5 EXPT #3: IMPROVING SYSTEM PERFORMANCE

5.1 OBJECTIVE

- To study how to improve the system performance by PD, PI and PID control.
- To learn how to eliminate the steady state error caused by a disturbance input.

5.2 INTRODUCTION

5.2.1 PID-CONTROL

In the first part of this experiment, improvement in the steady state response by means of proportional-integral (or PI) feedback is investigated, together with the subsequent extension to Proportional-Integral-Derivative (PID) control. The effectiveness of PI control in reducing or eliminating the output-offsetting effect of a 'disturbance' signal is also demonstrated. And finally, a PID-control configuration exhibiting an 'optimum' performance selected by the student is set up.

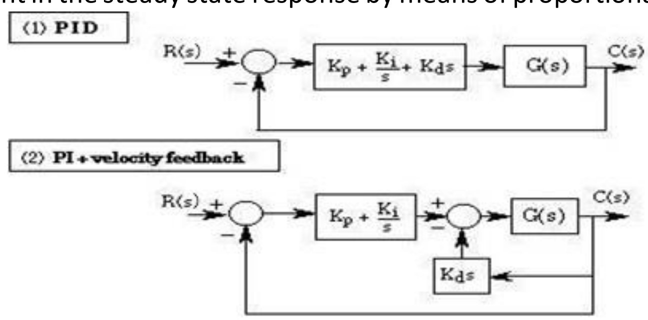


Figure 5.1: Closed-loop Configuration

The principle behind proportional-integral feedback is to increase the control effort (CE) with the objective of reducing or eliminating the steady state output offset, the added effort being obtained by *integrating* the error signal E(s). In the configurations #1 and #2 of above Figure, this integration is effective when a non-zero value for K_i is chosen. For stability considerations, integral control is always *added* to proportional control, resulting in the designation "PI-control".

In terms of 'error coefficients' adding an integral operation in the forward path is equivalent to increasing the system type number, thereby improving steady state error performance. In this experiment, the effects of adding integral feedback will be studied using the control scheme of #1 of Figure 5.1. The PI controller is implemented by selecting K_d =0 and non-zero values for K_p and K_i. Making the above selection is equivalent to introducing a transfer function:

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

in cascade with the plant which has the transfer function:

$$G(s) = \frac{K}{s(B + Js)}$$

In terms of the root-locus diagram, it can be seen that PI control adds a pole at the origin (increasing system 'type') and a zero at $s = -K_i/K_p$. If the added zero is close enough to the added pole, the effect on the transient response can be minimal, while still providing improvement in steady state performance due the increase in the 'type' number. It must be noted that the OL gain is multiplied by

K_p and a gain adjustment may be necessary. Considering the **unity feedback system** (UFS) OLTF of G_cG_p, however, it can be seen that the addition of integral control will also increase the system order. The increase in system order has the tendency of degrading the relative stability of the system which may be observed, for example, as an increase in the overshoot. This disadvantage can be overcome by introducing 'derivative control', thereby leading to what is known as "PID-control" (or "Three-Mode" control). For a PID controller, the transfer function is

$$G_C(s) = K_p + \frac{K_i}{s} + K_d s = K_d \frac{[s^2 + s\frac{K_p}{K_d} + \frac{K_i}{K_d}]}{s}$$

In terms of the root-locus diagram, a pole is introduced at the origin increasing the system 'type number' as before, and two finite zeros are introduced. The two zeros may be located on the real axis or may be complex conjugates in the 'left-half-plane'. (It can be shown that the latter condition will result if $K_p < 2\sqrt{K_i K_d}$). While the increase in system type improves the steady state error performance, the attendant degradation in relative stability is counteracted by the presence of the finite zeros in the 'left-half-plane'. PID control is commonly used in industrial control systems, the adjustment process for the three coefficients being called a "tuning" operation. Rather than determining the required settings analytically, various semi-empirical and well-established tuning procedures (such as the "Ziegler-Nichols tuning rules") are used in practice (See your course textbook for further information about this topic).

