

MATLAB Tools

SISO tool

Theory:

It is a graphical user interface (GUI) of MATLAB. It can be used to draw and manipulate the root locus plot of single input and single output system. It is a part of MATLAB control system toolbox. By default, root locus, and bode plot are drawn first. And this two are dynamically linked that is if we change the gain in root locus, it immediately affects the Bode Diagram as well.

Experimental transfer function:

G(s)=K/s(s+2)(s+20)

LTI view

Theory:

LTIVIEW opens a blank LTI Viewer. The LTI Viewer is an interactive graphical user interface (GUI) for analysingthe time and frequency responses of linear systems and comparing such systems. LTIVIEW(plottype,sys,extras) allows the additional input arguments supported by the various LTI model response functions to be passed to the LTIVIEW command. PLOTTYPE may be any of the following strings (or a combination thereof):

1) 'step' Step response
2) 'impulse' Impulse response
3) 'bode' Bode diagrams
4) 'bodemag' Bode magnitude plot
5) 'nyquist' Nyquist plot
6) 'nichols' Nichols plot
7) 'sigma' Singular value plot
8) 'pzmap' Pole/zero map

For example, LTIVIEW({'step';'bode'},sys1,sys2) opens an LTI Viewer showing the step and Bode responses of the LTI models SYS1 and SYS2.

Experimental transfer function:

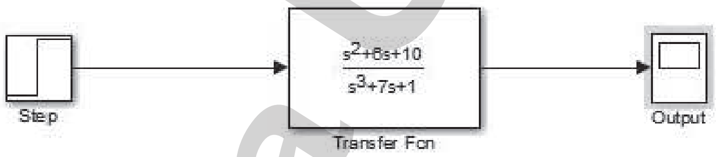
G(s)=K/s(s+2)(s+20)

MATLAB Simulink

Theory:

Simulink is a block diagram environment for multi-domain stimulation and Model Based Design. It supports system level design, simulation, automatic code generation and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modelling and simulating dynamic systems. It is integrated with MATLAB, enabling you to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis.

SIMULINK Model (example):

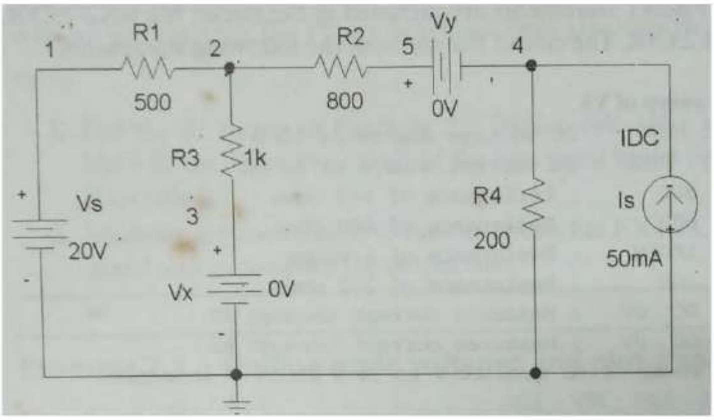


PSpice

Theory:

The PSpice Systems Option provides designers with a system-level simulation solution for their designs. Designers utilize PSpice simulation programs for accurate analcg and mixed-signal simulations supported by a wide range of board-level models. MATLAB and Simulink are a platform for multi-domain simulation and model-based design of dynamic systems The powerful combination enables developers to accelerate their PCB designs in a range of application areas, from automotive to internet of things (IoT) to industrial designs.

Schematic Diagram:



Experimental Code (Example):

```
VS   1 0 DC 20V        ; DC Voltage source of 10 V
 IS   0 4 DC 50MA      ; DC Current source of 50mA
 R1  1 2   500        ; Resistance of 500 ohm.
 R2  2 5   800        ; Resistance of 800 ohm.
 R3  2 3  1KOHM       ; Resistance of 1 kohms.
 R4  4  0  200        ; Resistance of 200 ohms.
 VX 3  0 DC    0V     ; Measures current through R3
 VY 5  4 DC    0V     ; Measures current through R2
   OP                 ; Directs the bias point to the output file
END                   ; End of circuit file.
```

Discussion:

Discuss about your experience and knowledge gathered through this experiment.

The Neotia University

Subject: Control System Lab **Subject Code: PC-RE/P/502**

Experiment No: 02

Name of the experiment: Determination of step response of first order and/or second order system with unity feedback andcalculation of various control system specifications.

Objective: Design an underdamped 2nd order system in MATLAB Simulink, analyse the step response and find out the following parameters.

Rise time

Settling time

Peak time

Peak overshoot

Theory:

An underdamped system moves quickly to equilibrium, but will oscillate about the equilibrium point as it does so.

The various response characteristics are as follows:

Rise Time- It is denoted by t_r. The rise time is the time required for the response to rise from 10% to 90%, 5% to 95%, or 0% to 100% of its final value. For underdamped second order systems, the 0%to 100%rise time is normally used.

Settling Time- It is denoted by t_s. The settling time is the time required for the response curve to reach and stay within a range about the final value of size specified by absolute percentage of the final value (usually 2% or 5%). The settling time is related to the largest time constant of the control system.

Peak Time- It is denoted by t_p. The peak time is the time required for the response to reach the first peak of the overshoot.

Peak overshoot- Peak overshoot M_p is defined as the deviation of the response at peak time from the final value of response. It is also called the maximum overshoot.

