8.1. 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.

8.2. BACKGROUND

8.2.1. 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/input becomes \( 1/(\sqrt{2}) \) (-3 dB).

Example:
Consider the proportional control system shown below. Let \( K_m = 5.5 \), \( T_m = 0.13 \), and \( K_p = 10 \).

\[
\frac{\Theta(s)}{R(s)} = \frac{K_p K_m}{s^2 T_m + s + K_p K_m}
\]

The bode plot of the above transfer function is shown in Figure 10. We can see from the bode plot that the –3dB point occurs around 30rad/s. This is considered the bandwidth of the closed-loop system. In other words, this controller can track signals that vary with frequencies up to 30rad/s (4.7Hz) without significant attenuation.
8.2.2. Experimental determination of bandwidth

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

Figure 10. Bode plot of the closed loop transfer function of the control system shown in Figure 9.

8.2.3. Position control

Figure 11 shows a position control loop for the lab DC servomotor. A variety of control strategies, represented by the Laplace function $C(s)$, 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.

8.2.3.1. PI controller

$$C(s) = K_p + \frac{K_i}{s}$$

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**

```c
error = (reference position - actual position)
integrated_error = integrated_error + error*Ts (Ts is the sampling period)
control voltage = Kp*error + Ki*integrated_error
```

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

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

does this, but remember to initialize `integrated_error` to zero to start.

8.2.3.2. PD controller

$$C(s) = K_p + K_d s$$

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**

```c
error = (reference position - actual position)
error_derivative = (error - previous_error)/Ts
control voltage = Kp*error + Kd*error_derivative
```
8.2.3.3. PID controller

$$C(s) = K_p + \frac{K_i}{s} + K_d s$$

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*T_s
- error_derivative = (error – previous_error)/T_s
- control voltage = $K_p*error + K_i*integrated_error + K_d*error_derivative$

8.2.3.4. 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

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

The controller gains $K_p$, $T_i$, and $T_d$ are calculated using the following design criteria.

**Ziegler-Nichols design rule**

<table>
<thead>
<tr>
<th>Type of controller</th>
<th>$K_p$</th>
<th>$T_i$</th>
<th>$T_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.5*K_crr</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PI</td>
<td>0.45*K_crr</td>
<td>$P_crr/1.2$</td>
<td>-</td>
</tr>
<tr>
<td>PID</td>
<td>0.6*K_crr</td>
<td>0.5*K_crr</td>
<td>0.125*K_crr</td>
</tr>
</tbody>
</table>

$K_crr$ and $P_crr$ are obtained by experiment. $K_crr$ is the critical gain where the shaft exhibits sustained oscillations. $P_crr$ 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 $K_d = K_p T_d$.

8.3. PRELAB

1. Modify the program you wrote in the Lab 7 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 $K_p$ 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_mfile`.

52
2. Determine the gain values for a PI controller and a PID controller using the Ziegler-Nichols tuning rule. Use the values for $K_{cr}$ and $P_{cr}$ that you experimentally determined in Lab 7.

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

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

8.4. LAB PROCEDURE

8.4.1. Exercise 1

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

<table>
<thead>
<tr>
<th>Input frequency (Hz.)</th>
<th>output/input</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

8.4.2. Exercise 2

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

8.4.3. Exercise 3

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

8.5. POSTLAB ASSIGNMENT AND LAB REPORT

1. 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.

2. 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?

3. 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).

**Lab 8 report requirements**

Solutions to the questions in Section 8.5.

All experimental plots obtained during the lab with appropriate titles.