Build a PID Controller out of Stuff Already on Your Workbench

Nearly all modern process actuators (valves, agitators, heaters, motors, etc.) and process meters are electronic, which makes them easily integrated into Proportional-Integral-Derivative (PID)-based control systems. PID control is at the heart of many processes, and it's important to get it right.

WHEN designing a process system, it can be helpful to test and refine PID algorithms before the equipment is commissioned or while the system is being designed. Fortunately, a PID controller can be created using instruments that most engineers have on their workbench. This article explains how to build a PID controller out of a digital multimeter (to measure the process variable) and a power source (to drive an actuator).

It is based on a real-world application: a PID temperature control scheme using a Peltier device (also known as a thermoelectric cooler), a thermocouple, a digital multimeter (DMM), a DC power supply (20V, 5A), and a PID algorithm created and run in the LabVIEW development environment. This example uses a Keithley Model 2000 6½-digit DMM and a Keithley Model 2200-20-5 unipolar Programmable DC Power Supply. The temperature controller consists of the Peltier device sandwiched between an aluminum plate (bottom) and an insulator. The bottom plate is immersed in an ice-water bath and the PID loop controls the temperature of a small aluminum plate underneath the insulator (***Figure 1***). This example demonstrates the effectiveness of a PID loop even when it's run from a PC. This control scheme allows controlling the temperature to within ±0.05°C under most operating conditions with no shielding of the signal line.

In this control scheme, the DMM collects voltage measurements from the thermocouple, converting them to temperature measurements, which are then fed as the measured variable (MV) into the PID loop running in a LabVIEW application on a PC. The loop compares these measurements against the user-selected set point and the PID algorithm determines an appropriate output. The output is converted to a current (ranging from 0A to 5A), which is sourced to the Peltier device by the power supply.

The algorithm running in LabVIEW is summarized by the following:

1. Determine the set point of the measured variable.
2. Make a measurement of the measured variable.
3. Measure the present time, call it t_n, and calculate the time difference between the present measurement and the previous measurement: $\Delta t_n = t_n - t_{n-1}$
4. Calculate the error: $e(t_n) = SP(t_n) - MV(t_n)$
5. Determine each of the PID component outputs:
   a. $u_P(t_n) = e(t_n)$
   b. $u_I(t_n) = u_I(t_{n-1}) + 3(t_n) * \Delta t_n$
   c. $u_D(t_n) = \dfrac{e(t_n) - e(t_{n-1})}{\Delta t_n}$
6. Determine the total PID output from user-supplied gain coefficients:
   a. $u(t_n) = K_P u_P(t_n) + K_I u_I(t_n) + K_D u_D(t_n)$
7. Begin again from step 1.

The top aluminum plate is cooled passively. The rate of passive cooling may be significantly different than the rate of active heating unless the controller's designer chooses a region of control in which the control midpoint is at the center of the power supply's output range (in this case, 2.5A).

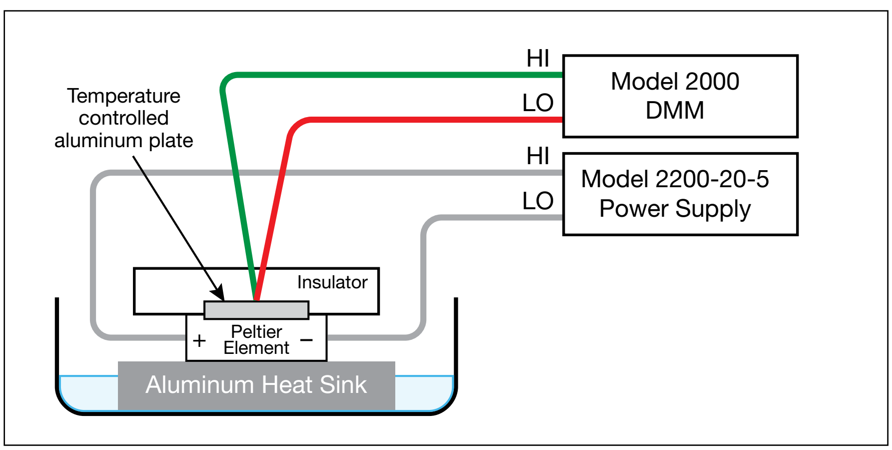

Figure 1. PID controller illustration

The Fundamentals of PID Control

Proportional, Integral, and Derivative (PID) feedback control is widely used because of its high effectiveness and ease of implementation. It's also economical; although it doesn't provide "perfect" control, its performance is generally good enough that most users find it preferable to the added cost and complexity of an optimum control scheme.