Experiment: Take a suitable transfer function and complete the experiment.

Discussion:

Discuss about your experience and knowledge gathered through this experiment.

The Neotia University

Subject: Control System Lab **Subject Code: PC-RE/P/502**

Experiment No: 03

Name of the experiment: Determination of step responses of 1st and 2nd order system with unity feedback on CRO and calculation of control system specification for verification of control system.

Objective: To determine step responses of 1st and 2nd order system with unity feedback on CRO and to calculate control system specification for verification of control system.

Theory:

Step response of 1st order system:

Let us consider a generic first order transfer function given by

$$H(s) = \frac{b \cdot s + c}{s + a}$$

where, a, b and c are arbitrary real numbers and either b or c (but not both) may be zero. To find the unit step response, we multiply H(s) by 1/s

$$Y_\gamma(s) = \frac{1}{s}H(s) = \frac{1}{s}\frac{b \cdot s + c}{s + a}$$

and take the inverse Laplace transform using Partial Fraction Expansion.

$$\begin{aligned} Y_\gamma(s) &= \frac{A}{s} + \frac{B}{s+a} \\ &= \frac{c}{a}\frac{1}{s} + \frac{ba - c}{a}\frac{1}{s+a} \\ &= \frac{c}{a}\frac{1}{s} + \left(b - \frac{c}{a}\right)\frac{1}{s+a} \end{aligned}$$

So,

$$y_\gamma(t) = \frac{c}{a} + \left(b - \frac{c}{a}\right)e^{-at}, \quad t > 0$$

We now note several features about this equation, namely

$$y_\gamma(0^+) = H(\infty) = b$$

$$y_\gamma(\infty) = H(0) = \frac{c}{a}$$

$$\tau = \frac{1}{a}$$

Thus, we can write the general form of the unit step response as:

$$\begin{aligned} y_\gamma(t) &= y_\gamma(\infty) + \left(y_\gamma(0^+) - y_\gamma(\infty)\right)e^{-\frac{t}{\tau}} \\ &= H(0) + \left(H(\infty) - H(0)\right)e^{-\frac{t}{\tau}} \end{aligned}$$

This last equation is important. It states that if we can determine the initial value of a first order system (at t=0+), the final value and the time constant, that we don't need to actually solve any equations (we can simply write the result). Likewise, if we experimentally determine the initial value, final value and time constant, then we know the transfer function.

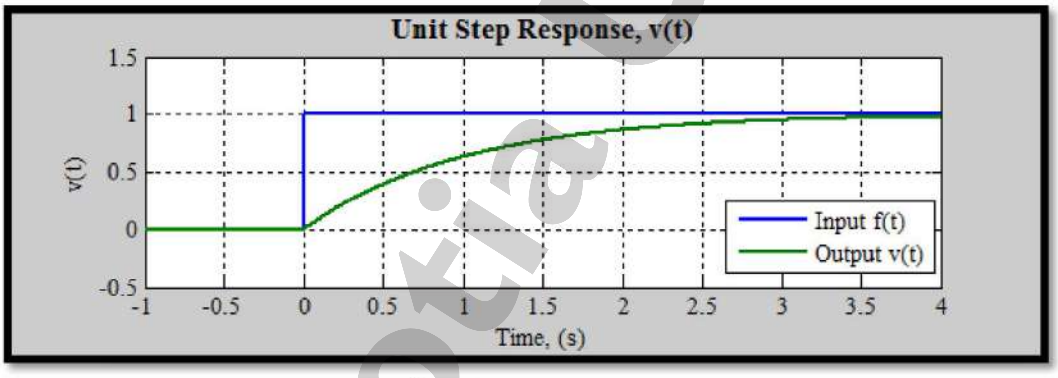


Figure 1: unit step response of 1st order systems

The value of the **unit step response, c(t)** is zero at t = 0 and for all negative values of t. It is gradually increasing from zero value and finally reaches to one in steady state. So, the steady state value depends on the magnitude of the input.

Step response of 2nd order system:

As you would expect, the response of a second order system is more complicated than that of a first order system. Whereas the step response of a first order system could be fully defined by a time constant (determined by pole of transfer function) and initial and final values, the step response of a second order system is, in general, much more complex. As a start, the generic form of a second order transfer function is given by:

$$\frac{Y(s)}{X(s)} = H(s) = \frac{as^2 + bs + c}{s^2 + ds + e}$$

where a, b, c, d and e are arbitrary real numbers and at least one of the numerator terms is non-zero.

Step Response of Prototype Second Order Lowpass System

It is impossible to totally separate the effects of each of the five numbers in the generic transfer function, so let's start with a somewhat simpler case where a=b=0. Then we can rewrite the transfer function as

$$H(s) = \frac{c}{s^2 + ds + e} = K\frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$$

Where, we have introduced three constants,

$\omega_0 = \sqrt{e}$, the natural (or resonant) frequency (rad/sec),

$\zeta = \frac{d}{2\sqrt{e}}$, the damping ration (unitless), and

$K = \frac{c}{e}$, the gain (same units as y/x).