STEADY STATE ERROR

The 'error' $e(t)$ is defined as the difference between the command input $r(t)$ and the output $c(t)$; i.e. $e(t) = r(t) - c(t)$. The steady state or 'static' error (e_{ss}) is the final value of $e(t)$ that is reached as time t approaches infinity, i.e. $e(\infty)$. In the frequncy domain, the "Final Value Theorem" (FVT) can be used to obtain e_{ss}:

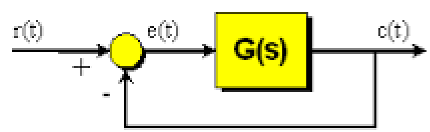


Figure 5.2: Unity Feedback System

$$e_{ss} = e(\infty) = \lim_{s\to 0} sE(s)$$

For a unity feedback system (UFS) in Figure 5.2, assuming it is stable, we have

$$E(s) = R(s) - R(s)T(s) = R(s)\left(1 - \frac{G(s)}{1+G(s)}\right) = R(s)\left(\frac{1}{1+G(s)}\right) \qquad (5.1)$$

where $G(s)$the transfer is function of the system and $R(s)$ is the input. e_{ss} can result in values of zero, a finite number, or ∞. For a given input, e_{ss} depends on the form of $G(s)$, especially on the presence and number of s terms in the denominator of $G(s)$ (i.e. the poles of $G(s)$ at the origin). For this reason, systems are classified in terms of the number of poles of the OLTF at the origin, which is called the system "Type number" (N). Thus, e_{ss} depends on both N and $R(s)$.

For example, consider the UFS OLTF $G(s) = 10/s(s+2)$. The system Type is 1 since there is one s term in the denominator of the OLTF. However, the order of this system is 2 since it has two poles. Now consider unit step ($R(s) = 1/s$) and unit ramp ($R(s) = 1/s^2$) inputs to the closed-loop system which has this OLTF.

For the step input,

$$e_{ss} = \lim_{s \to 0} s(\frac{1}{s})(\frac{1}{1+\frac{10}{s(s+2)}}) = \frac{1}{1+\infty} = 0 \quad \text{(i.e. zero error)}$$

For the ramp input,

$$e_{ss} = \lim_{s \to 0} s(\frac{1}{s^2})(\frac{1}{1+\frac{10}{s(s+2)}}) = \lim_{s \to 0} (\frac{(s+2)}{s(s+2)+10}) = \frac{2}{10} = 0.2 \quad \text{(i.e. a finite error)}$$

If the input is extended to a unit parabolic function $r(t) = t^2/2$ ($R(s) = 1/s^3$) and the application of the FVT as above will yield a $e_{ss} = \infty$ i.e. a Type 1 system will have zero error for step inputs, a finite error for ramp inputs and infinite error for parabolic inputs. Static error analyses, such as the above, can be standardized by expressing e_{ss} in terms of "error coefficients" (or error constants) k_j defined as follows:

$$\text{(General) Error coefficient } k_j = \lim_{s \to 0} s^j G(s) \quad \text{where} \quad j = 0,1,2,\ldots \qquad (5.2)$$

Following historical terminology used for early positional servomechanisms, the error coefficients corresponding to $j = 0,1, \text{and } 2$ are called the *positional, velocity* and *acceleration* error coefficients respectively. These are designated in this manual as k_{pos} , k_{vel} , and k_{acc} respectively, in order to avoid possible confusion with variables such as k_p used elsewhere in the manual. The three constants could alternatively be called the *step*, *ramp* and *parabolic* error coefficients. Thus

$$k_{pos} = \lim_{s \to 0} s^0 G(s) = G(0) \quad \text{and} \quad e_{ss}(step) = \frac{1}{1+k_{pos}}$$

$$k_{vel} = \lim_{s \to 0} s^1 G(s) \quad \text{and} \quad e_{ss}(ramp) = \frac{1}{k_{vel}} \qquad (5.3)$$

$$k_{acc} = \lim_{s \to 0} s^2 G(s) \quad \text{and} \quad e_{ss}(parabolic) = \frac{1}{k_{acc}}$$

The error shown is for a unit input and must be multiplied by the magnitude *A* of the actual input used. Steady state error can be expressed as the absolute value or as a percent of the input magnitude. In the latter case, the unit magnitude given by Equations (5.2) is in fact the decimal value of the percent figure. A table giving e_{ss} for the three different inputs as a function of system Type N is shown in table 5.1.

