PID CONTROL

PID control strategies are by far the most widely employed of all strategies used in the automatic feedback control of industrial systems. The acronym PID stands for "proportional, integral, derivative." Although the central concept of PID control can be gleaned from an examination of how these three terms are blended to form a control signal, the intelligent application of PID control in any given case requires an understanding of linear and nonlinear properties of the system that is being controlled, and practical PID controllers incorporate features such as bumpless transfer and anti reset windup.

The *proportional* term of the PID controller forms a part of the control signal based on a proportional gain times a system error. The *derivative* term of the PID controller forms a part of the control signal based on a gain times the rate of change of a system error. Similarly, the *integral* term of the PID controller forms a part of the control signal based on a gain times the integral of a system error. This integral term basically forces the error from a nonzero value (a value that would exist without integral action present) to a zero value—the integral term in effect resets the error to zero. Because of this phenomenon, the integral term of the PID controller is sometimes called the *reset* term. This terminology is especially prevalent in older literature; Eckman, for example, uses reset consistently in place of integral (1).

Much of this article is devoted to practical issues of PID controllers. Several alternative forms of PID algorithms are examined, and their relative merits are discussed. Rules for tuning the parameters of the controller are considered, the literature base devoted to this topic is exceedingly large. Tuning rules are concerned with ways of assigning controller parameters (proportional gain, integral gain, and derivative gain) to achieve good performance based on various process model assumptions. That there are a variety of tuning rules should come as no surprise; generally there are tradeoffs to be made in alternative performance criteria and different tuning rules weight some performance criteria more than others. Tuning rules tend to emphasize system performance in the neighborhood of an operating point, and therefore generally are based on linearized models of the process being controlled.

Tuning rules are but one aspect of PID controller design. Of equal importance is how the system performs when large input changes occur. For most systems, the control signal input to the system is limited by maximum and minimum bounds; when the control signal is saturated at one of these bounds for any length of time, the error between the desired system output and the actual system output can be quite large. Within the PID controller, this error signal is being integrated, and unless some mechanism is employed to control the integration process (either by selectively turning it off and on or by selectively forcing the integrator output to track another signal) the output by the PID integrator can literally wind up to exceedingly large values, requiring large periods of time to unwind and leading to unacceptably long transient response. There are a variety of ways to counter integrator wind up, all of which are classified under the heading of anti-reset-windup strategies; some of the more common ones are described in this article.

Prior to the 1960s, most PID controllers were implemented with analog techniques. With the steady advance in digital controller technology since that time, the majority of industrial PID controllers are now implemented digitally to take advantage of flexibility of programming and decreased sensitivity to environmental changes. A major section of this article therefore is devoted to digital implementation issues.

PID controllers are the workhorses of industrial process control, and they provide excellent results under a variety of conditions. There are circumstances, however, where other controllers (or at least enhanced PID controllers) offer superior results. Systems that are highly oscillatory often are difficult to control with PID alone, as are systems that exhibit long pure time delays. Systems having three dominant modes of comparable time constants generally can be controlled better by using a third-order controller, perhaps one that augments the desirable properties of a basic PID controller.

This article is arranged in sections: (1) characteristics of controlled processes; (2) the essence of the PID terms; (3) alternative PID forms; (4) practical derivative action; (5) velocity or incremental PID form; (6) proportional band; (7) anti-reset-windup; (8) bumpless transfer; (9) digital PID control (with subsections on analogous implementation of digital PID, incremental digital PID, recursive digital PID, signal property considerations, other digital PID issues, and an emulation method of digital PID design); and (10) PID tuning.

CHARACTERISTICS OF CONTROLLED PROCESSES

Before a controller for a process is specified, the process to be controlled should be characterized, at least in some broad sense. There are many different types of processes that are controlled automatically. Examples range from fluid levels in tanks to read–write heads of computer disk storage devices. The control inputs to a process are supplied through one or more *actuators.* For example a motor driven valve can be an actuator for fluid flowing into a tank. Process outputs that are being controlled (e.g., the fluid level in a tank) are measured using appropriate sensors. The basic control of one output variable by the use of one control input variable is called single-input single-output (SISO) control. For more complex systems, multi-input multi-output (MIMO) control may be required. PID control was developed initially as an SISO control strategy, but it has been extended in various ways to MIMO control.

Consider a process that responds to an increase in the control signal by having an increase in the controlled output. The output response generally is delayed from that of the controlled input because of time lags caused by system dynamics. Both dynamic and static characteristics of the process are of interest. Static (steady-state) characteristics of the process often can be measured in the following way: first, the control input is set at a specific value; next, after the output of the process has settled to a steady-state value, this steady-state

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering.*

value is tabulated next to the associated input value; and the preceding two steps are then repeated over the entire range of inputs that are realizable to create a steady-state or static input–output plot. Although static characteristics generally are nonlinear, they often exhibit regions of operation over which linear approximations apply, and it is possible to include nonlinear gain in the control to broaden the effective linear range of operation.

The preceding approach must be modified in those cases where a constant control input results, after transients have settled, in a constant rate of change in the controlled output. In that case, the process contains a pure integration term, and the "static" characteristic of interest is a plot of the rate of change of the controlled output as a function of the control input. Disturbance signals can also influence the behavior of a process. In the static measurements described in the preceding paragraph, it is assumed that disturbance signals have had negligible influence.

The controller for an SISO system is supplied with real-time information concerning the relationship of the controlled output of the system to some desired controlled output. The desired controlled output generally is called the *reference input* to the system. It is preferable to have the controller supplied with real-time values of both the reference input $r(t)$ and the controlled output $c(t)$; in some cases, however, only the error signal $e(t)$, $e(t) = r(t) - c(t)$, is available to the controller. The control problem is classified as a *regulator* problem if the reference input remains constant for long periods of time, and the controller strives to maintain the controlled output at a constant value in spite of disturbance inputs. The control problem is classified as a *tracking* problem if the controller is required to make the controlled output track a time-varying reference input such that the error between the two is maintained near zero, even in the presence of disturbance inputs. PID controllers are used in both regulator and tracking control applications. As with other types of controllers, PID controllers also are expected to reduce the sensitivity of the controlled system to changes in parameters in the process being controlled.

THE ESSENCE OF THE PID TERMS

Figure 1 shows the PID parts of a PID-controlled process (the PID parts alone do not constitute a practical PID controller, as will be evident later). The essence of how the PID parts work is as follows.