Name	Value of ζ	Roots of s	Characteristics of "s"
Overdamped	ζ>1	$s=-\zeta\omega_0\pm\omega_0\sqrt{\zeta^2-1}$ s=−ζω0±ω0ζ2−1	Two real and negative roots
Critically Damped	ζ=1	s = −ω0−ω0	A single (repeated) negative root
Underdamped	0<ζ<1	$s=-\zeta\omega_0\pm j\omega_0\sqrt{1-\zeta^2}$ s=−ζω0±jω01−ζ2	Complex conjugate ($j = \sqrt{-1}$);
Undamped	ζ=0	s=±jω0s=±jω0	Pure imaginary (no real part)
Exponential Growth	ζ<0	$s=-\zeta\omega_0\pm\omega_0\sqrt{\zeta^2-1}$ s=−ζω0±ω0ζ2−1	Roots may be complex or real, but the real part of s is always positive

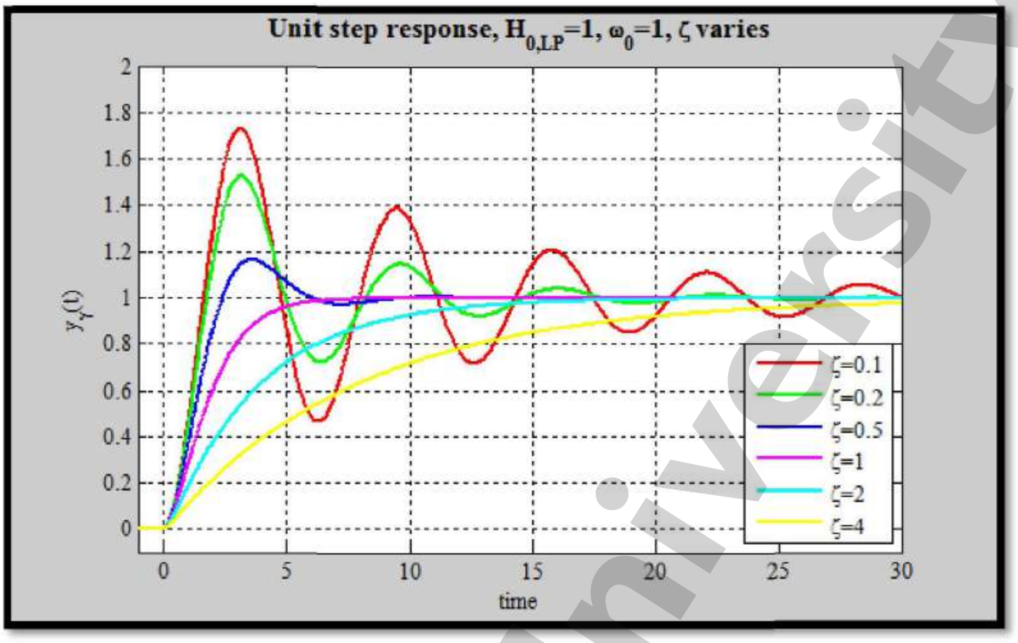


Figure 2: Unit step responses for 2nd order systems

There are number of common terms in transient response characteristics and which are

1. **Delay time** (t_d) is the time required to reach at 50% of its final value by a time response signal during its first cycle of oscillation.

2. **Rise time** (t_r) is the time required to reach at final value by a under damped time response signal during its first cycle of oscillation. If the signal is over damped, then rise time is counted as the time required by the response to rise from 10% to 90% of its final value.

3. **Peak time** (t_p) is simply the time required by response to reach its first peak i.e. the peak of first cycle of oscillation, or first overshoot.

4. **Maximum overshoot** (M_p) is straight way difference between the magnitude of the highest peak of time response and magnitude of its steady state. Maximum overshoot is expressed in term of percentage of steady-state value of the response. As the first peak of response is normally maximum in magnitude, maximum overshoot is simply normalized difference between first peak and steady-state value of a response.

$$Maximum\ \%\ Overshoot = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\%$$

5. **Settling time** (t_s) is the time required for a response to become steady. It is defined as the time required by the response to reach and steady within specified range of 2% to 5% of its final value.

6. **Steady-state error** (e_{ss}) is the difference between actual output and desired output at the infinite range of time.

$$e_{ss} = \lim_{t \to \infty} [r(t) - c(t)]$$

➢ **Tools and apparatus required:**

Sl. No.	Items	Quantity	Maker's name	Range
1.	CRO	1	GWINSTEK	30 MHz
2.	Linear system simulator	1	—	—
3.	Connecting probe	few	—	—

Experimental tables:

Table I: 1st order system

Gain	Overshoot	Rise time (ms)	Peak time (ms)	Steady time (ms)	Delay time (ms)	Steady state error
0						
1						
2						

Table II: 2st order system

Gain	Overshoot	Rise time (ms)	Peak time (ms)	Steady time (ms)	Delay time (ms)	Steady state error
0						
1						
2						

Discussion:

Discuss about your experience and knowledge gathered through this experiment.

The Neotia University

Subject: Control System Lab **Subject Code: PC-RE/P/502**

Experiment No: 04

Name of the experiment: Draw the **Root Locus**of a given system using MATLAB.

Objective: To draw the **Root Locus**using MATLAB for the following transfer functions and comment on the stability.

i) $G(s) = K(s+1)/s(s+2)(s^2+2s+5)$
ii) $G(s) = K/s(s+2)(s^2+4s+8)$

Theory:

Root locus is a graphical method used to find the positions of the roots of the characteristicequations or the poles of closed loop transfer function. By plotting root locus, we can know the behaviour of the closed loop system by varying the parameters and hence, can get the information about the system's stability. Any physical system is represented by a transfer function in the form of

$$G(s) = k \times \frac{numerator\ of\ s}{denomerator\ of\ s}$$

We can find poles and zeros from G(s). The location of poles and zeros are crucial, keeping view stability, relative stability, transient response and error analysis.

