

Ahsanullah University of Science and Technology
Department of Electrical and Electronic Engineering

LABORATORY MANUAL
FOR
ELECTRICAL AND ELECTRONIC SESSIONAL COURSES

Student Name :
Student ID :

Course no : EEE 4106
Course Title: Control System-I Laboratory

For the students of
Department of Electrical and Electronic Engineering
4th Year, 1st Semester

Experiment No: 1

Name of the Experiment: Study of the Unit Step Response of a Second Order System Simulated on a PC Using the 'MATLAB' Software

Objectives:

- 1) Introduction to Second Order System.
- 2) To determine the response $c(t)$ of a system for a unit step input $r(t) = 1.0$ and display the same in time domain graphically.
- 3) To determine various criteria of time-response such is undamped natural frequency ω_n , damping ratio ζ , frequency of damped oscillation ω_d , peak overshoot c_p , time to reach the peak overshoot t_p , per unit overshoot $M_0 = (c_p - c_{ss}) / c_{ss}$, first time to reach the steady state value t_0 response settling time t_s and envelop settling time T_s .
- 4) To observe the effects of varying system parameters or gain up to the response.

Second Order Systems :

In the speed control system, the plant was characterized by its time constant, which is determined by the inertia of the rotor and the viscous friction. This arrangement can be described mathematically by a first order differential equation.

The position control system has an integration effect between velocity and position. This makes the position control servomechanism into a second order system.

As a first order system is characterized by its time constant, it could be expected that a second order system would be characterized by two time constants. Although some second order systems can be described in this way, most of the systems dealt with in closed loop control cannot be described so simply.

Following Fig shows how the step response changes as the proportional gain is increased in a typical second order system. It is obvious that, as the gain increased, the position control system became more and more oscillatory.

When the gain is low, the response is sluggish and is said to be Over damped. An over damped response is characterized by two separate time constants.

Curve b shows the fastest response this system can have without any oscillation. This response is said to be Critically Damped. A critically damped response is characterized by two time constants both of the same value.

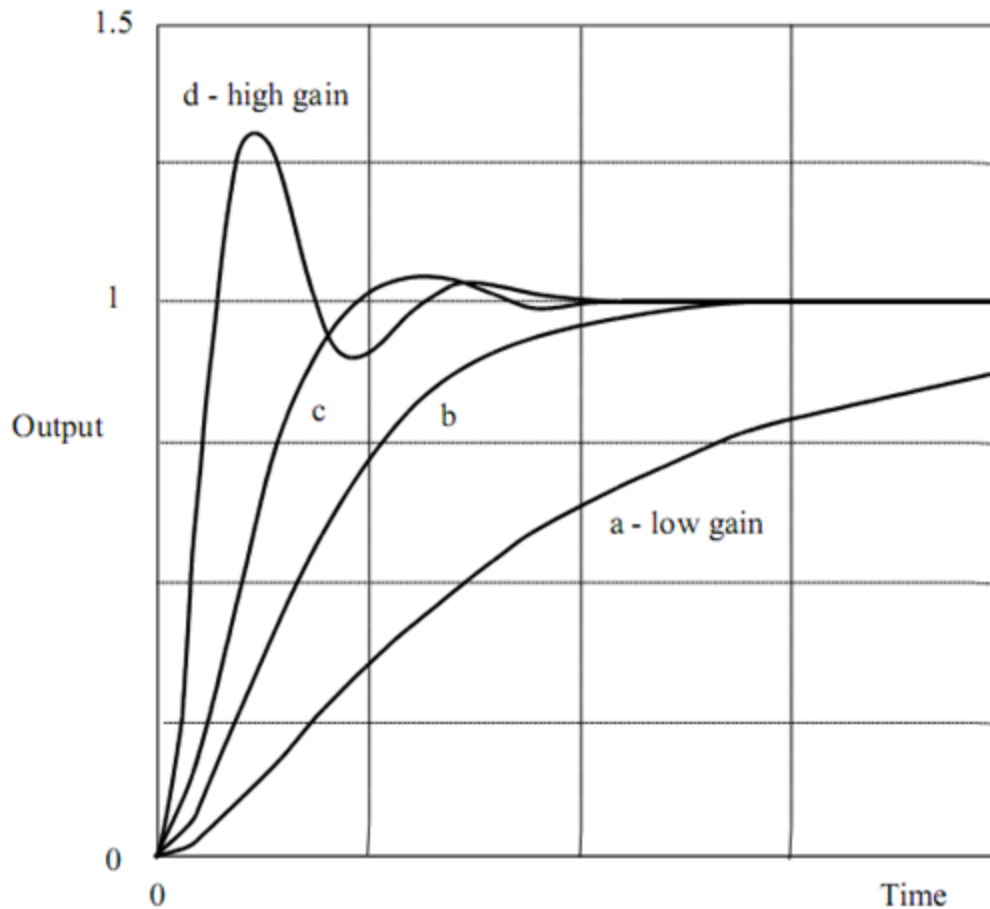


Figure 1.1 : Response of a second order system with different gain.

With higher gain, the response overshoots and oscillates. This type of response is said to be Under damped. An under damped response cannot be characterized by time constants. Mathematically it is described by a decaying sinusoid.

Examining the above Fig, the "best" response would appear to be somewhere between curves b and c. Before we can predict the gain necessary to give a specified response, we need to know how to describe the behavior of a second order system.

Under damped systems are often described by the amount the response overshoots and by the frequency at which it oscillates.

There are two other parameters used to describe second order systems - Damping Factor and Natural Frequency of Oscillation.

To be able to predict the overshoot and frequency of oscillation of a closed loop system, we must develop how the different parameters are related to the gain and time constant of the plant being controlled.

Overshoot and Damping Factor :

Overshoot

Overshoot is the amount by which a response goes beyond the steady state value before settling down.

Overshoot can be measured from the step response. It is the ratio :

$$\frac{\text{Peak Output Change} - \text{Steady State Output Change}}{\text{Steady State Output Change}}$$

Overshoot is usually stated as a percentage, which is the above ratio multiplied by 100.

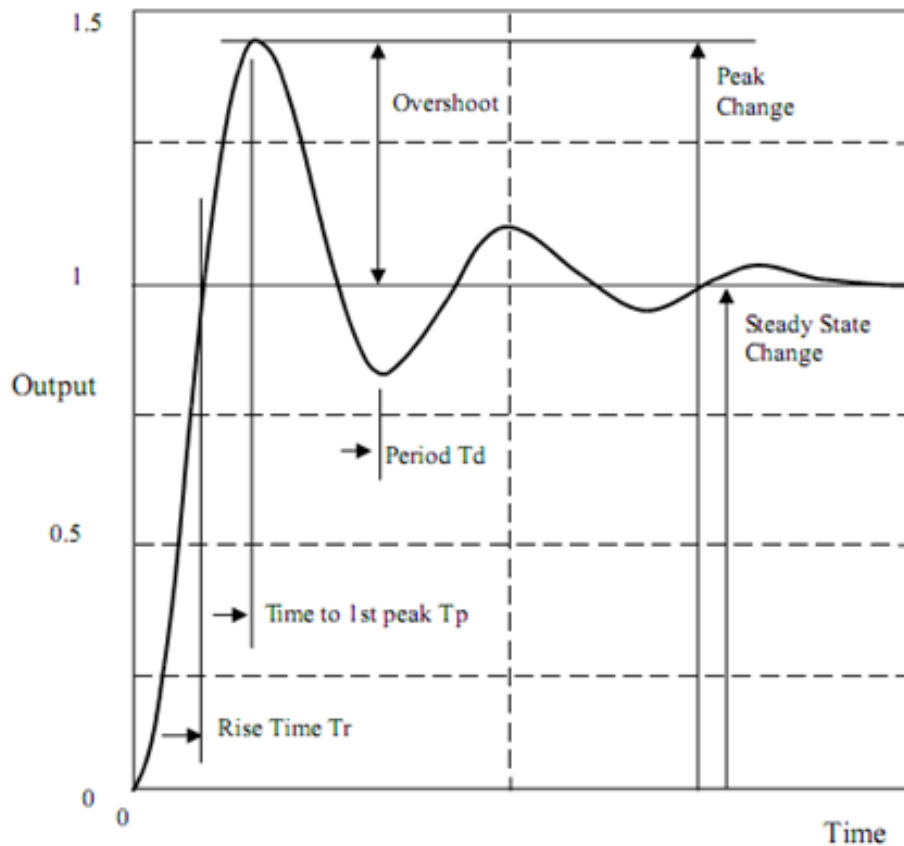


Figure 1.2: Parameters to describe a second order behavior.

Damping Factor

In the equations describing system behavior, overshoot is not an easy parameter to handle. Another parameter called Damping Factor is used and this gives an indication of the amount of overshoot in a system. Damping Factor has the symbol ζ (Zeta). ζ has a value of 1 when the system is critically damped, less than 1 when underdamped and greater than 1 when overdamped.

$$\begin{aligned}\zeta < 1 & \quad \text{Underdamped – decaying oscillations} \\ \zeta = 1 & \quad \text{Critically damped – just no overshoot} \\ \zeta > 1 & \quad \text{Overdamped – system sluggish}\end{aligned}$$

The objective of a control system design is often to achieve a fast response without any overshoot or with just a little overshoot. Systems are usually designed for ζ in the range 0.7 to 1.

The amount of overshoot is wholly dependent on the Damping Factor. Measuring overshoot allows the damping factor to be calculated and knowing ζ allows the overshoot to be calculated. They are linked by the equations:

$$\zeta = \frac{1}{\sqrt{1 + \left[\frac{\pi}{\ln(\text{overshoot ratio})} \right]^2}}$$

$$\text{Overshoot ratio} = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \Rightarrow \% \text{ Overshoot ratio} = 100 \times e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}}$$

Damped and Natural Frequencies of Oscillation

Damped Frequency

The frequency at which an underdamped system oscillates is called the Damped Frequency, ω_d . This can be determined by measuring the time between successive positive peaks if, as shown in Fig 2.2, there is more than one cycle.

The inverse of the period of a cycle is its frequency in Hertz:

$$f_d = \frac{1}{T_d} \Rightarrow \omega_d = 2\pi f_d = \frac{2\pi}{T_d}, \text{ where } T_d \text{ is the period of the oscillation.}$$

The time to the first peak, T_p , is half the period. The damped frequency can then be found by measuring the time to the first peak:

$$\omega_d = \frac{\pi}{T_d}$$

Natural Frequency

If there was no damping at all ($\zeta = 0$), the system would continuously oscillate at a frequency which is called the Natural Frequency of the system. This is given the symbol ω_n . The relationship between Natural and Damped frequencies is:

$$\omega_n = \frac{\omega_d}{\sqrt{1 - \zeta^2}}$$
$$\omega_d = \omega_n \sqrt{1 - \zeta^2}$$

It can be seen from Fig 2.1 that ω_d increases as the gain increases, which means that increasing the gain makes the system work faster but at the expense of increasing the overshoot.

Simulation:

Firstly, The MATLAB commands used here can compute step response for a system as in Fig (a) below. However, for a closed loop system as in Fig.(b) the overall transfer function has to be expressed in the form of a single transfer function $\frac{G(s)}{1 + G(s)H(s)}$ by the user using *loop* command (for unity feed back system) or *feedback* command.

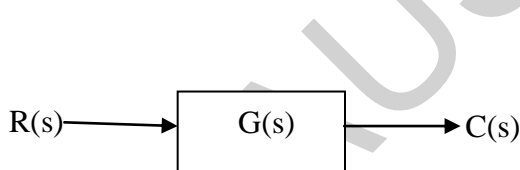


Figure 1.3 (a): Open Loop System

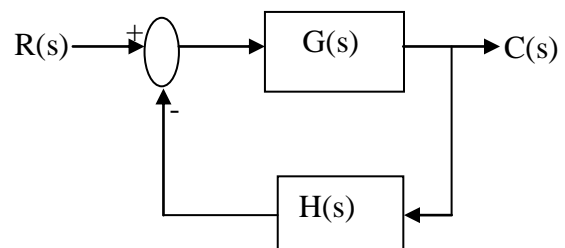


Figure 1.3 (b) : Close Loop System

Procedure:

1. Run the MATLAB package. Note that “»” is the prompt displayed automatically for the MATLAB user.
2. Enter each of the numerator and denominator factors of chosen system’s transfer function as follows
 - »num 1 = [coefficient of sⁿ coefficient of sⁿ⁻¹..... coefficient of s⁰ constant term];
 - »num2 = [similarly];

»den 1 = [similarly];

»den 2 = [similarly];

and so on.

