

Relay Method Automates PID Loop Tuning

A traditional loop tuning technique has been modified to simplify its implementation.

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

A proportional-integral-derivative (PID) controller operating in a feedback loop can be very effective at driving a measured *process variable* towards a desired *setpoint*, but only if the controller is adjusted or *tuned* to accommodate the behavior of the controlled process. For a PID controller using the *ISA standard* form of the PID formula, tuning is a matter of selecting appropriate values for the *tuning parameters* P , T_I , and T_D shown in the formula to the right.

Like all PID formulas, the ISA standard form computes the error between the process variable and setpoint as well as the integral and derivative of that error. It then weights the *proportional term* by a factor of P , the *integral term* by a factor of P/T_I , and the *derivative term* by a factor of $P \cdot T_D$, where P is the *controller gain*, T_I is the *integral time*, and T_D is the *derivative time*.

Tuning the loop—or more correctly, tuning the controller—requires adjusting those weights so that the controller can eliminate errors quickly without making matters worse. If the controlled process happens to be relatively sluggish, the controller can be tuned to take immediate and dramatic action whenever a random disturbance changes the process variable or an operator changes the setpoint. A relatively large value for P and a small value for T_I would be appropriate in this case.

Conversely, if the process happens to be particularly sensitive to the controller's efforts, the loop must be tuned to apply more conservative corrective efforts over a longer interval. The essence of loop tuning is identifying just how dramatically the process reacts to the controller's efforts and how aggressive the controller can afford to be as it tries to eliminate errors.

Unfortunately, loop tuning is not as simple as

ISA standard PID formula

$$CO(t) = P \left[e(t) + \frac{1}{T_I} \cdot \left(\int e(t) dt \right) + T_D \cdot \left(\frac{d}{dt} e(t) \right) \right]$$

Proportional term
Integral term
Derivative term

The PID formula determines the control effort (a.k.a. the controller output) $CO(t)$ from the process variable $PV(t)$ and the error $e(t)$ between the process variable and the setpoint. The controller can be made more or less aggressive by modifying the three tuning parameters – the controller gain P , the integral time T_I and the derivative time T_D . Source: Control Engineering

consulting a table to find a value of P that produces a desired degree of overshoot or a specific settling time. The best choice for each of the tuning parameters P , T_I , and T_D depends on the values of the other two as well as the behavior of the controlled process. Furthermore, modifying the tuning of any one term affects the performance of the other two since the modified controller affects the process and the process in turn affects the controller.

Ziegler-Nichols tuning

John G. Ziegler and Nathaniel B. Nichols of Taylor Instruments (now part of ABB) addressed these problems in 1942 when they proposed a two-step method for tuning a loop. They devised tests for quantifying the behavior of a process in terms of how fast and how much the process variable changes when the control effort changes. They also developed a set of empirical formulas for translating the

ONLINE

Read more from Vance VanDoren at www.controleng.com/archive:

Time proportional control: More from an on/off switch, May 2009

Sorting out PID controller differences, Feb. 2009

The basics of numerical filtering, Oct. 2008.

Process controllers predict the future, March 2008.

Three faces of PID, March 2007.

Also, search on "Vance" at www.controleng.com

results of those tests into appropriate settings for the controller's tuning parameters

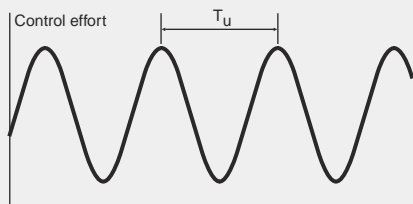
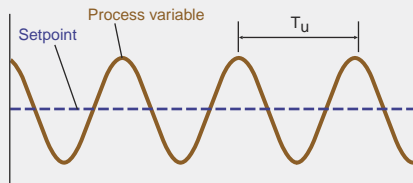
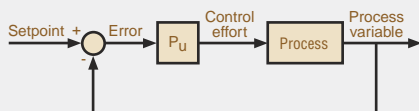
The Ziegler-Nichols *open-loop tuning test* is conducted with the controller temporarily shut off and feedback disabled. The control effort is changed manually and the resulting changes in the process variable are analyzed for clues about the process's behavior. (See "Loop Tuning Fundamentals," *Control Engineering*, July 2003, for more details.)