The elements of PID control act on the difference between some desired value or *set point* (SP) of a process variable (PV) and that variable's current value. The PV of interest is often referred to as the *measured* (or *manipulated*) variable (MV). The instantaneous difference between the SP and MV is the error. The goal of the PID controller is to eliminate this error. In other words, the controller works to ensure the process is operating to keep the MV always at the set point.

The PID controller has three elements: the *proportional* element (which drives the output in proportion to the instantaneous error), the *integral* element (which drives the output in proportion to the accumulated error), and the *derivative* element (which drives the output in proportion to the instantaneous rate of change of the error). Each element has a weighted contribution to the total output signal of the controller. *Tuning* is the process of establishing those weights (or gains) to get the best or desired response from the controller.

The PID controller takes the error as an input parameter and outputs a signal with a magnitude that's determined by the sum of the P, I, and D elements. As stated previously, the error is the difference between the desired value of some process variable and that variable's current value or the set point minus the measured value:

$$e(t) = SP(t) - MV(t)$$

The "P" contribution, $u_p(t)$, is determined by the following relation:

$$u_p(t) = e(t)$$

By similar notation, the "I" and "D" contributions are defined thus:

$$u_I(t) = \int_0^t e(\tau)d\tau$$

$$uD(t) = \frac{de(t)}{dt}$$

The integral is run from the start of the controller to the present time. The total output, $u(t)$, is the weighted sum of each of the contributions:

$$u(t) = K_P u_P(t) + K_I u_I(t) + K_D u_D(t)$$

where K_P, K_I, and K_D are the gain coefficients for each of the components as indicated by their subscripts. The coefficients are user-supplied parameters and are determined through tuning the controller for a particular use.

This occurs at about 80°C for this system. At this temperature, any current lower than 2.5A that's fed into the Peltier device will cause the top plate to cool down.

The PID loop operates at 20Hz or 50ms per cycle. The rise time associated with this system is tens of seconds, which is much longer than the PID loop period. The loop period can be increased to about one second before any significant performance degradation becomes apparent.

The PID output is linearly mapped so that an output of –1000 correlates to 0A and an output of 1000 correlates to 5A. This gives the 0A to 5A range with a 2.5A midpoint. The controller's arbitrary ±1000 output dictates the order of magnitude of the user-supplied PID gains.

Tuning the Controller

The first step in tuning a PID controller is deciding the kind of control response desired. Is a fast rise time needed? How much oscillation is tolerable? Is the process to be controlled too fast for the equipment used? Start by using only the proportional component, then add the integral and derivative elements if it becomes apparent that proportional-only control is inadequate.

Tuning the PID controller involves giving each element the proper weight to introduce desirable effects into the controller response while minimizing their individual drawbacks. The process being controlled dictates what is and isn't desirable in the controller response. For example, some processes don't accommodate oscillation well, which puts additional limits on the rise time of the measured, or manipulated, variable (MV). On the other hand, a process may tolerate some oscillation, which allows for a faster rise time of the MV.

Tuning requires understanding how each PID element affects the response. A poorly designed PID controller will display one or more of the following performance problems: oscillation, poor damping, overshoot, slow rise/fall times, kick, droop, and excessive windup.

Oscillation

Oscillation is a complicated performance issue that can result from several factors. For example, the physical system being controlled could be oscillatory in nature, so the PID controlling it may have to accept some oscillation. The most common cause of oscillation is too large a proportional gain (***Figure 2***) for the loop iteration rate, or frequency, of the PID controller. Note, however, that an all-integral controller is an oscillator.