In the numerator is only a constant value $c \neq 0.0$ then use:

»num=c

If the transfer function has more than one numerator or denominator factors, multiply them as follows to obtain a single numerator and denominator factor respectively. As

for example: if $TF = \frac{(num1)(num2)}{(den1)(den2)(den3)}$ then,

»num = conv(num1, num2)

»den4 = conv(den1, den2)

»den = conv(den3, den4)

If the number of such factors is less in number the multiplication can be done manually to obtain a single numerator or denominator factor.

If the system is a unity feedback system, then to find out the close loop transfer function we use

»[num,den]=cloop(numg,deng)

And for any feed back system we use

»[num,den]=feedback(numg,deng,numh,denh,+/-1)

Where $G(s) = \frac{numg}{deng}$, $H(s) = \frac{numh}{denh}$ and +/-1 represent positive/negative feedback.

3. Find undamped natural frequency ω_n and damping ratio ζ for the defined system using the command as follows:

»[wn,z] = damp(den)

Here. wn stores the value of ω_n and z stores ζ .

4. Define the time axis for $c(t)$ vs. t from 0 to 30 seconds at an interval of 0.2 seconds as follows:

»t = 0:0.2:30;

5. To compute $e(t)$ at the defined times use the command:

»y = step(num,den,t)

where the vector stores the response values.

6. Plot and display on the screen $c(t)$ vs. t using the command:

»plot (t,y)

The plot can be made to have grid boxes on it as follows:

»grid

The plot can also be displayed along with labels using the following commands:

» title ('Unit Step Response'), ylabel('c(t)', xlabel('Time in seconds')

7. To highlight a certain portion of the displayed plot define the portion on the x and y axes as follows and then display.

»A = [X_{min}, X_{max}, Y_{min}, Y_{max}]

»axis (A)

To restore back the original full plot just use the command:

»axis auto.

Warning: You must request your teacher to be with you before issuing the print command of the next step (8)

8. After plotting the step response for the chosen system, get its hard copy by selecting 'Print' command from the 'File' of the Figure window.
9. Estimate from each system's response curve and calculations: the frequency of damped oscillation $\omega_d = \omega_n \sqrt{1 - \zeta^2}$, peak overshoot c_p , time to reach the peak overshoot t_p , per unit overshoot $M_0 = (c_p - c_{ss}) / c_{ss}$, first time to reach the steady state value t_0 , response settling time t_s and envelope settling time T_s for a tolerance margin of 3% and 5%

$$T_s = \frac{\text{number of time constants}}{\zeta\omega_n} \approx \frac{4}{\zeta\omega} (2\% \text{ error}) = \frac{3}{\zeta\omega} (5\% \text{ error})$$

Feedback Connection:

M = feedback(M1,M2) computes a closed-loop model M for the feedback loop.for the following system

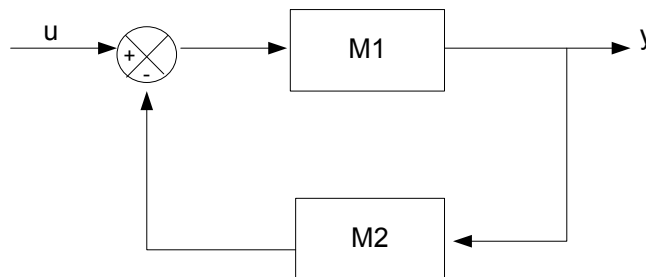


Figure 1.3 (c) : Close Loop System

Negative feedback is assumed and the model M maps u to y. To apply positive feedback, use the syntax $M = \text{feedback}(M1, M2, +1)$.

$[\text{Wn}, \text{Z}] = \text{damp}(M)$ returns vectors Wn and Z containing the natural frequencies and damping factors of the linear system M.

Report:

1) Record the response curve. ω_n , ζ , ω_d , c_p , t_p , m_0 , t_0 , t_s (2% and 5% tolerance), and T_s (2% and 5% tolerance) for the systems having the following transfer functions:

a) $G(s) = \frac{1}{s^2 + b_1s + b_0}$; $H(s) = 0$ i.e. open loop system with $b_1=0.4$, $b_0=1$

b) $G(s) = \frac{1}{s^2 + b_1s + b_0}$; $H(s) = 1$ with $b_1=0.4$, $b_0=1$

c) $G(s) = \frac{K_1}{s(1+s)(1+0.2s)}$; $K_1 = 0.83$, $H(s) = 1$

2) Record the same for the system in 1(b) for various cases with $b_1=0.2$, 0.6 and 1.0 respectively. Comment on the effects of variation of the parameter b_1 which represents the damping coefficient for a second order system.

3) Record the same for the system in 1(c) for various cases with $K_1=0.2$, 0.4 and 5.5 respectively. Comment on the effects of variation of the forward gain K_1 .

Experiment No: 2

Name of the Experiment: Determination of Time Response of a PC based DC motor

Objectives:

Having completed this experiment, students will be able to:

- Measure the parameters of a plant using step tests
- State the time model of the DC motor

Introduction:

We wish to determine a model which describes the time behavior of the plant using the 'Black Box' approach. To do this, you will ask the motor to change speed and infer the relationship between input voltage and output speed from the way in which the motor responds. You will be measuring the Step Response of the motor.

There are two parts to any output time response when there is a change in input:

- A Transient period which occurs immediately the input changes and during which the system seems to be dominated by something other than the input.
- A Steady State condition which is reached after the transient has died out. The system seems to have settled down to the influence of the input.

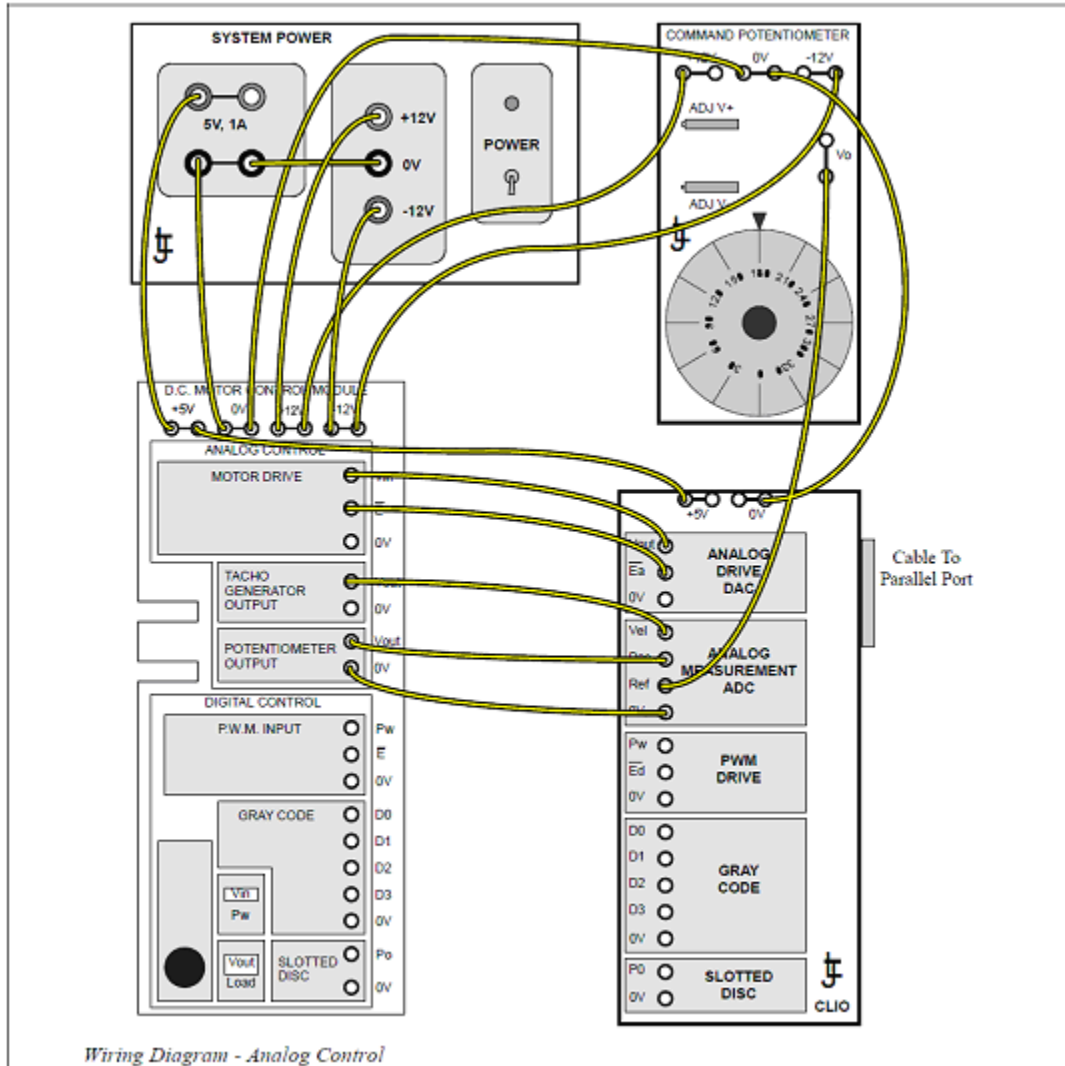
The transient situation is produced by elements within the plant which cannot respond instantly. Mass in a mechanical system and capacitance in an electrical system both store energy so it takes time to change the velocity of a mass or to change the voltage across a capacitor.

In the DC motor, it is the mass of the motor armature and all the disks and dials connected to the motor shaft which require energy to get them moving or stop them moving. Actually it is the inertia of these elements, not mass, since we are dealing with rotating bodies.

The two parameters that define the model are Gain and Time Constant. Gain (K) is the Steady State relationship between input and output. Time Constant (τ) defines the Transient Time.

Procedures:

1. Make the Hardware connection as shown in the following figure:



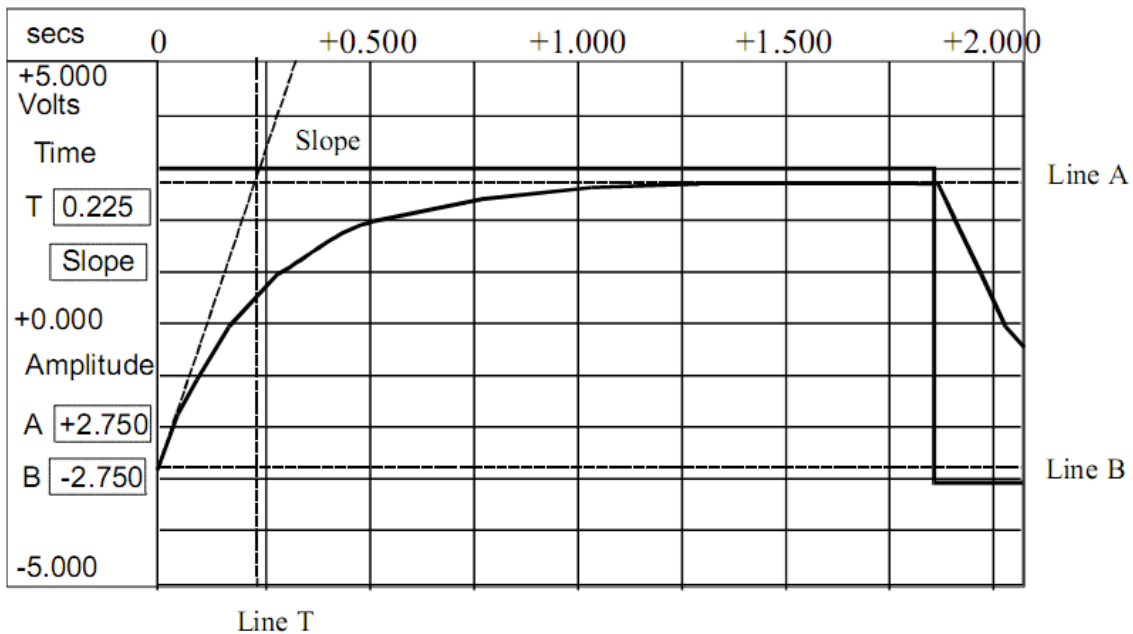
2. Start VCL and load CA06PE03. Make sure that following conditions are fulfilled.