The above results indicate that:

(1) to follow a step input with finite error, at least a Type 0 system is required
(2) to follow a ramp input with finite error, at least a Type 1 system is required
(3) to follow a parabolic input with finite error, at least a Type 2 system is required

Also, it is worth mentioning that whenever a finite error results, its value will depend on k_j which in turn is dependent upon the system parameters such as the gain. In many cases, an attempt towards extreme minimization of e_{ss} by increasing gain, for example, may fail because the system becomes unstable at the selected gain.

Table 5.1. Steady state error for specified input and type number

Unit Input r(t) \ Type Number N	0	1	2
Unit Step u(t)	$\frac{1}{1+K_{pos}}$	0	0
Unit Ramp t u(t)	∞	$\frac{1}{K_{vel}}$	0
Unit Parabolic $[t^2/2]u(t)$	∞	∞	$\frac{1}{K_{acc}}$

5.2.2 DISTUTBANCE CONTROL

In regulatory systems, the error introduced by a disturbance input d(t) can be reduced to zero by PI control.

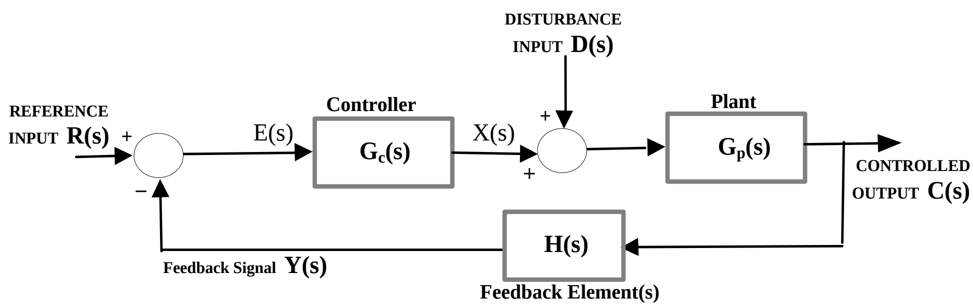


Figure 5.3: Disturbance control

The response of the closed-loop system to a disturbance D(s) can be investigated by considering D(s) as the input with R(s) set to zero. With R(s)=0, the effective 'disturbance transfer function' being given by

$$T_D(s) = \frac{C(s)}{D(s)} = G_p / (1 + G_p G_c H) = \frac{G_p(s)}{1+G(s)H(s)}$$

where $G(s)=G_p(s)G_c(s)$. It is seen that a high 'loop gain' G(s)H(s) will reduce the effect of the disturbance D(s).

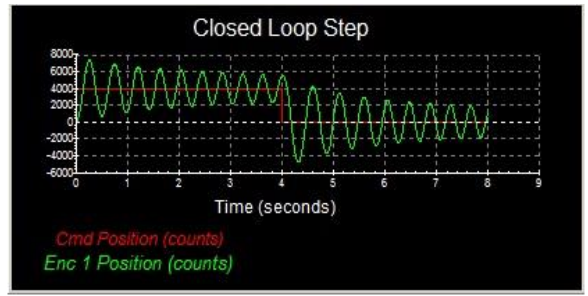
REMINDER: to use only the continuous domain specifications and procedures, ie under 'SETUP', all that is needed is to check the "Continuous Time" button and then "Set Up Algorithm", and then click on "Implement Algorithm" after setting the controller coefficients.

