

University of Utah
Electrical & Computer Engineering Department
 ECE 3510 Lab 5b
Position Control
Using a
Proportional - Integral - Differential (PID) Controller

M. Bodson, A. Stolp, 2/19/08
 rev, 2/24/09, 2/28/17 B. Mukunda, 3/2/19

Note: Bring the lab 2 and Lab 5a handouts as dSPACE reference material.

Objectives

- Analyze a type of PID control system.
- Modify the dSPACE programming to implement the PID control law.
- Observe the step response with different zero locations (as set by the feedforward gain).
- Compare the tuned PID response to the simple proportional response.

Equipment and materials from stockroom:

- DC Permanent-magnet Motor (If you can, get the same motor you used last time.)
- Dual Power Amp
- DSPACE cable

Experiment 3, PID Position Control

In this experiment you will construct a position controller, so you'll need to modify the model of the DC motor to be angular position over voltage rather than angular velocity over voltage. Write your motor transfer-function as:

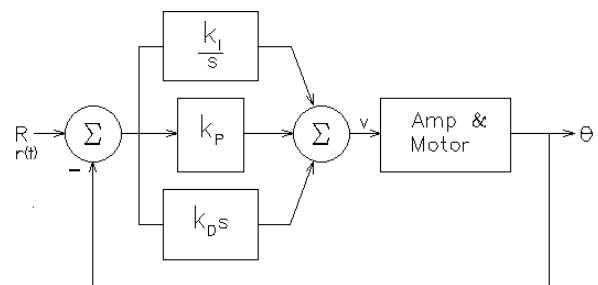
$$P(s) = \frac{\theta(s)}{V(s)} = \frac{k_1}{s(s+a_1)(s+a_2)}$$

In lab 3 you used a proportional control law to control the motor position.

$$v = k_p (r - \theta)$$

In this experiment you will use a variation of the *proportional-integral-derivative* control law.

$$v = k_P (r - \theta) + k_I \int (r - \theta) dt + k_D \frac{d}{dt} (r - \theta)$$

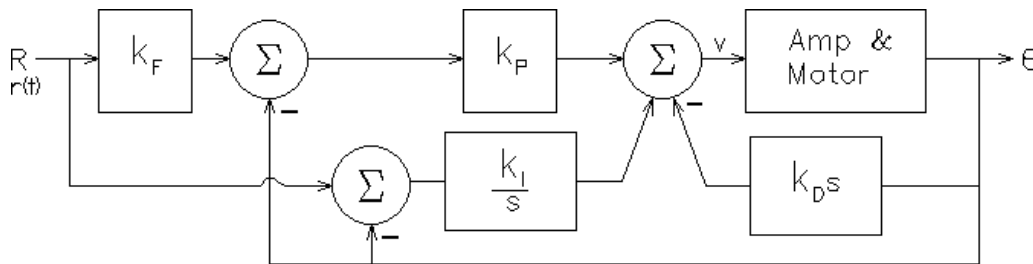


The integral term will minimize the steady state error as you saw with velocity control in experiment 2. The derivative term can be viewed as a proportional feedback acting on the velocity error. Responding to this additional error can increase the speed of the response-- a good thing. Ah, would that life were so simple... Think what would happen if you input a step response? What is the derivative of a step? Mathematically no big deal, the control system delivers an impulse to the motor which certainly speeds up the response. The problem arises with the limits of a system implemented with real parts. No real amplifier can deliver

an impulse, so instead, it “rails” for a very short time and has a minimal impact on the response. (Actually, this might be a blessing in disguise, given that an impulse would mean an infinite voltage— probably not too safe.)