The controller's frequency is the rate at which the control loop operates. The power supply output is updated once each cycle. However, if the proportional gain coefficient is too high, a small error will produce a large output, causing a large change in the MV, even in one cycle. If that change takes the MV past the set point, then the output reverses polarity for the next cycle. If the MV keeps jumping across the set point from cycle to cycle, then the controller is oscillating. If the gain is high enough and causes the MV to jump across with increasing error magnitude, then the controller is deemed unstable. Although reducing the proportional gain is the quickest fix, increasing the controller frequency, if possible, will also help.

A digital controller will always produce some oscillation. In addition to the loop frequency problem already described, it's important to realize the output of the power supply can only be resolved to some finite value, so there's a limit to how much the oscillation can be reduced. Oscillation still occurs in analog PID controllers because the analog circuits still have an effective

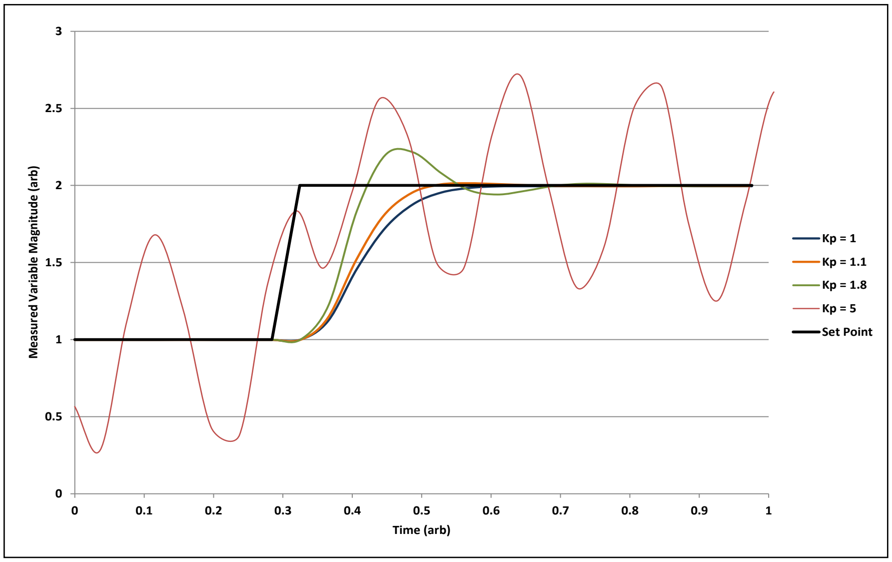

Figure 2. Measured variable response to proportional gain coefficient.

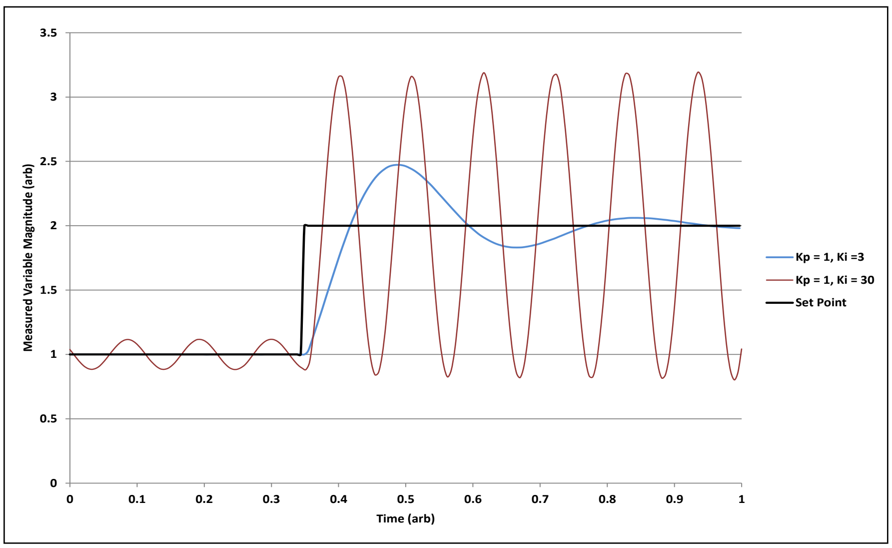

Figure 3. Oscillation due to high integration gain coefficient.

operation frequency due to reactive components that cause a time lag between output and input.