5.3 PID CONTROL

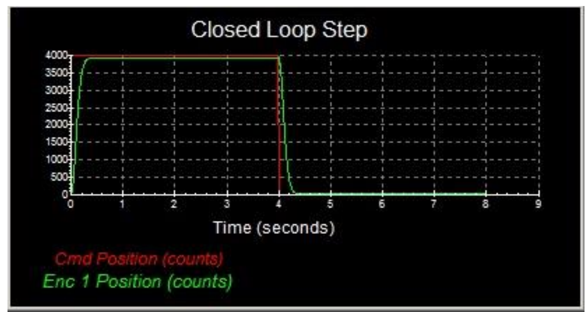
5.3.1 PI - CONTROL

The step response of a system with proportional controller is obtained first by setting $K_i = K_d = 0$ and K_p to some specific value. Viscous friction (input disturbance) is then introduced through the disturbance motor and its value is increased resulting in steady-state error in step response. Integral feedback is then introduced and its effect of reducing the steady state error is seen.

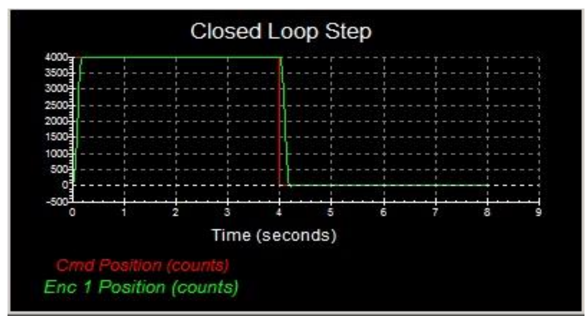
1) Reset the controller from the Utility menu. From the Set Up Control Algorithm implement the PID configuration (#1 in Figure 2.5) with $K_i = K_d = 0$ and with $K_p = 0.2$. Set up a closed-loop step response input using a step size of 4000 Counts, dwell time of 8000 ms. number of repetitions 1. RUN the step test and use axis scaling. A response similar to that shown below will be obtained. Determine* values for the overshoot (using suitable axis scaling of the plot) as well as the damped natural frequency ω_d.



2) Next, from the 'Command' menu, select 'Disturbance configuration', 'Viscous Friction' and set it to 4 volt-sec/radian. Then, after checking 'Implement' and OK, RUN the step test again. <u>Make sure to check the</u> "<u>Include Viscous friction" box</u> as well before clicking on 'RUN'. A step response, similar to the one shown below, should result. The response shows better damping but a finite offset in the steady state. Measure the offset error.



3) Re-implement the controller with K_i set to successive values from 0.02 to 0.1 in steps of 0.02 (you need to do it five times. For example 0.02, 0.04, 0.06, 0.08, and 0.1) and RUN the step test at each value. Select and note* the value of K_i which will minimize the steady state error to some acceptable value within 5%. Record* your observations. A typical response with K_p= 0.2, VF = 4 Volt-sec/radian and K_i = 0.03 is shown below.



RESULTS

1) Describe the effect of increase in K_i on the offset error and on the overshoot.
2) Derive the CLTF for the PID system (#1 in Figure 5.1) and obtain the relation between coefficients required for stability (Note that for *a cubic equation*

$$D_T(s) = a\, s^3 + b\, s^2 + c\, s + d = 0,$$

no root will have a positive real part if the condition $b \times c > a \times d$ is satisfied)

