LAB 6

CLOSED LOOP POSITION CONTROL

1. LAB OBJECTIVE

In this lab you will implement position control using simple proportional controllers. The response of the system to a step input will be experimentally determined and compared with theoretical values.

2. BACKGROUND

2.1 Block diagram

![Block diagram of closed-loop position control system of a DC servomotor.](image)

**Figure 1** Block diagram of closed-loop position control system of a DC servomotor.

The block diagram of a closed loop position control system is shown in Figure 1. The closed loop system transfer function is given by:

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

2.2 Step Response

Depending on system parameters and the controller gain that we use, the step response could be oscillatory (under damped), critically damped or over damped.

Critical damping condition

For critical damping, the denominator of the closed loop transfer function should have equal roots. The roots will be equal if and only if
\[ 1 - 4K_p K_m T_m = 0 \]  

or

\[ K_p^{cr} = \frac{1}{4K_m T_m} \]  

### 2.2.1 Step response using MATLAB

We can simulate the step response of a system using MATLAB. The following lines of code will work, assuming \( K_m, K_p, \) and \( T_m \) have been set to their appropriate values.

```matlab
numerator = Kp*Km; denominator = [Tm 1 Kp*Km];
sys = tf(numerator, denominator); step(sys);
```

Consider a second order system of the form as given above with \( K_m = 10 \) and \( T_m = 0.25 \). Then \( K_p = K_p^{cr} = 1/(4K_m T_m) = 0.1 \) for critical damping.

The following graphs (Figures 2, 3 and 4) show the step response for \( K_p = K_p^{cr} \), \( K_p < K_p^{cr} \) and \( K_p > K_p^{cr} \) respectively.

Try it yourself in MATLAB.

![Figure 2: Step response for a critically damped case](image-url)
**Figure 3** Step response for an underdamped case

**Figure 4** Step response for an overdamped case
3. PRELAB

Write a program that implements closed loop position control on the DC servomotor system. Use a step input of 500 counts (half a revolution). Use a sampling frequency of 1000 Hz. Use a gain $K_p$ of 0.008. Collect and store encoder data for 2 seconds, i.e. $2F_s$ or 2000 data points. Store the data in an integer array.

**Note 1:** A subroutine called `print_matlab` can be found in `step_plot.c` on the course website. This subroutine will accept an array of floats that represents time for the x-axis and an array of integers that represents encoder readings for the y-axis. An m-file called `step_plot.m` will be created that can be called from within MATLAB. This will produce a plot of the data. The function prototype for this subroutine looks like:

```c
int print_matlab(int N_POINTS, float *time, int *p_data);
```

**Note 2:** The control algorithm should be implemented to run in an infinite loop that is exited by hitting a ‘q’ on the keyboard. Remember that we want to collect data for only a half a second, though. Also make sure you reset the output of all DAC channels to zero when you exit the control loop. Otherwise the motor will keep rotating even after you stop your program execution.

You can use the following lines of code for inputting the reference values.

```c
while(key != 'q') /*stop when 'q' is hit*/
{
    /*read encoder*/
    /*calculate error*/
    /*write out voltage*/

    if(kbhit()) /*if you hit a key keyboard then read input*/
    {
        key = getch();
        if(key == 'M') /*if 'right arrow key' is hit ref_position += 500;*/
            ref_position += 500;
        else if(key == 'K') /*if 'left arrow key' is hit ref_position -= 500;*/
            ref_position -= 500;
    }

    /*timing calls*/
}
```

Every time you hit the right arrow key, this program will increment the reference position by 500 counts. When the left arrow key is hit, it decrements the reference position by 500 counts.
4. LAB PROCEDURE

4.1 Exercise 1

Implement your position control program. Connect the DAC0 channel to the motor input. Determine the experimental step response for $K_p = 0.008$. Write down the steady state error value. Remember to test as much of your program as you can using the scope before actually sending voltage signals to the motor.

Repeat the above steps for $K_p = 0.004$ and $K_p = 0.001$.

4.2 Exercise 2

In your position control program, gradually increase the gain $K_p$ in steps of 0.001 until the shaft exhibits sustained oscillations. Write down the gain value $K_{cr}^s$ (indices represent critical $K$ of the sustained oscillations) at this point. Estimate the period of sustained oscillations $P_{cr}^s$ of the shaft from the step response plot.

![Figure 5 Inducing sustained oscillations for estimating $K_{cr}^s$ and $P_{cr}^s$](image)

The value of $K_{cr}^s$ and $P_{cr}^s$ will be used in designing the gain value for a PID controller in the next lab. (Please note that $K_{cr}^s$ denotes the gain that induces constant oscillation. It is NOT the gain of critical damping).
5. POSTLAB ASSIGNMENT AND LAB REPORT

i. Using the open-loop transfer function obtained in the previous lab, derive the closed loop transfer function.

ii. For the above transfer function, determine the value of the gain that results in a critically damped system.

iii. Using MATLAB plot the theoretical step response of the system for the three gain values you tried in the lab.

iv. Find the pole locations of the transfer function (roots of the denominator) for the three different gain values that you used in the lab.

v. What is the experimental steady state error? Explain the reason for the steady state error if there is any.

Lab report requirements

- Solutions to questions in Section 5.
- Hard copy of your commented C-Program.
- Experimental step response plots.
- Plot of sustained oscillations and values of $K_{cr}^s$ and $P_{cr}^s$. 