File	Controller	Plant	Display
CA06PE03	Open-loop	MS15 Analog	Graph
Signal Generator		Graph	
Signal	Step	1 Input	ON
Level	60%	2 Position	OFF
Offset	0%	4 Velocity	ON
Rate	20 msec		
Reference	Internal		
DC Motor		Output Potentiometer	Disengage
Brake	0	Command Potentiometer	180°

3. Disengage the output potentiometer then switch power ON and Enable the motor. The output velocity trace (purple) on the PC shows what is called the Step Response (what happens when there is a step change in the input).

4. The purple trace is the Step Response of motor speed. Observe that the speed does become constant after a time but initially lags behind the input. Expand the time scale by decreasing the rate to 10msec and click the x2 time multiplier. Click Freeze | Freeze. This freezes the display at the end of the current cycle. The 'Frozen' control box appears when the cycle ends. The motor can now be disabled and measurements made from the screen.

You are going to measure the Gain and Time Constant which characterize the motor. The following figure overleaf shows the measurements to be made. The measurement facility is activated from the Frozen control box by clicking Time ON. The measurement lines and value boxes appear on the graph.



5. Measuring Steady State Response :

The steady state relationship between input and output is characterized by the Gain of the plant. Gain, or Magnitude Ratio or Amplitude Ratio, is the ratio between input and output when they have reached a steady state. The spans have been measured when the output has reached a steady state so:

$$Gain = \frac{Output\ Span}{Input\ Span}$$

Input Span

Input span is the amount by which the input changes.

- Select channel 1/Input/Dark Blue. The scale will show the input channel scale.

- Select Line A by clicking within the A box. The box and the line will change color.
- Move the mouse until the pointer is pointing at the upper dark blue trace in the graph area. Click the left button and line A will move to where you are pointing. You can click again if you did not position the line exactly the first time. The A box indicates the level of the line.
- Click in the B box and, in the same way, position line B over the lower part of the dark blue trace.

The difference between A and B is the Input Span.

Output Span

Output span is the amount by which the output changes in response to the input changes.

Change to channel 4/Velocity/purple and repeat the measurements on the purple trace. Line B should be positioned where the trace can be seen starting at the left of the graph.

6. Measuring Transient Response :

There are a number of ways to characterize the transient response. These come under the general heading of 'Rise Time' but there are many different definitions of Rise Time. You will measure three different times then we will see how these are related.

Initial Slope Method

- Make sure that lines A and B are the final and initial values of trace 4 respectively.
- Click in the Slope box. The line from the beginning of the transient sloping up to the right has changed to blue. This allows you to measure the initial slope of the velocity trace. The slope of the line can be changed by clicking in the graph area. The top of the line will move to the time at which you clicked.
- Move the slope line until its slope is the same as that of the initial part of the transient, such that the blue line covers the initial part of the purple velocity trace line.
- Click in the Time box. The vertical time line is highlighted.
- Click where the slope line crosses line A. The time shown is the Time Constant measured by the initial slope method.

Time Constant $t_1 = \text{_____}$ seconds.

Settling time method

The time constant can also be calculated from the time it takes the transient to reach the final value.

- Move the Time line to the time at which the velocity trace first reaches its final value (when the purple trace reaches line A).

The time shown is 5 time constants from the start of the transient .

Time Constant $t_2 =$ _____ seconds.

63% Method

Another time measurement is the time it takes for the transient to change by 63%.

- Click the A box to highlight Line A and move it to the 63% level. You may not be able to set the line exactly owing to the screen resolution. Expanding the scale using the Magnify and Shift controls may help. The traces require to be redrawn using Freeze | Redraw option after Magnify or Shift are changes.

- Now click the Time box and move the time line to the time at which the velocity trace reaches its 63% level. The time shown is the Time Constant measured by the 63% method.

Time Constant $t_3 =$ _____seconds.

Note: Experience has shown us that the 63% measurement is more accurate than the other two techniques so use t_3 as the time constant in your model.

7. The total Time response can be expressed mathematically as,

$$\text{Change in Output} = \text{Change in Input} \times \text{Gain} \times \left[1 - e^{-t/\tau}\right]$$

or,

Change in Output

$$= (\text{Change in Input} \times \text{Gain}) - (\text{Change in Input} \times \text{Gain} \times e^{-t/\tau})$$

Change in Output = Steady State Response – Transient Response

Report:

1. Draw the output response along with the step input as found in this experiment using graph paper.
 - a. Calculate gain by showing input & output Span clearly.
 - b. Calculate time constant using three different methods.
2. A plant has a gain of 0.8 and time constant of 3 seconds. Using the normalized sketch of a step response, determine the output response (in volts) to a 2 volt step input after 1.5 seconds.

Experiment No: 3

Name of the Experiment: Speed control of a DC servo system

Basic Theory:

A simplified diagram of a closed loop constant motor speed control system is shown in Figure 3-1. As the reference or control voltage is applied to the input of the comparator, and the Tacho generator produces a signal which is equivalent to the speed of the motor, the two, signals are compared at the input of the summing amplifier through addition of two signals with opposite polarity. The output of the comparator is, then, an error signal which represents the difference between the preset and actual speed. Because the error signal is out of phase to the reference signal, this signal compensates the motor speed in the Direction to achieve a constant speed.

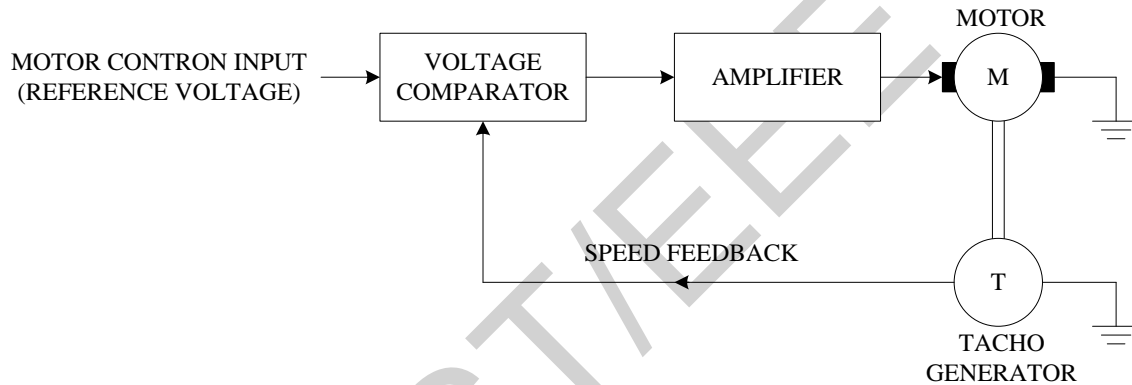


Figure 3.1 : Basic constant speed feedback loop system.

In general, the speed of a motor and the error signal, have the following relationship.

$$\theta_0 = KE \quad (3-1)$$

Where, θ_0 = the motor speed

E = error signal

K = system gain

The error signal is defined as:

$$E = V_{\text{ref}} - K_g \theta_0 \quad (3-2)$$

Where, V_{ref} = reference voltage

$K_g \theta_0$ = output of the Tacho generator

Replacing E in (3-1) with (3-2) yields;

$$\theta_0 = K(V_{\text{ref}} - K_g \theta_0) \quad (3-3)$$

$$\theta_0 = K \cdot V_{\text{ref}} - K \cdot K_g \theta_0$$

$$1 = \frac{K.V_{ref}}{\theta_0} - K.K_g$$

$$\frac{K.V_{ref}}{\theta_0} = 1 + K.K_g$$

$$\theta_0 = \frac{K.V_{ref}}{1 + K.K_g} \quad (3-4)$$

In case the K is very large in forward direction, Equation (3-4) is reduced to:

$$\theta_0 = \frac{V_{ref}}{K_g} \quad (3-5)$$

From equation (3-5), it's clear that for a given Tacho generator constant K_g , the motor speed is linearly proportional to V_{ref} only and is not dependent on the deviation of the system gain. This is the most beneficial advantage of a closed loop motor speed control system.

Similar relationships can be developed for the error signal in a closed loop system. Replacing θ_0 in (3-2) with (3-5).

$$E = V_{ref} - K_g.K.E \quad (3-6)$$

$$1 = \frac{V_{ref}}{K_g} - K_g.K$$

$$\frac{V_{ref}}{E} = 1 + K.K_g$$

$$E = \frac{V_{ref}}{1 + K.K_g} \quad (3-7)$$

Equation (3-7) indicates that the error voltage E can be reduced when the gain K is increased.

In a practical system, maintaining a high system gain means reduction of the dead band, as well as desensitizing motor speed to the load changes. Although large system gain is desired in general, the gain should be limited to an acceptable level. When the gain is beyond the acceptable level, the transient characteristics of the system will suffer, and it will cause irregular motor rotation.

The relationships between load, error and motor speed are shown in Figure 3-2 at two different system gain levels.

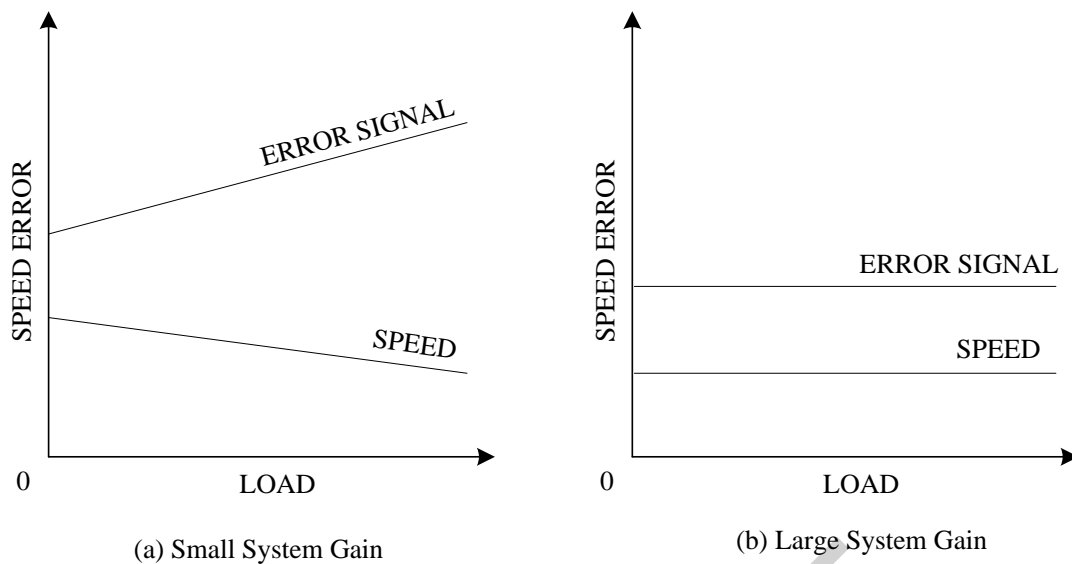


Figure 3.2 : Relationships between load, error and motor speed.

In the equivalent system diagram of Figure 3-3, the output of the Frequency-to-Voltage converter U-155 should be large enough to provide sufficient feedback signal. Otherwise, the motor will not run at constant speed. Also, when the gain of the amplifier U-153 is low, the system response will be slow and the “dead band” effect will get worse. However, in case the gain is too high, the system will become unstable.

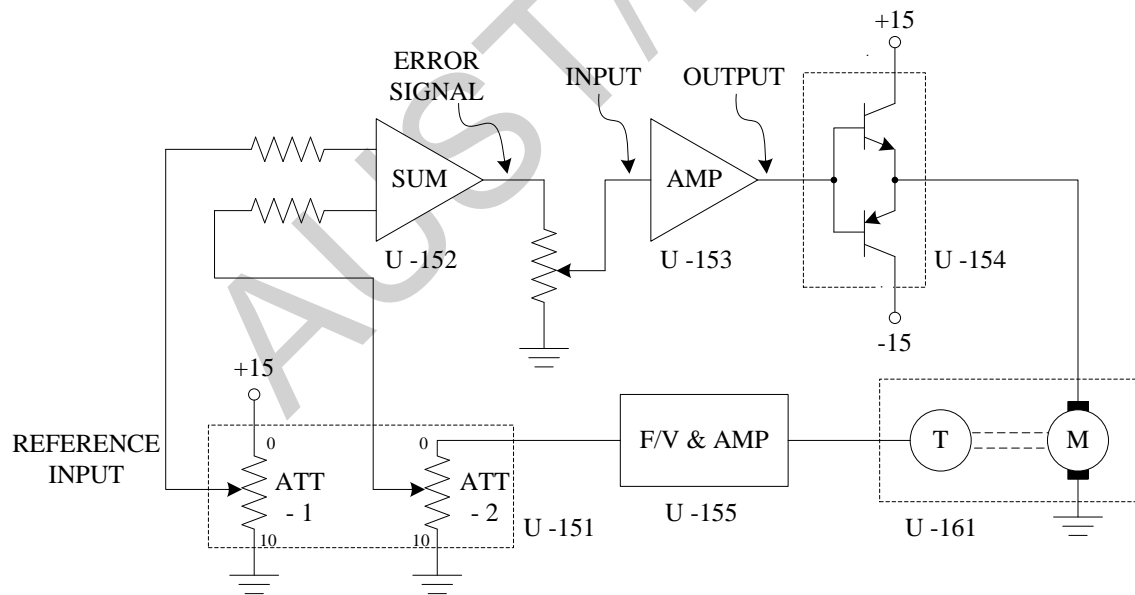


Figure 3.3 : Equivalent system diagram of the experiment.