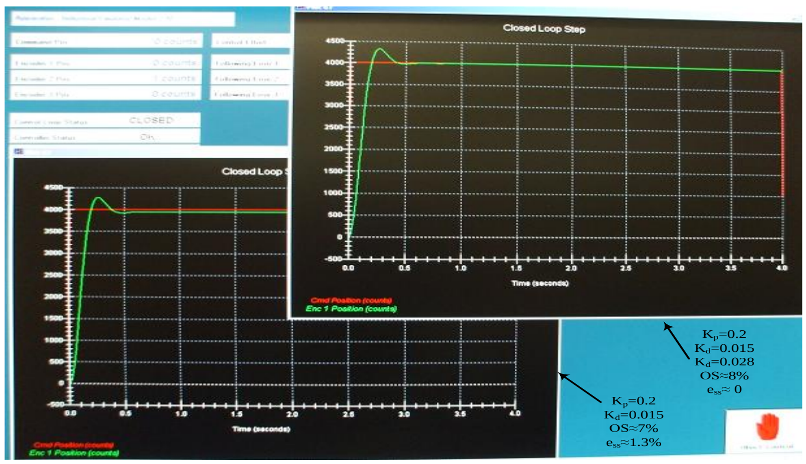
5.3.2 PID-CONTROL

1) From the Utility menu, reset the controller. Next, from the "Command" menu, remove the viscous friction and the disturbance input. Implement the PID controller with K_p= 0.2, and $K_i = K_d = 0$. Obtain the closed-loop step response using a step size of 4000 Counts, dwell time of 4000 ms, number of Repetitions 1. Display the response within the dwell period and reduce the display for later comparison. A typical plot that results from this step is shown below.

2) Successively increase the value of K_d in a small enough step, e.g. 0.002, and set its value so that a desirable overshoot (in your opinion) is obtained. To find the best value of $K_{d,}$ you could increase it up to 0.015. This value may vary around 0.015. Overshoot values in the 5 ~10% range are generally considered acceptable. An offset error will be noticed. Record the chosen value of K_d , the overshoot (OS) and the steady state error.
3) Next, successively increase the value of K_i in steps of 0.01 until the offset in the step response is almost reduced to zero. Record the chosen value of K_i. Typical displays that result from above Steps and the current one (after setting a suitable values) are shown below.

Results

Tabulate your choice of controller coefficients for PID control.

