

1 of 1 DOCUMENT



March 1, 2007

The Three Faces of PID;

The three elements of PID control each have a distinct personality, but they can be made to work together. Loop tuning becomes easier when you understand the interactions.

BYLINE: Vance VanDoren, Ph.D., P.E., Control Engineering.

Vance VanDoren is consulting editor to Control Engineering. Reach him at controleng@msn.com.

SECTION: FEATURES; Cover Story; Pg. 34

LENGTH: 1728 words

Although proportional-integral-derivative (PID) controllers dominate the process control landscape, they do have their limitations. PID loops can be difficult to configure properly, their behavior is not easily understood, and when they fail to produce the desired closed-loop performance, they can be difficult to troubleshoot. Part of the reason is that the three elements of PID control are vastly different and do not naturally interact well. Nonetheless, all three are necessary to successful control when they can be made to work together.

Fortunately, 60 years of practical experience has taught us much about the individual characteristics of PID. Engineers and technicians can learn the attributes of "proportional," "integral," and "derivative" actions and how they interact. With that knowledge, they can adjust each as needed to solve problematic control situations.

The earliest controllers were proportional-only, but users quickly identified a major drawback. Proportional controllers tended to quit too soon when attempting to eliminate errors between the process variable and setpoint. This *steady-state offset* is illustrated in the graphic on page 36. Proportional controllers settle for small, but non-zero errors, leaving the process variable at a value somewhat shy of the setpoint. This problem stems from the way that a proportional controller with a gain of P computes the control effort, which is by multiplying the error by a factor of P . A large gain produces a large control effort, which drives the process variable closer to the setpoint. That reduces the error between the two, which in turn reduces the control effort. Eventually the effort is too small to move the process any farther. No gain value can be large enough to generate a control effort large enough to eliminate the error completely.

Adding integral action

Operators discovered that they could overcome the "droop" resulting from waning proportional action by manually adding bias to the control effort—just enough to raise the process variable up to the setpoint. This exercise became known as *resetting* the loop.

The Three Faces of PID; The three elements of PID control each have a distinct personality, but they can be made to work together. Loop tuning becomes easier when you understand the interactions. Control Engineering March 1, 2007

Automatic reset was introduced to perform that function without operator intervention. It would automatically increase the control effort in proportion to the running total or *integral* of past errors. That forced the control effort and process variable to continue upwards as long as the error remained positive or downwards as long as the error remained negative. Today, automatic reset is known as *integral action*, although the gain that determines the strength of the integral action in a PID controller is still sometimes called the *reset rate*.

A high reset rate causes the controller to generate an aggressive control effort as long as the process variable and setpoint differ. But, unlike proportional action, integral action will continue to grow as long as the error remains non-zero. So if the proportional action starts to create a steady-state offset, the integral action will become more and more aggressive about eliminating it. In fact, integral action is so persistent that it will only give up once the error and steady-state offset have both been eliminated entirely.

Unfortunately, integral action comes with its own problems. If the process happens to be particularly sluggish, it may take a while for the error to disappear even if the controller employs an aggressive integral action. But if an operator sets the reset rate too high, the controller will become so ambitious that it overcompensates for every positive error and creates an even larger error in the negative direction (and vice-versa). The ensuing oscillations in the process variable will continue to grow until the control effort starts to cycle back and forth between 0% and 100%.

This *hunting* or *closed-loop instability* can also occur if the process is especially sensitive to control efforts or if the process includes its own integral action (such as when liquid accumulates in a tank). The proportional and derivative actions of a PID controller can also exacerbate hunting, depending on the behavior of the process.

Reset windup

Integral action comes into play in situations where a process has an actuator that is too small to implement an especially large control effort. That can happen, for example, when a burner isn't large enough to supply sufficient heat, a valve is too small to generate a high enough flow rate, or a pump reaches its maximum speed. The actuator is said to be *saturated* at some limiting value-either its maximum or minimum output.

When actuator saturation prevents the process variable from rising any further, the controller continues to see positive errors between the setpoint and process variable. The controller's running total of past errors continues to rise, and the integral action continues to call for an increasingly aggressive control effort. However, the actuator remains stuck at its maximum output, so the process variable can't get any closer to the setpoint.

Because the actuator is already fully on, this *reset windup* doesn't affect the controller's immediate performance. But, if an operator tries to fix the problem by reducing the setpoint back into the range that the actuator is capable of achieving, the controller will not respond.