Procedures:

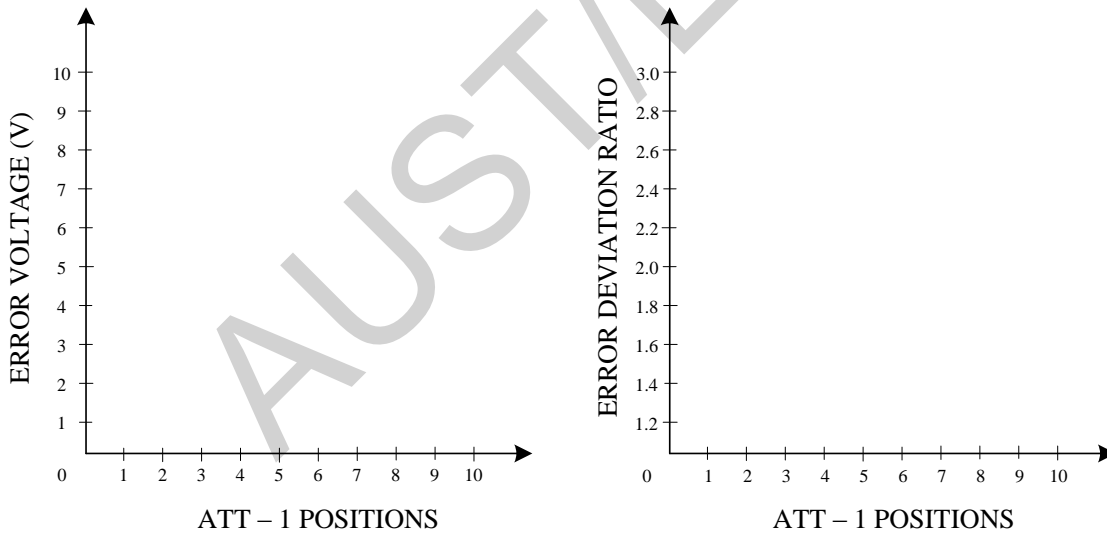
1. Referring to Figure 3-4, arrange all the modules and an oscilloscope and connect them together.
2. Set the selector switch of U- 152 to “a”.
3. Set ATT-1 of U-151 to “9” and ATT’-2 to “10”. This will minimize the reference setting, and the feedback will be almost zero.

4. Turn the power of U-156 on. Adjust U-157 to approximately one half of the maximum motor speed (2500 RPM).
5. Attach the disk brake to the high speed shaft of the servo motor, and set the brake to “0”. Raise the brake setting by one increment, and each time, press the brake button and measure the motor speed and the associated error signal.
6. Set the U-151 ATT-2 to “5”. Adjust the motor speed to 2500 RPM, and repeat Step 5 Plot the data obtained in Figure 3.5 (a).

Notes: The same motor speed can be obtained by increasing the reference signal level and decreasing the amplifier gain. However, this method will reduce the amount of feedback control signal and thus decrease the over-all ability to control the system.

7. Using U-157 set the motor speed to 2500 RPM. Set U-151 ATT-2 to “5”. Adjust ATT-1 from “0” to “9”, and measure the error voltage at each point.
8. For each point of ATT-1 setting, hold the high speed motor shaft by hand and repeat the experiments in Step 7. Compute the error deviation ratio as defined by the following equation and plot the results in Figure 3-4.

Note:
$$\text{Error Deviation Ratio} = \frac{\text{error measured with motor stalled}}{\text{error measured with motor running}}$$



(a) Error Voltage vs. ATT – 1 Setting

(b) Error Deviation Ratio vs. ATT – 1 Setting

(The higher ATT – 1 setting means the lower system gain)

Figure 3.4 : System Gain Vs. Error Voltage Characteristics.

Report:

1. Show all data in tabular form.
2. Plot the curves as stated in the procedure and comment on their shape.
3. Compare the experimental data with theoretical prediction.
4. List the advantages of using close loop speed control instead of open loop system.

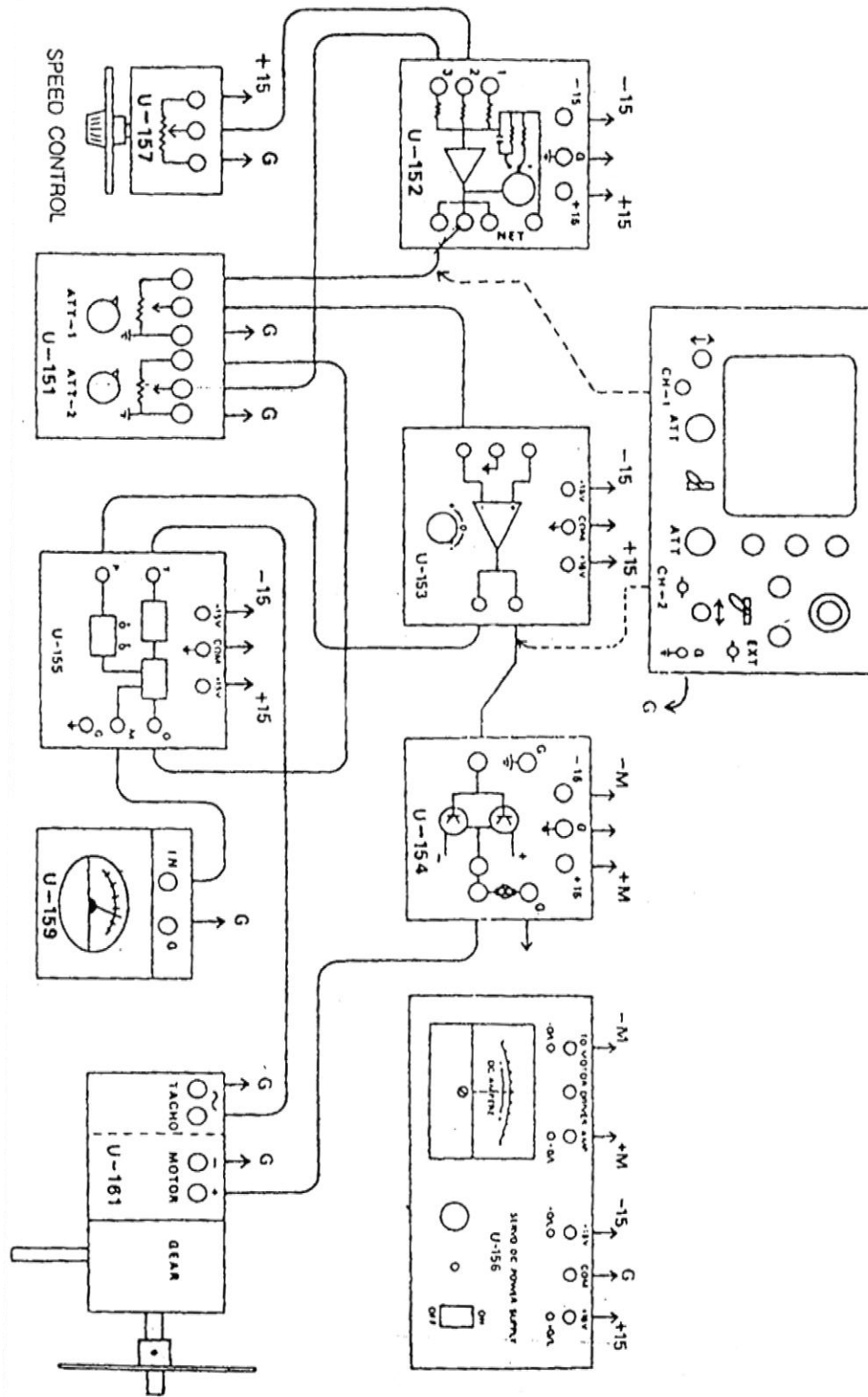


Figure 3.5: Wiring diagram of the Experiment

Experiment No: 04

Name of the Experiment: Control of a conveyor system using Programmable Logic Controller (PLC)

Introduction:

In industry, many manufacturing process demand a sequence of operation which are to be performed repetitively. Early automation systems were mechanical in design, timing and sequencing being affected by gears and cams. Slowly these design concepts were replaced by electrical drives which were controlled by relays. Even these days the relay control has not yet been obsolete rather is favorite to the engineers in many industries who understand a process and its cont better using Relay Ladder diagram.. However, the relays suffer from a number of problems viz: large size, slower operation, contact wear, inability to accept more than one input simultaneously, and the necessity of replacing the whole control panel in case another set of operation different from those for which the relay are hard wired are to be performed.

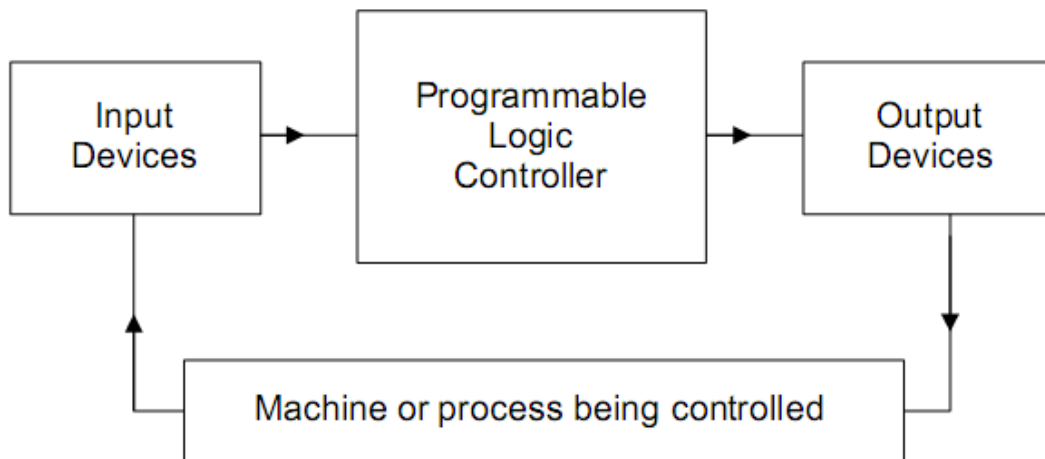
The Programmable Logic Controller

The necessity of controlling ever increasing systems, engineers turned to computers. The computer however was not suited to the industrial environment and the use of the computer on the factory floor, was not possible, unless costly interfacing filtering was used.

A programmable logic controller is a solid-state device, designed to operate in a noisy environment and perform all the logic function previously achieved using electromechanical relays, drum switches, mechanical timers and counters.

Basic PLC Operation:

Figure below shows how the PLC controls a machine or plant.



The PLC System:

The PLC, like a computer employs a microprocessor chip to do processing and memory chips to store the program.

The PLC consists of three sections:

1. A processor.
2. Input/Output (I/O).
3. A programming unit.

The inputs and outputs are connected via interface.

Basic PLC architecture:

Figure below shows the basic architecture of a PLC. It contains a processor (microprocessor chips), memory chips and an arithmetic logic unit (ALU). It also contains all the input and output interfacing. The programming device either hand held, dedicated terminal or desktop, are remote from the PLC controller.

