Whether the control implementation be pneumatic, electronic analog, digital computer, or digital controller, the objective function can be simply stated. Moreover, a logical sequence of reasoning leads naturally to the familiar PID format for control, points out author Pemberton. Why then should we strain to create more sophisticated algorithms, is his thesis. Why not understand PID more clearly? Why not use PID more effectively? In this roundup of PID thinking, new light is shed on the derivation of this algorithm; and in a sequel, a simple rule for tuning PID controllers is verified by simulation.

PID: The Logical Control Algorithm

T. J. PEMBERTON, Applied Automation

Instead of culling the PID algorithm from the pages of a book, let it be derived naturally by observing a feedback control system model and specifying its control element such that a basic control objective function shall be satisfied.

The well-known first- and second-order lag plus dead time models for "process output divided by controller output" are used for this purpose (Figure 1):

$$\frac{C(s)}{M(s)} = G_P(s) = \frac{K\exp(-\tau_d s)}{\tau_1 s + 1} \tag{1A}$$

for a first-order lag, and

$$G_P(s) = \frac{K\exp(-\tau_d s)}{(\tau_1 s + 1)(\tau_2 s + 1)} \tag{1B}$$

for a second order lag, where

τ_d is dead time
τ_1 is time lag of first-order system
τ_1 and τ_2 are time lags of second-order system
K is process gain
$G_P(s)$ is process transfer function
$C(s)$ is process output
$M(s)$ is controller output to process

In this approach, the control transfer function $G_c(s)$ in Figure 1 is specified by first defining a control objective function. Recognizing that $C(s)$ responds to a change in setpoint $R(s)$ only after a delay of τ_d sec, the control objective may be defined:

$$C(s) = R(s)\exp(-\tau_d s) \tag{2}$$

But the block diagram of Figure 1 shows this quantity to be:

$$C(s) = \frac{R(s)G_c(s)G_P(s)}{1 + G_c(s)G_P(s)} \tag{3}$$

It is necessary to find what algorithm (control function) should be implemented in $G_c(s)$. Combining Equations 2 and 3 and solving for $G_c(s)$ appears to be a logical procedure.

The first step yields

$$\exp(-\tau_d s) = \frac{G_c(s)G_P(s)}{1 + G_c(s)G_P(s)} \tag{4}$$

Before solving for $G_c(s)$, the expression for $G_P(s)$ from Equation 1B is substituted in Equation 4. The reason for taking the dual lag case first is that in the resulting algorithm the second lag τ_2 can always be set to zero to satisfy the system of Equation 1A. Equation 4 then becomes

$$\exp(-\tau_d) = \frac{G_c(s)\dfrac{K\exp(-\tau_d)}{(\tau_1 s + 1)(\tau_2 s + 1)}}{I + \dfrac{G_c(s)K\exp(-\tau_d)}{(\tau_1 s + 1)(\tau_2 s + 1)}} \tag{5}$$

which reduces with algebraic manipulation to

$$G_c(s) = \frac{(\tau_1 s + 1)(\tau_2 s + 1)}{K} + G_c(s)\exp(-\tau_d) \tag{6}$$

To illustrate the actual controller that now evolves (Figure 2), Equation 6 is restated in terms of $M(s)$, the controller output after the time delay τ_d has had its effect. This leads to

$$G_c(s) = \frac{M(s)}{E(s)} = \frac{(\tau_1 s + 1)(\tau_2 s + 1)}{K[1 - \exp(-\tau_d)]} \tag{7}$$

FIG. 1. Simplified feedback control system used in the text to define a control objective function for the purpose of deriving a suitable control algorithm.

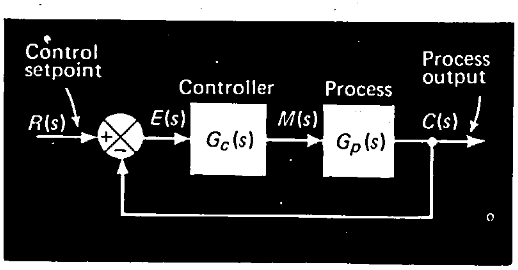
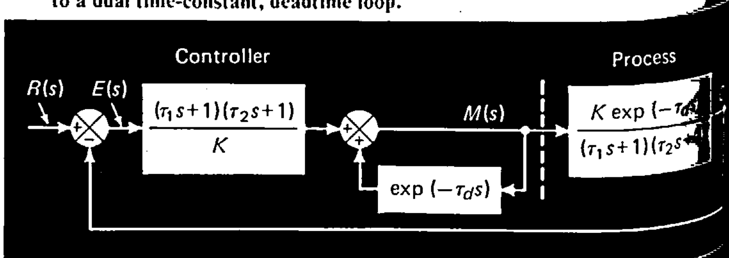


FIG. 2. As the derivation of a control algorithm continues, keeping the stated control objective in mind (Equation 2), the feedback control system of Figure 1 acquires parametric detail conforming to a dual time-constant, deadtime loop.