In practice, the control law is often modified in two ways. First, to avoid the problem with step inputs, the derivative action is applied only to the output (θ), and not to the reference input (r). This also eliminates a zero in the overall (closed-loop) transfer function. (No, I don’t expect you to just see that, it takes a bit of math to show that that is the case.) Second, the reference input to the proportional gain is modified by a feedforward gain (k_F). This is not done to adjust the DC gain (as in experiment 1), but rather to gain control over the placement of the remaining zero of the closed-loop transfer function. Again, this is not obvious, but you will see it come out in the math. The modified control law is given by:

$$v = k_P(k_F r - \theta) + k_I \int (r - \theta) dt - k_D \frac{d}{dt} \theta$$



Verify that the resulting closed-loop transfer function for the system is: (This may be done later if you’re short on time, but leave an empty page in your notebook.)

$$H_{CL}(s) = \frac{\theta(s)}{r(s)} = \frac{k_I k_P k_F (s + \frac{k_I}{k_P k_F})}{s^4 + (a_1 + a_2)s^3 + (a_1 a_2 + k_I k_D)s^2 + k_I k_P s + k_I k_I}$$

Now again, set the poles somewhat like those of critical damping, that is, make all the poles the same, say at $-p$. That would make:

$$\begin{aligned} s^4 + (a_1 + a_2)s^3 + (a_1 a_2 + k_I k_D)s^2 + k_I k_P s + k_I k_I &= (s + p)^4 \\ &= s^4 + 4ps^3 + 6p^2s^2 + 4p^3s + p^4 \end{aligned}$$

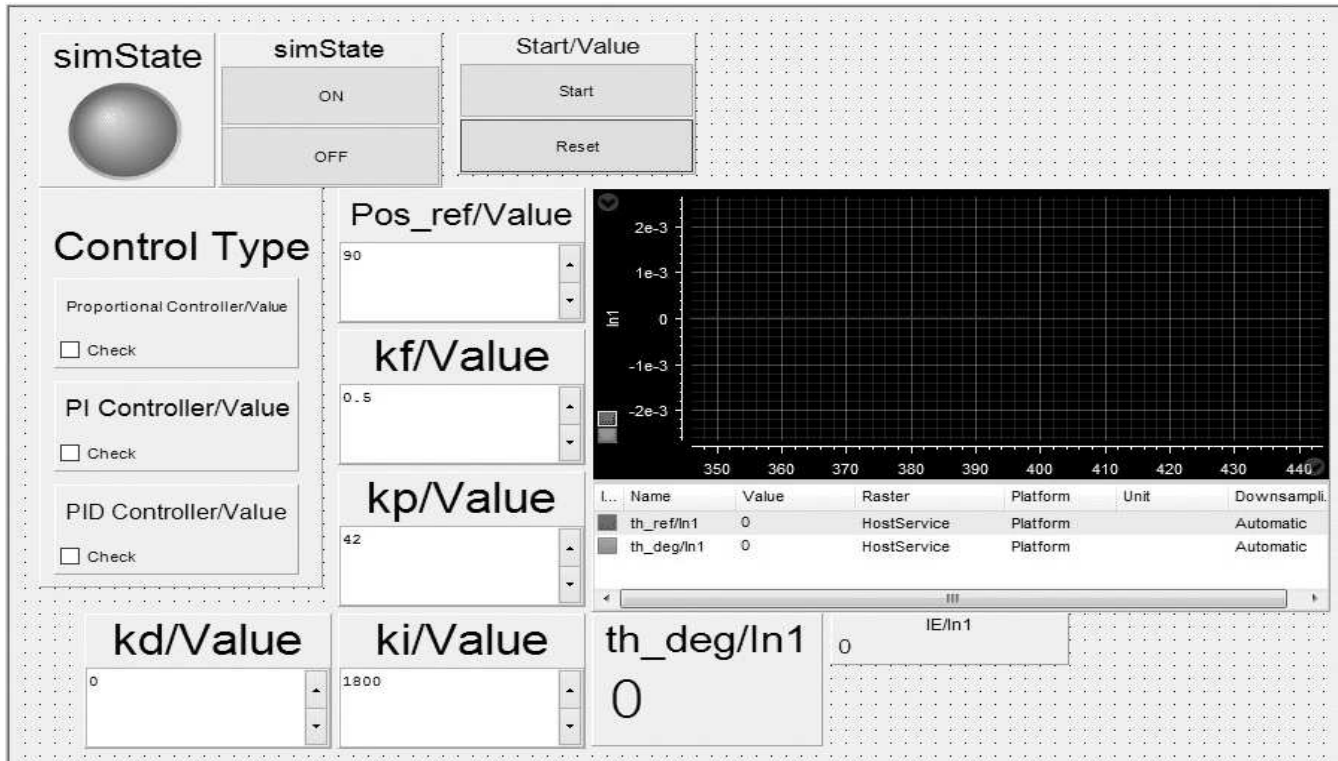
Find p , k_D , k_P , and k_I . They should be about 180, 0.2, 30-42 and 1000-1900 respectively.