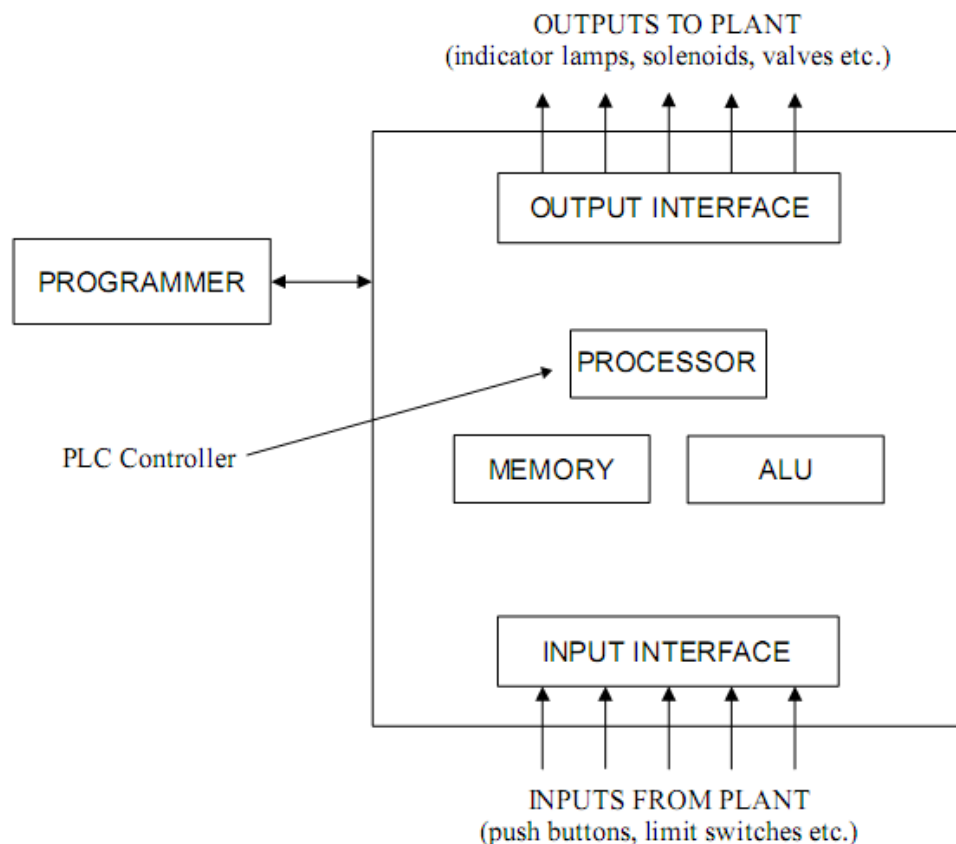


Table 1 to 3 show the connection that should be made between the PC45 trainer, MicroLogix 1500 PLC and Sequence Switch Module in order to complete the programming exercise.

Table 1:

PLC Input Addresses	PC45 Connection	Description
I0	SWG	Green Push Button
I1	SWR	Red Push Button
I2	S1	Sensor 1
I3	S2	Sensor 2
I4	R1	Micro Switch 1
I5	R2	Micro Switch 2
I6	R3	Micro Switch 3

Table 2:

PLC Input Addresses	PC45 Connection	Description
O4	E	Conveyor (Forward)
O5	R	Conveyor (Backward)
O6	LG	Green Lamp
O7	LR	Red Lamp
O8	C1	Cylinder 1
O9	C2	Cylinder 2
O10	C3	Cylinder 3

Table 3:

PLC Input Addresses	Switch Sequence Module Connection	Description
I1	S1	Switch 1
I8	S2	Switch 2
I9	S3	Switch 3

Other Connections:

DC COMMO and DCCOM1 are shorted and connected to 0 volt of upper block.

VAC VVDC4 and VAC DC5 are shorted and connection to 0 volt of lower block.

Vsw of sequence switch module to be connected to 24 volt of lower block.

Note:

When the cylinders of the PC45 Trainer are used, you will need to have the hand compressor pump connected.

As the cylinders on the PC45n Trainer are operated, air pressure drops in the hand compressor, so additional air will need to be pumped in as the system is used.

1. Short component should move the full length of the conveyor, being rejected by falling of the far end.
2. If the red push button is pressed, the conveyor should stop and the red lamp should lit.
3. Pressing green push button again should start the conveyor.

[**Note:** Before running the program make sure that the sensors are properly adjusted. The transmitter and receiver of the sensors should be aligned properly for accurate operation. The voltage between S1 br S2 arid 0 V should be around 23 volts if they are adjusted properly. Check the voltage of S1 and S2 in upper block and adjust the sensors if necessary.]

Procedures:

PART 1 To run the conveyor in forward and backward direction.

1. Open the RSLogix programming software by the following instruction.
Start →All Programs →Rockwell Software →RSLogix500English→
RSLogix500English.
2. Open a new project. You will get a window with a list of processor name. Select a processor of Bul.1764 LSP Series C by scrolling down. A window will appear to develop the ladder logic diagram.
3. Develop the ladder logic diagram as instructed and save it properly.
4. Now you have to download the file. To download the file click on the drop-down arrow of the top-left dialogue box appearing offline and select download option. Before downloading the file make sure that the MicroLogix1500 PLC and the PC45 trainer is switched on.
5. When the program has been downloaded, you will be asked if you want to go online. Click yes to go online.
6. The top left dialogue box will change color and read REMOTE PROG. Click on the down arrow to the right of this and select Run from the drop down menu. Click on yes to change to run mode.
7. Place the green push button to run the conveyor in forward direction. Press the Red push button to stop the conveyor.

8. Place the object at the right most position of the conveyor. When sensor 2 gets the object the conveyor should run in backward direction and it will continue to run in the backward direction unless you reprogram it.
9. Modify Rung 03 of the ladder logic diagram as suggested and observe the movement of conveyor after sensing the object at sensor 2. The conveyor should run back and forth with the movement of the object in between sensor 1 and 2.
10. To bring the program out of Run mode, click on the down arrow button to the right of the top left dialogue box and select program, then click on yes to return to program mode.
11. Click on the file menu and select close to close the project. Now you are safe to exit the programming software.
12. Remove the power from MicroLogix 1500 PLC and PC45 trainer.

PART 2: To distinguish tall object from short object placed on a running conveyor.

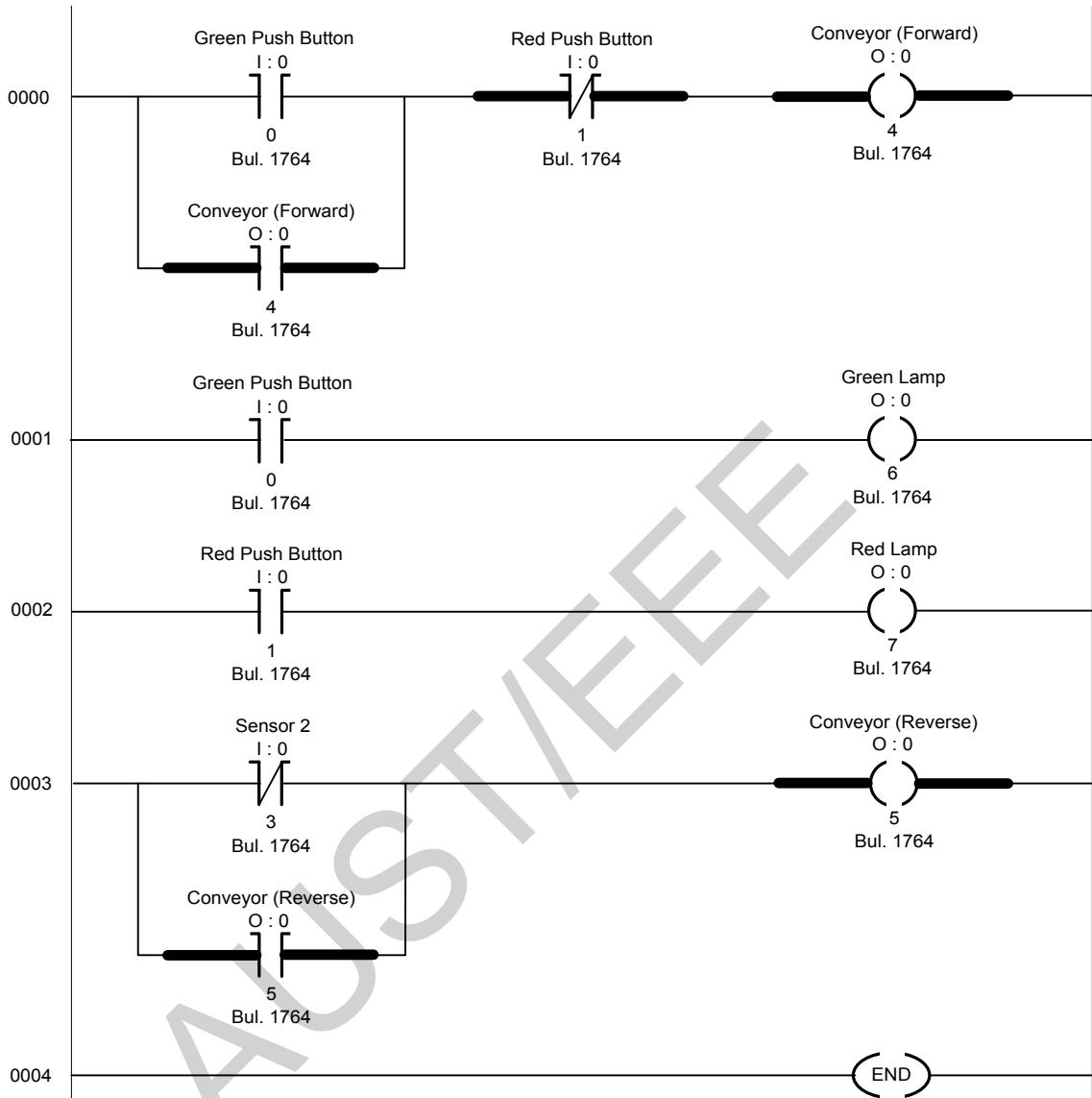
In this part the program should perform the following operation.

1. The Green push button should be used to start the conveyor in the forward, right to left direction.
2. While the conveyor is operating, the green lamp should be lit.
3. Use sensor 1 on the conveyor to determine if a component is tall or short. (Should be set to detect a tall component)
4. Sensor 2 on the conveyor is used to start timing for either a tall component or a short component. (Should be set to detect all components)
5. Cylinder 2 (the center cylinder) is operated when a tall reaches it. (Remember that micro switch 2 is activated when cylinder 2 is operated)

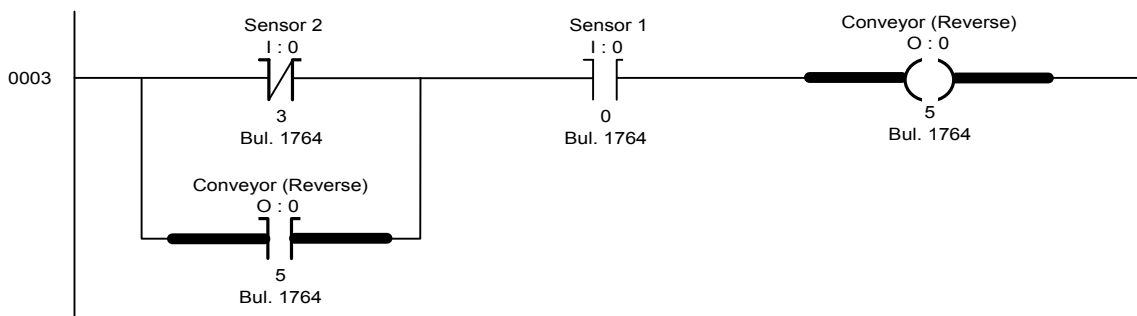
Report:

Explain the Logic of the ladder diagrams programmed in both parts.