Experimental transfer functions and commands to be used:

i) $G(s) = K(s+1)/s(s+2)(s^2+2s+5)$

ii) $G(s) = K/s(s+2)(s^2+4s+8)$

```
num=[];

den=[];

G=tf(num,den);

rlocus(G);
```

Discussion:

Discuss about your experience and knowledge gathered through this experiment.

The Neotia University

Subject: Control System Lab **Subject Code: PC-RE/P/502**

Experiment No.: 05

Name of the experiment: Position control of servo motor

Objective: To control the position of a servo motor

Theory:

Servo motors have been around for a long time and are utilized in many applications. They are small in size but pack a big punch and are very energy-efficient. These features allow them to be used to operate remote-controlled or radio-controlled toy cars, **robots** and airplanes. Servo motors are also used in industrial applications, robotics, in-line manufacturing, pharmaceutics and food services.

How does the little motor work?

The servo circuitry is built right inside the motor unit and has a positional shaft, which usually is fitted with a **gear** (as shown below). The motor is controlled with an electric signal which determines the amount of movement of the shaft.

What is inside the servo?

To fully understand how the servo works, you need to take a look under the hood. Inside there is a pretty simple set-up: a small **DC motor**, **potentiometer**, and a control circuit. The motor is attached by gears to the control wheel. As the motor rotates, the potentiometer's resistance changes, so the control circuit can precisely regulate how much movement there is and in which direction. When the shaft of the motor is at the desired position, **power** supplied to the motor is stopped. If not, the motor is turned in the appropriate direction. The desired position is sent via electrical pulses through the **signal wire**. The motor's speed is proportional to the difference between its actual position and desired position. So if the motor is near the desired position, it will turn slowly, otherwise it will turn fast. This is called **proportional control.** This means the motor will only run as hard as necessary to accomplish the task at hand, a very efficient little guy.

How is the servo controlled?

Servos are controlled by sending an electrical pulse of variable width, or **pulse width modulation** (PWM), through the control wire. There is a minimum pulse, a maximum pulse, and a repetition rate. A servo motor can usually only turn 90° in either direction for a total of 180° movement. The motor's neutral position is defined as the position where the servo has the same amount of potential rotation in the both the clockwise or counter-clockwise direction. The PWM sent to the **motor** determines position of the shaft, and based on the duration of the pulse sent via the control wire; the **rotor** will turn to the desired position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position. Shorter than 1.5ms moves it in the counter clockwise direction toward the 0° position, and any longer than 1.5ms will turn the servo in a clockwise direction toward the 180° position.

When these servos are commanded to move, they will move to the position and hold that position. If an external force pushes against the servo while the servo is holding a position, the servo will resist from moving out of that position. The maximum amount of force the servo can exert is called the **torque rating** of the servo. Servos will not hold their position forever though; the position pulse must be repeated to instruct the servo to stay in position.

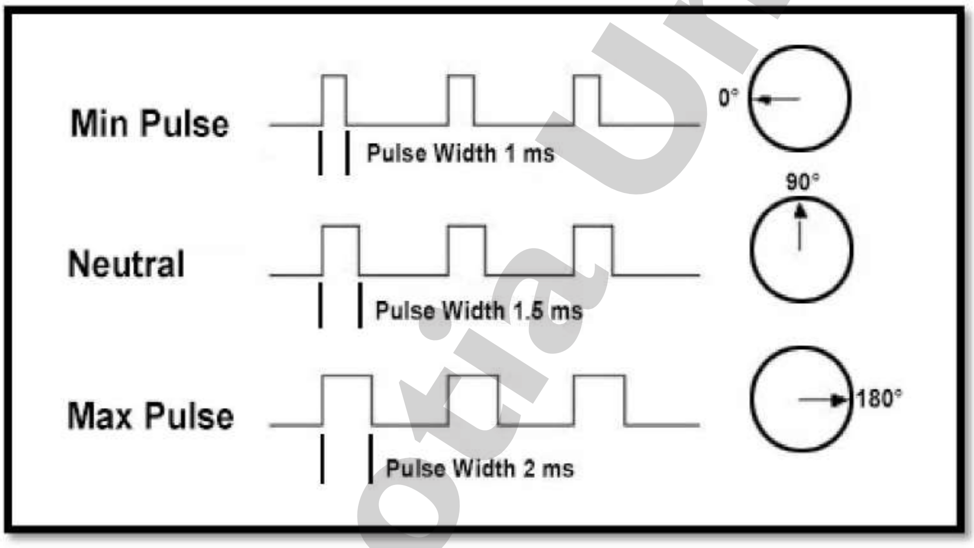


Figure 1: Servo motor control

Types of Servo Motors

There are two types of servo motors - AC and DC. AC servo can handle higher current surges and tend to be used in industrial machinery. **DC servos** are not designed for high current surges and are usually better suited for smaller applications. Generally speaking, DC motors are less expensive than their AC counterparts. These are also servo motors that have been built specifically for continuous rotation, making it an easy way to get your robot moving. They feature two ball bearings on the output shaft for reduced friction and easy access to the rest-point adjustment potentiometer.