The Proportional Term

First consider the proportional term $k_p e(t)$ with proportional gain k_p being a positive constant, and assume that only the proportional term is active (temporarily assume $k_i = k_d = 0$ and disturbance input $d(t) = 0$). In this case, the PID controller reduces to a proportional controller (P controller). With the system initially in equilibrium, suppose that the reference input $r(t)$ increases abruptly from zero to a constant steady-state level r_{ss}. The controlled output is then less than the reference input causing an increase in the error $e(t)$. The corresponding positive control signal $k_p e(t)$ results in an increase in the controlled output (the increase is not instantaneous and depends on the static and dynamic characteristics of the process). Assuming that the closed-loop system is stable, the error signal $e(t)$ approaches a steady-state level e_{ss} such that

$$e_{ss} = r_{ss} - K_s k_p e_{ss} \tag{1}$$

where the static process gain K_s generally is dependent on the value of $k_p e_{ss}$. Note that Eq. (1) can be rearranged to obtain e_{ss} as follows:

$$e_{ss} = r_{ss}/(1 + K_s k_p) \tag{2}$$

Assuming $K_s > 0$, it would appear that e_{ss} could be made arbitrarily small by making k_p arbitrarily large; however, for any given system, an upper limit k_u on k_p invariably exists such that the closed loop system is unstable for $k_p \geq k_u$, where the ultimate value k_u depends on the system dynamics.

The Integral Term

The steady-state error described above generally can be reduced to zero by the inclusion of the integral term. With $k_i > 0$, $k_p > 0$, and $k_d = 0$, the controller of Fig. 1 is a PI controller. In this case, consider what would happen to the integrator output if it started at 0 at $t = 0$, and the error signal e was constant at $e = e_{ss} > 0$; the integrator output would then equal $k_i e_{ss} t$, but this increases indefinitely as t increases! The logical conclusion is that e_{ss} must be zero—the output of the integrator is part of the control signal $u(t)$, and as $u(t)$ increases, so also does the controlled output $c(t)$ to the point where $r_{ss} - c_{ss} = e_{ss} = 0$ thereby inhibiting further increases in the output of the integrator.

The Derivative Term

In Fig. 1, if $k_i = 0$ with $k_p \neq 0$ and $k_d \neq 0$, the controller is a PD controller. If all three gain terms are nonzero, the controller is a three-term PID controller. The k_d term plays the role of an anticipatory or predictive element. That is, even if the $e(t_1)$ is zero, if $\dot{e}(t_1) = de(t_1)/dt$ is large, the implication is that the error for $t > t_1$ is going to increase. By including $k_d \dot{e}$ in the control, the controlled output is forced to increase sooner than it would otherwise, with the goal of reducing the anticipated future error. This action is useful in compensating for dynamic lags that invariably are present in the process being controlled. However, if significant measurement noise is present in the measured output signal, the k_d term can have a detrimental effect on performance (the derivative of rapidly changing high-frequency noise can be extremely large even if the magnitude of the noise is small). Because of this, and because the output of any real circuit is band-limited, practical implementation of the k_d term requires the use of a *frequency-dependent differentiator,* as is described later in this article.

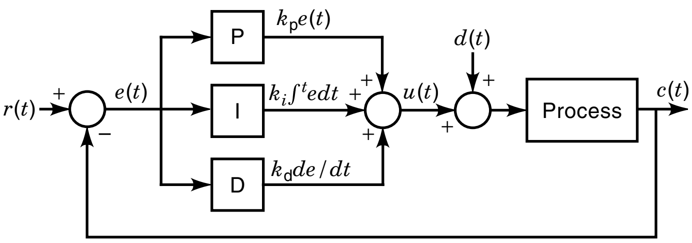

Figure 1. A simplified version of PID control, with one degree of control freedom and without anti-reset-windup protection.

Performance Objectives

The values of k_p, k_i, and k_d should be selected to provide good system response under design conditions, and to reduce harmful effects of process changes that lead to off-design conditions. System response is often described by step response characteristics such as: (1) the 10–90% rise time resulting from a step change in the reference input; (2) the percent overshoot in response to a step input; (3) the peak time associated with the peak overshoot in response to a step input; and (4) the time required for the system output to settle to within 2% of its final value in response to a step input. Of interest are response characteristics caused by process disturbances (represented by $d(t)$ in Fig. 1) in addition to responses activated by reference inputs $r(t)$. Note that for constant disturbance values, with $d(t) = d_{ss}$, the reason given in the previous Subsection on the integral term for e_{ss} being 0 still holds, and the controlled output $c(t)$ approaches the steady-state reference input r_{ss} independent of the d_{ss} value, assuming of course that practical upper and lower bounds on the output of the integrator are not exceeded.

If the reference input, in addition to the error signal is available to the controller, a *modified reference input* may be beneficial. For example, if a controlled output exhibits too much overshoot in response to a step change in a reference input, a rate-limited version of the reference input can be generated and inserted in place of the actual reference input. Conversely, if a controlled output reacts too slowly in response to a change in a reference input, a rate-enhanced version of the reference input can be generated and inserted in place of the actual reference input. In some applications, future values of a reference input are available to the controller; in machine-tool control, for example, a desired contour to be traversed is known in advance. This look-ahead or preview information can be used to reduce the error between the reference input and the controlled output.

ALTERNATIVE PID FORMS

The control action of Fig. 1 is expressed as

$$u(t) = k_p e(t) + k_i \int_0^t e(\lambda)\,d\lambda + k_d \frac{de(t)}{dt} \tag{3}$$

where k_p is the proportional gain, k_i is the integral gain, and k_d is the derivative gain. In terms of Laplace transforms, the Laplace transform transfer function associated with Eq. (3) is

$$G_c(s) \triangleq \frac{U(s)}{E(s)} = k_p + \frac{k_i}{s} + k_d s \tag{4}$$

The above form of the PID terms is called the *parallel* form. An alternative form to that of Eq. (4) is called the *standard* or *noninteracting* form and is characterized by

$$G_c(s) = K\left(1 + \frac{1}{T_i s} + T_d s\right) \tag{5}$$

with the obvious relationships $k_p = K$, $k_i = K/T_i$, and $k_d = KT_d$. Although these two forms are equivalent, the standard form of Eq. (5) uses different terminology: the T_i constant is called the integral time; and the T_d constant is called the derivative time. Either Eq. (4) or Eq. (5) can be arranged as a ratio of polynomials in s. The two roots of the numerator polynomial [the zeros of $G_c(s)$] can be assigned to arbitrary real or complex conjugate values by proper assignment of the PID gains.

Another commonly used form of the basic PID is the *series* or *interacting* form:

$$G_c(s) = K'\left(1 + \frac{1}{sT_i'}\right)(1 + sT_d') \tag{6}$$

Note that the zeros of Eq. (6) are restricted to be real, and therefore the $G_c(s)$ of Eq. (6) is less general than those of Eqs. (4) or (5). However, this series form can be augmented in a simple way to counter windup (to be described in the section on anti-reset-windup). Also, from a historical perspective, the series form was more readily implemented in early pneumatic and hydraulic equipment, some of which is still in use.

Because there are several alternative forms of PID, manufacturers of PID controllers do not always use the same terminology. It is important therefore to determine the terminology and the particular PID controller forms adopted by suppliers of PID controller equipment under consideration. For consistency in this article, however, the parallel form of Eq. (4) is emphasized.