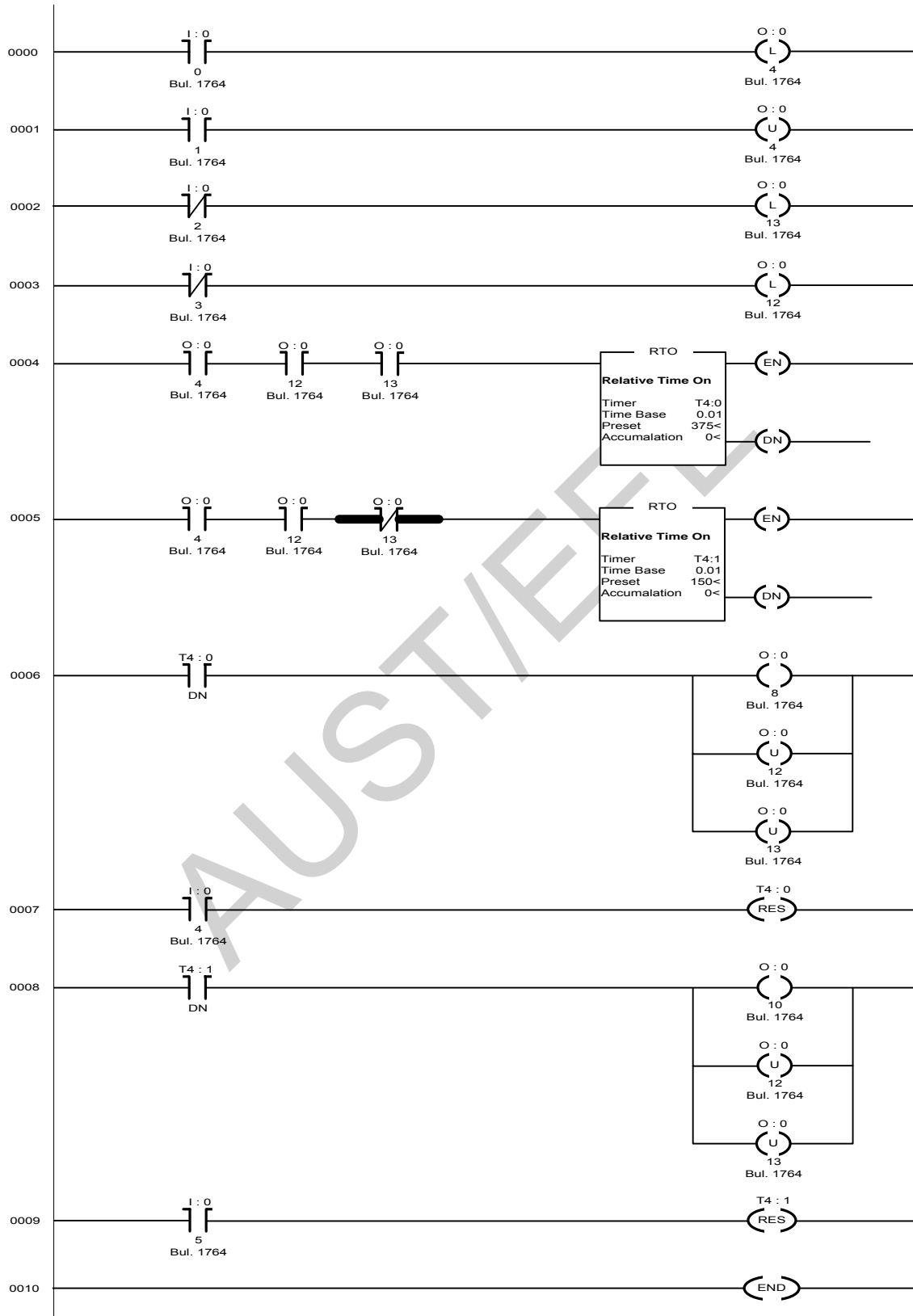
Ladder Logic Diagram for Part 1



Modification of Rung 03 :



Ladder Logic Diagram for Part 2



Experiment No: 5

Name of the Experiment: Study of the Root Locus of a System Simulated on a PC Using the 'MATLAB' Software

Objectives:

- a. To draw the root locus of a given system for a specified range of gain.
- b. To analyze system stability from the root locus with the variation of gain.

Introduction:

For a closed loop system. the control ratio or overall transfer function is

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

The stability of the system and the response $c(t)$ depend upon the poles of $C(s)/R(s)$ which are the roots or zeros of the characteristic equation $1+G(s)H(s)=0$. The root locus of the system is a plot of these roots as a function of the gain. To work with the MATLAB the product $G(s)H(s)$ is to be expressed as $G(s)H(s)=\frac{KP(s)}{Q(s)}$, where the variable K represents the gain, $P(s)$ and $Q(s)$ are functions of s either in a single polynomial or in factored form.

Procedures:

1. Run the MATLAB package. Note that '»' is the prompt. Displayed automatically for the MATLAB user.
2. Consider the following systems. K and K are the gain terms. Carry Out steps 3 to 11 for each system.
 - a. $G(s) = \frac{K(s+1)}{s^2+3s+3.25}$, $H(s) = 1$.
 - b. $G(s) = \frac{K}{s(s^2/2600 + s/26+1)}$, $H(s) = \frac{1}{0.04s+1}$.
3. Enter each of the numerator and denominator factors of the product $G(s)H(s)$ as follows:
 - »num1 = of s coefficient of s^n , coefficient of s^{n-1} , ...coefficient of s^0 or constant term];
 - »num2 = [similarly];
 - »den2 = [similarly];
 - »den1 = [similarly];and so on

If the numerator is only a constant value $c \neq 0.0$ then use

```
»num = [c];
```

If the transfer function has more than one numerator (for the given systems there is only one numerator factor) or denominator factors, multiply them as follows to obtain a single numerator and denominator factor respectively. As for example:

```
»num = conv(num1,num2)
```

```
»den4 = conv(den1 ,den2)
```

```
»den = conv(den3,den4)
```

4. Define a vector of desired gain values e.g., 0 to 5.0 at an interval of 0.5 as follows:

```
»k = 0:0.5:5.0;
```

5. Obtain the root locus for the specified gain range using the following command:

```
»r = rlocus (num, den, k);
```

6. To plot and display the root locus on the screen with the roots marked 'x' use:

```
»plot (r, 'x')
```

The plot can be made to have grid boxes on it as follows:

```
»grid
```

The plot can be labeled as follows:

```
»title ('Root locus'), xlabel ('Real parts of the roots'), ylabel ('Imaginary parts of the roots')
```

7. To highlight a certain portion of the displayed plot define the portion on the x and y axes as follows and then display.

```
»A=[Xmin, Xmax, Ymin, Ymax]
```

```
»axis (A)
```

To restore back the original full plot just use the command:

```
»axis auto
```

8. Verify that the number of branches in the root locus is equal to v , the order of the denominator of $G(s)H(s)$ i.e., for each value of gain k there will be v number of roots of $1 + G(s)H(s) = 0$.

9. To see the effects of increased or decreased system gain upon the root locus, redefine the gain vector k in step 2. Then repeat steps 4 to 6.
10. Check the root locus display if any branch has extended to the right of the origin i.e., the (0,0) coordinate. If so then it means that the corresponding gain values have produced roots with positive real part and will produce an unstable response. These roots can be identified displaying the whole matrix 'r' containing the roots as follows:


```
»r
```

The corresponding rows containing the roots with positive real parts can be redisplayed one by one using the command:

```
»r (m,n)
```

where row number corresponds to the m-th gain and the column number represents the n-th root for this gain. Then the gain value can be displayed as follows:

```
»k(m)
```
11. After plotting the root locus for the chosen sy get its hard copy by selecting 'Print' command from the 'File' menu of the Figure window.

Observing response of a value from the root locus:

How do we design a feed-back controller for the system by using the root locus method where our design criteria are 5% overshoot and 1 second rise time?

$$H(s) = \frac{Y(s)}{U(s)} = \frac{s + 7}{s(s + 5)(s + 1)(s + 20)}$$

Enter the transfer function and the command to plot the root locus:

```
num=[1 7];
den=conv (conv ( [ 1 0], [1,5] ),conv ( [ 1 15], [ 1,20 ] ) );
rlocus(num,den)
axis( [-22 3 -15 15] )
```

The plot shows all possible closed-loop pole locations for a pure proportional controller. Obviously not all of those closed-loop poles will satisfy our design criteria. To determine what part of the locus is acceptable, we can use the command `sgrid(Zeta,wn)` to plot lines of constant damping ratio and natural frequency. Its two arguments are the damping ratio(Zeta) and natural frequency(Wn) [these may be vectors if you want to look at a range of acceptable values]. In our problem, we need an overshoot less than 5% (which means a damping ratio Zeta of greater than 0.7) and rise time of 1 second (which means a natural frequency Wn greater than 1.8) Enter in the matlab command window:

```
zeta=0.7;
wn=1.7;
Sgrid(zeta, wn)
```

In the plot, the two white dotted lines at about 45 degree angle indicate pole locations with $Zeta=0.7$; in between these lines, the poles will have $Zeta > 0.7$ and outside of the lines $Zeta < 0.7$. The semi circle indicates pole locations with natural frequency $W_n=1.8$; inside the circle, $W_n < 1.8$ and outside the circle > 1.8 .

Going back to our problem, to make the overshoot less than 5%, the poles have to be in between the two white dotted lines and to make the rise time shorter than 1 second, the poles have to be outside of the white dotted semicircle. So now we know only the part of the locus outside of the semicircle and in between the two lines are acceptable. All the poles in this location are in left half plane, so the closed-loop system will be stable.

From the plot we see that there is part of the root locus inside the desired region. So in this case we need only a proportional controller to move the poles to the desired region. We can use `rlocfind` command in Matlab to chose the desired poles on the locus:

```
[kp,poles]=rlocfind(num,den)
```

Click on the plot the point where you want the close loop pole to be. Since the root locus may has more than one branch, when you select a pole, you may want to find out where the other pole (poles) are. Remember they will affect the response too.

In order to find out the step response, you need to know the closed loop transfer function. You could compute this using the rules of block diagrams or let Matlab do it for you:

```
[numc, denc]=cloop((kp)*num,den)
```

The two arguments to the function `cloop` are the numerator and denominator of the open-loop system. You need to include the proportional gain that you have chosen. Unity feedback is assumed.

If you have non unity feedback situation, look at the help file for the `feedback`, which can find the closed-loop transfer function with a gain in the feedback loop.

Check out the step response of you closed ;oop system:

```
step(numc,denc)
```

As we expected, this response has an overshoot less than 5% and a rise time less than 1 second.

Warning: You must request your teacher to be with you before issuing the above print command.

Report:

Show the root locus for each system and find the range of gain which will produce system instability. Comment on the results obtained.

Experiment No: 6

Name of the Experiment: Study of steady state error analysis of different Types of system.

Theory:

In many control system designs, we are specifically interested in the final, or steady state value of the output. This is known as steady state accuracy. Ideally, in the steady state, the output, $y(t)$, equals the command signal, $r(t)$, and the error is zero. This ideal situation is rarely met, and so we need to be able to determine the steady state error for any system. The steady state error is defined as:

$$e_{ss} = \lim_{t \rightarrow \infty} e(t) = \lim_{t \rightarrow \infty} [r(t) - y(t)]$$

Only for unity feed back systems, the error is the comparator output signal and we can determine the steady state error by examining the open loop transfer function $KG(s)$. By using final value' theorem it can be shown that

$$e_{ss} = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} \frac{s}{1 + G(s)} R(s)$$

We are interested in the steady state error for step, ramp, parabolic, and higher order polynomial inputs, i.e

$$r(t) = \frac{t^n}{n!} \rightarrow R(s) = \frac{1}{s^{n+1}} \quad n = 0, 1, 2 \dots \dots$$

Therefore eurn

$$e_{ss} = \lim_{s \rightarrow 0} \frac{s}{s^n + s^n G(s)}$$

For a unit step input ($n=0$), $e_{ss} = \lim_{s \rightarrow 0} \frac{1}{1+G(s)}$

For higher order polynomial inputs, $e_{ss} = \lim_{s \rightarrow 0} \frac{1}{s^n G(s)}$

If $G(s)$ has no poles at origin, the $G(0)$ is finite, which means that the step response error is finite and all other responses infinite. We define the system *Type* as the order of the input polynomial that the closed loop system can track with finite error. If $G(s)$ has no poles at the origin, the closed system is *Type 0* and can track a constant, one pole at origin results in a *Type 1* system that can a ramp; two poles at the origin result in a *Type 2* system that can track a parabola etc.

Because we deal with many electronechani systems, control engineers also define position, velocity and acceleration error constants as follows.

$$K_p = G(0), \quad K_v = \lim_{s \rightarrow 0} sG(s), \quad K_a = \lim_{s \rightarrow 0} s^2 G(s)$$

Following Table shows the 3 steady state errors for Type 0, 1, and 2 systems.

Steady State Errors for Polynomial Inputs			
System Type → Polynomial Degree ↓	0	1	2
Step Input	$\frac{1}{1 + k_p}$	0	0
Ramp Input	∞	$\frac{1}{k_v}$	0
Parabolic Input	∞	∞	$\frac{1}{k_a}$

Procedures:

1. Execute the following program for the system

$$K = 1.5 \text{ and } G(s) = \frac{1}{s(s+1)(s+2)}$$

The program will give three plot windows. Figure-1 is for step response, Figure-2 is for ramp response and Figure-3 is for parabolic response. Each figure will have subplots of original response and the steady state error. We use 'step' command for finding the step response and 'Isim' command for finding the ramp and parabolic responses.

