EE443L Lab 5: DC Motor Position Control

Introduction:

Applications such as robot joint control often require precise positioning of DC motors. This lab investigates the use of proportional position control for DC motor positioning as shown in figure 1.

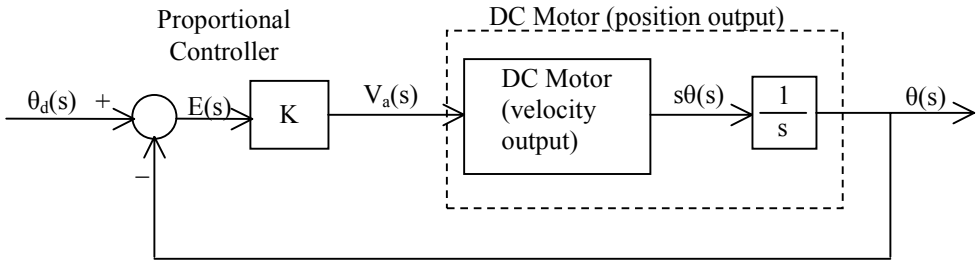

Figure 1: Proportional Position Control of a DC Motor

Prelab:

1. Determine the motor position transfer function $\theta(s)/V_a(s)$ (in general without numerical motor parameters and controller) that relates motor position to input voltage. How many poles does this transfer function have at s = 0? The number of poles at s = 0 is referred to as the system type and directly influences the steady state error of the closed loop control system. What type was the DC motor model with velocity output used in the first few labs?

$\theta(s)/V_a(s)$ = _____ position system type = _____
velocity system type = _____

2. Determine the closed-loop system transfer function $\theta(s)/\theta_d(s)$ in terms of the motor parameters in general and controller gain K. Note that this transfer function relates the motor position to the desired motor position.

$\theta(s)/\theta_d(s)$ = _____

3. What values of K will yield zero steady state error for a fixed desired motor position? Explain why this result occurs by comparing the proportional closed loop motor position control system of this week with the proportional closed loop motor speed control system of last week.

4. What effect will varying K have on the system's transient and steady state responses?

5. Simulate your proportional position controlled DC motor with Simulink for $\theta_d(t)=3u(t)$ rad to determine a range of controller gains K that correspond to peak motor armature voltages V_a between 5V and 15V. Plot both the motor voltage $V_a(t)$ and position $\theta(t)$ for both minimum and maximum gain values and indicate your selected K values.

Laboratory Procedure:

1. Download the LabVIEW VI *lab5.vi* and its associated subVI *Counter_Setup.vi* from the directory *N:\ee443l\Lab 5*. This VI utilizes a counter on the PCI-6025E data acquisition board connected to a LS7084 quadrature clock converter as shown in the lab motor connections. This implementation utilizes the rising and falling edges and phase difference of the two quadrature encoder signals to yield 2000 counts per revolution of the motor from 500 count encoders. Run the VI without motor power and turn the motor one full revolution in both directions to verify the 2000 counts per revolution.

2. Create a proportional closed loop position controller as shown in figure 1 using the components of lab5.vi for position feedback and components of previous labs for PWM and hardware-timed loops. Keep in mind the following:
   a) the 2000 counts/revolution will need to be converted to radians,
   b) PWM generation and position and error plotting will need to be added,
   c) the analog input setup of past labs can be utilized to achieve accurate timing and as long as the analog input is activated motor velocities from frequency to voltage converter can be plotted for observation,
   d) ensure positive PWM corresponds to increasing position, otherwise we'll have positive feedback,
   e) ensure neatness as VI will become complicated,
   f) sampling rate may need to be adjusted (try to keep it as fast as possible) with control loop performing more operations.

3. Rotate the motor to 0 rad (0 deg on the motor label) by hand let $\theta_d(t) = \pi\, u(t)$ rad which turns on after the control loop starts (use loop counter to switch on desired value after 50 loops) to get a nice view of the step response. Run your controller for three values of K within the range determined from the prelab. Printout responses (position, velocity, and error) for these three K values and note how varying K affects the steady state error, rise time, and settling time.

4. Questions:
   a) After implementing proportional control for both velocity and position, which works better? Support your answer through observed performance.
   b) They are both proportional control, so how can one work better than the other?
   c) Does a good system model help you predict how well your proportional controller will work?