The Ziegler-Nichols *closed-loop tuning test* is conducted with the controller operating in automatic mode but with the integral and derivative terms temporarily shut off. The controller gain is increased until even the slightest error causes a sustained sinusoidal oscillation in the process variable as shown in the "Ziegler-Nichols closed-loop test" graphic.

The smallest controller gain that can cause such an oscillation is called the *ultimate gain* P_u . The period of those sinusoidal oscillations is called the *ultimate period* T_u . The appropriate tuning parameters are computed from these two values according to the Ziegler-Nichols *tuning rules*:

$$\begin{aligned} P &= 0.6 \cdot P_u & T_I &= 0.5 \cdot T_u \\ T_D &= 0.125 \cdot T_u \end{aligned}$$

Ziegler-Nichols closed-loop test



Forcing the closed loop system into sustained oscillations with a proportional-only controller reveals the process's ultimate gain P_u and ultimate period T_u . Unfortunately, doing so can also cause dramatic and sometimes dangerous swings in the control effort and the process variable. Source: Control Engineering

Caveats

But even the Ziegler-Nichols methods aren't so straightforward in actual practice. Different PID controllers use different versions of the PID formula, and each must be tuned according to a different set of tuning rules. The rules also change when:

- The controller is not equipped with integral and/or derivative terms;
- The process itself is inherently oscillatory;
- The process behaves as if it contains its own integral term (as with level control); and/or
- The process deadtime is very small or significantly larger than the process time constant.

For many years, the Ziegler-Nichols methods were strictly manual operations executed whenever a new control loop was commissioned. An engineer would run a test, tune the loop according to the results, then start production with the new loop in automatic mode. It was tedious and repetitive work to commission every loop in the plant this way, and the results weren't always satisfactory. Several iterations were often necessary to generate tuning parameters that produced acceptable closed-loop performance.

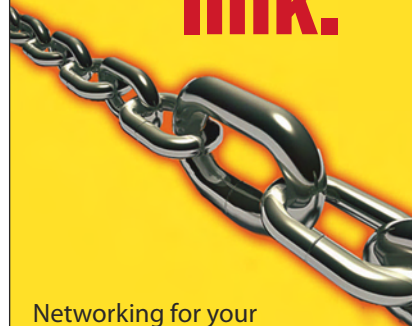
Worse still, the closed-loop test deliberately forces the process variable to fluctuate between two extreme values that can not be predicted beforehand. Many control engineers have held their breath while ramping up the controller gain, hoping that the process variable would reach a sustained oscillation without destroying the process.

Automating the tuning process

Then in 1984, Karl Åström and Tore Hägglund of the Lund (Sweden) Institute of Technology proposed a less risky alternative to the Ziegler-Nichols open-loop test. Their *relay method* generates a sustained oscillation of the process variable but with the amplitude of those oscillations restricted to a safe range.

The Åström-Hägglund method works by forcing the process variable into a *limit cycle* as shown in the "relay test" graphic. With all three PID terms temporarily disabled, the controller uses an on/off relay to apply a step-like control effort to the process. It then holds the control

The strongest link.



Networking for your industrial system should be the easy part. Partnering with GarrettCom makes it simple and rock solid. Rely on our 20 years of expertise and proven solutions that include:

- Hardened products
- Resilient networking
- Ethernet, serial, WAN
- Integrated PoE
- Outstanding support

With all the things you need to worry about, your network shouldn't be one of them.



