LAB 7

PID CONTROL

1. LAB OBJECTIVE

The objective of this lab is to design and implement a PID (Proportional-Integral- Derivative) controller and to study the effect of integral and derivative control actions on the system. We will also learn about the importance of the bandwidth of a controller.

2. BACKGROUND

2.1 Closed-loop bandwidth

The frequency range of input signals that a closed-loop system can track successfully without significant attenuation is called the closed-loop bandwidth of the system. It is defined as the frequency at which output amplitude becomes $1/\sqrt{2}$ (or -3 dB)

Example:
Consider the proportional control system shown below. Let $K_m = 5.5$, $T_m = 0.13$, $K_p = 10$.

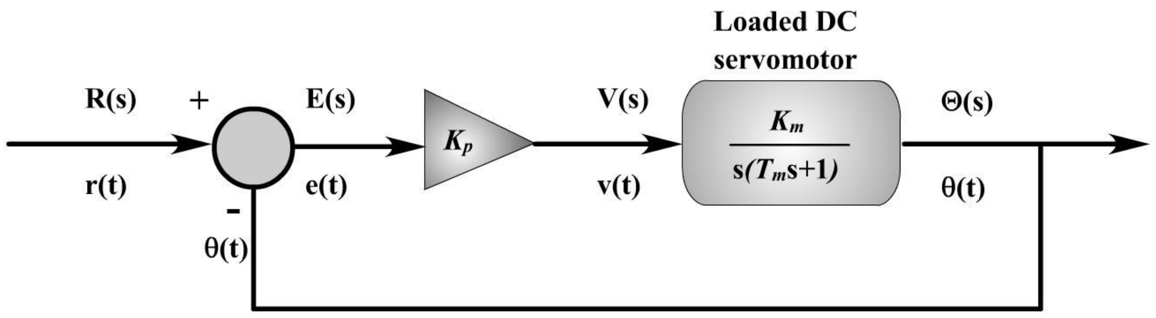

Figure 1: Block diagram of closed-loop proportional control system that controls the shaft position of a DC servomotor.

The closed loop transfer function is given by

$$\frac{\Theta(s)}{R(s)} = \frac{K_p K_m}{s^2 T_m + s + K_p K_m} \tag{1}$$

The bode plot of the above transfer function is shown in Figure 2. We can see from the Bode plot that the –3dB point occurs around 30 rad/s. This is considered to be the bandwidth of the closed-loop system. In other words, this controller can track signals that vary with frequencies up to 30 rad/s (4.7Hz) without significant attenuation.

2.2 Experimental determination of bandwidth

We can determine the bandwidth experimentally by observing how well the plant tracks sinusoidal input signals of different frequencies.

