5 Lab 5: Position Control Systems

5.1 Introduction

In this lab, you will convert the DC motor to an electromechanical positioning actuator by properly designing and implementing a proportional (P) and a proportional-plus-derivative (PD) controller, both are variations of a proportional, derivative, and integral (PID) controller. An example of this positioning actuator is an automatic door of an accessible entrance.

This lab has three major components: analytical design, simulation, and implementation. The analytical design asks you to design each controller in the Laplace domain by simplifying the provided block diagrams and solving for unknowns. Next, you will simulate the design in Simulink to see if the gain values determined analytically meet the design specifications. Finally, you will implement these controllers in LabVIEW, test the gain values with the DC motor, and verify that your controller meets the specified design requirements.

5.2 Objectives

By the end of this lab, students will be able to:

- Design proportional, proportional + derivative, and proportional + speed controllers to certain specifications
- Implement and test the above controllers

5.3 References

See Section 7.10 and 9.8 of *Feedback Control Systems* (2011), Fifth ed. by Charles L. Phillips and John M. Parr. Note: the same information is in *Feedback Control Systems* (2000), Fourth ed. by Charles L. Phillips and Royce D. Harbor.

5.4 Lab Exercises

5.4.1 Design

In order to design the controllers, we must use previous lab results as well as an understanding of the proposed closed loop system. Refer to the block diagrams in Figure 1, Figure 2 and Figure 3 for guidance. Ensure to take into account all the gains!

1. The general first order approximation for the transfer function of a DC motor:

$$\frac{\Omega(s)}{E_a(s)} = \frac{K_m}{\tau_m s + 1}$$

   Using integration in the Laplace domain, derive the transfer function for motor position from the velocity transfer function above.

2. Place the position transfer function in a unity feedback loop with a proportional gain controller (Figure 1). Calculate the closed loop transfer function of this system. Hint: it should simplify into the form of an ideal second order system.

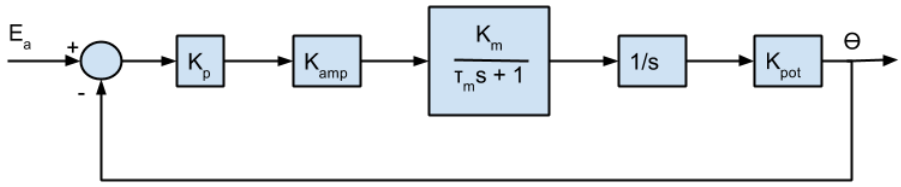

Figure 1: Proportional controller block diagram

Using the following values for K_m and τ_m, choose K_p (using Equation 1 and Equation 2) to satisfy the following designs with their constraints. If not possible, explain why:

K_m: 1.5 τ_m: 0.05

Design 1:

i. Peak overshoot < 10 % (Use Eq.1 below or text Fig.4.16)
ii. Settling time < 800 ms

Design 2:

i. Peak overshoot < 10 % (Use Eq.1 below or text Fig.4.16)
ii. Settling time < 180 ms

Equation 1: Peak overshoot

$$M_p = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}}$$

Equation 2: Settling time

$$t_s = \frac{4.6}{\zeta\varpi_n}$$

3. Change the controller in the previous question to a P-D as shown in Figure 2. Calculate the closed loop transfer function and choose values of K_d and K_p so as to satisfy the same design constraints as in question 2, Design 2. Hint: the closed loop transfer function's denominator that you find should simplify into the form of an ideal second order system.

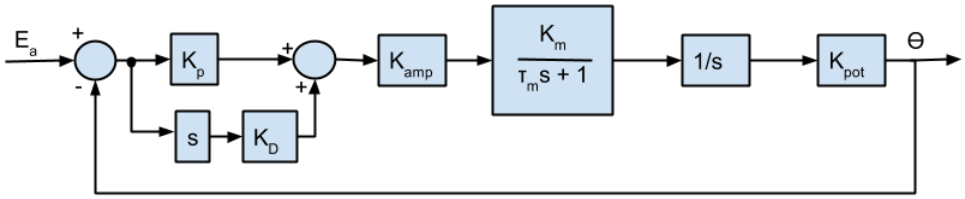

Figure 2: Proportional + Derivative Controller Block Diagram

4. Adjust the system to feedback the voltage from the tachometer as well as from the output potentiometer as in Figure 3. Then choose K_p and K_{p1} to satisfy the same design and the constraints from question 2, Design 2. Hint: the closed loop transfer function should simplify into the form of an ideal second order system.

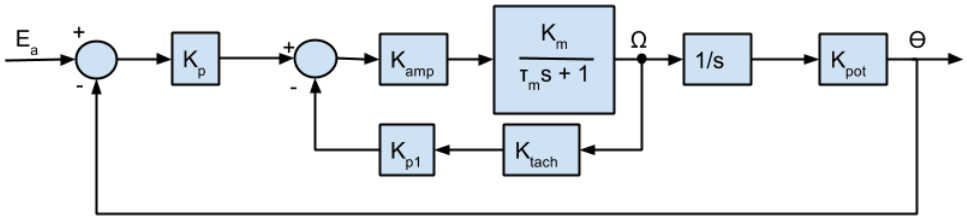

Figure 3: Proportional + Speed Controller Block Diagram

5.4.2 Simulation

Using MATLAB's Simulink, create three block diagrams to simulate the systems in questions 2, 3, and 4 using the gains you calculated in those questions. See Figures 4, 5 and 6. Load the starter file "GE320_lab4starter.mdl" by typing its name at the matlab command prompt. Make sure to save this file with a new name. In addition, use Figures 1, 2, and 3 as a guide to create your Simulink simulations. Ensure that you include all the following gains and system components:

Motor Potentiometer Gain:	Use the value you found in Lab 1 (but also given to you in the figures below)
Signal Generator:	Use a square wave with a 0.25Hz frequency and amplitude of 1V.
Constant:	Offset the signal input by 2.5V.
Gains:	K_p, K_d and K_{p1} depending on the controller.
Amplifier Gain:	It has a gain of 2.4.
Derivative Approximation:	Use a transfer function from the Continuous set of blocks in the Simulink library.
Transfer function:	Use the Gain of 1.5 and time constant of 0.05. For this lab, leave the friction value at 0.
Saturation:	Limit the control input signal from -100 to 100.
Random Number:	Set Mean to 0, Seed to 0, Sample Time to -1 (inherited). Initially set Variance to 0.

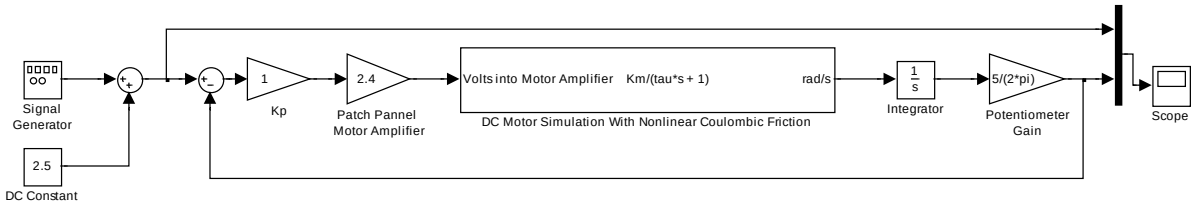

Figure 4: Proportional Gain Control Simulink Block Diagram

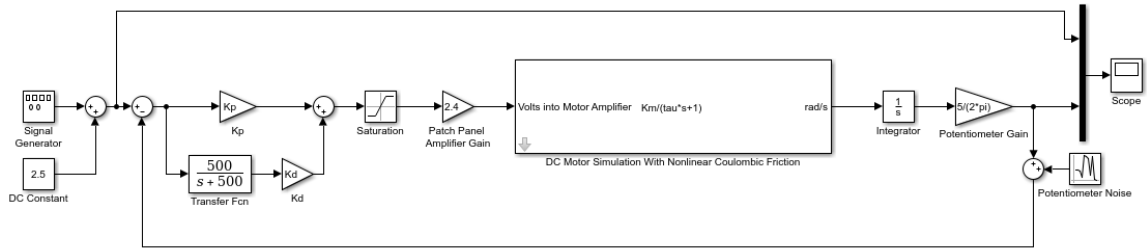

Figure 5: Proportional-Derivative Controller Simulink Block Diagram

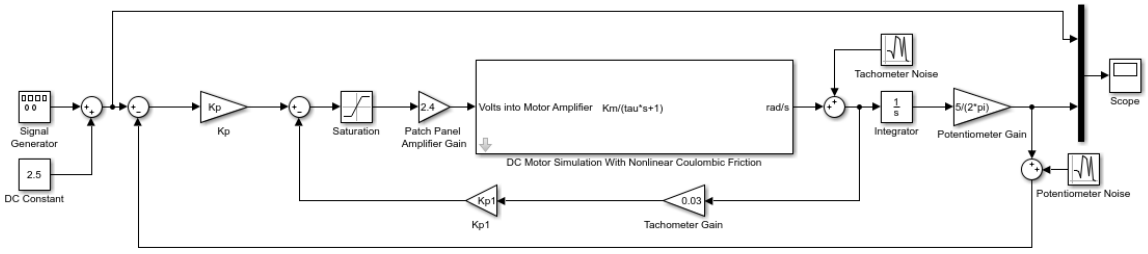

Figure 6: Proportional and Speed Controller Simulink Block Diagram

Show all the responses to the TA. With the control gains computed, check if all the design specifications were met.
After testing the values for the gains you obtained in the design part of the lab, we will examine the effect of noise on the performance of the Proportional-Derivative and Proportional + Speed Controller. In each random number block, set Variance to 0.01. Look at the scope of the output and compare the effect of noise on each controller. Experiment with different values for noise. Does either controller seem to be more effected by noise than the other?

Record the values for the gain and general comments regarding overshoot and settling time in your data sheet.

5.5 Post-lab

1. What are the performance differences between the controllers? What about their settling time? Percent overshoot?
2. Which of these controllers do you see having problems in a noisy (electrical noise/interference) environment? Why?

5.6 Data Sheet

	Design and Simulation
Kp	
% Overshoot	
Settling time	
Comparison to simulation's Overshoot and Settling Time	

	Design and Simulation
K_p	
K_d	
% Overshoot	
Settling Time	
Comparison to simulation's Overshoot and Settling Time	

	Design and Simulation
K_p	
K_{p1}	
% Overshoot	
Settling time	
Comparison to simulation's Overshoot and Settling Time	