

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
DEPARTMENT OF MECHANICAL ENGINEERING

2.004 *Dynamics and Control II*

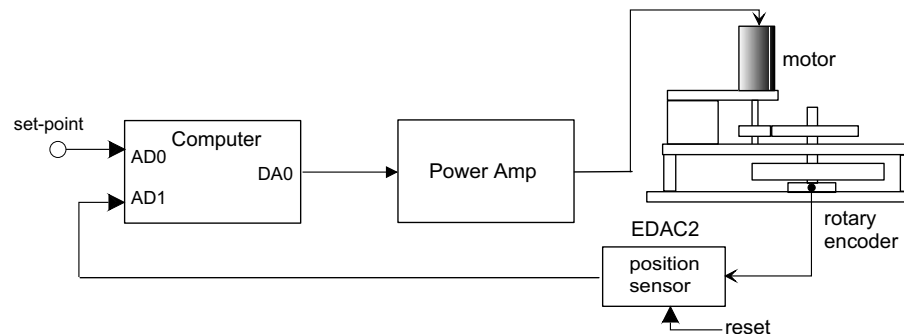
Laboratory Session 6:
Closed-Loop Position Control, and the Effect of Derivative Control Action¹

Laboratory Objectives:

- (i) To investigate closed-loop control of the angular *position* (θ) of the rotational plant.
- (ii) To investigate the manipulation of closed-loop dynamic response through the use of derivative (D), and proportional + derivative (PD) control.
- (iii) To compare your experimental results with a Simulink digital simulation.

Introduction: In all of our laboratory sessions to date we have investigated closed-loop control of the angular velocity of the rotational plant. In this session we switch to position control, and our goal is to command the flywheel to move to a given angular displacement. We will see that, for this particular plant, proportional control does not generate satisfactory transient behavior, and that the use of PD (proportional + derivative) control allows us to manipulate the closed-loop poles to achieve much improved response characteristics.

The Experimental Setup: The set-up is very similar to that of Labs. 4 and 5, except for the substitution of a US Digital EDAC2 position sensor for the ETACH tachometer module:



The EDAC2 produces an output voltage that is proportional to the angular position of the flywheel. Optical encoders, such as those used in this system, are *incremental* – that is, they do not have an inherent absolute zero position. You will notice a *RESET* button on the side of the module; this is used to set the current position of the flywheel as the “home” or zero position.

Experiment #1: Calibrate the Position Sensor: Use the DVM to calibrate the EDAC2 position sensor. Obtain the gain factor K_{pos} (with appropriate units), and investigate the

¹December 14, 2006

range of angular displacement over which there is a linear relationship between angle and output voltage. What happens when you exceed this limit?

Experiment #2: Proportional Control of Position: Using the information in Appendix A, derive the closed-loop transfer function for the system with proportional control.

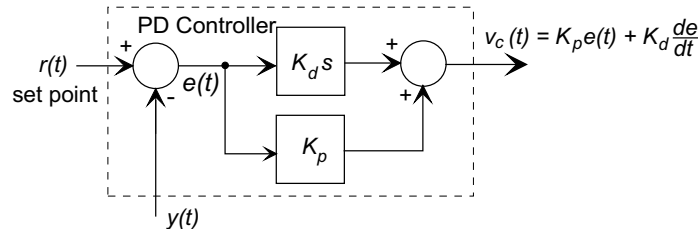
Set up your plant with proportional control ($K_i = 0$, $K_d = 0$), with a single damping magnet, using a 1 v DC signal from the function generator, and investigate the step response for $K_p = 1, 2, 3, 4, 5$. Remember to reset the EDAC2 between tests. Make a single plot of all the responses. Would you classify the response as “satisfactory”?

Examine your closed-loop characteristic equation. Is it possible to move the closed-loop poles deep into the left half-plane so as to control the settling time T_s ?

Introduction to PD Control: Proportional + derivative (PD) control is used to manipulate the closed-loop poles to improve the transient response of closed-loop systems. A PD controller has a transfer function

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

with a block-diagram



and a time-domain response

$$v_c(t) = K_p e(t) + K_d \frac{de(t)}{dt}$$

where $v_c(t)$ is the controller output.

A description of how the derivative component acts to modify the transient response is given in Appendix B. In practice pure derivative action is rarely used. Appendix B explains why, and briefly describes the use of a *band-limited differentiator*, or *pseudo-differentiator*. Please take a few minutes to read through and understand the Appendices.

Experiment #3: Verification of the Differentiator Performance The 2.004 PID controller uses a band-limited differentiator to reduce the effects of the high frequency noise at the output of the position and velocity sensors. Verify that this approximate differentiation produces satisfactory results using the following steps:

- (a) Connect the PID computer-based controller, but keep the power amp turned off.
- (b) Start the controller, and select a sampling rate of 100 samples/sec. (Maintain this value for all parts of the lab.)
- (c) Disconnect the position sensor from AD1 and ground the input connection using a 50Ω terminator resistor (*NOT* the one from your function generator) to prevent noise from contaminating the derivative action.

- (d) Set $K_p, K_i = 0$ and $K_d = 1$ on the front panel. For each of the following input waveforms (from the function generator with an offset of 0) start the controller and observe the output trace (grey/blue). Visually confirm that the output trace approximates the derivative of the input. Either save and plot the output, or make a sketch of it. Point out any differences between what you expect to see and the results you obtain.
- A 2 v p-p triangular wave at 1 Hz.
 - A 1 v p-p square wave at 1 Hz.
 - A 1 v p-p sinusoid at 0.1, 0.5 and 1 Hz.

