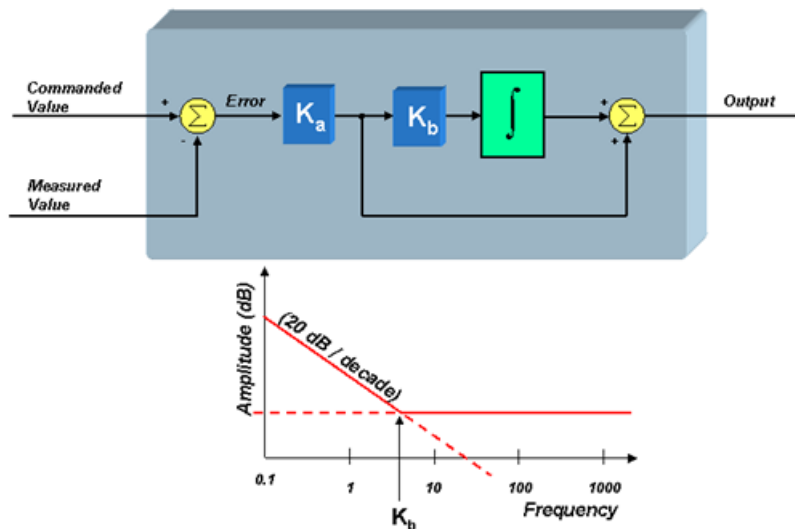


# Teaching Your PI Controller to Behave (Part II)



Dave Wilson

In my previous blog on this topic, we briefly reviewed the history of the PI controller and presented two forms that are commonly used today. Regardless of which form you use, the frequency responses look identical. But for our analysis, let's focus on the *series* form of the PI controller as shown below.

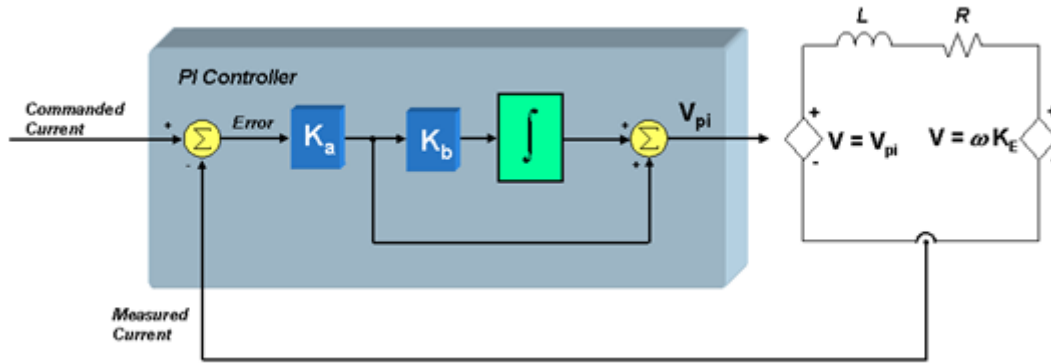


If you have ever done a frequency stability analysis of a control system, you probably understand why  $K_a$  is so important, since it sets the gain of the PI control loop, which in turn has a pronounced effect on system stability. But it turns out that the inflection point in the graph (the “zero” frequency) also plays a significant but perhaps more subtle role in the performance of the system. To understand this, we will need to dive into a little math (hopefully not too much) to derive the transfer function for the PI controller, and understand how the controller’s “zero” plays a role in the overall system response. We can define the “s-domain” transfer function from the error signal to the controller output as:

$$PI(s) = \frac{K_a K_b}{s} + K_a = \frac{K_a K_b \left(1 + \frac{s}{K_b}\right)}{s}$$

Figure 1. Equ. 1

From this expression, we can clearly see the pole at  $s = 0$ , as well as the zero at  $s = K_b$  (rad/sec). So, why is the value of this zero so important? To answer this question, let's drop the PI controller into the heart of a current mode controller which is controlling a motor, as shown below.



We will use a first-order approximation of the motor winding to be a simple series circuit containing a resistor, an inductor, and a back-EMF voltage source. Assuming that the back-EMF voltage is a constant for now (since it usually changes slowly with respect to the current), we can define the small-signal transfer function from motor voltage to motor current as:

$$\frac{I(s)}{V(s)} = \frac{\frac{1}{R}}{\left(1 + \frac{L}{R}s\right)}$$

Figure 2. Equ. 2

If we also assume that the bus voltage and PWM gain scaling are included in the  $K_a$  term, we can now define the “loop gain” as the product of the PI controller transfer function and the V-to-I transfer function of the RL circuit:

$$G_{loop}(s) = PI(s) \cdot \frac{I(s)}{V(s)} = \left( \frac{K_a K_b \left(1 + \frac{s}{K_b}\right)}{s} \right) \left( \frac{\frac{1}{R}}{\left(1 + \frac{L}{R}s\right)} \right)$$

Figure 3. Equ. 3

To find the total system response (closed-loop gain), we must use the following expression which you probably remember from your college control systems class:

$$G(s) = \frac{G_{loop}(s)}{1 + G_{loop}(s)}$$

Figure 4. Equ. 4

assuming that the feedback gain  $H(s) = 1$ .