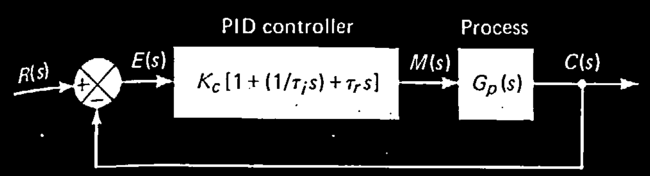


FIG. 3. The feedback control system that finally emerges has a relatively simple controller that implements a PID algorithm.

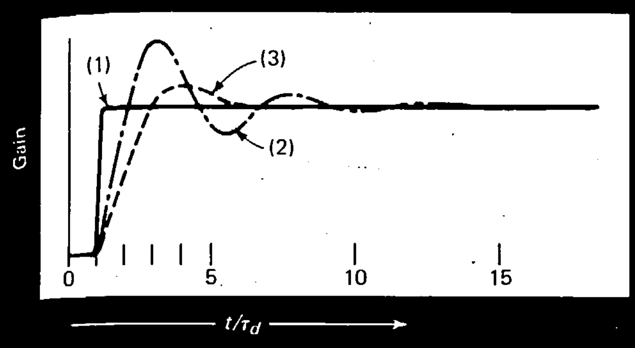


FIG. 4. Setpoint responses of three controller implementations described in the text. (1) Ideal controller (Equation 7), (2) PID controller (Equation 11), (3) PID controller (Equation 11, with gain modified by a factor of 2/3).

which in terms of controller output as affected by delay yields

$$M(s) = \frac{(\tau_1 s + 1)(\tau_2 s + 1)}{K} + M(s)\exp(-\tau_d) \qquad (8)$$

Satisfaction of the control objective function of Equation 2 may be verified by substituting $G_c(s)$ and $G_P(s)$ into Equation 3 from Equations 7 and 1B respectively.

An additional step leads to the conventional PID algorithm. Substitution of the truncated series expansion of the exponential function

$$\exp(-\tau_d s) \cong 1 - \tau_d s \qquad (9)$$

into Equation 7 yields

$$G_c(s) = \frac{1}{K\tau_d s}[\tau_1\tau_2 s^2 + (\tau_1 + \tau_2)\, s + 1] \qquad (10)$$

Multiplying numerator and denominator by $(\tau_1 + \tau_2)s$ derives the familiar PID algorithm

$$G_c(s) = \frac{\tau_1 + \tau_2}{K\tau_d}\left[1 + \frac{1}{(\tau_1 + \tau_2)s} + \frac{\tau_1\tau_2 s}{\tau_1 + \tau_2}\right] \qquad (11)$$

where proportional gain is $K_c = (\tau_1 + \tau_2)/K\tau_d$
reset time is $\tau_i = \tau_1 + \tau_2$
rate time is $\tau_r = \tau_1\tau_2/(\tau_1 + \tau_2)$

Letting $\tau_2 = 0$ makes these results applicable to the first-order system of Equation 1A.

In terms of PID parameters (see Figure 3):

$$G_c(s) = K_c\left(1 + \frac{1}{\tau_i s} + \tau_r s\right) \qquad (12)$$

The amount of error introduced by the approximation of Equation 9 depends on the magnitude of $\tau_d s$. From the phase-shift point of view, the error is very small when $\tau_d s$ is less than ½. Thus if the significant frequency components of $M(s)$ are below $1/2\tau_d$, no appreciable error is introduced. However, step changes in setpoint violate this limited bandwidth restriction with observable effects.

The three traces of Figure 4 are

- Ideal response resulting from the use of $G_c(s)$ as specified in Equation 7 (curve 1)
- Response resulting from the use of $G_c(s)$ as specified in Equation 11 (curve 2)
- Response resulting from the use of $G_c(s)$ as specified in Equation 11, but with proportional gain K_c reduced by a factor of 2/3 (curve 3)

Curve 2 shows the effects of introducing the approximation of Equation 9. Note how the overshoot of curve 2 is reduced by the 2/3 factor in curve 3. If the proportional gain were reduced to four-tenths of its value in curve 2, there would be no overshoot, but rise time would be correspondingly longer, (Ref.).