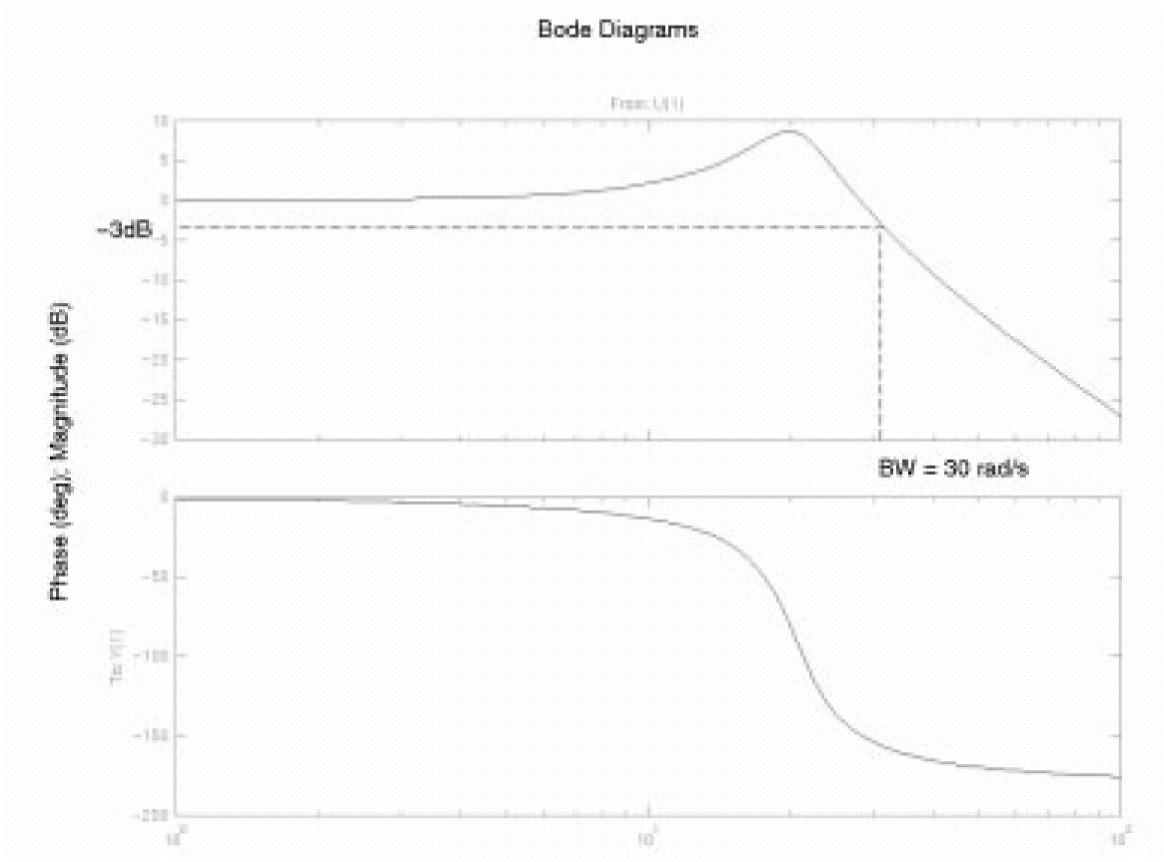

Figure 2: Bode plot of the closed loop transfer function of the control system shown in Figure 1

2.3 Position control

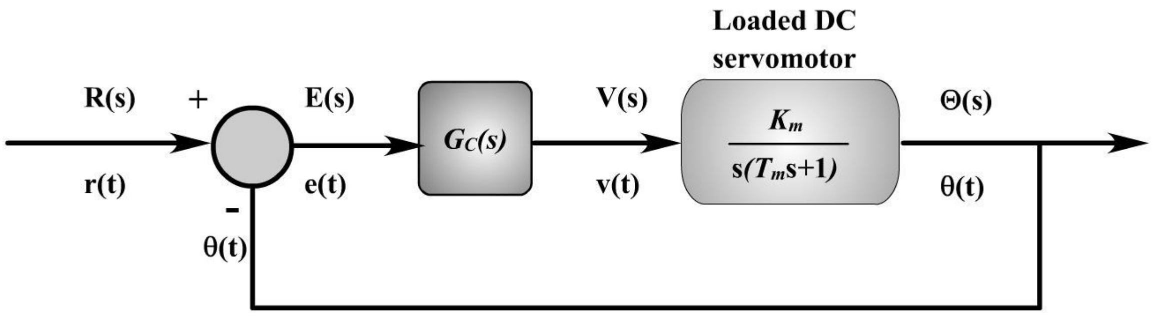

Figure 3: Position control loop.

Figure 3 shows a position control loop for the lab DC servomotor. A variety of control strategies, represented by the Laplace function *$G_C(s)$*, can be used to control the shaft position. We have already seen proportional (P) control. In this section we will consider three other types of control that use not only the error of the shaft position, but also the integral and/or the derivative of this error. The three additional types of control to be considered include PI, PD, and PID control.

2.4 PI controller

$$G_c(s) = K_p + \frac{K_i}{s} \tag{2}$$

Proportional-integral (PI) control considers both the magnitude of the system error and the integral of this error. For the DC servomotor, by integrating the error of the shaft position over time, scaling the integral, and adding this term to the control voltage, steady-state errors in position can be eliminated that P control alone may not be able to cancel. This is the primary reason to add integral control action, to eliminate steady state error. The drawback of adding integral action is that it may have a destabilizing influence on the system response if K_i is not properly selected.

Implementation

error = (reference position – actual position)
integrated_error = integrated_error + error x T_s (T_s is the sampling period)
control voltage = K_p x error + K_i x integrated_error

The integrated_error term is calculated by summing the error at each time instant. In C the statement

```
integrated_error += error*Ts;
```

does this. It is necessary to initialize *integrated_error* to zero before the start of the integration process.