An integral-heavy PID controller will also oscillate (***Figure 3***) because it takes some time while the MV is over the set point to correct the error accumulated while the MV was less than the set point. The integral component then starts to accumulate error in the opposite direction, which won't be corrected until the MV crosses the set point again. It's possible to reduce this type of oscillation by either reducing the integral gain or increasing the proportional gain. The proportional action will dampen the oscillations caused by the integral action. However, either action—decreasing the integral gain or increasing the proportional gain—may cause other problems. This example illustrates the challenges that associated with tuning and importance of deciding which characteristics of controller responses are required, desired, or unacceptable.

Droop and Windup

Most physical systems have some sort of mechanism that causes a loss or reduction of the MV. For example, a heater/cooler system must take heat loss/gain through system boundaries into consideration. The effort of a proportional-only control system will always reduce to the point where the controller output is just enough to bring the MV to the set point minus system losses. This point is where the controller output is insufficient to change the error but just enough to keep it constant (that is, a steady-state error). This phenomenon, known as droop (***Figure 4***), prevents the MV from reaching the set point.

One's first instinct may be to increase the proportional gain to attempt to minimize the droop, but a proportional-only controller will always encounter droop because there's no mechanism to account for system losses. In systems in which a smaller droop is acceptable, a proportional-only controller remains useful. For other systems, droop is undesirable and increasing the proportional gain may introduce more problems, such as oscillation and set point overshoot.

The heater/cooler example described previously is a generic temperature control system, complete with boundaries and temperature gradients across them. Controlling the system temperature and accounting for droop accurately would require a complicated algorithm relating temperature gradients and time constants. Alternatively, the proportional-only controller may expand to a PI controller and allow the integral component to account for droop (***Figure 5***). The output of the controller will continually increase while the MV is less than the set point. The integral component builds up a *buffer* and establishes it as an offset of the controller output that compensates for system losses. The time required to build this buffer from the start of the controller is known as *windup*. There can also be a significant windup period following a large set point change. Stopping the integral mechanism from accumulating while the output is already at a maximum can reduce this effect. This is because the integral term isn't affecting the output but is still accumulating error that will have to be cancelled out later.

Rise Time and Overshoot

The MV's rise time is the time it takes to reach the set point after a change or disturbance. Although a fast rise time is obviously desirable, too fast a rise time will cause an overshoot of the MV across the set point, resulting in damped oscillation as the MV settles out to the set point. Although some overshoot is an acceptable tradeoff for a short rise time in some applications, in others, no overshoot can be tolerated. Rise time is principally affected through the proportional

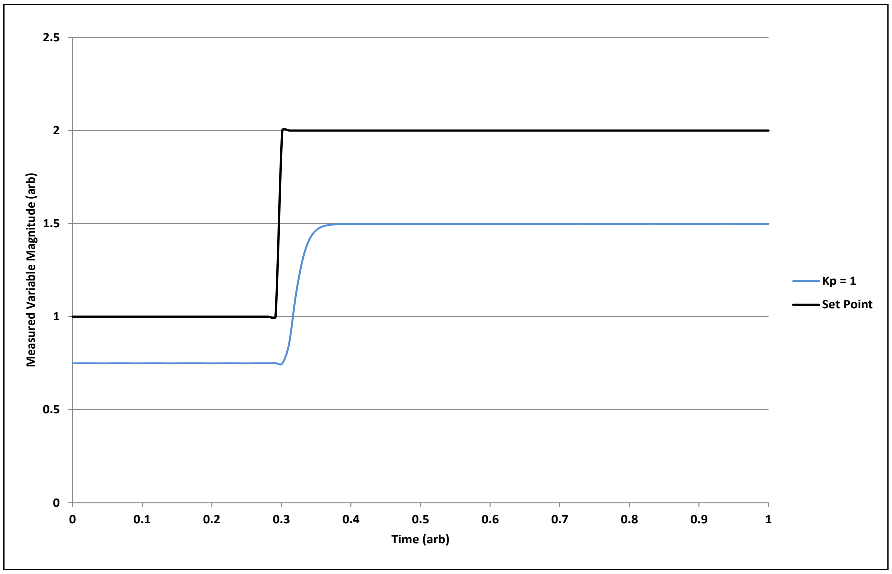

Figure 4. PID droop.