Tools and apparatus required:

Sl. no.	Items	Quantity	Maker's name	Range
1	CRO	1	GWINSTEK	30 MHz
2	Servomotor	1	--	--

3	Potentiometer setup	few	--	--

Experimental tables:

Gain = 3

Sl. No	Angle(degree)	Ref.	V_R(v)	V_O(V)	V_E(V)	Peak time(ms)	Rise time(ms)	Settling time	Max. overshoot
1									
2									

Gain = 6

Sl. No	Angle(degree)	Ref.	V_R(v)	V_O(V)	V_E(V)	Peak time(ms)	Rise time(ms)	Settling time	Max. overshoot
1									
2									

Discussion:

Discuss about your experience and knowledge gathered through this experiment.

The Neotia University

Subject: Control System Lab **Subject Code: PC-RE/P/502**

Experiment No: 06

Name of the experiment: Draw the **Bode Diagram** of a given system using MATLAB.

Objective: Draw the **Bode Diagram** (magnitude and phase) using MATLAB for the following transfer functions. Find out the gain margin, phase margin, gain cross over frequency, phase cross over frequency and also comment on the stability of the system.

i) $G(s)H(s)= K(1+0.2s)(1+0.025s)/s^3(1+0.001s)(1+0.005s)$

ii) $G(s)H(s)= 100(1+0.025s)/(1+s)(1+0.1s)(1+0.01s)^2$

Theory:

The Bode plot or the Bode diagram consists of two plots −

- Magnitude plot
- Phase plot

In both the plots, x-axis represents angular frequency (logarithmic scale). Whereas, y-axis represents the magnitude (linear scale) of open loop transfer function in the magnitude plot and the phase angle (linear scale) of the open loop transfer function in the phase plot.

The magnitude of the open loop transfer function in dB is -

$M=20\log|G(j\omega)H(j\omega)|$

The phase angle of the open loop transfer function in degrees is -

$\phi=\angle G(j\omega)H(j\omega)$

<u>Experimental transfer functions and commands to be used:</u>

i) G(s)H(s)= K(1+0.2s)(1+0.025s)/s3(1+0.001s)(1+0.005s)

ii) $G(s)H(s)= 100(1+0.025s)/(1+s)(1+0.1s)(1+0.01s)^2$

```
num=[ ];
den=[ ];
G=tf(num,den);
bode(G);
margin(G);
```

Discussion:

Discuss about your experience and knowledge gathered through this experiment.

The Neotia University

Subject: Control System Lab **Subject Code: PC-RE/P/502**

Experiment no: 07

Name of the experiment: Determination of approximate Transfer Function experimentally using Bode Plot.

Objective: To determine approximate Transfer Function experimentally using Bode Plot.

Theory:

In electrical engineering and control theory, a **Bode plot** is a graph of the frequency response of a system. It is usually a combination of a **Bode magnitude plot**, expressing the magnitude (usually in decibels) of the frequency response, and a **Bode phase plot**, expressing the phase shift.As originally conceived by Hendrik Wade Bode in the 1930s, the plot is an asymptotic approximation of the frequency response, using straight line segments.

Definition:

The Bode plot for a linear, time-invariant system with transfer function H(s) (s being the complex frequency in the Laplace domain) consists of a magnitude plot and a phase plot.

The **Bode magnitude plot** is the graph of the function |H(s=jw)| of frequency w (with j being the imaginary unit). The w-axis of the magnitude plot is logarithmic and the magnitude is given in decibels, i.e., a value for the magnitude |H| is plotted on the axis at 20 log|H|.

The **Bode phase plot** is the graph of the phase, commonly expressed in degrees, of the transfer function arg(H(s=jw)) as a function of w. The **Bode phase plot** is the graph of the phase, commonly expressed in degrees, of the transfer function w-axis as the magnitude plot, but the value for the phase is plotted on a linear vertical axis.

Frequency Response:

This section illustrates that a Bode Plot is a visualization of the frequency response of a system.

Consider a linear, time-invariant system with transfer function H(s). Assume that the system is subject to a sinusoidal input with frequency w,

$$u(t) = \sin(\omega t)$$

that is applied persistently, i.e. from a time -∞ to a time t. The response will be of the form

$$y(t) = y_0 \sin(\omega t + \varphi)$$

i.e. a sinusoidal signal with amplitude y_0 shifted in phase with respect to the input by a phase φ.

It can be shown that the magnitude of the response is

$$y_0 = |H(\mathrm{j}\omega)|$$

and that the phase shift is

$$\varphi = \arg H(\mathrm{j}\omega)$$

Tools and apparatus required:

Sl. No.	Item	Quantity	Range	Maker's name
1	CRO	1	30 MHz	GWINSTEK
2	Signal generator	1	3 MHz	GWINSTEK
3	Linear system simulation	1	--	--
4	probes	few	--	--

Experimental Table:

TABLE-I (First Order System)

Frequency (Hz)	A(V)	B(V)	X0(V)	Y0(V)	Gain(dB)	Phase(deg.)
20						
50						
80						
110						
140						
170						
200						
230						
260						
290						

310						
340						
370						
400						

TABLE-II (Second order System)

Frequency (Hz)	A(V)	B(V)	X0(V)	Y0(V)	Gain(dB)	Phase(deg.)
20						
50						
80						
110						
140						
170						
200						
230						
260						
290						
310						
340						
370						
400						

Discussion:

Discuss about your experience and knowledge gathered through this experiment.