For digital PID systems, the approximation of Equation 9 may be avoided, and Equation 8 may be implemented directly.

PID aftermath

It is noteworthy that the PID algorithm of Equation 12 followed directly from

- Process control models that are well accepted (Equations 1A, 1B.)
- A concisely stated control objective (Equation 2)
- A commonly used approximation to the exponential function (Equation 9)

The derivation leading to Equation 12 also yielded rather simple formulas for tuning PID controllers in terms of time constants, deadtime, and gain.

Four critical observations relate to Equation 12:

- Even though its "ideal" derivative action is only approximated by commercial controllers, the objective of Equation 2 is substantially achieved.
- Derivative action is applied to the error, that is, to both the setpoint change and the process response to load changes.
- To the extent that the process parameters τ_1, τ_2, τ_d, and K are known, tuning the PID controller according to Equation 11 will enhance the quality of control provided by Equation 12.
- The smaller the error in the approximation of Equation 9, the more closely the control action of Equation 12 will approach the Equation 2 objective.

Simulation tests conducted to verify these derivations yielded results comparing favorably with those of the Ziegler-Nichols Ultimate method. It would seem that the need is not to search for better control algorithms, but to improve tuning procedures for the basic PID algorithm.

In a sequel, the subject will be explored in greater depth, with the accent on realistic tuning practice. □

REFERENCE

"Adjusting Controllers for a Deadtime Process," A. Haalman, *Control Engineering*, July '65, pp. 71-73.

The natural derivation of a control algorithm to satisfy a basic control objective function was presented in a previous article (CE, May, '72), and led quite simply and directly to the well-known PID algorithm. In the present discussion, author Pemberton further expounds on his thesis that control sophistication can stem from skilled application of the PID algorithm more simply than from a search for more complex analog or digital algorithms. He uses computer simulation to substantiate his point.

PID: The Logical Control Algorithm—II

T. J. PEMBERTON, Applied Automation

Regardless of the form of the algorithm that is mechanized, it is important to tune or adjust an analog or a digital controller to the process which the controller is to regulate. Automatic control texts abound with tuning methods (e.g., Refs. 1, 2), each of which requires its own type of process knowledge. Moreover, each of these methods, when properly applied, is structured to produce a desired form of process response.

The present discussion examines ways of simplifying the tuning procedure, and is addressed as much to the several methods appearing in the literature as to the set of tuning formulas presented in the previous article (CE, May '72).

The recommendations that are presented are based on a study involving a simulated, second-order lag plus deadtime process. Both an analog-simulated and a commercial PID controller were used to regulate the simulated process. The PID algorithm was used exclusively, conforming with these conclusions reached in the previous article: that PID is a naturally derived algorithm satisfying the purpose of control, that it is amenable to either analog or digital implementation, and that it is capable of control sophistication if properly married to the process.

So many effective tuning methods exist that it seemed pointless to strain beyond present status to develop other procedures, aside from the small set of formulas that grew naturally from previous analysis. Rather, study effort was devoted to making tuning procedures easier to use, on the premise that control loops might then be more frequently and consistently updated to compensate for process drift.

Proportioning rate time to reset time

The full capability of many PID controllers is not utilized because rate action is so often tuned out (rate time set to zero). Why should derivative control with its beneficial anticipatory effects be thus casually eliminated? One reason is that operators adjusting controller knobs seem able to more easily visualize and understand the effects of proportional and reset action than rate action. Adjusting the reset and proportional knobs quite often results in what appears to be good, stable control. The tendency then may well be to "let well enough alone", and the additional control sophistication that could come from a fine setting of the rate knob is not realized.

The results discussed below show that not only is rate action beneficial but PID controllers can be tuned with only two knobs: proportional and reset. To understand this seeming paradox, the relationships between reset time and rate time prescribed by several well-known tuning methods are examined (table). Recurrence of the value 0.25 is apparent, suggesting the possibility that for many applications the rate time might be arbitrarily set at one-fourth the reset time.

To verify this assumption, numerous setpoint responses were recorded for a control loop containing a PID controller and a second-order lag plus deadtime process. Experiments were varied between a Motorola Model 55RC and an analog simulation of an ideal PID controller. The Ziegler-Nichols Ultimate Period (Z-N) formulas were selected—for no reason other than their ease of use—to provide a basis for comparison of setpoint responses, Figure 1.