Notice that the closed-loop transfer function has a zero at $-k_I/(k_F k_P)$. With a neutral feedforward gain ($k_F = 1$) the zero may have a small magnitude compared to the closed-loop poles, yielding overshoot in the step response even if the closed-loop poles are well-damped. (See section 3.3.3 (p.52-3) in the text.) Find the position of the zero if $k_F = 1$ then show that if $k_F < 1$, the zero will be further in the left-half plane and the step response will be better.

DSPACE

Download the current “PID_Control_DC_Motor_xxx.zip” file from the class website. Set up the experiment just like you did in Lab 5a with some minor differences. Import the new

recorder file and use the new Trigger Rule 1 for the start trigger. A Time-Limit stop condition of 1 second will work well. You may want to change the stop condition later if you don't like your plots. Your layout should now look something like the next figure.



Proportional-Integral-Derivative (PID) Control

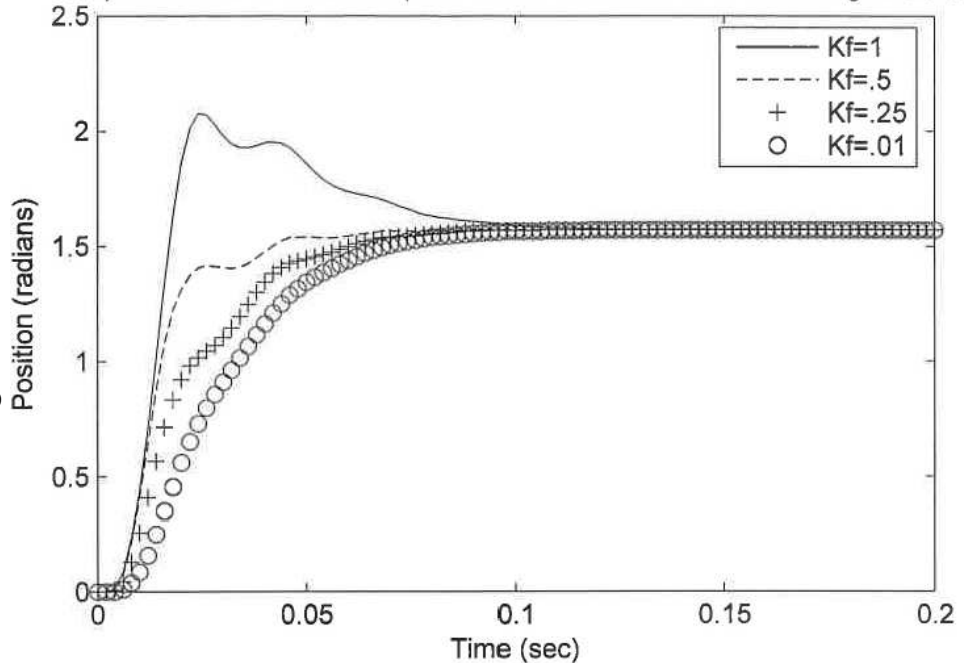
- Set up your recorder but don't hit "Start Triggered" for now.
- Once online and Measuring, turn ON the simulation (simState) and make sure the "Reset" under "Start/Value" is blue, otherwise, click it.
- Enter the k_D , k_P , and k_I values you calculated earlier and set the "Feed-forward gain", k_F , to 1.
- Set the "Pos_ref" to 90. This is the reference input angular position in degrees.
- Find the boxes under the graph labeled "th_deg/ln1" and "IE/ln1". If they are not both zero hit "Start" and "Reset" alternately until both are zero. "th_deg/ln1" is the measured motor position and "IE/ln1" is the integrator output before multiplication by k_I .
- Check the "PID Controller" in the Control Type panel. **NOTE:** If the motor ever starts to oscillate or run on in one direction, uncheck the "PID Controller". Then re-zero the boxes under the graph like you did in the previous step. Before trying again, check and possibly reset your k values. Still having trouble? Try conservative values of 0.2, 30 and 1000 for k_D , k_P , and k_I .
- As soon as you hit "Start", the motor should spin to 90 degrees and stop, giving you a step response with considerable overshoot due to the zero location. The settling time should be about 100ms. Hitting "Reset" turns the motor back to the start position (0 degrees).
- Try some different k values and position values and play around a bit. Problems? See NOTE, 2 steps above. Reset your values to what they were.
- Adjust the parameter k_F to get a better response (make k_F smaller). Try several values of k_F with two goals in mind: 1. Note some interesting k_F values to record in the next step.