TWO-DEGREE-OF-FREEDOM PID

Figure 2 is a block diagram of a linear system model of a PID-controlled system. Anti-reset-windup and bumpless-transfer features are not included in the linear model. In Fig. 2, $G(s)$ is the Laplace transform transfer function associated with a particular operating point of the process. Note that the integral action operates on the error signal $e(t) = r(t) - c(t)$ as previously described, but that now different proportional and derivative gains apply to the reference input $r(t)$ and the controlled output $c(t)$. In terms of Laplace transforms of the associated signals,

$$U(s) = \left[k_p' R(s) - k_p C(s)\right] + s\left[k_d' R(s) - k_d C(s)\right] + \frac{k_i E(s)}{s} \tag{7}$$

Whereas the feedback system of Fig. 1 has one degree of control freedom, that of Fig. 2 has two degrees of control freedom: (1) the k_p, k_i, and k_d gains can be selected to achieve desired goals regarding closed-loop stability, disturbance rejection, and sensitivity reduction; and (2) the k_p' and k_d' gains can be selected to achieve performance goals associated with the re-

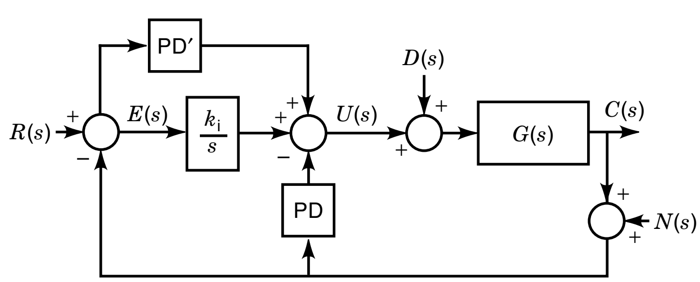

Figure 2. Linear system features of two-degree-of-freedom PID control.

sponse of the system to the reference input. If $G(s)$ in Fig. 2 were a second-order system, with $c(t)$ and $\dot{c}(t)$ viewed as state variables, the PID control form in Fig. 2 would be exactly the control form that would be expected from a modern state-variable viewpoint. For higher-order systems, the $c(t)$ and $\dot{c}(t)$ states are often the most significant ones, so that PID control in many cases can be tuned to give excellent results. Clearly, however, there are cases where higher-order transfer functions should be used in place of the PD and PD′ terms of Fig. 2.

PRACTICAL DERIVATIVE ACTION

The derivative term of the PID controller is often described as simply $k_d s$. However, this term has an output response of $\omega A\cos(\omega t)$ to the sinusoidal input $A\sin(\omega t)$. For large values of ω, corresponding to high-frequency noise, the response $\omega A\cos(\omega t)$ can be overly large even if the input amplitude A is reasonable. Thus, not only is pure derivative action not achievable, it generally is undesirable and cannot be tolerated in practical systems. The $k_d s$ term invariably is replaced by a filtered version, as follows:

$$G_d(s) = \frac{k_d s}{1+\tau s} \tag{8}$$

where the time-constant τ is chosen such that measurement noise [$N(s)$ in Fig. 2] does not have a significant impact on the control action $U(s)$. The transfer function from $N(s)$ to $U(s)$ is

$$G_{NU}(s) \triangleq \frac{U(s)}{N(s)} = \frac{-G_{PID}(s)}{1+G_{PID}(s)G(s)} \tag{9}$$

where

$$G_{PID}(s) = k_p + \frac{k_i}{s} + \frac{k_d s}{1+\tau s} \tag{10}$$

In Eq. (9), at high frequencies where both $\tau\omega > 1$ and $|G(j\omega)| \approx 0$, we have

$$G_{NU}(j\omega) \approx -G_{PID}(j\omega) \approx -\left(k_p + \frac{k_i}{j\omega} + \frac{k_d}{\tau}\right) \tag{11}$$

The k_d/τ term in Eq. (11) replaces a corresponding term $j\omega k_d$ that would have resulted from the use of the pure derivative form. The PID transfer function of Eq. (10) has four adjustable parameters and is therefore a four-term compensator.

VELOCITY OR INCREMENTAL PID FORM

In some applications, the integral part of the PID controller is an attribute of the actuator that drives the process. In that case, the *velocity* form of PID is appropriate, as displayed within the dashed lines of Fig. 3. Note that time constants τ_p and τ_d are included to keep the high-frequency gain of the controller bounded. At low frequencies, the output of the top block in Fig. 3 is approximately $k_p\dot{e}_p(t)$, where

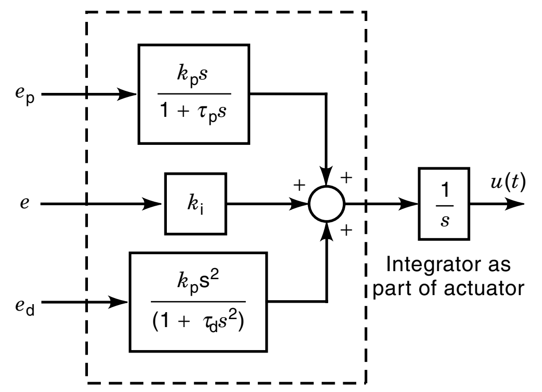

Figure 3. Linear aspects of a velocity form PID controller.

$$e_p(t) \triangleq \frac{k'_p}{k_p} r(t) - c(t) \tag{12}$$

and the low-frequency output of the lower block in Fig. 3 is approximately $k_d\ddot{e}_d(t)$, where

$$e_d(t) \triangleq \frac{k'_d}{k_d} r(t) - c(t) \tag{13}$$

Equations (12) and (13) correspond to the two-degree-of-freedom control loop of Fig. 2.

A potential problem with the velocity PID form is that the second-derivative of an error signal must be approximated; for the same reasons described in the preceding section, this process leads to difficulties in the presence of significant measurement noise. Also, with the integrator external to the controller, it is essential that k_i not be zero in Fig. 3 so that no attempt is made to cancel the $1/s$ of the integrator by the s inherent in a velocity form of the PD controller. This means that the velocity PID of Fig. 3 has the integral term as an essential term of the controller. However, if the control signal $u(t)$ is available as a measured signal, Åström and Hägglund (2) show how to use it to obtain a valid PD form from the velocity PID form.

The velocity form of the PID controller does have some advantages: (1) if the integrator in the actuator is implemented in a way that limits the output of the integrator at the saturation bound (when either the upper or lower saturation bound is intercepted), then no integrator windup can occur, and no special anti-reset-windup strategy need be implemented; (2) if the control is transferred from automatic to manual or visa versa, the signal supplied to the actuator is of an incremental control form, and the resulting transfer in control generally will not cause a large swing in the process output—the transfer of control will be bumpless; and (3) when the velocity PID is implemented digitally (in which case it is called an *incremental* PID algorithm), the PID values being accumulated (corresponding to those at the summation point in Fig. 3) are incremental in nature, and often can be accommodated by a digital word length that is shorter than that required by a corresponding parallel PID algorithm.

PROPORTIONAL BAND

The control signal associated with any practical actuator has a realistic lower bound $u_{\min}$ and an upper bound $u_{\max}$. For any $u(t) > u_{\max}$ the effective control is $u_{\max}$, and for any $u(t) < u_{\min}$ the effective control is $u_{\min}$. When either of the preceding conditions exist, the system is said to be in saturation. The term *proportional band* refers to conditions on the error signal such that the effective control signal is not saturated.