Each set of setpoint responses corresponds to a unique adjustment of process dynamics. In each case

Methods for Tuning A Controller to the Process

Method	Rate time / Reset time
Ziegler-Nichols (open loop) [1,2]	0.25
Cohen-Coon (open loop) [1]	0.125 – 0.175
3-Constraint (open loop) [1]	0.25 – 0.50
Ziegler-Nichols Ultimate Period (closed loop) [1,2]	0.25
Damped Oscillation (closed loop) [2]	0.25
Gallier-Otto formulas (IAE criterion) [3]	0.12 – 0.25
Murrill-Smith formulas (ITAE criterion) [1]	0.22 – 0.32
Murrill-Smith formulas (ISE criterion) [1]	0.35 – 0.60
Ideal PID tuning formulas (CE, May 1972)	0.10 – 0.25

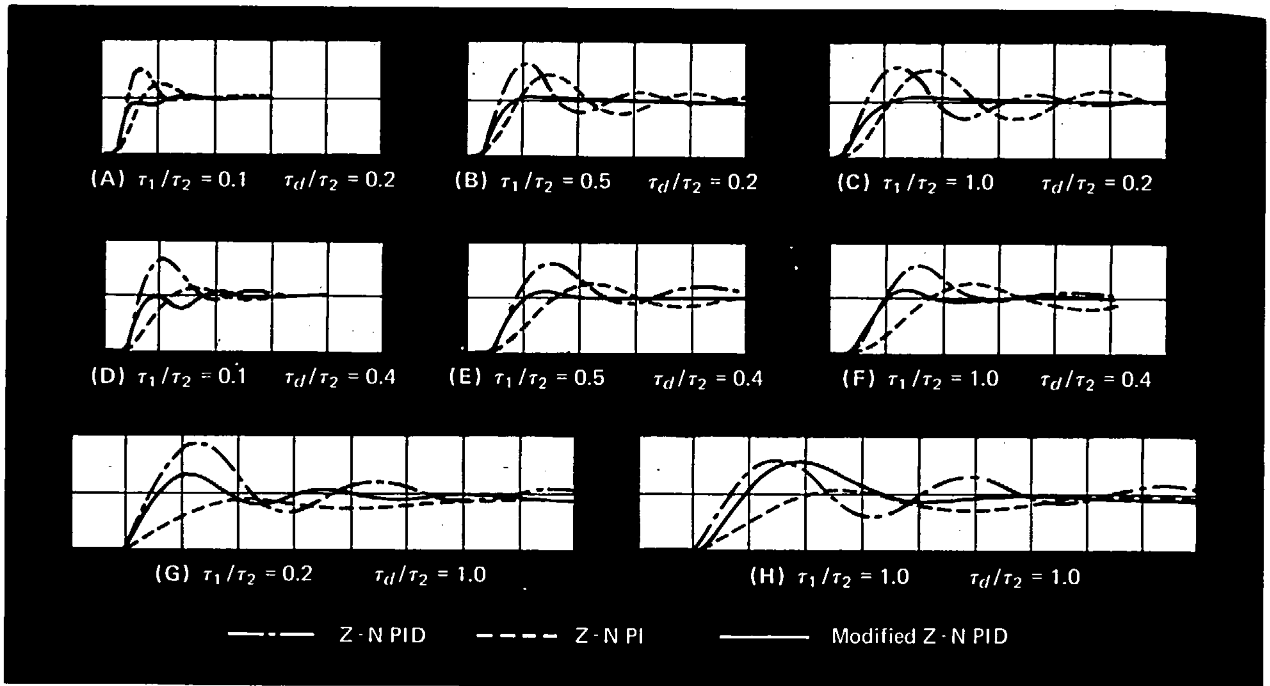


FIG. 1. Responses to setpoint changes of a simulated second-order lag plus deadtime process under control of a simulated analog controller using three variations on the Ziegler-Nichols Ultimate Period tuning method listed in the table.

the three curves represent responses of the process to Z-N PID, Z-N PI, and Mod Z-N PID formulas. The Mod Z-N PID formula, producing superior response in seven out of eight dynamic profiles, was obtained by modifying the Z-N PI formula to include a rate term equal to one-fourth the reset time.

