ME 576 Lab 12 Digital Control of Servo Trainer and Command Generation

1. PD control implementation of the Servo Trainer

Using c=0 and J=0.002937 obtained from the system ID, the objective of this lab is to implement both continuous and discrete PD controllers as shown in Figure 1 to meet the following specifactions. (Ts=0.00442 sec).

- the closed loop system natural frequency $\omega_n = 4$ Hz,
- and the damping ratio of 0.5. Design the controller for both continuous and discrete implementations

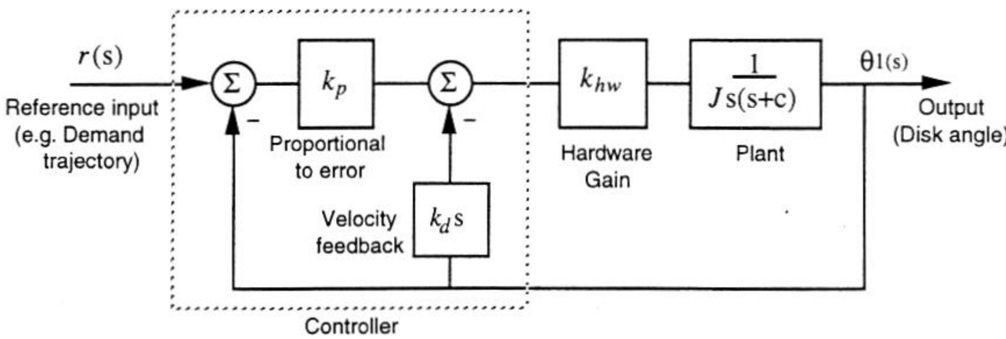

Figure 1 Controller Configuration for Plant Identification

Control Implementation Procedure

Procedure

Part A: Procedure (Continuous Time Control)

1. Enter the **Control Algorithm** box under the **Set-up** menu and set **Ts=0.00442 s** and select **Continuous Time Control**. Select **PI with Velocity Feedback** and **Set-up Algorithm**. Enter the control gain values you design and select **OK**.

   In this and all future work, be sure to stay clear of the mechanism before doing the next step. Selecting Implement Algorithm immediately implements the specified controller: if there is an instability or large control signal, the plant may react violently.

2. Select **Implement Algorithm** and then **OK**.

3. Enter the Command menu, go to **Trajectory** and select **Step**, Set-up. Select **Closed Loop Step** and input a step size of **1000 counts**, a duration of **1000 ms** and **1 repetition**. Exit the Background Screen by consecutively selecting **OK**. This puts the controller board in

a mode for performing a pair of closed loop steps (one forward and then one backward) of one second duration. This procedure may be repeated and the duration adjusted to vary the maneuver and data acquisition period.

4. Go to **Set up Data Acquisition** in the **Data** menu and select **encoder #1** and **Command Position** as data to acquire and specify data sampling every 1 (one) servo cycles, i.e., every Ts. Usually it is not necessary to acquire data at such a high frequency. Here however we wish to have high resolution data to make fairly precise measurements of the response frequencies. Select **OK** to exit. Select Zero Position from the Utility menu to zero the encoder positions.

5. Select **Execute** from the **Command** menu and select **Run**. The drive disk will step, oscillate, and attenuate, then return. Encoder data is collected to record this response. Select **OK** after data is uploaded.

6. Select **Set-up Plot** from the **Plotting** menu and choose **Encoder #1 Position** then select **Plot Data**. You will see the drive disk time response.

Part B: Procedure (Discrete Time Control)

1. Enter the **Control Algorithm** box under the **Set-up** menu and set **Ts=0.00442 s** and select **Discrete Time Control**. Select **PID** or **PI with Velocity Feedback** depending on your design and **Set-up Algorithm**. Enter the control gain values you design and select **OK**.

   In this and all future work, be sure to stay clear of the mechanism before doing the next step. Selecting Implement Algorithm immediately implements the specified controller: if there is an instability or large control signal, the plant may react violently.

2. Repeat the step 2, 5, and 6 shown in continuous time control part.

2. Command Generation II

INTRODUCTION

This week's lab assignment is to continue the experiment with more advanced features pertaining tomotion control of the servo table. In the last week's lab, you were exposed to simple command structures for the compumotor PC23 indexer. You will be required to use the handout given to you during the last lab along with the attached documents in order to complete this week's assignment.

Motion synchronization of multiple axes

If individual axes are controlled separately, the beginning and ending times of each axis would not be same. In order to avoid this problem, coordinating velocity profiles is necessary as we discussed in the lecture. The PC23 provides three ways to achieve this goal: using normal command mode, immediate velocity streaming mode and the distance command mode (see the attached document).

In this experiment, the normal command mode will be used to synchronize the motions of two axes in order to have a straight line motion in x-y plane. Assume that the maximum velocities for axis 1 and 2 are 5 and 10 rad/sec and maximum accelerations for axes 1 and 2 are 5 and 8 rev/sec2 respectively. The distances to travel are 1 and 0.5 inches in x and y directions. First, generate synchronized trapezoidal velocities of both axes (to be done with a pencil and a paper). The number of pulses per one revolution is 25,000 for both axes. The pitches are 0.2 in/rev and 0.02 in./rev. for axes 1 and 2 respectively.

Note : Modify the program EXAMPLE2 from line 5000 on in order to perform this task.

First use motion commands which control each axis separately. Name this program as TRAJSYNC.bas. While executing this program, measure the velocity profiles of axes 1 and 2 using the Tektronix (or HP) and plot the profiles.

<u>Items to be included in the report</u>

a) Printout of the measured velocity profiles.
b) Theoretically calculated velocity profiles.