EE443L Lab 6: DC Motor Trajectory Tracking

Introduction:

Many applications require a motor to accurately track a position profile that varies smoothly over time rather than a sharp transition from one fixed value to another. This type of control objective is referred to as trajectory tracking and can serve two purposes. The first application of trajectory tracking is for tasks such as robotic welding, robotic spray painting, or radar target tracking where a motor driven system needs to be precisely positioned for each instant of time. The second usage is for transitioning a motor driven system such as a large antenna from one fixed position to another without the sudden motions and excessive control action often required for following a step transition. This lab investigates the use of proportional and integral (PI) position control along with proportional (P) velocity control for DC motor trajectory tracking as shown in figure 1. Making use of both position and velocity information in the control system gives the controller more information about the task and should improve tracking accuracy.

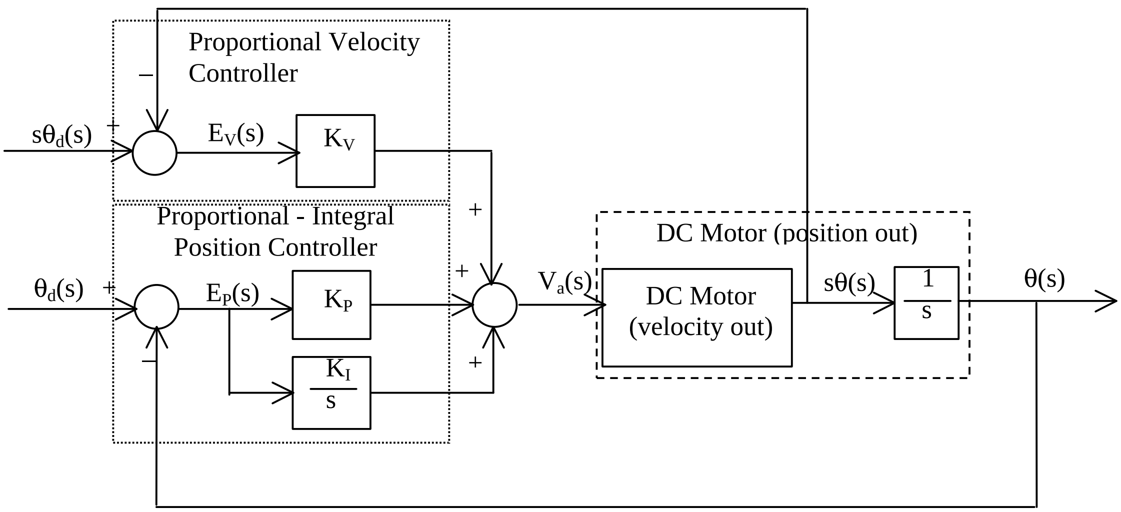

Figure 1: Proportional-Integral Position plus Proportional Velocity Control of a DC Motor

Prelab:

1. Trajectory generation is the first step in tracking control and many techniques exist for connecting points with a smooth curve. One of the simplest is to use sines and cosines for desired position $q_d(t)$ and velocity $\dot{q}_d(t)$ profiles as shown

$$\begin{aligned} q_d(t) &= q_0 + \left(\frac{q_f - q_0}{2}\right)\left(1 - \cos\left(\frac{pt}{t_f}\right)\right) \\ \dot{q}_d(t) &= \left(\frac{q_f - q_0}{2}\right)\left(\frac{p}{t_f}\right)\sin\left(\frac{pt}{t_f}\right) \end{aligned} \tag{1}$$

where q_0, q_f are the initial motor position and desired final motor position, respectively, and t_f is the final (stop) time of the trajectory assuming the start time is zero.

a) Verify $\dot{q}_d(t)$ is the time derivative of $q_d(t)$ as given in equation 1, so that you're convinced these are compatible position and velocity trajectories.

b) Evaluate $q_d(t), \dot{q}_d(t)$ in equation 1 for time values of $t = 0$ and $t = t_f$ to verify the desired trajectory starts with zero velocity at position q_0 and ends with zero velocity at position q_f.

c) Plot $q_d(t), \dot{q}_d(t)$ as specified in equation 1 using simulink for parameter values of $q_0 = 0\text{ rad}$, $q_f = 3p/2\text{ rad}$, and $t_f = 1\sec$. Verify smooth profiles and beginning and ending points.

2. The control system depicted in figure 1 has two inputs and one output making system analysis difficult; therefore, for ease of investigation, the control system can be redrawn in true PID form as shown in figure 2. Explain what was done in this reduction.