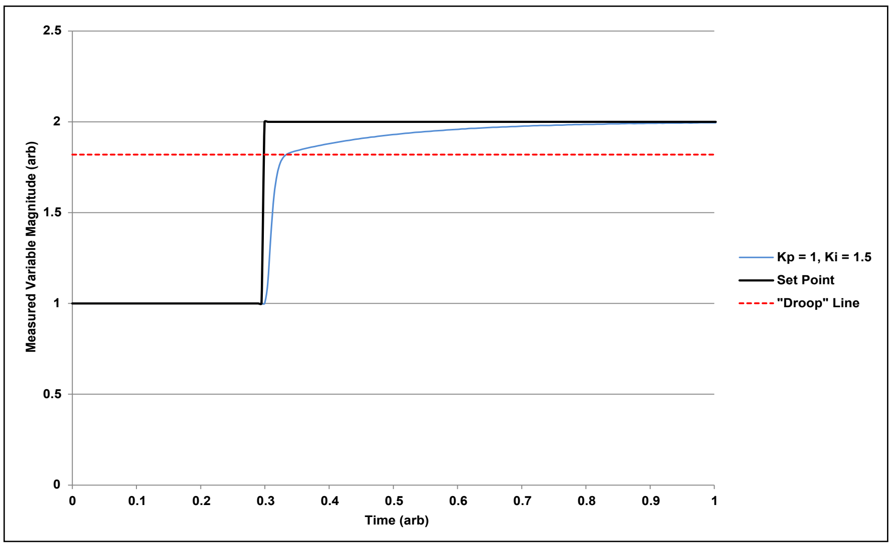

Figure 5. Integration correction to droop.

and derivative gains. The derivative action responds strongly to an abrupt change but slows the approach to the set point. The integral component doesn't affect rise time appreciably because of its error accumulation nature. Furthermore, oscillation will result from a large integral gain.

Disturbance Robustness

A disturbance occurs when anomalies in any of the process variables produce an abrupt or unusual change in the MV. Think of a feedstock pipe burst that causes a dramatic drop in vessel pressure (the MV) or a malfunction of a pre-heating phase in a process, which causes the system temperature to drop much lower than during normal operation. A PID controller's *disturbance robustness* is a measure of its ability to handle these changes effectively.

Disturbances aren't always accounted for in a process scheme, so attempting to control a process using a controller without a feedback loop would certainly fail to account for a disturbance. On the other hand, a PID controller can be tuned to reject disturbances, even if they originate with something seemingly unrelated to the process at hand.

An abrupt change in the MV can be viewed as functionally equivalent to a step change of the set point (***Figure 6***), so the same tuning performed to ensure the best response to a step change of the set point does much the same for disturbance rejection, unless the disturbance affects the PID controller's ability to control the MV either because the actuation becomes compromised or the disturbance brings the MV near or outside of the actuation boundaries. The burst pipe suggested previously is an example of compromised actuation; controlling the pressure inside a broken pipe would surely overtax the controlling pump.

Kick, Noise, and the Derivative Action

Figure 6 also illustrates the usual controller response to a step change or disturbance, an abrupt spike known as kick. Although unavoidable, kick can be undesirable for several reasons: it may be stressing the power supply or actuator unnecessarily or the controller output could be instructing a power supply to force a large current into an inductive load, such as a motor.

When kick is undesirable, one of the following steps can help prevent it:

- Hard code a limit on how much the output can change from iteration to iteration. This might not affect the controller performance appreciably if detrimental kick occurs only under certain extreme conditions. Otherwise, the limit will affect the rise time of the MV.
- If the derivative action has the potential to cause a very high kick due to division by a very small time step, removing the derivative component completely may be the solution.

Noise is a related problem, particularly with the derivative action. Small, rapid changes in the MV can be magnified into large controller responses solely as a result of the derivative action. If the derivative action has a gain high enough to be doing something meaningful, the more adversely it will act on noise. The easiest solution is to include a dead-band of error change to which the derivative component doesn't respond (that is, has a set output of zero).