Special Case of Proportional Band

First, consider the case where only proportional control (P control) is applied [i.e., Fig. 1 with $k_i = k_d = 0$, and $d(t) = 0$]; in this case, there will be a range from $e_{\min} \triangleq u_{\min}/k_p$ to $e_{\max} \triangleq u_{\max}/k_p$ in which the control signal will not be saturated, and the corresponding proportional band PB is

$$\mathrm{PB} = \frac{u_{\max} - u_{\min}}{k_p} \tag{14}$$

Alternatively, we say in this case that the error is in the proportional band if $e_{\min} < e(t) < e_{\max}$.

General Case of Proportional Band

When all three terms are present in the PID controller, but $d(t) = 0$ in Fig. 1, the control signal $u(t)$ will be out of saturation only if the instantaneous $e(t) = r(t) - c(t)$ satisfies

$$\frac{u_{\min} - k_i \int^t e(\lambda)\,d\lambda - k_d \dot{e}(t)}{k_p} < r(t) - c(t), \text{ and}$$

$$r(t) - c(t) < \frac{u_{\max} - k_i \int^t e(\lambda)\,d\lambda - k_d \dot{e}(t)}{k_p} \tag{15}$$

Thus, the proportional band varies in a complicated way, depending both on the integral and the derivative of the error.

ANTI-RESET-WINDUP

The control signal of a given system can be in saturation for long periods of time for many reasons. For example, if a large step increase in the reference input occurs at $t = 0$, the error signal for some time after $t = 0$ also will be large, and its direct effect on the control signal will be to force it into saturation until the system output is able to match to some degree the reference input. Although the effective control signal is in saturation, the integrator in the controller accumulates the area under the large $e(t)$ curve, unless restricted to do otherwise. If the integrator output is not restricted in some way, it takes a relatively long time for the integrator output to reduce to a normal level; it does not even start to decrease until after the controlled output $c(t)$ has overshot the desired reference input value [causing $e(t)$ to turn negative and allowing the output of the integrator to start to decrease]. Thus it is essential that the integrator output be managed in some effective way whenever the control signal is in saturation. It is of interest to note that *windup* is not limited to PID controllers—any controller that has a *lag* term may exhibit windup, and ways of controlling the windup should be considered to enhance performance.

Many anti-reset-windup methods have been developed. They are based on one of two approaches: (1) conditional integration, in which integration is interrupted and/or modified when control signal saturation is likely; or (2) integrator tracking, whereby the output of the integrator is forced to track a signal usually with the objective of reducing the control signal magnitude to the saturation boundary. In either case, the anti-windup mechanism must avoid getting locked-up, such that the integrator does not perform as required under normal conditions.

A Conditional Integration Example

Conditional integration schemes tend to be heuristic and application dependent. A given approach may work reasonably well in one application, only to fail in another. Because of their use of logic, conditional integration schemes are readily incorporated in digital implementations of PID. One approach is to condition the integration on both $|k_p e(t)|$ and $|k_d \dot{e}(t)|$: if either one of these values exceeds preassigned bounds, the integration process is suspended and the output of the integrator is reset to a desirable level (perhaps zero); on the other hand, if both values are less then their respective bounds, the integration process is uninhibited; and in either case, the control signal $u(t)$ can be formed on the basis of the three PID terms.

Anti-reset-windup for Two-degree-of-freedom Control

A basic tracking implementation of anti-reset windup is depicted in Fig. 4. The output from the PD and PD′ block is

$$q(t) = k_p' r(t) - k_p c(t) + k_d' \dot{r}(t) - k_d \dot{c}(t) \tag{16}$$

In practice, both derivative terms in the above expression would be filtered versions of the derivatives [for example, $k_d \dot{c}(t)$ would be replaced by the output of the $G_d(s)$ filter of Eq. (8) with the input to the filter being $c(t)$].

The saturation block in Fig. 4 is characterized by

$$u = \begin{Bmatrix} w,\ u_{\min} < w < u_{\max} \\ u_{\max},\ w \geq u_{\max} \\ u_{\min},\ w \leq u_{\min} \end{Bmatrix} \tag{17}$$

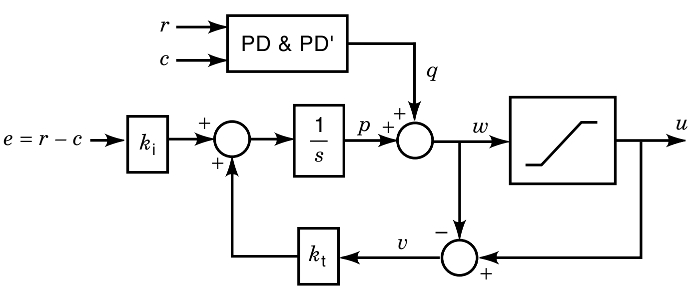

Figure 4. A PID controller with anti-reset-windup provided by a tracking loop. The nonlinear block has a gain of 1 for w in the range $u_{\min} < w < u_{\max}$.

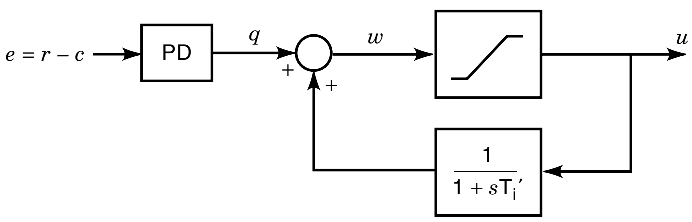

Figure 5. A series-form PID controller with anti reset windup. The nonlinear block has the same characteristics as that in Fig. 4, and the indicated positive feedback results in the linearized transfer function from q to u being $(1 + sT_i')/(sT_i')$.

From the above equation, when w is in the range $u_{\min} < w < u_{\max}$, $u = w$ and the tracking error v in Fig. 4 is zero, resulting in normal operation of the PID algorithm. When $w > u_{\max}$, however, with the tracking gain $k_t > 0$, the additional input to the integrator is $k_t(u_{\max} - w) = k_t v < 0$, causing $p(t)$ to tend to decrease until w reduces to $u_{\max}$. The larger the tracking gain k_t, the faster the tracking process. In high noise environments, the tracking gain should be reduced to a reasonable level to avoid over zealous tracking. When $w < u_{\min}$, the tracking error v becomes positive, and the integration process forces w to tend to increase towards $u_{\min}$. Thus, whenever the control is in saturation, the output of the integrator is automatically adjusted to a level that renders w near the saturation boundary, and any change in $r(t)$ or $c(t)$ that tends to move w away from saturation results in normal PID control.