For the second-order lag plus deadtime process model,

$$G_P(s) = \frac{K\exp(-\tau_d s)}{(\tau_1 s + 1)(\tau_2 s + 1)}$$

the range of parameters was τ_1/τ_2 varied from 0.1 to 1.0, and τ_d/τ_2 varied from 0.2 to 1.0 for each value of τ_1/τ_2.

Trying out the "ideal" PID algorithm

The PID algorithm (Equation 12 in the previous article),

$$G_c(s) = K_c\left(1 + \frac{1}{\tau_i s} + \tau_r s\right)$$

is referred to as "ideal" in this article because it has ideal derivative action that is only approximately implemented in commercial controllers. For tuning purposes, the previous article (Equation 11) gives

$$K_c = (\tau_1 + \tau_2)/K\tau_d$$
$$\tau_i = \tau_1 + \tau_2$$
$$\tau_r = \tau_1\tau_2/(\tau_1 + \tau_2)$$

Figure 2 presents a comparison of responses resulting from Z-N PI, Ideal PID, and Mod Ideal PID. Again, the modified PID is formulated by using Z-N PI with rate time added, scaled to one-fourth of the reset time.

It is apparent that the PID formulations are acceptable, although perhaps some people would find a bit more overshoot than desired. Figure 3 shows the effect on overshoot of reducing proportional gain to two-thirds and to one-half of the value prescribed by tuning formulas. While the plot of Figure 3 applies to the Ideal PID, similar damping will be experienced for the Mod Ideal PID.

On the basis of the traces of Figure 3, the recommended formulas are

	Ideal PID	Mod Ideal PID
Proportional gain K_c	$\dfrac{2(\tau_1 + \tau_2)}{3K\tau_d}$	$\dfrac{2(\tau_1 + \tau_2)}{3K\tau_d}$
Reset time τ_i	$\tau_1 + \tau_2$	$\tau_1 + \tau_2$
Rate time τ_r	$\dfrac{\tau_1\tau_2}{\tau_1 + \tau_2}$	$\dfrac{\tau_i}{4}$

Note that in both Figures 1 and 2 the rate action yields faster rise times and shorter settling times than are possible with PI algorithms alone.

Curves shown in Figures 1, 2 and 3 were obtained on an analog computer simulation of PID and PI controllers. Many of these experiments were repeated using the Motorola Model 55RC, with results fully supporting the simulation findings.

How to handle commercial controllers

The idea of proportioning rate time to reset time is applicable to PID controllers whether they be electronic, pneumatic, or digital. The means for implementing this rate/reset ratio will vary from one type to another, as well as being affected by the type of process. Although the proportion of one-fourth is