These should illustrate the affect of k_F . 2. Find the “best” k_F which results in less than 5% overshoot and the shortest time to get within 5% of the final position.

- When ready, reset all your values (or use the conservative ones) and record some data like you have in the past. The “Start” button should trigger the recording. Hit “Reset”. Set several of your interesting k_F values and record them as well. Take note of which data files correspond to which k_F values for later plotting. When you have good data (you may wish to check in Matlab), Stop recording.

- Use Matlab to plot the results for a few values of k_F on a single graph. (This may be done later if you’re short on time.) Print the plot and indicate what value of k_F gives the best response. Sample plots are shown here.

Comparison of PID controller plots based on various feed-forward gain values



- Reset all your gains to your best values and hit “Start”.
- Grab the coupler of the motor in such a way that if it unexpectedly begins to spin, it won’t hurt you. If this worries you, temporarily remove the motor coupler. The naked shaft is pretty harmless. Try to turn the motor shaft. (Don’t use too much force or do this too long, it will heat up the amplifier.) Can you feel the integral control fight the change and eliminate the steady-state error? Can you see it work in the “th_deg/ln1” box? Does the angle keep returning to 90 (or 0 if “Reset” is blue)? Comment in your notebook. OK, now we’ll see just how much better this system is than the one from Lab 3.
- Uncheck the “PID Controller”. You should be able to turn motor shaft freely now. Set $k_F=1$, $k_p=10.5$, and both $k_I=0$ and $k_D=0$. This should recreate the underdamped system of Lab 3. Check either the “Proportional Controller” or the “PID Controller”, hit “Start”, and “Reset” a few times to see the overshoot. Repeat the previous step. Does it fight as hard to stay at 90 degrees? How about the steady-state error? If you release the shaft, it can be almost 2 degrees. Compare to the PID system.
- Set $k_p=2.09$. This should recreate the overdamped system of Lab 3 and eliminate the overshoot, but now it holds it’s position like a wet noodle. Repeat the previous step to feel and comment. Record one 90-degree step at this setting and another at $k_p=10.5$.
- Use Matlab to plot these two along with the best PID results on a single graph. Print the plot. Compare and comment in you notebook.
- To shut the experiment down, uncheck the PID controller, zero out your Pos_ref, Hit “Reset”, turn off the simulation and go offline to shut the experiment down.

Conclusion

Check-off and conclude as always.