Teaching Your PI Controller to Behave (Part III)

Dave (Wisconsin) Wilson Mar 9, 2013

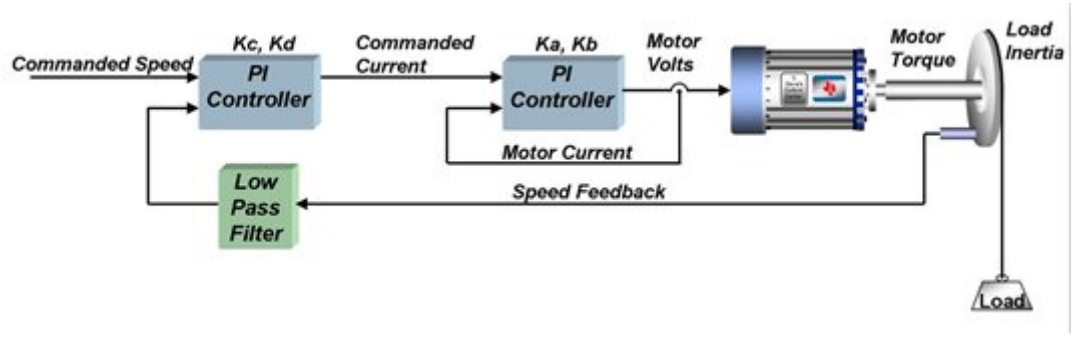
Dave Wilson, Motion Products Evangelist, Texas Instruments

In my last blog, I explained how to calculate the P and I coefficients (actually the Ka and Kb coefficients in a series structure) for a current loop PI controller for a motor. We saw that Kb could be used to eliminate the zero in the closed-loop system response, resulting in a system having only one real pole (i.e., well behaved and stable). Ka sets the bandwidth of the closed-loop system response.

Today, let's back out and take a look at a speed control loop which contains two PI controllers; one to control the motor's current, and another to control the motor's speed. From the figure below, notice that the output of the velocity PI controller is connected as the input reference signal for the PI current controller. This makes sense when you think about it. If the velocity is too low, we want to increase the motor's current to produce more torque to speed it up. Conversely if the motor is going too fast, we want to decrease the motor's torque (or even drive it negative) to slow the motor down. Together, these two PI controllers form a cascaded control loop. By "cascaded," I mean a control system that consists of an outer loop with one or more inner loops. In my last blog, we saw that setting the coefficients for the PI current controller is a fairly deterministic process. Can designing the speed PI controller be just as simple? Do the coefficient values perform the same system functions they did with the current controller?

It turns out that closing the speed loop is a little more complicated than closing the current loop. To properly design the speed loop, we need to know more system parameters than we did for the current loop. This can be seen in the figure below which shows all of the

components that comprise a typical cascaded speed control loop.



Notice that in our system, the speed feedback signal is filtered. Unless you have an analog tachometer mounted to your motor shaft, you are probably synthesizing the speed signal by measuring other system parameters. Even if you are using an incremental encoder on the motor shaft, you will need to synthesize velocity by measuring position. For this reason, the speed signal often needs to be filtered before it is usable by the control system. For our purposes, let's just assume that we are using a single-pole low pass filter of the form:

$$Vel_{filter}(s) = \frac{1}{\tau s + 1}$$

Equ. 1

where τ is the time constant of the velocity filter.

Assuming we design the current loop as discussed in my previous blog, its closed-loop transfer function is:

$$G_{current}(s) = \frac{1}{\frac{L}{K_a}s + 1}$$

Equ. 2

where Ka is the error gain term in the current regulator's PI structure. Recall that Kb is not visible to the speed loop since it is used internally to the current loop to achieve pole-zero cancellation in its closed-loop transfer function.

I hate to do this to you again, but we are going to have to wade through a little bit more math to understand how to set the two coefficients for the speed PI controller. To avoid confusing the coefficients of the speed controller with those of the current controller, we will call the speed controller's coefficients Kc and Kd as shown in the figure above. In the series form of the PI controller, Kc will be the error gain term (Kc = Kp), and Kd is the integrator gain term (Kd = Ki/Kp). So let's use the same equation I did in my last blog for the current

controller to define the transfer function for the speed PI controller:

$$PI_{speed}(s) = \frac{K_c K_d}{s} + K_c = \frac{K_c K_d \left(1 + \frac{s}{K_d}\right)}{s}$$ Equ. 3

The transfer function from motor current to motor torque will vary as a function of what type of motor you are using. Assuming we are using a Permanent Magnet Synchronous Motor under Field Oriented Control, the transfer function between q-axis current and motor torque is:

$$Mtr(s) = \frac{3}{2}\frac{P}{2}\lambda_r = \frac{3}{4}P\lambda_r$$ Equ. 4

where P = the number of rotor poles

λ_r = the rotor flux (which is also equal to the back-EMF constant (Ke) in SI units)

For reference sake, an AC Induction motor has a little bit different transfer function between q-axis current and motor torque:

$$Mtr(s) = \frac{3}{4}P\frac{Lm^2}{Lr}I_d$$

where P = the number of stator poles

Lm is the magnetizing inductance

Lr is the rotor inductance

Id is the component of current that is lined up with the rotor flux

But for now, let's do the analysis using a Permanent Magnet Synchronous Motor (Equ. 4). Moving on to the load, we see that the transfer function from motor torque to load speed (in radians/second) is:

$$Load(s) = \frac{\omega(s)}{T(s)} = \frac{1}{Js + k_v} = \frac{1}{k_v\left(\frac{J}{k_v}s + 1\right)}$$