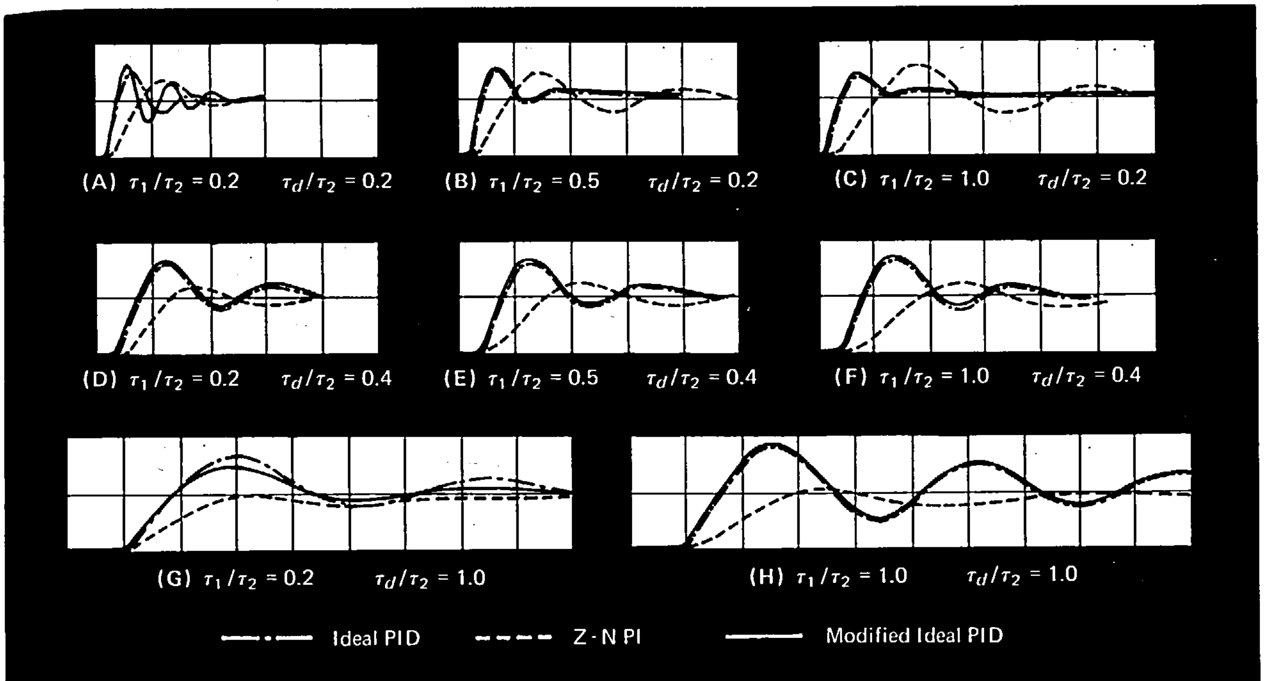


FIG. 2. Responses to setpoint changes of a simulated second-order lag plus deadtime process under control of a simulated analog controller using variations of the Ideal PID algorithm developed in the previous article (CE, May '72).

freely recommended, judgment should be used in a particular situation to ensure that shading one way or the other will not improve response.

Electronic analog control. Figure 4 shows a typical electronic analog arrangement. The adjustable rate time and reset time are accomplished by potentiometers or rotary contact switches that alter network resistance. Thus rate time may be proportioned automatically to reset time by using ganged potentiometers or switches. It follows that R_1 should track R_2 so as to cause the RC product relation

$$R_2C_2 = R_1C_1/4$$

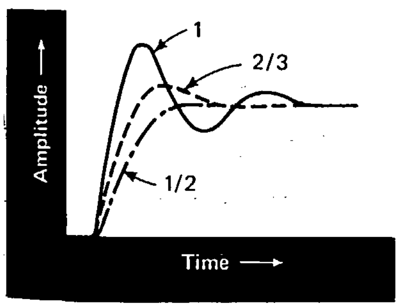
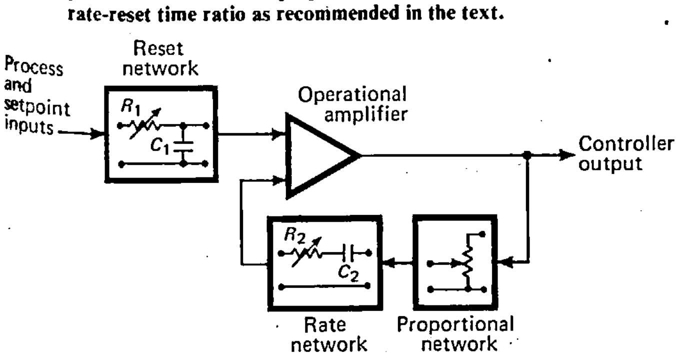


FIG. 3. Example of how overshoot shown in Figure 2 may be reduced by applying factors of two-thirds and one-half to the proportional-gain value prescribed by tuning formulas.

FIG. 4. Typical electronic analog PID controller showing the RC products that should be proportioned in order to achieve a constant rate-reset time ratio as recommended in the text.



Pneumatic analog controllers. Installing proportionate relation between reset and rate restrictors has not been attempted. It appears that mechanical linkages are required, so that the rate restrictor would be altered by the knob that alters the reset restrictor.

DDC with modified PID. In terms of direct digital control, the modified PID algorithm may be written:

$$M_n = M_{n-1} + K_c[e_n + e_{n-1} + e_n\Delta t/\tau_i + \tau_i(e_n + 2e_{n-1} + e_{n-2})/4\Delta t]$$

Only two tuning parameters, K_c and τ_i, are needed to adjust this three-mode algorithm.

Three of the nine tuning methods studied actually called out the rate-reset time ratio as 0.25; another four out of nine called for a range of values that includes this figure; only two failed to include this figure in their stated range of values. In all cases where ranges were given, the upper limit was no more than double the lower limit. Guided by this general agreement among the several tuning methods, a simulation study verified that good responses are consistently obtained by using the rate-reset time ratio of one-fourth. □

REFERENCES

1. *Automatic Control of Processes*, P. W. Murrill, International Textbook, Scranton, Pa., 1967.
2. *Process Control*, Peter Harriott, McGraw-Hill, New York, 1966.
3. "A Self-Tuning Method for Direct Digital Control," P. W. Gallier and R. E. Otto, *Instrumentation Technology*, Feb. '68, pp. 65-70.