2.5 PD controller

$$G_c(s) = K_p + K_d s \tag{3}$$

Proportional-derivative (PD) control considers both the magnitude of the system error and the derivative of this error. Derivative control has the effect of adding damping to a system, and, thus, has a stabilizing influence on the system response.

Implementation

error = (reference position – actual position)
error_derivative = (error – previous_error)/T_s
control voltage = K_p x error + K_d x error_derivative

2.6 PID controller

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \tag{4}$$

Proportional-integral-derivative control (PID) combines the stabilizing influence of the derivative term and the reduction in steady-state error from the integral term.

Implementation

error = (reference position – actual position)
integrated_error = integrated_error + error x T_s
error_derivative = (error – previous_error)/T_s
control voltage = K_p x error + K_i x integrated_error + K_d x error_derivative

2.7 Ziegler-Nichols rule for tuning a PID controller

The Ziegler-Nichols tuning rule is widely used to tune PID controller gains in process control systems. For a PID controller of the form

$$G_c(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right) \tag{4}$$

The parameters K_p, T_i and T_d are calculated using the following design criteria.

Ziegler-Nichols design rule

Type of controller	K_p	T_i	T_d
P	**0.5 x** K_{cr}^s	-	-
PI	**0.45 x** K_{cr}^s	P_{cr}^s **/1.2**	-
PID	**0.6 x** K_{cr}^s	**0.5 x** P_{cr}^s	**0.125 x** P_{cr}^s

K_{cr}^s and P_{cr}^s are obtained by experiment. K_{cr}^s is the critical gain where the shaft exhibits sustained oscillations (marginal stability). P_{cr}^s designates the period of these oscillations.

For the PID control law given in Section 8.2.3.3, $K_p = K_p$, $K_i = K_p/T_i$, and $K_d = K_p T_d$.

3. PRELAB

a) Modify the program you wrote in the Lab 6 (Proportional Control) prelab (question 1) so that the reference displacement signal is a sine wave instead of a step input. Use a magnitude of 250 counts and a frequency of 5 Hz. Use Kp of 0.004. You will vary the frequency of the reference signal and observe tracking performance. As before, store encoder data in an array variable for plotting with *print_into_matlab*.

b) Determine the gain values for a PI controller and a PID controller using the Ziegler-Nichols tuning rules. Use the values for K_{cr}^{s} and P_{cr}^{s} that you experimentally determined in the previous lab.

c) Write a program to implement PI position control. Use gain values obtained from Question 2. Use a step input of 500 counts.

d) Write a program to implement PID position control. Use gain values obtained from Question 2. Use a step input of 500 counts.

4. LAB PROCEDURE

Exercise 1

Implement Prelab Question 1. Vary the reference signal frequency according to the following table. Plot the experimental response using *print_into_matlab48*. Determine the value of the output magnitude from the plots obtained. You already know the input amplitude.

Input frequency (Hz.)	output amp./input amp. (MAGNITUDE)
4	__________
8	__________
10	__________
12	__________
14	__________
16	__________
20	__________

Exercise 2

Implement PI position controller (Prelab Question 3). Plot the experimental step response.

Exercise 3

Implement PID position controller (Prelab Question 4). Plot the experimental step response.

5. POSTLAB ASSIGNMENT AND LAB REPORT

a) Use MATLAB to obtain the bode plot of the closed loop transfer function that you determined in Postlab Question 1 of Lab 7 for $K_p = 0.004$. Estimate the bandwidth of the controller from the bode plot. Show your results on the bode plot and include the plot in your postlab.

b) Create a bode plot from your experimental data. What is the bandwidth of the controller based on the experimental data? How well does your experimental value match the theoretical prediction in question 1? Turn the experimental bode plot in with your postlab.

c) Compare the experimental step response curve that you obtained for position control experiments with P, PI and PID controllers. Comment on the steady state error value, oscillatory behavior (stability), and speed of response (rise time). If necessary show your reasoning on the plots.

Lab Report Requirements

Solutions to and all the requirements of the questions in Section 5.

All experimental plots obtained during the lab with appropriate titles, except for the sinusoidal response plots for each frequency for Exercise 1.