This problem stems from the enormous value that the total integrated error will have achieved during the time that the actuator was stuck at 100%. That total will remain very large for a very long time, no matter what the current value of the error happens to be. The integral action will remain very high, and the controller will keep pushing the actuator against its upper limit.

Fortunately, the error will turn negative if the operator drops the setpoint low enough, so the total integrated error will start to fall. Still, a long series of negative errors will be required to cancel the long series of positive errors that had been accumulating in the controller's running total. Until that happens, the integral action will remain large enough to continue forcing the actuator fully on, as shown on page 38.

Several solutions have been devised to protect against reset windup. Most involve shutting off the integrator when the actuator saturates, though additional measures are generally necessary to prevent the controller from requesting impossible actuator moves in the first place.

The Three Faces of PID; The three elements of PID control each have a distinct personality, but they can be made to work together. Loop tuning becomes easier when you understand the interactions. Control Engineering March 1, 2007

Derivative dilemma

The derivative action of a PID controller can also be a mixed blessing. Derivative action decreases the control effort in proportion to the error's rate of change, so it can decelerate the descent of a process variable that is approaching the setpoint too rapidly. That, in turn, reduces the likelihood of overshoot and hunting.

However, if the derivative action is particularly aggressive, it can put on the brakes so hard that it causes hunting all by itself. This effect is particularly pronounced in processes that react quickly to the controller's efforts, such as motors and robots.

The derivative action also tends to add a dramatic spike or "kick" to the control effort when the error changes abruptly during a setpoint change. This forces the controller to act immediately without waiting for proportional or integral action to take effect. Compared to a two-term PI controller, a full three-term PID controller can even appear to anticipate the level of effort that will ultimately be required to maintain the process variable at the new setpoint (which is why derivative action was originally known as *pre-act*).

While such predictive capability is generally a good thing, dramatic spikes in a control effort can be troublesome in applications that require slow and steady changes in the process variable, such as room temperature control. A blast of hot air following every thermostat adjustment would not only be uncomfortable for the room's occupants, but hard on the furnace as well.

For such applications, it is advantageous to forgo derivative action altogether or calculate the derivative from the negative of the process variable rather than directly from the error. If the setpoint is constant, the two calculations will be identical. If the setpoint only changes in a stepwise manner, the two will still remain identical except at the instant when each step change is initiated. The negative derivative of the process variable will lack the spike present in the derivative of the error.

Derivative action is also a problem for applications that involve noisy measurements. If the controller's derivative gain or rate parameter is too high, a large control effort will ensue every time the noise fools the controller into thinking that the process variable has changed. The controller could end up generating a large derivative action even if the actual process variable has already reached its setpoint. Fortunately, it is relatively easy to filter out measurement noise before computing the derivative of the process variable.

Fixes abound

Fortunately, all of these problems have been addressed if not fixed altogether in modern PID controllers. Reset windup protection, calculating derivative action from just the process variable, and noise filtering have all become standard features on most commercial PID controllers. These help control the internal dissonance and use the characteristics of each element to its best advantage.

Loop tuning is the art of selecting appropriate values for the proportional, integral, and derivative gains so as to achieve a rapid response to setpoint and load changes without making the closed loop system unstable. This tuning is arguably the most challenging aspect of configuring a PID controller. Fortunately, innumerable loop-tuning techniques and software packages have been developed to facilitate that chore. Loop tuning manually is still a challenge, but even that is becoming easier.

See "Loop Tuning Fundamentals," *Control Engineering* , July 2003; and also "Auto-Tuning Control Using Ziegler-Nichols," *Control Engineering* , October 2006.

For more information, visit:

www.controleng.com and search on PID to find:

The Three Faces of PID; The three elements of PID control each have a distinct personality, but they can be made to work together. Loop tuning becomes easier when you understand the interactions. Control Engineering March 1, 2007

"Loop Optimization Software"

"Auto-Tuning Control Using Ziegler-Nichols"

...and many related articles.

resource.controleng.com under References Library has the "Control Engineering Reference Guide to PID Tuning."

Vance VanDoren is consulting editor to Control Engineering. Reach him at controleng@msn.com.

LOAD-DATE: March 19, 2007

LANGUAGE: ENGLISH

PUBLICATION-TYPE: Magazine

Copyright 2007 Reed Business Information, US, a division of Reed Elsevier Inc.
All Rights Reserved