Anti-reset-windup for the Series Form of PID

In the special case of the series PID form, anti-reset-windup can be implemented in a particularly simple way, as shown in Fig. 5. The saturation block in Fig. 5 is characterized by Eq. (17). Note that the feedback in the loop in Fig. 5 is positive. When the forward path is not saturated, $u(t) = w(t)$ and linear feedback theory can be used to show under these conditions that

$$U(s) = \left(1 + \frac{1}{sT_i'}\right) Q(s) \tag{18}$$

The PD block in Fig. 5 provides (approximately)

$$Q(s) = K'(1 + sT_d')E(s) \tag{19}$$

The combination of Eqs. (18) and (19) implements the series PID form given in Eq. (6). When the forward path in Fig. 5 is saturated at $u = u_{\max}$, the contribution that u makes to w tends to $u_{\max}$. Similarly, when u is saturated at $u = u_{\min}$, the contribution that u makes to w tends to $u_{\min}$. Thus the integration process is effectively bounded and windup is avoided.

An alternative and somewhat more desirable anti-reset-windup strategy for the series PID can be obtained by moving the saturation block from the forward path in Fig. 5 to the feedback path, but keeping the saturation block to the right of the first-order lag block. This enables the full effect of the PD action to be supplied to the process while yet limiting the integral action.

BUMPLESS TRANSFER

When a feedback loop is switched from an automatic mode of control to a manual mode of operation or vice versa, it is often important to avoid large instantaneous jumps in the control action. Certain forms of PID implementation lend themselves in a natural way to smooth transitions from one mode of control to another. Velocity (incremental) PID implementations supply incremental values to be accumulated by an integrating actuator, so large transitions in effective control are not incurred when the source of the incremental changes is switched.

For a general parallel implementation of PID control, tracking can be used to obtain bumpless transfer in addition to anti-reset-windup. Figure 6 is a diagram of such a system. The relationship between w and u in this diagram is that given by Eq. (17). When the switch is in the manual mode, as shown, the human operator supplies a value e_m to drive the system: if the nonlinear block is not saturated, the control u increases (or decreases) at a rate given by $k_m e_m$; at the same time, the signal u_{PID} tracks u_m because of the upper tracking loop, with tracking gain k_t; and when the nonlinear block is saturated, the lower tracking loop insures that the output of the lower integrator will be maintained at either $u_{\max}$ or $u_{\min}$, as appropriate. When the switch to automatic control is made, the u_{PID} value starts with the most recent tracked value of u_m plus any additional component supplied by the PD terms operating on error signals involving $r(t)$ and $c(t)$. Similarly, when the switch is in the PID position, the manual mode output tracks the control $u(t)$ supplied by the PID unit because of the presence of the lower tracking loop, also with tracking gain k_t.

Considerations of bumpless transfer also apply when parameters in the controller are changed abruptly. Such changes are especially easy to make in digital implementations of PID control. Real-time changes in the controller parameters can be motivated by monitored changes in the dynamics of the system being controlled. When these changes are done automatically, the resulting control system is classified as an adaptive control system. The particular PID parameter that is most readily conditioned for bumpless transfer is

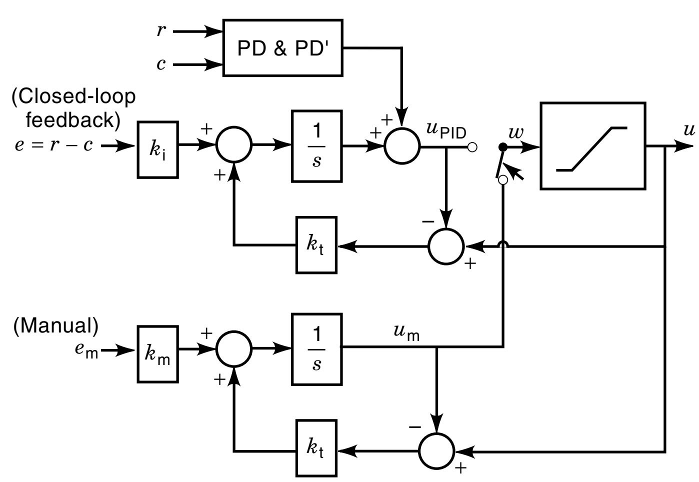

Figure 6. A general analog PID controller featuring anti-reset-windup and bumpless transfer.

k_i. If the error signal $e(t)$ is integrated first and then multiplied by k_i, any abrupt change in k_i will result in a corresponding abrupt change in $u(t)$. On the other hand, if the weighted error $k_i e(t)$ is integrated in real time, the effect of an abrupt change in k_i will be smoothed by the integration process. In terms of a block diagram representation, the k_i block should precede the $1/s$ block for bumpless transfer with respect to changes in k_i.

DIGITAL PID CONTROL

Digital PID control often is implemented by having the system signals of interest [$r(t)$, $c(t)$, and $e(t)$] sampled periodically with sample period T; the sampled input signals are supplied to a microprocessor through an analog-to-digital converter (ADC), and the microprocessor is programmed to supply the control signal $u(t)$ to the system using a digital-to-analog converter (DAC). Usually $u(t)$ appears to the input of the process being controlled as a sampled-and-held signal with sample period T. Within the microprocessor, the linear aspects of PID control are based on the following calculations (using a parallel form):

$$u(kT) = K_\text{p} e(kT) + K_\text{i} \sum_{m=0}^{k} e(mT) + K_\text{d}\{[e(kT) - e[(k-1)T]\} \tag{20a}$$

To simplify notation, we use $u(k)$ to denote $u(kT)$, etc., thereby replacing the above equation by

$$u(k) = K_\text{p} e(k) + K_\text{i} \sum_{m=0}^{k} e(m) + K_\text{d}[e(k) - e(k-1)] \tag{20b}$$

The proportional term of course is $K_\text{p} e(k)$; the "integral" term is the sum $K_\text{i} \sum_{m=0}^{k} e(m)$; and the derivative term is now the first backward difference term $K_\text{d}[e(k) - e(k-1)]$. Using z transforms, the transfer function from $U(z)$ to $E(z)$ is given by

$$G_\text{PID}(z) \triangleq U(z)/E(z) = K_\text{p} + K_\text{i}\frac{z}{z-1} + K_\text{d}\frac{z-1}{z} \tag{21}$$

which can be placed over a common denominator to obtain the form

$$G_\text{PID}(z) = \frac{b_0 z^2 + b_1 z + b_2}{z(z-1)} \tag{22}$$

where b_0, b_1, and b_2 are related to K_p, K_i, and K_d in a straightforward way (to be described). Essentially all digital PID z-transform transfer functions can be based on Eq. (22), which has two poles, one at $z = 0$ and one at $z = 1$. The two zeros of Eq. (22) can be either two real zeros or a pair of complex conjugate zeros located to achieve performance objectives.