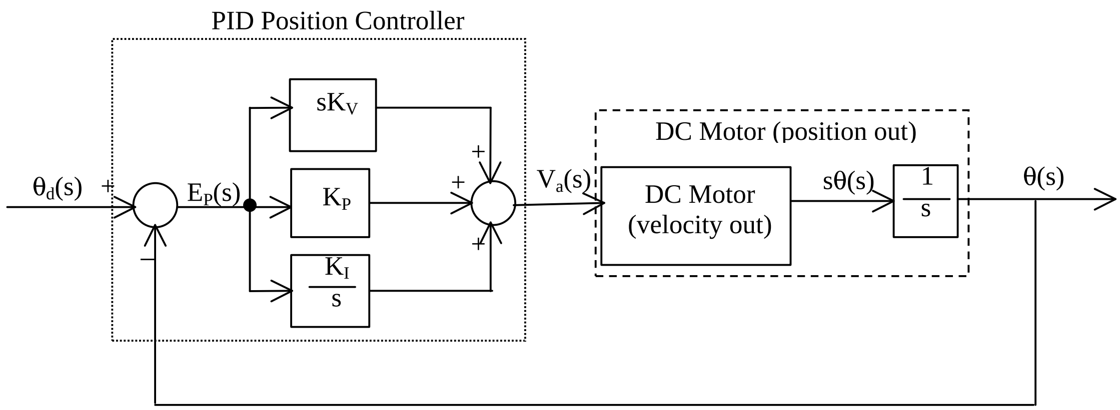

Figure 2: Proportional-Integral-Derivative (PID) Position Control of a DC Motor

3. For the PID control system of figure 2, find the forward path transfer function $q(s)/E_P(s)$ in general.

4. For the PID control system of figure 2, find the closed-loop transfer function $q(s)/q_d(s)$ both in general and with motor parameters substituted.

5. Note that with three controller gains (K_V, K_P, and K_I) in the characteristic polynomial, there is a lot of flexibility for pole placement. The problem becomes how to place them for stability and desired performance. Luckily, in the past two labs we have performed proportional velocity control and proportional position control which are essentially the derivative (K_V) and proportional (K_P) terms in the PID controller. Choose K_V to be a good gain value from lab 4 and K_P to be a good gain value from lab 5 as starting points. Check that these gains will yield a stable system and find a range for K_I that ensures stability by utilizing a Routh array.

Laboratory Procedure:

1. Download the matlab m-file *lab6.m* located in the network directory *N:/ee443l/Lab 6/*. This m-file plots the closed-loop system poles and zeros as gain K_I is varied (gains K_P, K_V held constant).
   a) Change the motor parameters to those you've determined for your motor.
   b) Enter your values of K_P, K_V chosen in the prelab and change the range of K_I to that found to yield system stability in the prelab.
   c) Check to make sure the numerator and denominator of the closed-loop transfer function utilized in the m-file are the same as the ones you found in the prelab.
   d) Run the m-file in matlab to view how the poles and zeros vary with K_I and verify that the system will become unstable as the maximum K_I value is approached.
   e) Choose K_I from your pole-zero plot such the dominant poles result in an overdamped system.

2. Simulate your motor tracking system in Simulink as shown in figure 1 using the trajectory generator created in the prelab, motor model developed in previous labs, and gains determined above and in the prelab. The simulation serves to confirm that the chosen gains will achieve reasonable performance.

3. Start the hardware portion of this lab with your LabVIEW VI created for lab 5. It should contain motor position input (converted to radians), motor velocity input(s) (converted to radians/second), and PWM output (converted from armature voltage and limited). Note: By the end of lab the VI will be very crowded, so use care to make it as neat as possible.

4. Add trajectory generation of the type given in equation 1 using either multiple blocks (sinusoids, multiplies, etc.) or use a *formula node* where you can enter formulas into one block. See the online help for more information on *formula nodes*. Use the same trajectory parameters as in the prelab.

5. Modify your VI such that desired and actual positions ($q_d(t), q(t)$) are plotted on one graph and desired and actual velocities ($\dot{q}_d(t), \dot{q}(t)$) are plotted on another. Run your VI with no controller (i.e., zero PWM) and verify that your desired trajectories are generated correctly.

6. Add the PI position and P velocity control algorithms shown in figure 1 to implement the control approach on the DC motor. Implement the integral as simple Euler integration shown in equation 2 noting that *shift registers* will need to be used to initialize and compute the sum. See the online help for more information on *shift registers* and use gains K_V, K_P, and K_I determined above. After running your VI and successfully controlling the motor, make any necessary adjustments to the gains to further enhance performance and print the final VI and motor position and velocity responses. Comment on any differences between simulated and actual responses, problems, or unforeseen difficulties.

$$\int_{t=0}^{t} e(t)\,dt \approx \sum_{i=0}^{n} e[i]\Delta t \qquad (2)$$

7. Questions:
   a) Give two examples (other than those listed above) of where trajectory tracking would be preferred over step responses.
   b) Why was the controller utilized in this lab (see figure 1) described as PI position with P velocity rather than PID (see figure 2)? Is there a difference?