Frequency

A control loop's frequency (the rate at which it operates) is important because it determines the type of system it can control adequately. The inverse of the frequency is the period (T), the time between iterations of the loop. Every physical system has a time constant that dictates how fast the MV changes with an associated change in the actuator. If

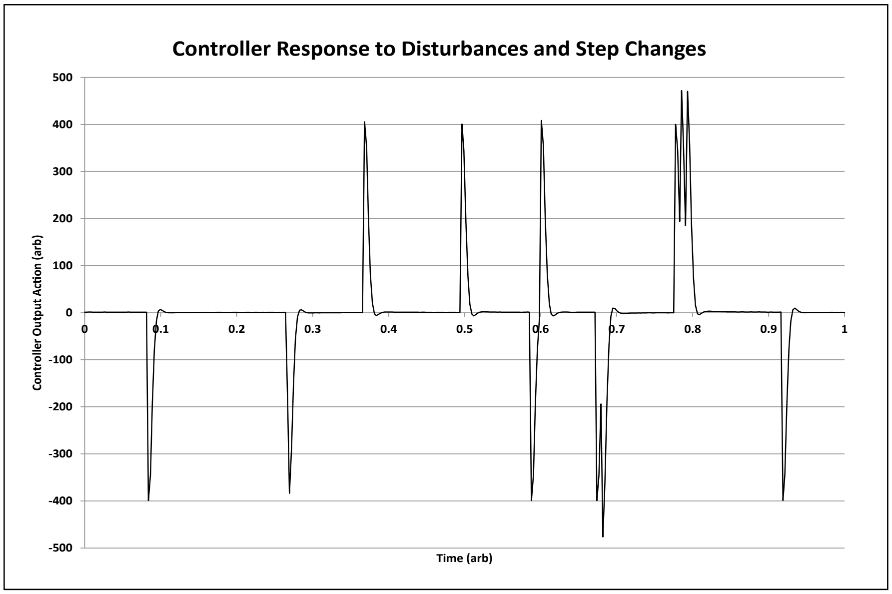

Figure 6. Controller response to disturbances and step changes. The two are indistinguishable.

the PID loop's T is much greater than the time constant of the MV, it isn't possible to bring the MV to the set point, if at all, on an order of time faster than T. Significant overshoot and oscillation will also occur. If the T is much smaller than the time constant of the physical system, the controller may be overdesigned for the application. Although this doesn't represent a system control drawback, the controller itself may be more expensive to build, maintain, and run due to specialized parts, faster clocks, etc.

A control loop's period is limited by the slowest running part of the system. For example, if the power supply requires 100ms to respond to commands from a host PC running a software PID, then the period is 100ms, assuming the meter and PID loop itself can respond much faster than 100ms.

Conclusion

For a more in-depth discussion of the challenges associated with creating a PID controller, download Keithley's PID Controller white paper. An example PID controller written in LabVIEW is also available for download. ■

KEITHLEY INSTRUMENTS ■ 28775 AURORA RD. ■ CLEVELAND, OH 44139-1891 ■ 440-248-0400 ■ Fax: 440-248-6168 ■ 1-888-KEITHLEY ■ www.keithley.com

BENELUX
+31-40-267-5506
www.keithley.nl

FRANCE
+33-01-69-86-83-60
www.keithley.fr

ITALY
+39-049-762-3950
www.keithley.it

MALAYSIA
60-4-643-9679
www.keithley.com

SINGAPORE
01-800-8255-2835
www.keithley.com.sg

BRAZIL
55-11-4058-0229
www.keithley.com

GERMANY
+49-89-84-93-07-40
www.keithley.de

JAPAN
81-120-441-046
www.keithley.jp

MEXICO
52-55-5424-7907
www.keithley.com

TAIWAN
886-3-572-9077
www.keithley.com.tw

CHINA
86-10-8447-5556
www.keithley.com.cn

INDIA
080-30792600
www.keithley.in

KOREA
82-2-6917-5000
www.keithley.co.kr

RUSSIA
+7-495-664-7564
www.keithley.ru

UNITED KINGDOM
+44-1344-39-2450
www.keithley.co.uk

For further information on how to purchase or to locate a sales partner please visit **www.keithley.com/buy**

Printed in the U.S.A

7.16.14