Analogous Implementation of Digital PID

The general continuous PID implementation of Fig. 6 can be converted to digital form by using the following approximations. Assume that the lower portion of the figure is left unchanged, but that u_PID is supplied from the output of a DAC and hold circuit. The derivative action is approximated by appropriate first backward differences, with a gain of $K_\text{d} = k_\text{d}/T$. Similarly, the integration is approximated by a summation of samples, and the summation gain $K_\text{i} = k_\text{i}T$ is used in place of the k_i block in Fig. 6. The upper tracking loop in the figure can be implemented digitally, but note that the z-transform model of the $1/s$ for the integrator is $z/(z-1)$. If the output of the k_t block is delayed by one sample period for convenience in the calculations for the tracking loop, the tracking loop will have the following characteristic equation:

$$1 + [k_\text{t}/(z-1)] = 0 \tag{23}$$

which places a pole of the tracking loop at $z = 1 - k_\text{t}$. For stability of the tracking loop, this pole must have a magnitude less than one; and a value of k_t between 0.5 and 1 is recommended.

Incremental Digital PID

Equation (20b) can be used to show that

$$\begin{aligned} u(k) - u(k-1) = {}& K_\text{p}[e(k) - e(k-1)] + K_\text{i} e(k) \\ & + K_\text{d}[e(k) - 2e(k-1) + e(k-2)] \end{aligned} \tag{24}$$

This falls naturally into the incremental PID form. If the actuator of the control loop contains the integrator, then during each sample period, the digital microprocessor simply has to perform the calculations given on the right-hand side of Eq. (24), normalize the result by dividing by T, and send the resulting approximate derivative value through the DAC to the integrating actuator. Of course a slight modification of the above is required if two-degree-of-freedom control is employed; namely, the value V,

$$V = (K'_\text{p} - K_\text{p})r(k) + (K'_\text{d} - K_\text{d})[r(k) - r(k-1)] \tag{25}$$

would need to be added to the right-hand side of Eq. (24).

Recursive Digital PID

Equation (24) can be rearranged in the recursive form

$$\begin{aligned} u(k) = {}& u(k-1) + (K_\text{p} + K_\text{i} + K_\text{d})e(k) \\ & - (K_\text{p} + 2K_\text{d})e(k-1) + K_\text{d} e(k-2) \end{aligned} \tag{26a}$$

Also, using the z-transform relationship of Eq. (22), it can be shown that the associated difference equation is

$$u(k) = u(k-1) + b_0 e(k) + b_1 e(k-1) + b_2 e(k-2) \tag{26b}$$

By comparing coefficients in Eqs. (26a) and (26b), it follows that $b_0 = K_\text{p} + K_\text{i} + K_\text{d}$, $b_1 = -(K_\text{p} + 2K_\text{d})$, and $b_2 = K_\text{d}$.

The recursive form of Eq. (26b) is an excellent way to implement digital PID. It lends itself naturally to anti-reset-windup: each time $u(k)$ is calculated, if it is outside the saturation bounds, it is reassigned the value of the nearest saturation bound. For real-time implementation, we use the following notation: $u \equiv$ current value of u; $u_1 \equiv$ most recent past value of u; $e \equiv$ current value of e; $e_1 \equiv$ most recent past value of e; and so on. The following sequence of events constitutes a flow diagram for computer code to implement the algorithm:

1. Initialize $e_1 = e_2 = u_1 = 0$.
2. Compute temp $= u_1 + b_1e_1 + b_2e_2$.
3. Wait for the sample period.
4. At the sample period, obtain $e = r - c$ using appropriate A/D converters.
5. Compute $u =$ temp $+ b_0e$.
6. If $u > u_{\max}$, assign $u = u_{\max}$, or if $u < u_{\min}$, assign $u = u_{\min}$.
7. Output u to the DAC.
8. Assign in proper order $e_2 = e_1$, $e_1 = e$, and $u_1 = u$.
9. Return to step (2).

Note that the above steps are arranged to minimize the number of calculations required between the instant that e is obtained from the ADC and the instant that u is sent to the DAC. This process minimizes the computational time delay that is introduced into the control loop; computational time delay can have a detrimental effect on system performance. Also, for two-degree-of-freedom implementations, the terms from Eq. (25) would need to be included appropriately in the code.

Signal Property Considerations

Both control signals and measured process signals are constrained in a variety of ways. Common examples of constrained control signals are (1) signals that are continuously adjustable between saturation levels; (2) control signals that can be assigned only a few values—as in the case of a relay with dead zone, with possible output values $-u_{\max}$, 0, or $u_{\max}$; and (3) control signals that are pulse-width modulated. Even in case (1) above, when the control is supplied by the output of a DAC, the finite word length of the DAC may have a noticeable quantization effect for small steady-state error signals. In case (2) above, when the control is supplied by an electromechanical relay (once a common practice, but now largely displaced by power electronics), a correct balance of P and D terms is needed to avoid excessive switching of the relay; in all-electronic implementations of such systems, rapid switching actually may be included intentionally to achieve a *sliding* mode of control, and nonlinear control strategies involving nonlinear functions of e and $\dot{e}$ are appropriate. Case (3) above is especially prevalent in electronic drives for dc motors. This is because a power amplifier dissipates the least energy if it is either full on or full off. To reduce heat buildup in the amplifier, the control is implemented as follows: if the desired control action from $t = kT$ to $t = (k + 1)T$ is u_o where $-u_{\max} \le u_o \le u_{\max}$, the effectively equivalent pulse-width control supplied is

$$u(t) = \begin{Bmatrix} u_{\max}\mathrm{sign}(u_o),\ kT \le t < kT + \Delta T \\ 0,\ kT + \Delta T \le t < (k+1)T \end{Bmatrix} \tag{27}$$

where $\Delta T = T|u_o|/u_{\max}$, and

$$\mathrm{sign}(u_o) = \begin{Bmatrix} 1,\ u_o > 0 \\ 0,\ u_o = 0 \\ -1,\ u_o < 0 \end{Bmatrix} \tag{28}$$

When the sample period T is larger than the major electrical time constant of the motor drive circuit, but is substantially smaller than the dominant mechanical time constant of the motor and its load, the above control provides smooth operation of the motor, and a conventional linear model of the motor-and-drive generally can be used in the design of the control system.

Measured process signals supplied to a digital controller are quantized, sometimes as a result of ADC conversions, but in other cases as a result of the types of measurement sensors employed. For example, an optical rotary position sensor provides a finite number of angular positions for one complete revolution of the sensor disk. In this case, when the magnitude of $e(t)$ is very small, the sensor error can switch abruptly from 0 to ϵ or $-\epsilon$, where ϵ is the effective quantization level of the sensor. To avoid very-low-level oscillations of the control under such conditions, it may be necessary to zero the input to the PID integrator when $|e(t)|$ is small, and to effect a small dead-zone notch in the control action.

Other Digital PID Issues

Other issues in digital controller design are sampling period selection, antialiasing, and process modeling. Ideally, the sampling period T should be selected to be an order of magnitude smaller than the dominant time constants of the system being controlled. However, extremely small values of T may lead to round-off error and computational problems on finite-word-length computers. The choice of sampling period also is affected by the amount of process and measurement noise. The folding frequency associated with T is $0.5/T$ Hz. When a signal that has a frequency component above the folding frequency is sampled, the sampled signal exhibits an associated aliased low-frequency component. Thus, for digital implementation, generally signals are filtered with an analog antialiasing filter prior to sampling. Often adequate for the purpose is a first-order low-pass filter with a corner frequency near $0.5/T$ Hz.