The Neotia University

Subject: Control System Lab **Subject Code: PC-RE/P/502**

Experiment No: 08

Name of the experiment: Draw the **Nyquist plot**of a given system using MATLAB.

Objective: Draw the **Nyquist Plot** using MATLAB for the following transfer functions. Find out the gain margin, phase margin, gain cross over frequency, phase cross over frequency and also comment on the stability (when K=1) of the system. Find the range of K for which the system will be stable.

i) G(s)H(s)= K(s+3)/s(s-1)
ii) G(s)H(s)= K/s(1+s)(1+2s)(1+3s)

Theory:

Nyquist plots are the continuation of polar plots for finding the stability of the closed loop control systems by varying ω from −∞ to ∞. That means, Nyquist plots are used to draw the complete frequency response of the open loop transfer function.

The Nyquist stability criterion works on the principle of argument. It states that if there are P poles and Z zeros are enclosed by the 's' plane closed path, then the corresponding G(s)H(s)G(s)H(s) plane must encircle the origin P−Ztimes. So, we can write the number of encirclements N as,

$$N=P-Z$$

- If the enclosed 's' plane closed path contains only poles, then the direction of the encirclement in the G(s)H(s)G(s)H(s) plane will be opposite to the direction of the enclosed closed path in the 's' plane.

- If the enclosed 's' plane closed path contains only zeros, then the direction of the encirclement in the G(s)H(s)G(s)H(s) plane will be in the same direction as that of the enclosed closed path in the 's' plane.

Experimental transfer functions and commands to be used:

i) G(s)H(s)= K(s+3)/s(s-1)

ii) G(s)H(s)= K/s(1+s)(1+2s)(1+3s)

```
num=[ ];
den=[ ];
G=tf(num,den);
nyquist(G);
```

Discussion:

Discuss about your experience and knowledge gathered through this experiment.

The Neotia University

Subject: Control System Lab **Subject Code: PC-RE/P/502**

Experiment no: 09

Name of the experiment: Determination of PI and PD controller action.

Objective: To determine the action of PI and PD controller.

Theory: A **PID controller** is an instrument used in industrial control applications to regulate temperature, flow, pressure, speed and other process variables. PID (proportional integral derivative) controllers use a control loop feedback mechanism to control process variables and are the most accurate and stable controller.

PID control is a well-established way of driving a system towards a target position or level. It's a practically ubiquitous as a means of controlling temperature and finds application in myriad chemical and scientific processes as well as automation. PID control uses closed-loop control feedback to keep the actual output from a process as close to the target or setpoint output as possible.

P-Controller:

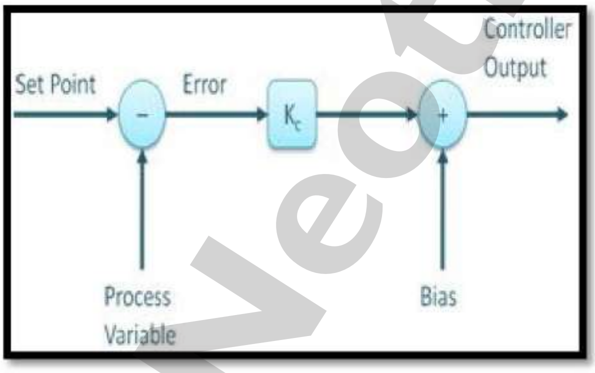


Figure 1: P-controller

Proportional or P- controller gives output which is proportional to current error e (t). It compares desired or set point with actual value or feedback process value. The resulting error is multiplied with proportional constant to get the output. If the error value is zero, then this controller output is zero.

This controller requires biasing or manual reset when used alone. This is because it never reaches the steady state condition. It provides stable operation but always maintains the steady state error. Speed of the response is increased when the proportional constant Kc increases.

I-Controller:

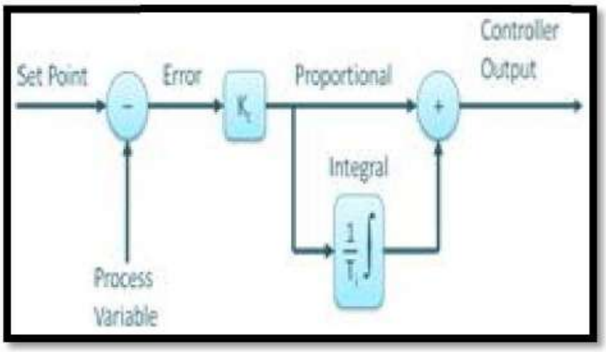


Figure 2: I-controller

Due to limitation of p-controller where there always exists an offset between the process variable and set point, I-controller is needed, which provides necessary action to eliminate the steady state error. It integrates the error over a period of time until error value reaches to zero. It holds the value to final control device at which error becomes zero.

Integral control decreases its output when negative error takes place. It limits the speed of response and affects stability of the system. Speed of the response is increased by decreasing integral gain Ki.

While using the PI controller, I-controller output is limited to somewhat range to overcome the integral wind up conditions where integral output goes on increasing even at zero error state, due to nonlinearities in the plant.