Equ. 5

where J equals the inertia of the motor plus the load

k_v is the viscous damping term

Normally, the viscous damping factor is denoted by k_d, but since we have already used k_d to represent one of the velocity PI coefficients, we will use k_v to avoid confusion. Finally, if we multiply equations 1 through 5 together, we achieve the composite open-loop transfer function:

$$GH(s) = \underbrace{\left(\frac{K_c K_d\left(1 + \frac{s}{K_d}\right)}{s}\right)}_{Velocity\ PI}\underbrace{\left(\frac{1}{\frac{L}{K_a}s + 1}\right)}_{Current\ Loop}\underbrace{\left(\frac{3}{4}P\lambda_r\right)}_{Motor}\underbrace{\left(\frac{1}{k_v\left(\frac{J}{k_v}s + 1\right)}\right)}_{Load}\underbrace{\left(\frac{1}{\tau s + 1}\right)}_{Filter}$$

Equ. 6

In order to untangle this mess we will make some assumptions and combine some terms. Let's start by assuming that the viscous damping term (k_v) is zero. Later in this blog series we will revisit the subject of viscous damping to see what effect it has on our system.

$$GH(s) = \underbrace{\left(\frac{K_c K_d\left(1 + \frac{s}{K_d}\right)}{s}\right)}_{Velocity\ PI}\underbrace{\left(\frac{1}{\frac{L}{K_a}s + 1}\right)}_{Current\ Loop}\underbrace{\left(\frac{3}{4}P\lambda_r\right)}_{Motor}\underbrace{\left(\frac{1}{Js}\right)}_{Load}\underbrace{\left(\frac{1}{\tau s + 1}\right)}_{Filter}$$

Equ. 7

Finally, let's combine all the motor and load parameters into a single constant K:

$$K = \frac{3P\lambda_r}{4J}$$

Equ. 8

Simplifying, we get:

$$GH(s) = \frac{K K_c K_d \left(1 + \frac{s}{K_d}\right)}{s^2 \left(1 + \frac{L}{K_a} s\right)(1 + \tau s)}$$

Equ. 9

From inspection of Equ. 9, we can determine the following characteristics of the speed controller's open-loop transfer function:

1. Two poles at s = 0, resulting in an attenuation rate at low frequencies of 40 dB per decade of frequency.

2. Two additional poles at s = Ka/L (the current controller's pole), and s = $1/\tau$ (the velocity filter pole).

3. One zero at s = Kd

For stable operation, the unity gain frequency should be higher than the zero at s = Kd, and lower than the two poles at s=Ka/L and s = $1/\tau$. Other than that, there is an infinite number of combinations of Kc and Kd which could yield acceptable system responses, depending on whether you want higher bandwidth or better stability. At this point, you are probably saying, "Yeah, tell me something I DON'T know!" Well, what if I told you there was a way to take the guesswork out of the process by defining a single parameter which is proportional to system stability and inversely proportional to bandwidth, which can be used to set both Kc and Kd automatically? In my next blog, I will introduce a parameter which can do just that. In the meantime…

Keep Those Motors Spinning,

Oldest Best Newest

Sivabalan Mohan *over 9 years ago*

Hi Dave,

How do you define the time constant for a velocity filter in the discrete(digital) domain? For ex., if the equation used to filter speed is filtered_speed = filtered_speed + k*(unfiltered_speed-filtered_speed), what is the time constant in terms of k and sampling time Ts. Also does the time constant change if a digital implementation of angle observer is used to calculate angle used in speed measurement?

Thanks

Siva

^ 0 v Reply

Dave (Wisconsin) Wilson *over 9 years ago*

Hi Siva,

The example low pass filter that you mentioned is a very common one that I use frequently. In this particular case, k equals the filter pole in radians/sec times the sampling period (Ts) of the filter. Solving for the filter time constant, tau = Ts/k. As long as Ts and k are fixed values, then the filter time constant will not change, regardless of how the speed measurement was obtained.

Regards,

Dave

^ 0 v Reply

Dimitra Vasiliki *over 8 years ago*

Hi Dave,

I have a question regarding the time constant.

How can I define the total time constant in a model of a synchronous machine with two PI for the current and one for the speed and how for a second order subsystem?

Thanks in advance.

Regards,

Dimitra

^ 0 v Reply

Dave (Wisconsin) Wilson *over 8 years ago*

Hi Dimitra,

The time constant I am talking about in my equations is the time constant of the speed filter. This is very much dependent on your application and something you set yourseslf to filter out the noise in your velocity signal.

Regards,Dave

^ 0 v Reply

Gregory John *over 7 years ago*

Good Day Dave,

What literature did you use(that is if you used any) to obtain the transfer function of the electromagnetic torque and current for the induction motor?

Thank you!

Regards

Greg

^ 0 v Reply

▼ View More

About TI

Quick links

Buying

Connect with us

Texas Instruments has been making progress possible for decades. We are a global semiconductor company that designs, manufactures, tests and sells analog and embedded processing chips. Our products help our customers efficiently manage power, accurately sense and transmit data and provide the core control or processing in their designs.