When a digital controller is designed using z-transform methods, it is helpful to have a representative z-transform transfer-function model of the process being controlled—for example, a model corresponding to a sample-and-hold circuit followed by the process transfer function $G(s)$ of Fig. 2. In the case of a conventional zero-order hold circuit, the required z-transform transfer function of the sample-and-hold in series with the process is

$$G_{\mathrm{hp}}(z) = (1 - z^{-1})Z\{G(s)/s\} \tag{29a}$$

or

$$G_{\mathrm{hp}}(z) = (1 - z^{-1})Z\left\{\int_0^t g(\tau)\,d\tau\right\} \tag{29b}$$

The characteristic equation of the feedback loop is then

$$1 + G_{\mathrm{PID}}(z)G_{\mathrm{hp}}(z) = 0 \tag{30}$$

with closed-loop poles of the system being roots of the characteristic equation. The PID gains can be adjusted to obtain desirable pole locations inside the unit circle of the z plane, and a variety of other digital controller design methods can be employed.

Emulation Method of Digital PID Design

The process of selecting of PID gains to achieve desirable goals is called *tuning.* Because most PID tuning methods have been developed for analog PID controllers, a natural question is can we design an analog PID using established tuning rules and then translate the analog PID gains into meaningful digital PID gains? Of course, the answer is generally yes. However, there are several factors that should be considered in doing this. In the first place, the sample-and-hold operation that is inserted in the control loop has the effect of introducing a time delay, of approximately $T/2$ s into the loop. Also, computational delay, denoted by δT s where $0 < \delta < 1$, is introduced. These delays need to be taken into account when converting from analog PID gains to digital PID gains.

Consider the following analog $G_{\mathrm{PID}}(s)$:

$$G_{\mathrm{PID}}(s) = k_{\mathrm{p}} + \frac{k_{\mathrm{i}}}{s} + \frac{k_{\mathrm{d}}s}{1+\tau s} \tag{31}$$

In a general emulation approach developed by Pierre and Pierre (3), $G_{\mathrm{PID}}(s)$ is replaced by digital controller transfer function $G_{\mathrm{c}}(z)$:

$$G_{\mathrm{c}}(z) = G_{\mathrm{a}}(z)G_{\mathrm{PID}}\left(\frac{2}{T}\frac{z-1}{z+1}\right) \tag{32}$$

where the replacement of s by $(2/T)(z-1)/(z+1)$ in G_{PID} is the well known Tustin approximation, and where

$$G_{\mathrm{a}}(z) \triangleq \frac{[0.5(3-a)+(1-a)\delta]z-[0.5(1+a)+(1-a)\delta]}{z-a} \tag{33}$$

in which a is an additional tuning parameter (typically $-0.3 \le a \le 0.4$). It is readily shown that the dc gain of $G_{\mathrm{a}}(z)$ is $G_{\mathrm{a}}(1) = 1$ and that the zero in Eq. (33) is to the right of the pole thereby providing phase lead. $G_{\mathrm{a}}(z)$ compensates to some degree for sample-and-hold delay and for computational delay δT in the control loop.

As a specific example of $G_{\mathrm{a}}(z)$, the case where $a = -0.2$ and $\delta = 0$ in Eq. (33) gives

$$G_{\mathrm{a}}(z) = \frac{1.6z-0.4}{z+0.2} \tag{34}$$

It is important to have a in Eq. (33) bounded away from $z = -1$; poles of the controller near $z = -1$ often are ringing poles, generating oscillations of period $2T$ in $u(t)$ that are not readily observed in $c(t)$.

From Eqs. (31) and (32), along with assigning $\tau = T/2$, it follows that

$$G_{\mathrm{c}}(z) = G_{\mathrm{a}}(z)\left[k_p + \frac{k_{\mathrm{i}}T(z+1)}{2(z-1)} + \frac{k_{\mathrm{d}}(z-1)}{Tz}\right] \tag{35}$$

A controller based on the $G_{\mathrm{c}}(z)$ of Eq. (35) can be implemented using a variety of digital control programming methods; practical implementations insure that integrator windup is avoided.

PID TUNING

Numerous studies have been made to develop assignment rules to specify PID parameters on the basis of characteristics of the process being controlled. There are many representative sources that can be consulted for details on a wide variety of alternative tuning rules (2,4–7). Common tuning rules are described in this section. Most of these approaches give excellent results if the process being controlled is well damped and is dominantly second order. Systems that are essentially first order can often be tuned well with PI control only. Systems with large time delay can be controlled with PID control in combination with a Smith predictor (8). From a frequency response viewpoint, the PID integral term supplies 90° of phase lag at low frequencies, and the practical derivative term supplies somewhat less than 90° of phase lead at some intermediate to high frequencies. Thus, if the system requires more than 90° of phase-lead compensation over some frequency range to achieve good performance, PID control alone will not be adequate.

For lightly damped oscillatory open-loop systems, one method that is tempting (but should be approached with caution) is to place the zeros of the PID controller at the pole locations of the process, to cancel the oscillatory poles. This approach should be avoided if the PID controller employed has one degree of freedom only: although the response mode of the canceled poles will not be excited by reference inputs, disturbance inputs will excite the oscillatory mode in the controlled output, and the PID controller with its zero gain at the oscillatory frequency will not supply damping. For two-degree-of-freedom systems, the PID gains within the loop can be assigned to add damping to the oscillatory mode, whereas the independent PD′ factors associated with the reference input can be adjusted to provide blocking zeros. Sensitivity of oscillatory pole-zero cancellations with respect to parameter changes also is of concern (9).

Ziegler–Nichols Tuning

A widely applied tuning rule, the *ultimate gain* rule, was developed in 1942 by Ziegler and Nichols (10). This rule is based on experimentation with the process to be controlled. First, with k_{i} and k_{d} set to zero in Eq. (3), the proportional gain k_{p} is increased until the system starts to oscillate; the value of k_{p} that starts the system oscillating is denoted as k_{u} and is called the ultimate gain. The period of the oscillation, T_{u}, also is recorded. As an alternative procedure, rather than determining k_{u} and T_{u} by forcing the actual system into instability, if the frequency response $G(j\omega)$ is available, it can be used to obtain both k_{u} and T_{u} analytically, as follows. First, the frequency ω_{u} at which the angle of $G(j\omega_{\mathrm{u}}) = 180°$ is determined, and the corresponding value of $|G(j\omega_{\mathrm{u}})| = A_{\mathrm{u}}$ is obtained. The value of k_{u} satisfies the characteristic equation

$$1 + k_{\mathrm{u}}G(j\omega_{\mathrm{u}}) = 0 \tag{36}$$

and therefore $k_{\mathrm{u}} = 1/A_{\mathrm{u}}$ and $T_{\mathrm{u}} = 2\pi/\omega_{\mathrm{u}}$. An ultimate gain tuning rule is then:

$$k_{\mathrm{p}} = 0.6k_u \tag{37a}$$

$$k_{\mathrm{i}} = 1.2k_u/T_u \tag{37b}$$

and

$$k_{\mathrm{d}} = 3k_uT_u/40 \tag{37c}$$

This ultimate-gain rule is but a rule-of-thumb; although it was developed with a nominal 20% step response overshoot goal, it is easy to find cases where it results in overshoots in excess of 50%. It should not be surprising that the rule does not apply well to all cases because process characteristics vary widely.