Experiment #4: PD Control: In this experiment, use PD Control, that is with

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

- (a) Start with $K_p = 5$, $K_d = 0.5$, and a single magnet for damping. Use a 1 v p-p square wave at 0.1 Hz and record and save the step response. What is the peak current driving the motor with the square wave input? Is the controller “saturating”? Use the pan and zoom tools to select a complete positive step section of the response before saving it to MATLAB. Is the response with PD control more satisfactory?
- (d) Repeat (a) with $K_d = 1$ and 2. In each case remember to reset the EDAC2 before beginning, save the response to MATLAB, and make a plot of the positive step response.

Compare your three plots. Briefly describe how the value of K_d has affected 1) any “overshoot” in the step response, 2) the time to the peak response, and the time to reach the steady-state response.

Experiment #5: Compare your results with a Simulink Simulation: Build a Simulink model of the closed-loop system. We suggest that you use the Simulink model given to you for Lab 5 as the starting point, and modify it for this week’s configuration.

- (a) The Lab. 5 Simulink model is contained in the file *PIControl.mdl* in the Lab 5 folder of the 2.004 Course Locker on the lab machines. Drag the file to your desktop or home directory (Z:). Double-click on the file to start MATLAB, and open the model.
- (b) Modify the model. Use the Library Browser to find a differentiator amongst the Continuous blocks. Modify the plant transfer function, and any gain values.
- (b) Run the simulation for the cases you studied in Experiment 4.
- (c) Compare your experimental and the simulated data, in particular compare the transient performance of each.

Appendix A: The Plant Transfer Function for Position Control

In previous labs we found the open-loop plant transfer function relating angular velocity to controller output to be

$$\frac{\Omega(s)}{V_c(s)} = \frac{K_a K_m / N}{J_{eq} s + B_{eq}}$$

where $\Omega(s)$ is the angular velocity of the flywheel, $V_c(s)$ is the controller output (input to the power amplifier), K_a is the power amplifier gain, K_m is the motor torque constant, and N is the gear train ratio. We have measured or calculated the following values:

- $J_{eq} = 0.03 \text{ N.m}^2$
- $B_{eq} = 0.014 \text{ N.m.s/rad}$ (lab average)
- $K_a = 2.0 \text{ A/v}$
- $K_m = 0.0292 \text{ N.m/A}$ (lab average)
- $N = \frac{44}{180} = 0.244$

Since angular displacement θ is the integral of angular velocity Ω ,

$$\frac{\theta(s)}{\Omega(s)} = \frac{1}{s},$$

the transfer function for this week's experiments is

$$G_p(s) = \frac{\theta(s)}{V_c(s)} = \frac{1}{s} \cdot \frac{\Omega(s)}{V_c(s)} = \frac{K_a K_m / N}{J_{eq} s^2 + B_{eq} s},$$

which is a second-order system with a pole at the origin of the s -plane.

Appendix B: Introduction to Derivative Control Action

Consider the experimental second-order system with a transfer function

$$G_p(s) = \frac{K}{Js^2 + Bs}$$

With PD control, ie

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

the closed loop transfer function is

$$\begin{aligned} G_{cl}(s) &= \frac{G_c(s)G_p(s)}{1 + K_{pos}G_c(s)G_p(s)} \\ &= \frac{K(K_p + K_d s)}{Js^2 + (B + KK_d K_{pos})s + KK_p K_{pos}} \end{aligned}$$

During this lab session you will see that with proportional control, the system is very lightly damped, leading to an unsatisfactory transient response. However, with PD control the characteristic equation is

$$Js^2 + (B + KK_d K_{pos})s + KK_p K_{pos} = 0.$$

Notice that each of the feedback gains affects only a single term in the polynomial. For an underdamped system the real part of the poles is affected only by K_d . Therefore, K_d may be used to control the settling time of the system. K_p may be used to control the undamped natural frequency of the system, thus with the two parameters the designer has much greater ability to set the transient response of the closed-loop system.

We note in passing, that the PD control action has introduced a zero at $s = -K_p/K_d$, which must be taken into account when computing the transient response.

The most obvious way of computing a derivative in a digital computer is by using the backward difference, that is

$$\frac{de}{dt} \approx \frac{e_n - e_{n-1}}{\Delta T}$$

where e_n is the error measured at the n th iteration, and ΔT is the time step.

In practice pure differentiation is rarely used, because of amplification of high frequency noise. Consider the derivative of a sinusoid, $f(t) = \sin(\omega t)$,

$$\frac{df}{dt} = \omega \cos(\omega t)$$

and note that the amplitude of the output is proportional to the frequency ω . If there is any unwanted high frequency noise on a signal, it will be highly amplified by a differentiator, and if passed on to the plant can cause all sorts of operational problems

Therefore, practical PID controllers take preventative action to combat the noise. It is common to

- Use a smoothing (averaging) filter to minimize high frequency components in the error signal before differentiation, or

- Use a *band-limited* or *pseudo*-differentiator algorithm to approximate a differentiator at low frequencies, but not respond to high frequency inputs.

The 2.004 PID Controller uses a band-limited differentiator. It is designed to work up to a frequency of 0.1 of the sampling frequency, and attenuate all frequencies above that frequency. The design and implementation of the differentiator requires knowledge of digital signal processing, and is therefore beyond the scope of 2.004.