D-Controller:

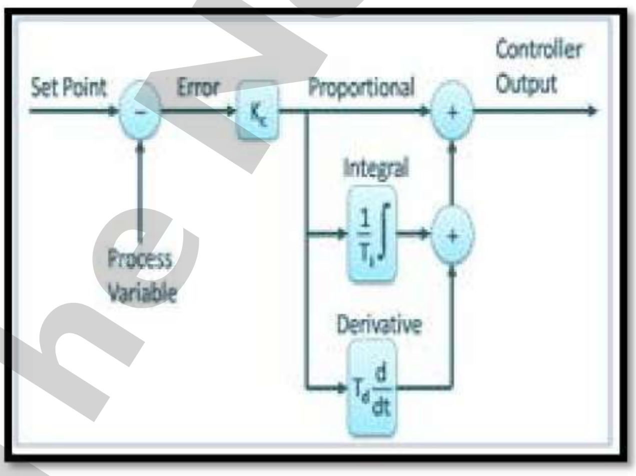


Figure 3: D (in PID)-controller

I-controller doesn't have the capability to predict the future behaviour of error. So, it reacts normally once the set point is changed. D-controller overcomes this problem by anticipating future behaviour of the error. Its output depends on rate of change of error with respect to time, multiplied by derivative constant. It gives the kick start for the output thereby increasing system response.

Working principle of PID controller:

PID controllers are found in a wide range of applications for industrial process control. Approximately 95% of the closed loop operations of industrial automation sector use PID controllers. PID stands for Proportional-Integral-Derivative. These three controllers are combined in such a way that it produces a control signal.

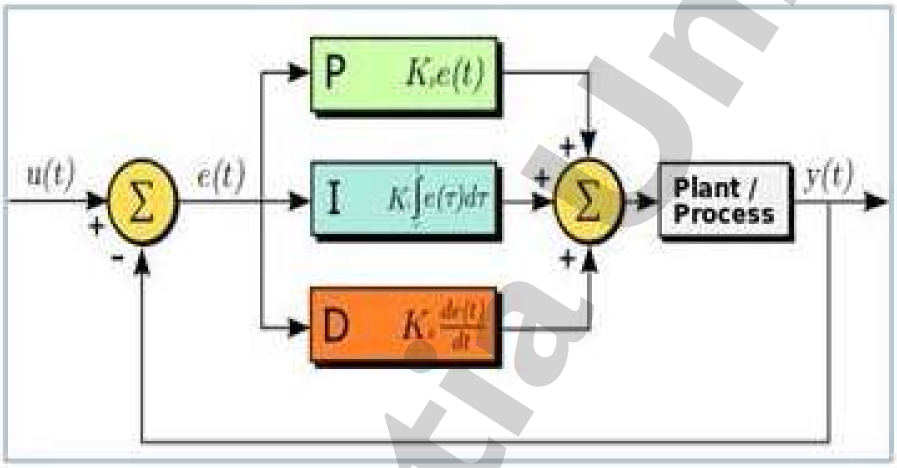


Figure 4: PID controller

As a feedback controller, it delivers the control output at desired levels. Before microprocessors were invented, PID control was implemented by the analog electronic components. But today all PID controllers are processed by the microprocessors. Programmable logic controllers also have the inbuilt PID controller instructions. Due to the flexibility and reliability of the PID controllers, these are traditionally used in process control applications.

Tools and apparatus required:

Sl. no.	Item	Quantity	Range	Maker's name
1	PID Controller	1	--	Techno Instruments
2	CRO	1	30MHz	GWINSTEK
3	Probes	Few	--	--

Experimental Table:

PI Controller

	K_P	K_I	V0
Output 1			
Output 2			

PD Controller

	K_P	K_D	K_D
Output 1			
Output 2			

Discussion:

Discuss about your experience and knowledge gathered through this experiment.

The Neotia University

Subject: Control System Lab **Subject Code: PC-RE/P/502**

Experiment no:10

Name of the experiment: Determination of PID controller action.

Objective: To determine the action of PID controller.

Theory: A **PID controller** is an instrument used in industrial control applications to regulate temperature, flow, pressure, speed and other process variables. PID (proportional integral derivative) controllers use a control loop feedback mechanism to control process variables and are the most accurate and stable controller.

ID control is a well-established way of driving a system towards a target position or level. It's a practically ubiquitous as a means of controlling temperature and finds application in myriad chemical and scientific processes as well as automation. PID control uses closed-loop control feedback to keep the actual output from a process as close to the target or setpoint output as possible.

P-Controller:

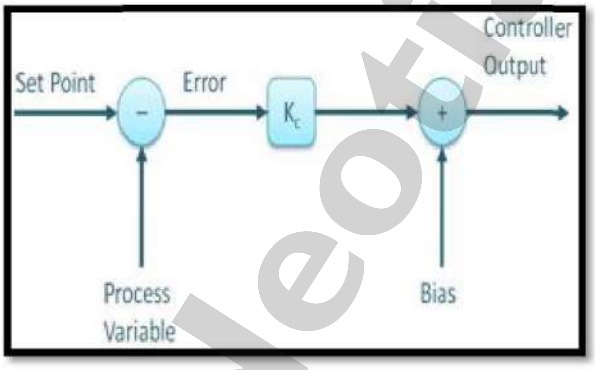


Figure 1: P-controller

Proportional or P- controller gives output which is proportional to current error e (t). It compares desired or set point with actual value or feedback process value. The resulting error is multiplied with proportional constant to get the output. If the error value is zero, then this controller output is zero.