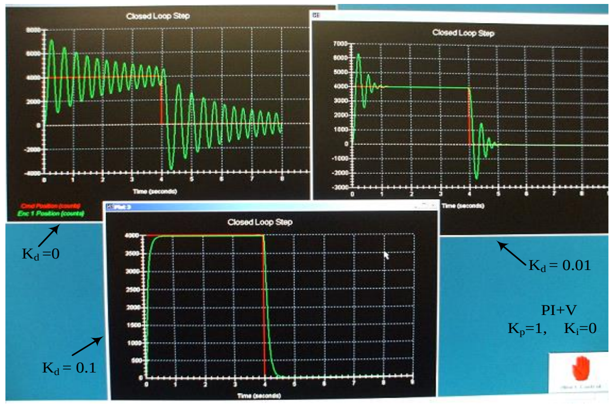


5.4 STEADY STATE ERROR ANALYSIS

5.4.1 USE STEP SIGNAL WITH PD

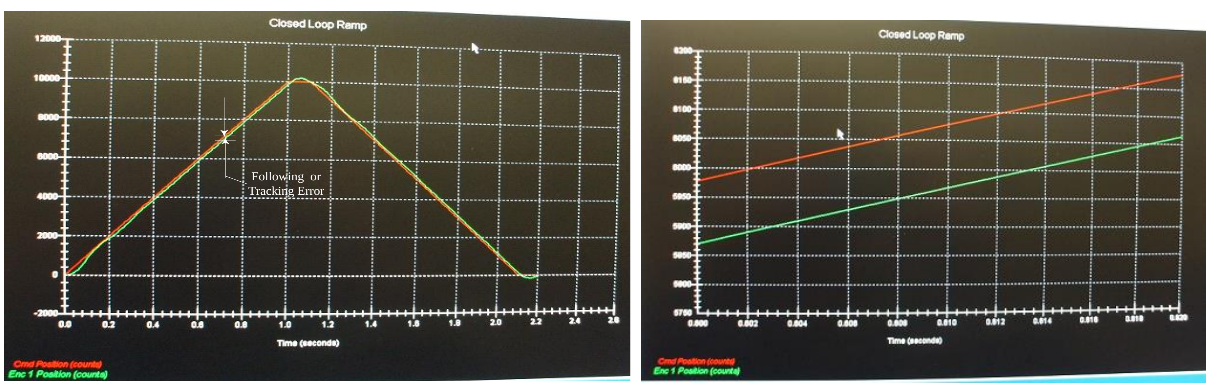
1) Reset the controller from the Utility menu. From the Set Up Control Algorithm menu implement the "PI + velocity feedback" with the following settings: T_s = 0.00442s, K_p = 1, K_i = 0 and K_d = 0.01 with Continuous Time type .
2) Perform a closed-loop step test using a Step Size of 4000 Counts, Dwell Time of 4000 ms and Number of Repetitions 1.
3) Observe the output which is highly oscillatory.
4) Click on Plot and view the waveform. Using axis scaling, obtain* the steady state step error (tracking error) in counts using the increasing part of the response.
5) Repeat above steps with K_p equals 0.5 and 0.2. Keep K_i and K_d unchanged. It will be seen that the system is now stable and an underdamped step response results. Obtain* the percent overshoot (PO) and the damped natural frequency ω_d using suitable axis scaling. Increase K_d to 0.1 and observe the effect on the step response.

Typical results show the stabilizing effect of increasing K_d are seen in the screen below.



5.4.2 USE RAMP SIGNAL

1) Reset the controller from the Utility menu. From the Set Up Control Algorithm menu implement the “PI + velocity feedback” with the following settings: $T_s = 0.00442s$, $K_p = 1$, $K_i = 0$ and $K_d = 0$ with Continuous Time type.
2) Set up and perform a RAMP test with a Ramp Size of 10000 counts, Velocity 10000 Counts/sec, Dwell Time of 100 ms, number of Repetitions 1 (these are the default values).
3) Observe the output which is highly oscillatory.
4) Click on Plot and view the waveform. <u>Using axis scaling</u>, obtain* the steady state Step error (tracking error) in counts using the increasing part of the response.



Expanded display using Axis Scaling to find the tracking error

e_{ss} can be readily found from the y-axis intercepts.

5) Repeat Steps 2-4 above using the same gains (K_p= 1, K_i = 0, and K_d= 0.01) but with the “PID” controller (#1 of Figure 2.5) implemented instead of the “PI +Velocity feedback” configuration. Note that in this case, there is no minor loop.

RESULTS

- Tabulate the results of the step and ramp tests for the two different controller implementations. Is there any change in the step response between the two cases? What is the difference in the ramp response between the two cases?

- The equivalent UFS OLTF and the CLTF for the “PI+Velocity feedback” system were already given. for both the systems. Derive the corresponding transfer functions for the system using the “PID” controller with K_i =0. Show that in both cases, the system Type number is 1.

Calculate the ‘velocity-error’ coefficients K_{vel} for the two cases.

The steady state ramp error is given by: e_{ss}(ramp) = Ramp magnitude / K_{vel}

Hence, for a ramp input of 10,000 Counts/sec,

$$K_{vel} = 10000 / e_{ss}(\text{Counts/sec})$$

Show that the velocity error-coefficients for the configurations #1 (PID) and #2 (PI+V) of Figure 5.1 with K_i =0 are respectively

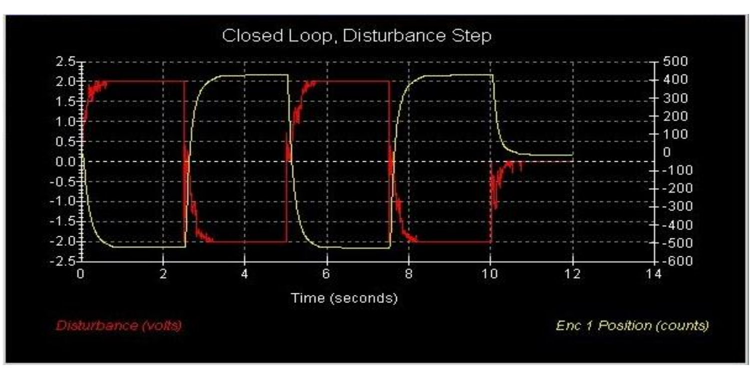
$$K_{vel\,(PID)} = KK_p/B \quad \text{and} \quad K_{vel(PI+V)} = KK_p/[B+KK_d].$$