GarrettCom®

(510) 438-9071

www.garrettcom.com



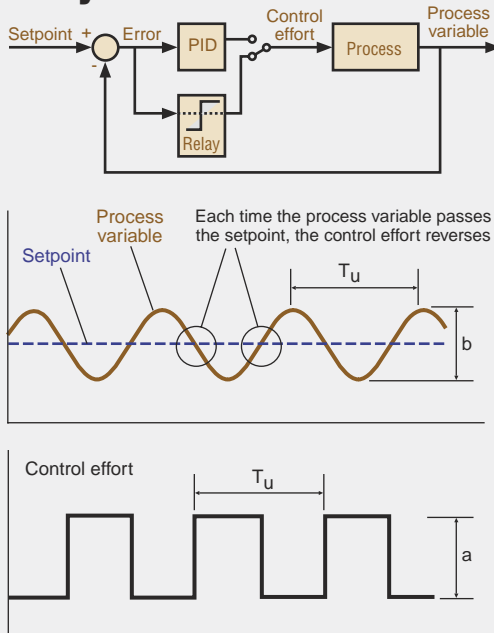
effort constant and waits for the process variable to exceed the setpoint.

At that point, it applies a negative step and waits for the process variable to drop back below the setpoint. Repeating this procedure each time the process variable passes the setpoint in either direction forces the process variable to oscillate out of sync with the control effort but at the same frequency.

Although the process variable's oscillations aren't strictly sinusoidal, their period turns out to be a close approximation of the ultimate period that Ziegler and Nichols used for their tuning rules. And the amplitude of the process variable's oscillations relative to the amplitude of the control effort's oscillations approximates the process's ultimate gain when multiplied by $4/\pi$. So once the ultimate period and ultimate gain have been

determined, tuning the loop becomes a simple matter of plugging those two values into the Ziegler-Nichols tuning rules.

Relay test



To identify the ultimate period T_u and ultimate gain P_u of the process, the control engineer running the test temporarily disables the PID block and replaces it with an on/off relay that forces the process variable to oscillate. The engineer then measures T_u directly from the square-wave control effort and computes the value of P_u according to

$$P_u = (4 \cdot b) / (\pi \cdot a)$$

where a is the amplitude of the controller's square wave and b is the amplitude of the process variable's resulting oscillations. By fixing a at a relatively small value, the engineer can also limit the value of b and thereby prevent the wild swings that sometimes plague the original Ziegler-Nichols open-loop test.

Advantages

But unlike the original Ziegler-Nichols closed-loop test, the relay test can be configured to limit the amplitude of the process variable's oscillations by fixing the amplitude of the control effort's oscillations at a user-defined value. This allows the controller to force the process variable to oscillate just enough to distinguish the process's behavior from measurement noise. The process variable needs not swing so wildly as to endanger the process.

Better still, the controller can be configured to conduct the relay test and tune the loop without operator intervention. Theoretically, even an operator unfamiliar with the fundamentals of tuning theory can press a button and let the controller conduct its own relay test and select its own tuning parameters accordingly. If the resulting closed-loop behavior proves unacceptable, the operator can simply push the tune button again.

Whether based on the Åström-Hägglund relay method or a similar technique, such *auto-tuning* and closely related *self-tuning* functions have become standard features on most commercial PID controllers. (See "Fundamentals of Self-Tuning Control," *Control Engineering*, July 2007, and "Applications of Self-Tuning Control," *Control Engineering*, September 2007.) **ce**

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

Windows XP / XP Embedded based - Panel PC

CUPC

- Windows XP based Touch Controller
- Touch panel & Touch controller
- AMD Geode LX800 (500MHz)
- 512M DDR RAM
- 4G FLASH HDD
- USB, ETHERNET, RS232, RS485 port
- SD card support
- Windows XP embedded, Windows 2000 / XP, Linux support



www.cubloc.com
Toll-Free: 1.888.928.2562

COMFILE
TECHNOLOGY