Modified Ziegler–Nichols Methods

Many variations of the ultimate gain rule have been developed. One popular one credited to Harriott in (4) is as follows. First, with k_i and k_d set to zero in Eq. (3), k_p is adjusted until the step-response of the closed loop exhibits a decay ratio of 0.25 (the percent overshoot of the second peak in the step response is one-fourth of the percent overshoot of the first peak in the step response). Let k_c denote the critical value of k_p corresponding to this 0.25 decay ratio. Also, let T_c denote the difference between the time of occurrence of the second peak and that of the first peak. Next, assign k_i and k_d on the basis that

$$k_i = 1.5k_c/T_c \tag{38a}$$

and

$$k_d = k_c T_c/6 \tag{38b}$$

And finally, while conducting additional closed-loop step-response tests, adjust all three gains by the same percentage until desirable overshoot conditions are achieved.

More Advanced Tuning Methods

When the transfer function $G(s)$ of Fig. 2 is known and is a reasonably good model of the process being controlled, an array of tuning methods are available, depending on the form of $G(s)$. For example, variations of the Ziegler–Nichols methods are based on cases where $G(s)$ assumes forms such as

$$G(s) = \frac{K_0 e^{-sT_0}}{\tau_0 s + 1} \tag{39}$$

or

$$G(s) = \frac{K_0 e^{-sT_0}}{s} \tag{40}$$

A pole-placement approach has been developed (2) for those cases where $G(s)$ is of the form

$$G(s) = \frac{b_1 s + b_2}{s^2 + a_1 s + a_2} \tag{41}$$

The characteristic equation for the closed loop is

$$1 + G_c(s)G(s) = 0 \tag{42}$$

When the PID transfer function of Eq. (4) is substituted for $G_c(s)$, and $G(s)$ of Eq. (41) is substituted into Eq. (42) also, the resulting equation reduces to a third-order polynomial having coefficients that are functions of k_p, k_i, k_d, b_1, b_2, a_1, and a_2. A desirable placement of the poles of the system can be achieved by equating coefficients of the above polynomial to coefficients of $s^3 + \beta_1 s^2 + \beta_2 s + \beta_3$, where the roots of this polynomial are assigned to achieve desirable damping and natural frequency. Explicit relations for k_p, k_i, and k_d can be developed in terms of β_1, β_2, β_3, b_1, b_2, a_1, and a_2.

Tuning methods also can be based on frequency response characteristics and the sensitivity function

$$S(j\omega) \triangleq \frac{1}{1 + G_c(j\omega)G(j\omega)} \tag{43}$$

Generally the values of k_p, k_i, and k_d should be such that

$$\max_{\omega} |S(j\omega)| < 1.5 \tag{44}$$

Values smaller than 1.5 in Eq. (44) usually correspond to well damped systems.

Tuning methods based on optimization techniques can be applied directly to a particular system if a valid $G(s)$ representation of the controlled process is available. Åström and Hägglund (2) do a systematic evaluation of many cases to generate tuning diagrams that can be used to obtain desirable PID gains for a variety of $G(s)$ forms.

CONCLUSION

In his classical paper on governors, Maxwell (11) in 1868 clearly expressed the difference in effects produced by proportional and integral control. However, there is no one person that can be credited with having *invented* PID control. Many of the early developments, those involving anti-reset-windup for example, were company proprietary and therefore were not available in the open literature. With the advent of computer control in the 1960s, many of the traditional PID techniques were reassessed and translated into the digital control algorithms.

In this article, the basics of PID control have been described. Operational features of both linear PID terms and nonlinear characteristics have been examined. Digital PID algorithms have been described and have a special significance in the modern era of digital control.

Today, PID controllers can be purchased from many manufacturers [see Åström and Hägglund (2) for a recent listing of companies and features available]. In addition to being used for SISO control, PID controllers are used in a variety of other applications. In *cascade control,* one control variable is generated on the basis of several measured variables. In *selector control* one actuator can be driven from a selected PID unit, and then automatically switched to a different PID unit based on a max/min selector to control some other process signal if it starts to deviate from a desired range. In *ratio control* two outputs are controlled with one of the outputs required to be a fixed percentage of the other output. In *split-range control* and *multiactuator control* there may be fewer measured signals than actuators, and the amount of control from each actuator has to be automatically balanced to achieve the desired overall process performance goals. Feedforward control of disturbances, when they can be measured, often leads to vastly improved disturbance rejection. Automatic gain scheduling of PID parameters can be based on measured process conditions. And interactions of coupled control loops can be reduced by properly designed coupled PID controllers. In many cases, aspects of PID control are blended with other control concepts, leading to higher-order controllers. PID developments in the future will be, and already have been to some extent, coupled with fuzzy control, neural-network control, and adaptive control.

BIBLIOGRAPHY

1. D. P. Eckman, *Principles of Industrial Process Control,* 11th ed., New York: Wiley, 1961.
2. K. J. Åström and T. Hägglund, *PID Controllers: Theory, Design, and Tuning,* 2nd ed., Research Triangle Park, NC: Instrument Society of America, 1995.
3. D. A. Pierre and J. W. Pierre, Digital controller design—alternative emulation approaches, *ISA Trans. J. Instr. Soc. Amer.,* **34**: 219–228, 1995.
4. B. G. Liptak, (ed.), *Instrument Engineers Handbook, Vol. II, Process Control,* New York: Chilton, 1970.
5. R. Isermann, *Digital Control Systems,* Vol. 1, 2nd ed., Berlin: Springer-Verlag, 1989.
6. C. L. Smith, *Digital Computer Process Control,* Scranton, PA: International Textbook, 1972.
7. C. C. Hang, K. J. Åström, and W. K. Ho, Refinements of the Ziegler–Nichols tuning formula, *IEE Proc.,* **D138** (2): 111–118, 1991.
8. T. Hägglund, A predictive PI controller for processes with long dead times, *IEEE Control Syst. Mag.,* **12** (1): 57–60, 1992.
9. D. A. Pierre, Root locus near isolated pole-zero dipoles and a counterintuitive case of digital control compensation, *IEEE Trans. Auto Control,* **AC-29**: 439–441, 1984.
10. J. G. Ziegler and N. B. Nichols, Optimum settings for automatic controllers, *Trans. ASME,* **64** (8): 759–768, 1942.
11. J. C. Maxwell, On governors, *Proc. R. Soc. Lond.,* **16**: 270–283, 1868. Also published in R. Bellman and R. Kalaba (eds.), *Mathematical Trends in Control Theory.* New York: Dover Publications, 1964, pp. 3–17.

DONALD A. PIERRE
Montana State University