PID Control System Analysis and Design

PROBLEMS, REMEDIES, AND FUTURE DIRECTIONS

By YUN LI, KIAM HEONG ANG, and GREGORY C.Y. CHONG

With its three-term functionality offering treatment of both transient and steady-state responses, proportional-integral-derivative (PID) control provides a generic and efficient solution to real-world control problems [1]–[4]. The wide application of PID control has stimulated and sustained research and development to "get the best out of PID" [5], and "the search is on to find the next key technology or methodology for PID tuning" [6].

This article presents remedies for problems involving the integral and derivative terms. PID design objectives, methods, and future directions are discussed. Subsequently, a computerized, simulation-based approach is presented, together with illustrative design results for first-order, higher order, and nonlinear plants. Finally, we discuss differences between academic research and industrial practice, so as to motivate new research directions in PID control.

STANDARD STRUCTURES OF PID CONTROLLERS

Parallel Structure and Three-Term Functionality

The transfer function of a PID controller is often expressed in the ideal form

1066-033X/06/$20.00©2006IEEE

$$G_{\mathrm{PID}}(s) = \frac{U(s)}{E(s)} = K_{\mathrm{P}}\left(1 + \frac{1}{T_{\mathrm{I}}s} + T_{\mathrm{D}}s\right), \tag{1}$$

where $U(s)$ is the control signal acting on the error signal $E(s)$, K_{P} is the proportional gain, T_{I} is the integral time constant, T_{D} is the derivative time constant, and s is the argument of the Laplace transform. The control signal can also be expressed in three terms as

$$\begin{aligned} U(s) &= K_{\mathrm{P}}E(s) + K_{\mathrm{I}}\frac{1}{s}E(s) + K_{\mathrm{D}}sE(s) \\ &= U_{\mathrm{P}}(s) + U_{\mathrm{I}}(s) + U_{\mathrm{D}}(s), \end{aligned} \tag{2}$$

where $K_{\mathrm{I}} = K_{\mathrm{P}}/T_{\mathrm{I}}$ is the integral gain and $K_{\mathrm{D}} = K_{\mathrm{P}}T_{\mathrm{D}}$ is the derivative gain. The three-term functionalities include:

1) The proportional term provides an overall control action proportional to the error signal through the allpass gain factor.
2) The integral term reduces steady-state errors through low-frequency compensation.
3) The derivative term improves transient response through high-frequency compensation.

A PID controller can be considered as an extreme form of a phase lead-lag compensator with one pole at the origin and the other at infinity. Similarly, its cousins, the PI and the PD controllers, can also be regarded as extreme forms of phase-lag and phase-lead compensators, respectively. However, the message that the derivative term improves transient response and stability is often wrongly expounded. Practitioners have found that the derivative term can degrade stability when there exists a transport delay [4], [7]. Frustration in tuning K_{D} has thus made many practitioners switch off the derivative term. This matter has now reached a point that requires clarification, as discussed in this article. For optimum performance, K_{P}, K_{I} (or T_{I}), and K_{D} (or T_{D}) must be tuned jointly, although the individual effects of these three parameters on the closed-loop performance of stable plants are summarized in Table 1.

The Series Structure

If $T_{\mathrm{I}} \geq 4T_{\mathrm{D}}$, the PID controller can also be realized in a series form [7]

$$G_{\mathrm{PID}}(s) = (\alpha + T_{\mathrm{D}}s)\,K_{\mathrm{P}}\left(1 + \frac{1}{\alpha T_{\mathrm{I}}s}\right) \tag{3}$$

$$= G_{\mathrm{PD}}(s)G_{\mathrm{PI}}(s), \tag{4}$$

where $G_{\mathrm{PD}}(s)$ and $G_{\mathrm{PI}}(s)$ are the factored PD and PI parts of the PID controller, respectively, and