Note that $K_{vel(PID)} > K_{vel(PI+V)}$ and hence the corresponding steady state ramp error for the PID controller will be the smaller of the two.

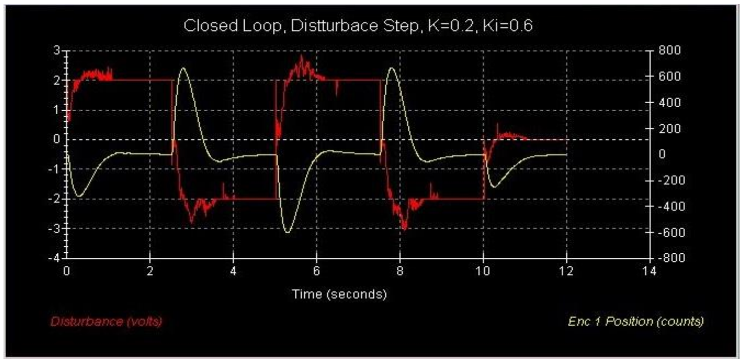
5.5 DISTURBANCE ATTENUATION

The effectiveness of PI control in eliminating the steady state error caused by a disturbance input will be demonstrated**.** Please note that the settings given are only typical since gain and inertia and friction vary from unit to unit.

1) Set up a PID control scheme with Continuous Time type, with K_p = 0.2 and K_i = K_d = 0 and with viscous friction introduced though the disturbance motor and set at 4 Volt-sec/radian.
2) From the ‘Command’ menu, select an input STEP trajectory with a magnitude of zero counts and 6000ms dwell, number of Repetitions 1. This configures the system to acquire data over a period of 12 seconds.
3) From the ‘Command’ menu again, select a disturbance step trajectory of 2 Volts magnitude, 2500ms dwell, and number of Repetitions 2. From Set Up Plot, select Encoder 1 Position for the Right axis and Disturbance Effort for the Left axis, after making sure that the latter has been added under Data Acquisition. Then RUN the system, after making sure that both “Include Viscous friction” and “Include Disturbance” boxes are checked. A plot similar to the one shown below should result.



4) Now, re-enter "Set up algorithm" and set K_i = 0.6. Implement the controller and repeat Step 3 above with the integral action included. The resulting plot will be similar to the one shown below. It is seen that the integral action attempts to 'force' the error to zero.



The error-reducing force can also be felt by hand: Re-enter the Command menu and set the disturbance to zero. Then implement the system and click on 'RUN'. With the thumb and forefinger of the left hand, gently attempt to disturb the load disk by a small angle, say ± 10º to 15º. The integral action resisting the displacement can be felt on your hand.

Since data has been uploaded by the attempted disturbance, the data-acquisition sequence will proceed and the disturbance that you manually introduced, as well as the resulting corrective action can be displayed. Note that if a larger disturbance is given, the system will overload and open the feedback loop. A typical plot of the action following two 'hand-induced' disturbances is shown below.

5) Repeat Step 4 above with lower and higher values of K_i and describe your observations. Use the results from Step 5 above to evaluate the maximum value of K_i that can be used with the given settings.

RESULTS

1) Obtain the equivalent OLTF $G_{ufs}(s) = T(s) /[1–T(s)]$ and sketch the root locus diagram for the system. Attempt a correlation with the location of the closed-loop poles.
2) Comment on any problems or differences encountered in this correlation.
3) [Optional] Use Simulink to simulate the system response in PID-CONTROL of 5.4 PID Control.

================================