This controller requires biasing or manual reset when used alone. This is because it never reaches the steady state condition. It provides stable operation but always maintains the steady state error. Speed of the response is increased when the proportional constant Kc increases.

I-Controller:

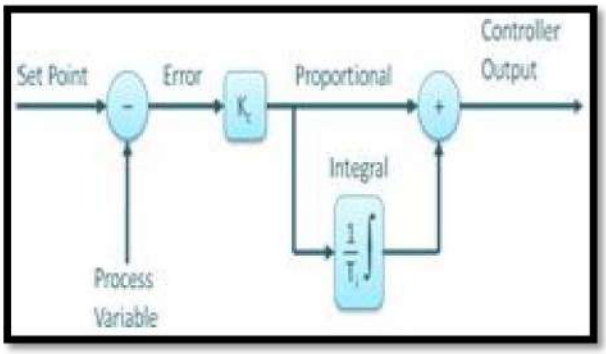


Figure 2: I-controller

Due to limitation of p-controller where there always exists an offset between the process variable and set point, I-controller is needed, which provides necessary action to eliminate the steady state error. It integrates the error over a period of time until error value reaches to zero. It holds the value to final control device at which error becomes zero.

Integral control decreases its output when negative error takes place. It limits the speed of response and affects stability of the system. Speed of the response is increased by decreasing integral gain Ki. While using the PI controller, I-controller output is limited to somewhat range to overcome the integral wind up conditions where integral output goes on increasing even at zero error state, due to nonlinearities in the plant.

D-Controller:

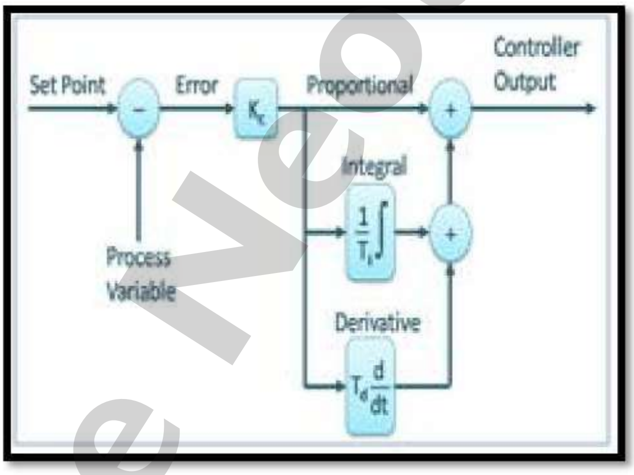


Figure 3: D-controller

I-controller doesn't have the capability to predict the future behaviour of error. So it reacts normally once the set point is changed. D-controller overcomes this problem by anticipating futurebehaviour of the error. Its output depends on rate of change of error with respect to time, multiplied by derivative constant. It gives the kick start for the output thereby increasing system response.

Working principle of PID controller:

PID controllers are found in a wide range of applications for industrial process control. Approximately 95% of the closed loop operations of industrial automation sector use PID controllers. PID stands for Proportional-Integral-Derivative. These three controllers are combined in such a way that it produces a control signal.

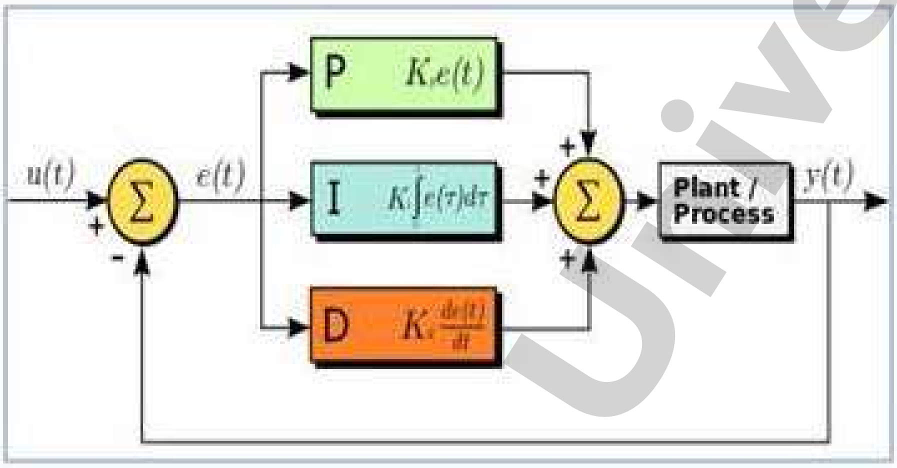


Figure 4: PID controller

As a feedback controller, it delivers the control output at desired levels. Before microprocessors were invented, PID control was implemented by the analog electronic components. But today all PID controllers are processed by the microprocessors. Programmable logic controllers also have the inbuilt PID controller instructions. Due to the flexibility and reliability of the PID controllers, these are traditionally used in process control applications.

Tools and apparatus required:

Sl. no.	Item	Quantity	Range	Maker's name
1	PID Controller	1	--	Techno Instruments
2	CRO	1	30 MHz	GWINSTEK
3	Probes	Few	--	--

Experimental Table:

	K_P	K_I	K_D
Output 1			
Output 2			

Discussion:

Discuss about your experience and knowledge gathered through this experiment.