

## 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 specifications. ( $T_s=0.00442$  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

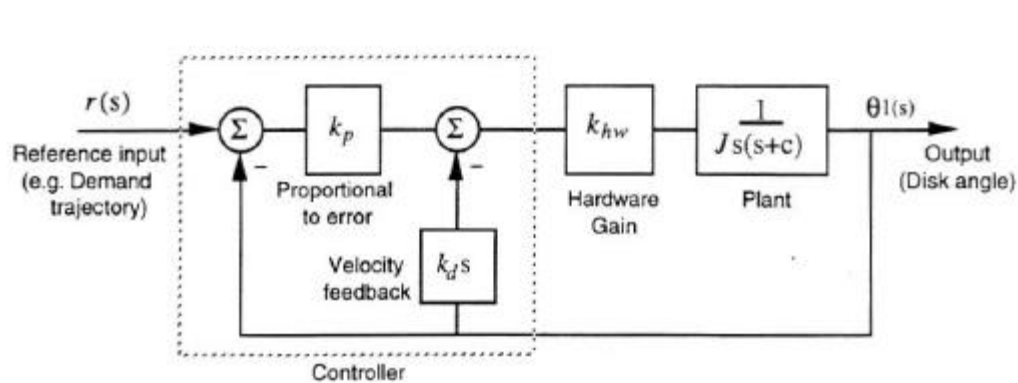


Figure 1 Controller Configuration for Plant Identification

## Control Implementation Procedure

### Procedure

#### Part A: Procedure (Continuous Time Control)

1. Enter the **Control Algorithm** box under the **Set-up** menu and set  $T_s=0.00442$  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  $T_s$ . 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  **$T_s=0.00442$  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. PID Control Implementation

The final part of this experiment is to design a PID controller for the table to meet the following performance criteria. The control implementation is shown for continuous and discrete versions in Figure 1. The sampling time interval is 0.01 sec.

Specifications:  $e_{ss} \leq 5\%$   
 $T_s \leq 0.2$  sec  
 % Overshoot  $\leq 10\%$

Use the following model to design the controller:

$$\frac{Vel(s)}{V_{in}(s)} = \frac{95.4(0.6)}{\left(\frac{s}{35} + 1\right)\left(\frac{s}{150} + 1\right)}$$

The gain of 95.4 relates the velocity of the table (ipm) to the feedback voltage.

Comments: Try to design the controller directly in the discrete domain. The control input panel shown in Figure 2 can take up to five digits (including decimal point). Therefore, the number smaller than 0.0001 will be considered zero for the controller parameter.

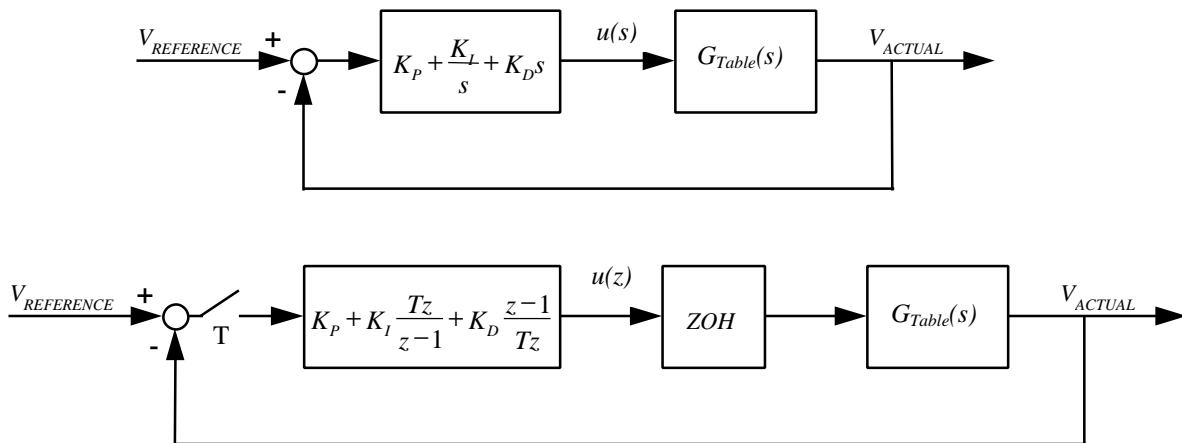


Figure 1 - Block Diagrams for PID Control

High Speed Mill

Start Table

Stop Program

Stop Table

Reference Velocity

Testing  
 Saving File  

File Name

P Gain

I Gain

D Gain

Control Signal

(v)				

Sample Number

Actual Velocity

(ipm)				

Sample Number

Figure 2 User Interface for Digital Controller Implementation (*MillPID*).