Substituting [Figure 3](#) into [Figure 4](#) yields:

$$G(s) = \frac{\left( \frac{K_a K_b \left( 1 + \frac{s}{K_b} \right)}{s} \right) \left( \frac{\frac{1}{R}}{\left( 1 + \frac{L}{R} s \right)} \right)}{1 + \left( \frac{K_a K_b \left( 1 + \frac{s}{K_b} \right)}{s} \right) \left( \frac{\frac{1}{R}}{\left( 1 + \frac{L}{R} s \right)} \right)}$$

**Figure 5. Equ. 5**

I realize the expression is getting bigger instead of smaller, but don't freak out just yet. With a little bit of algebraic prowess, we can reduce this expression to the following:

$$G(s) = \frac{\left( 1 + \frac{s}{K_b} \right)}{\left( \frac{L}{K_a K_b} \right) s^2 + \left( \frac{R}{K_a K_b} + \frac{1}{K_b} \right) s + 1}$$

**Figure 6. Equ. 6**

The denominator is a second order expression in "s" which means there are two poles in the transfer function. If we are not careful with how we select Ka and Kb, we can easily end up with complex poles (Yuck!) Depending on how close those complex poles are to the *jw* axis, our system could have some really nasty resonant peaks. So let's assume right away that we want to select Ka and Kb in such a way as to avoid complex poles. In other words, can we put the denominator in factored form so that it looks like this:

$$\left( \frac{L}{K_a K_b} \right) s^2 + \left( \frac{R}{K_a K_b} + \frac{1}{K_b} \right) s + 1 = (1 + Cs)(1 + Ds)$$

**Figure 7. Equ. 7**

where C and D are real numbers.

If we multiply out the expression on the right side of the equation, and compare the results with the left side of the equation, we see that in order to obtain real poles, the following conditions must be satisfied:

$$\frac{L}{K_a K_b} = C \cdot D$$

**Figure 8. Equ. 8**

and,

$$\frac{R}{K_a K_b} + \frac{1}{K_b} = C + D$$

**Figure 9. Equ. 9**

As a first attempt to solve [Figure 8](#) and [Figure 9](#), let's simply equate the terms on both sides of [Figure 9](#). In other words,

$$\frac{R}{K_a K_b} = C, \text{ and } \frac{1}{K_b} = D$$

**Figure 10. Equ. 10**

The reason I recommended these substitutions will now become clear. If we replace the denominator of [Figure 6](#) with its factored equivalent expression as shown in [Figure 7](#), and then make the substitutions recommended in [Figure 10](#), we get the following:

$$G(s) = \frac{\left(1 + \frac{s}{K_b}\right)}{\left(1 + \frac{R}{K_a K_b} s\right) \left(1 + \frac{s}{K_b}\right)}$$

**Figure 11. Equ. 11**

Notice anything interesting about this expression? It turns out that the “D” substitution results in a pole which cancels out the zero in the closed-loop gain expression! By choosing C and D correctly, we not only end up with real poles, but we can create a closed-loop system response that has only ONE real pole and NO zeros! No peaky frequency responses or resonant conditions! Just a beautiful single-pole low-pass roll off response!

But wait...there's more! (At this point I feel like an announcer for a cheap infomercial!) By substituting the expressions for C and D recommended in [Figure 10](#) back into [Figure 8](#), we get the following equality:

$$K_b = \frac{R}{L}$$

**Figure 12. Equ. 12**

Recall that  $K_b$  is the frequency at which the controller zero occurs. So in order to get the response described in [Figure 11](#), all we have to do is to set  $K_b$  (the controller zero frequency) to be equal to the pole of the plant it is controlling!

So, now that we know how to set  $K_b$ , what about  $K_a$ ? Let's rewrite the closed-loop system response  $G(s)$ , making all of the substitutions we have discussed up to now, and see what we get:

$$G(s) = \frac{1}{\frac{L}{K_a} s + 1} \Rightarrow K_a = L \cdot \text{Bandwidth (rad/sec)}$$

**Figure 13. Equ. 13**

So we see that  $K_a$  directly sets the *bandwidth* of the control system which is scaled by the motor's inductance. It turns out that numerically,  $K_a$  equals the inductive impedance at whatever the bandwidth cutoff frequency is.

In summary, what have we learned? Even if you didn't follow all the math, there are some simple rules you can use to help you design your PI controller for your current loop:

1.  $K_b$  sets the zero of the PI controller. When controlling a plant parameter with only one real pole in its transfer function (e.g., the current in a motor),  $K_b$  should be set to the value of this pole. Doing so will result in pole/zero

cancellation, and create a closed-loop response that also only has a single real pole. If the PI controller is part of a larger control loop,  $K_b$  is not visible to the outside world since it is only used for pole-zero cancellation within the PI control loop itself.

2.  $K_a$  sets the BANDWIDTH of the closed-loop system response. As seen by [Figure 13](#), the higher  $K_a$  is, the higher the current loop bandwidth will be. We will discuss how to select an appropriate bandwidth in a later blog.

Whew! I think I blew out enough brain cells for one blog. I realize that this got a little “mathy,” but hopefully you could follow along with my equations. More importantly, I hope you understand what these equations are trying to tell you about how to design your PI current control loop. In my next blog, I want to discuss how to design a cascaded PI velocity loop which contains a PI current controller within it. In the meantime...

Keep Those Motors Spinning,



[www.ti.com/motorblog](http://www.ti.com/motorblog)

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](https://www.ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2023, Texas Instruments Incorporated