2. The program also gives the steady state error for our considered each type. of input. Note down the steady state error and Y compare with the theoretically obtained steady. State error using the above table. Assignment:

Reports:

1. What is the 'Type number' of our system? Discuss your findings of each type of obtained plot
2. Run the program for the following systems:

$$\text{System-1 : } K = 1, G(s) = \frac{10}{(0.2s+1)(0.5s+2)}$$

$$\text{System-2: } K = 1, G(s) = \frac{12}{s(s+5)(s^2+2s+2)}$$

What are the type numbers of the above two systems? Get the hard copy of the plots for the above two systems and discuss the plot in each case.

3. What is the effect of stability of the closed loop system on the value of steady state error?


```

% To observe the step, ramp and parabolic response and
% their errors, steady state errors with respect to time.
% Given transfer function,  $G=1.5/s(s+1)(s+2)$ 
% Numerator=1.5
% Denominator=  $s(s+1)(s+2)$ 
% The sign '%' indicates a comment in matlab. Matlab
% ignores the lines having % at the time of simulation.

% Detail program is given below.
num1=1.5;
den1=[1 3 2 0];
[num, den]=cloop(num1,den1);
% To get the close loop transfer function.
H=tf(num,den);
% H is the overall transfer function taking close loop
% numerator and denominator.
t=linspace(0,60,2000)';
% Defines time vector from 0 to 60 having 2000 values.
% This time is transposed by the ' ' sign to get a column
% vector to match the matrix dimension for addition or subtraction.

% Step response
u=ones(length(t),1);
% Defines a unity vector of row size length(t) and single column.

% length(t) returns the number of t values, defines the step signal.

ys=step(H,t);
% Gives step response
figure(1);
subplot(2,1,1), plot(t,u,t,ys,'--');
% Divides the window in two row and column and plots in row
% 1 for this
% command.
% Plots the input and response vs time.
grid
eu=u-ys;
% error= input-response
eu_ss=eu(length(eu))
% Finds the final value of the error, that is the steady
state value.
subplot(2,1,2), plot(t, eu);
% Plots in 2nd row, Plots error vs time signal
grid

```

```

% Ramp Response
r=t;
% defines the ramp signal
yr=lsim(H,r,t);
% lsim is the keyword for the ramp and parabolic analysis.
% Basically for the analysis of linear system
figure(2);
subplot(2,1,1), plot(t,r,t,yr,'--');
grid
er=r-yr;
er_ss=er(length(er))
subplot(2,1,2), plot(t,er);
grid

% Parabolic Response
p=(t.*t)/2;
yp=lsim(H,p,t);
figure(3);
subplot(2,1,1), plot(t,p,t,yp,'--');
grid
ep=p-yp;
ep_ss=ep(length(ep))
subplot(2,1,2), plot(t,ep);
grid

```

AUSTRIE

Experiment No: 7

Name of the Experiment: Position Control Using a DC Servo System

Theory:

Error Signals in a Position Controller

The basic function of an angular position controller is to provide an output angular position signal which precisely follows the input angular position signal. The input or output position information is expressed in terms of the selected angle around a circle.

To achieve the control function, it is necessary to rotate a motor until the signal detected for the motor position is equal to the signal representing the reference or the input position. A potentiometer is used to convert the angular position to an equivalent electrical signal. Figure 7.1 shows a circuit diagram which utilizes potentiometers as an angle-to-voltage converter.

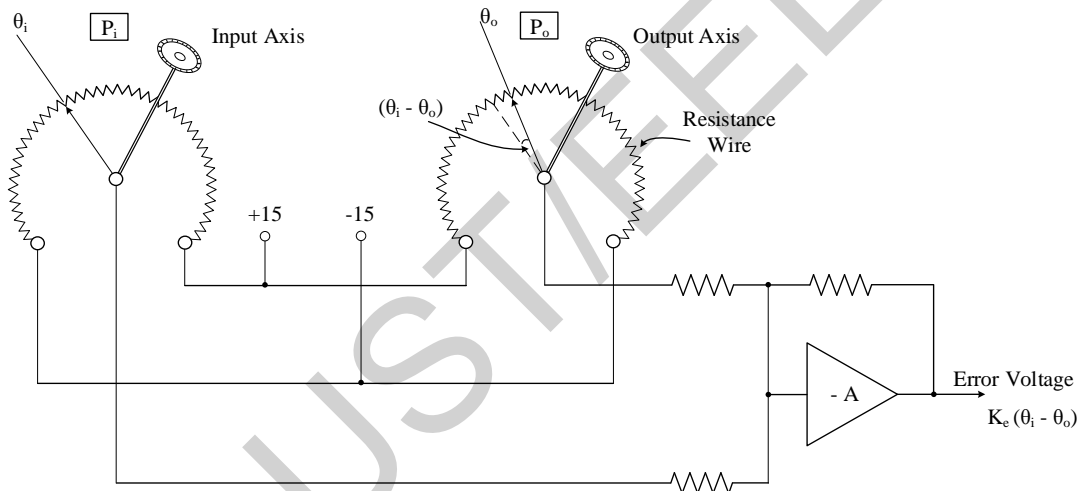


Figure 7.1: Circuit Diagram of an Angular Error Detector Using Potentiometers.

The P_i in the figure is the input potentiometer, and P_o the output potentiometer. The amplifier ($-A$) is configured as an inverting amplifier. Due to the polarity applied to P_i and P_o when the input and output positions are identical, the output of the amplifier becomes zero.

In general, when the angular position of P_i is θ_i and θ_o is the angular position of P_o . Also the relative angular position error between P_i and P_o is defined as $(\theta_i - \theta_o)$. The converted and amplified output of the error from the amplifier can be set to $K_e(\theta_i - \theta_o)$, where K_e represents a conversion factor. K_e can be determined for a given system when the actual output voltage of the amplifier is measured.

A closed loop control system can be formed when the error signal is further amplified and applied to a motor. As the motor reacts to the incoming error signal, and also the motor is coupled to the output potentiometer P_o the loop is closed. As the loop is closed,

error detection and associated motor reaction processes continue until the error signal is reduced to zero.

Closed Loop Position Controller:

In a closed loop position controller system, the positional information from an output potentiometer (P_0) which is mechanically coupled to a motor is fed back to a control amplifier. Then, the reference position input from the input potentiometer (P_i) is combined with the feedback signal at the input of the amplifier which drives the motor in proportion to the difference between the two, signals. When the two positions are identical, the output of the amplifier becomes zero.

A simplified system diagram of a closed loop position controller which will be used in this experiment is shown in Figure 7.2.

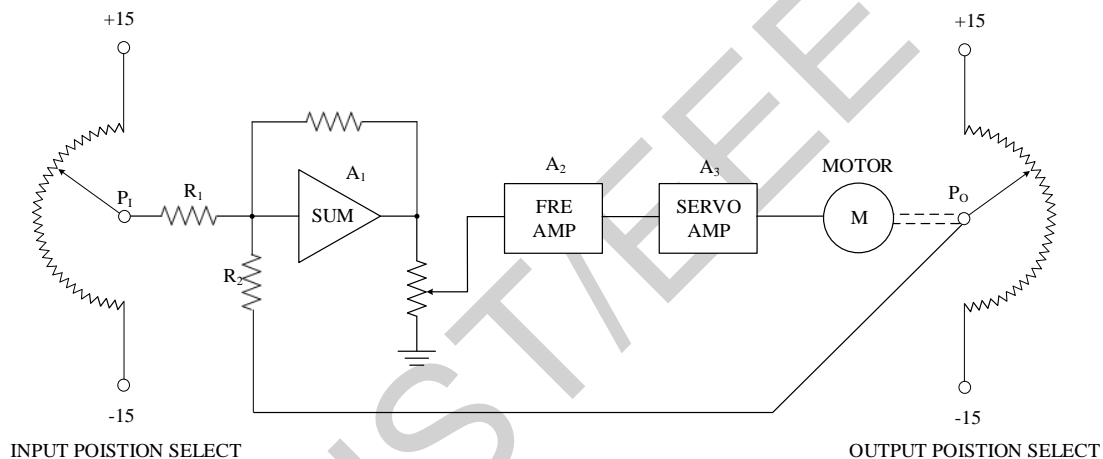


Figure 7.2: A Closed Loop Position Controller

There are three amplifiers in Figure 7.2. The A_1 is an error signal generator, A_2 is an error signal amplifier and A_3 is the driver for the motor M . As P_i is turned away from P_0 , the difference between the two potentiometer voltages becomes an error signal which appears at the input of A_1 . The error signal is further amplified through A_2 and A_3 , and drives the motor in the direction to reduce the error voltage between P_i and P_0 . Therefore, as P_i is turned clockwise, P_0 follows the same direction. This feedback action continues until the output of A_1 is reduced to zero. At this point, the voltage measured at P_i and P_0 are same but in opposite polarity. For example, if P_i is at $+3V$, then P_0 is at $-3V$, making the sum of two zero.

The final relative position between P_i and P_0 depends upon the gain of the amplifiers. For a large gain, the position of P_0 can be almost equal to the position of P_i . But when the gain is not sufficient, there can be an offset in the relative position, and also there may be no output initially until the input exceeds a certain value. The range of input for which there is no output is the “deadband and for a position controller.

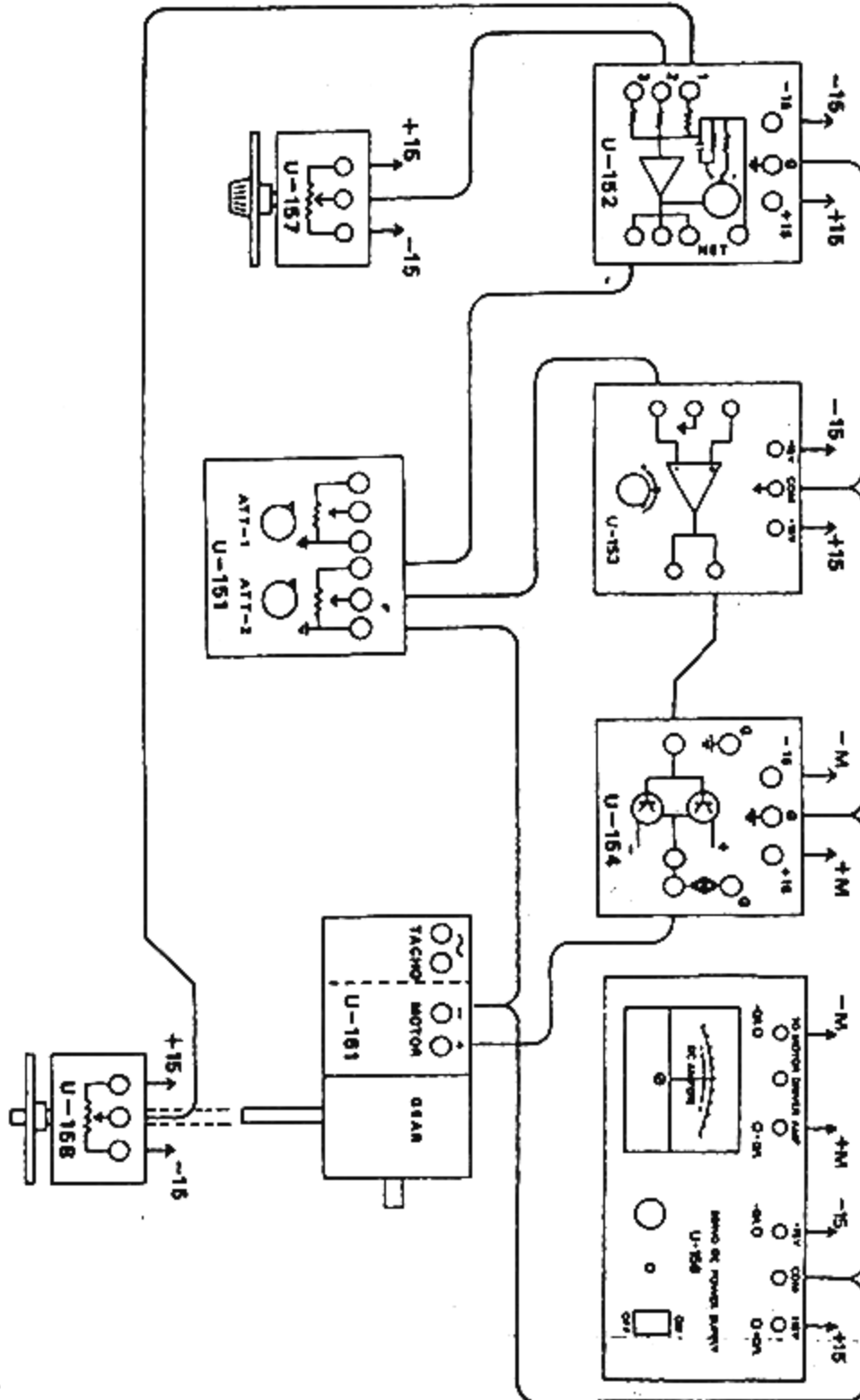


Figure 7.3: Wiring diagram of the Experiment

Procedures:

- 1) Referring to Figure 7-3, arrange the modules, including coupling of U- 158 to U-161, and connect them together.
- 2) Set U-152 switch to “a” and U-151 to “10”. Turn the power of U-156 on. Set U-157 dial to 180 degrees.
- 3) Adjust U-153 to make the output of U-154 zero. Once the adjustment is done, do not alter U-153 setting.
- 4) 4 Set U-151 to ‘9’ Within ± 20 degrees from the original 180 degree setting turn U-157 either clockwise or counterclockwise, and see if U-158 follows the movement U-158 motion should lag U-157. In case U-158 leads U-157, switch the wires of U-161 motor.
- 5) Turn U-157 clockwise from 0 degree position by 10 degree increment up to 150 degrees. Measure the angle of U-158 (θ_0) at each position of U-157 (θ_i) Note down the deadband. Calculate the offset error angle between U-157 and U-158 at each position. Record all the results in a tabular form as given below. Repeat the measurements with U-157 turned counterclockwise from 0 degree position.
- 6) Increase the system gain by setting U-151 to 7, 5, 3 and 1. At each U-151 setting, repeat Step 5 experiment. Observe the change in offset error angle and de as a function of the system gain.

Data Taable:

U-151 Setting	θ_i	θ_0	Offset error angle= $\theta_i - \theta_0$	Deadband

Reports:

1. Show all the results in tabular form.
2. Plot θ_0 vs. θ_i for different values of U-151 setting (inversely related to system gain) on the same graph paper. Use separate graphs for clockwise and counterclockwise direction of rotation.
3. Plot dead band vs. U-151 setting.
4. From the results obtained, comment on the change in offset error angle as a function of the sys gain. Also comment on the relationship between system gain and dead band.

AUSTI/EEEE

Experiment No: 8

Name of the Experiment: Study of PID (Three-Term) Control Using PC Based Servo-System

Objectives:

Having completed this experiment students will be able to:

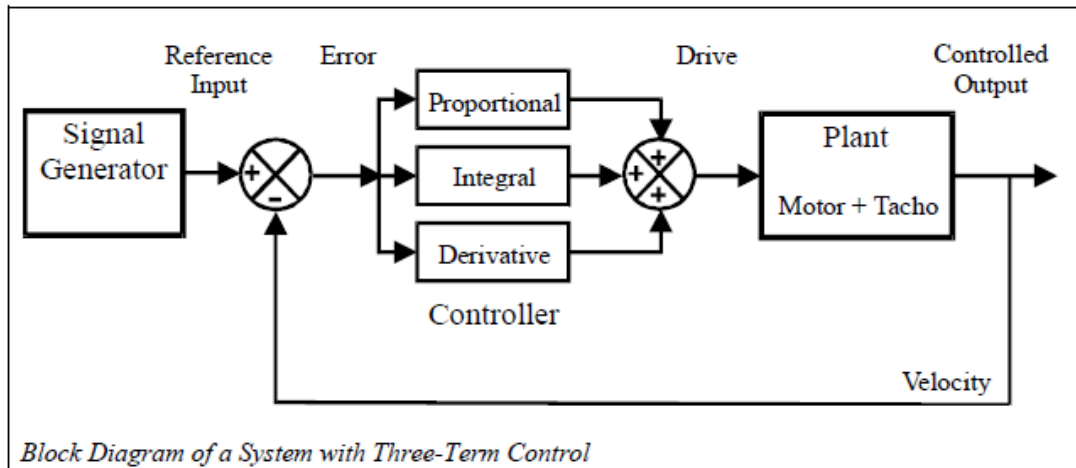
- Define the purpose of Three-Term Control
- Explain the effects of Proportional Band, Integral Action and Derivative Action

Introduction:

In the simplest feedback control system the control signal is directly proportional to the deviation (error) i.e. the difference between the reference input and the feedback signal. However in the sophisticated control systems to improve the steady state and transient performance respectively an integral and a derivative of the error (e) are added to the proportional term to make a composite control or drive signal (u). Mathematically,

$$u = k_c e + k_i \int e dt + k_d \frac{de}{dt}$$

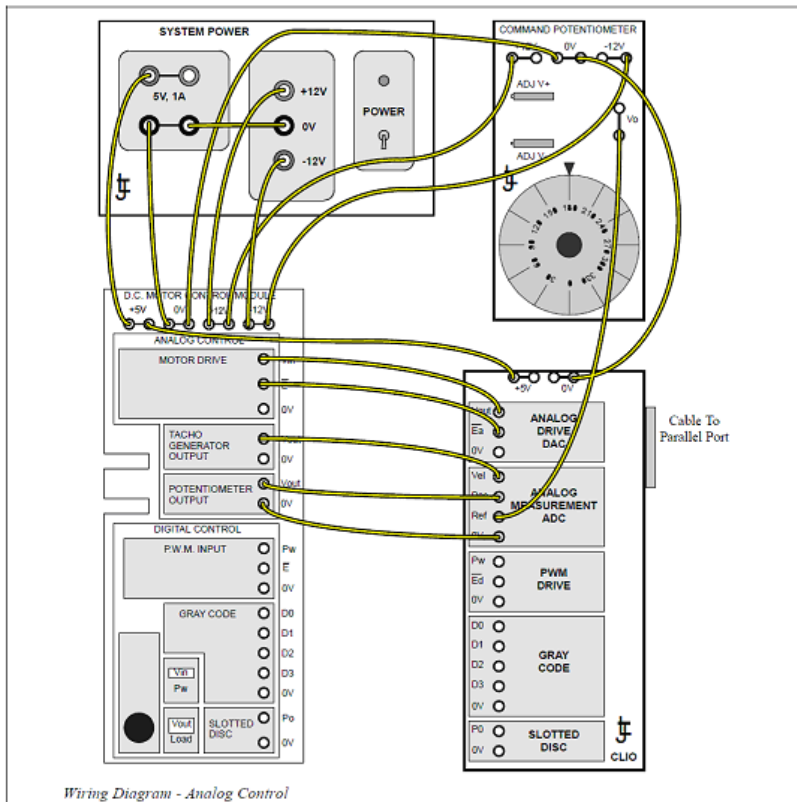
Where, k_c , k_i and k_d are gain of proportional, integral and derivative blocks respectively.



To improve the steady state performance means, the reduction of error (also termed offset) and improvement of transient response is the enhanced stability through reduction of oscillation and settling time.

Procedures:

1. Make the Hardware connection as shown in the following figure:



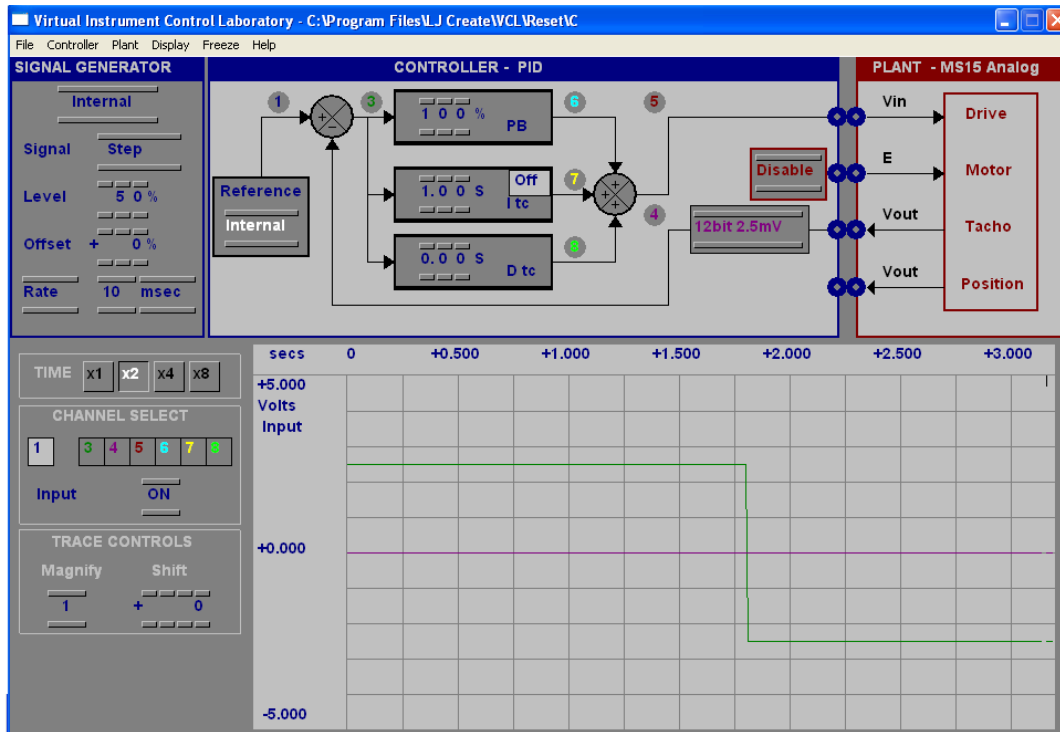
2. Start VCL and load CA06PE09. Make sure that following conditions are fulfilled.

File	Controller	Plant	Display
CA06PE09	PID	MS15 Analog	Graph
Signal Generator		Graph	
Signal	Step	1 Input ON	5 Drive OFF
Level	50%	3 Error ON	6 PB OFF
Offset	0%	4 Velocity ON	7 Integral OFF
Rate	10 msec		8 Deriv OFF
Reference	Internal		
DC Motor		Output Potentiometer	Disengage
Brake	0	Command Potentiometer	180°

3. Set **Integral time constant** (I_{tc}) Off and **Derivative time constant** (D_{tc}) to 0. The controller is now proportional only. Where $I_{tc} = T_i = 1/k_i$ and $D_{tc} = T_d = k_d$. The proportional control is marked PB. This stands for **Proportional Band**. Proportional Band is the inverse of gain. When expressed as a percent,

$$\%PB = \frac{100}{K_c}$$

This nomenclature is a result of the origins of PID control.



4. Set PB to 100% ($K_C = 1$). Switch on the system.

5. Measure the error between the input (ch1 dark blue) and the velocity (ch4 purple). Also measure the time constant of the output and the settling time. Note down the values in the table 8.1.

Observation No.	PB (%)	I _{tc}	D _{tc}	Error= input -output	Time constant (τ)	Settling time (t_s)

6. Repeat step 5 by setting PB to 40%. ($K_c = \underline{\hspace{2cm}}$)

7. Decrease the PB to 4% (controller gain, $K_c = \underline{\hspace{2cm}}$). Repeat step 5.

8. Decrease the input *Level* to 30% and set the PB to 40%. Integral time constant (I_{tc}) to 1 second and click the On/Off box to bring in the Integral controller. Repeat step 5.

9. Decrease the Integral time constant until a good response is obtained. Repeat step 5. If the integral goes off scale and the system will not respond, click the Itc to Off then back to On. This resets the integrator.

10. Change the *Plant* to **Process**. The computer is now simulating a more complicated plant. Set PB = 30% and Itc = 0.24s. The Output should be showing an oscillatory transient. Increase the Derivative time constant (Dtc) until only a small overshoot can be seen. Repeat step 5.

11. Set Itc = 0.24s. and Dtc=0.1 sec. Repeat step 4-7.

Reports:

1. Show all the results in tabular form.
2. Comment on effects of only proportional control on the steady state and transient response.
3. Comment on the effect of P+I control and effects of increasing and decreasing the value of T_i for a given PB.
4. Comment on the effect of P+I+D control of a second order process. Also comment on the effect of derivative control alone for a fixed value of integral and proportional block.
5. Comment on the effect of proportional control alone for a fixed value of integral and proportional block.