$$\alpha = \frac{1 \pm \sqrt{1 - 4T_{\mathrm{D}}/T_{\mathrm{I}}}}{2} > 0.$$

THE INTEGRAL TERM

Destabilizing Effect of the Integral Term

Referring to (1) for $T_{\mathrm{I}} \neq 0$ and $T_{\mathrm{D}} = 0$, it can be seen that adding an integral term to a pure proportional term increases the gain by a factor of

$$\left|1 + \frac{1}{j\omega T_{\mathrm{I}}}\right| = \sqrt{1 + \frac{1}{\omega^2 T_{\mathrm{I}}^2}} > 1, \text{ for all } \omega, \tag{5}$$

and simultaneously increases the phase-lag since

$$\angle\left(1 + \frac{1}{j\omega T_{\mathrm{I}}}\right) = \tan^{-1}\left(\frac{-1}{\omega T_{\mathrm{I}}}\right) < 0, \text{ for all } \omega. \tag{6}$$

Hence, both gain margin (GM) and phase margin (PM) are reduced, and the closed-loop system becomes more oscillatory and potentially unstable.

Integrator Windup and Remedies

If the actuator that realizes the control action has saturated range limits, and the saturations are neglected in a linear control design, the integrator may suffer from windup; this causes low-frequency oscillations and leads to instability. The windup is due to the controller states becoming inconsistent with the saturated control signal, and future correction is ignored until the actuator desaturates.

Automatic Reset

If $T_{\mathrm{I}} \geq 4T_{\mathrm{D}}$ so that the series form (3) exists, antiwindup can be achieved implicitly through automatic reset. The factored PI part of (3) is thus implemented as shown in Figure 1 [8], [9].

Explicit Antiwindup

In nearly all commercial PID software packages and hardware modules, however, antiwindup is implemented explicitly through internal negative feedback, reducing $U_{\mathrm{I}}(s)$ to [8]–[10]

TABLE 1 **Effects of independent P, I, and D tuning on closed-loop response. For example, while K_{I} and K_{D} are fixed, increasing K_{P} alone can decrease rise time, increase overshoot, slightly increase settling time, decrease the steady-state error, and decrease stability margins.**

	Rise Time	Overshoot	Settling Time	Steady-State Error	Stability
Increasing K_{P}	Decrease	Increase	Small Increase	Decrease	Degrade
Increasing K_{I}	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increasing K_{D}	Small Decrease	Decrease	Decrease	Minor Change	Improve

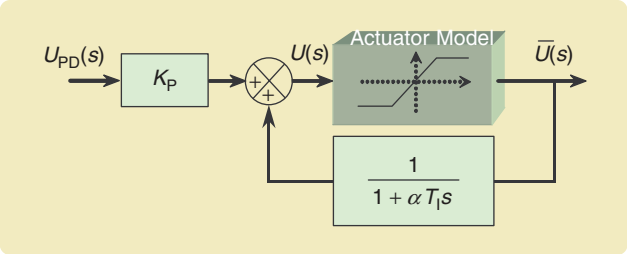

FIGURE 1 The PI part of a PID controller in series form for automatic reset. The PI part of a PD-PI factored PID transfer function can be configured to counter actuator saturation without the need for separate antiwindup action. Here, $U_{PD}(s)$ is the control signal from the preceding PD section. When $U(s)$ does not saturate, the feedforward-path gain is unity and the overall transfer function from $U_{PD}(s)$ to $\bar{U}(s)$ is thus $1 + 1/(\alpha T_I s)$, the same as the last factor of (3).

$$\tilde{U}_I(s) = \frac{1}{T_I s}\left[K_P E(s) - \frac{U(s) - \bar{U}(s)}{\gamma}\right], \tag{7}$$

where $U(s)$ is the theoretically computed control signal, $\bar{U}(s)$ is the actual control signal capped by the actuator limits, and γ is a correcting factor. A value of γ in the range [0.1, 1.0] usually results in satisfactory performance when the PID coefficients are reasonably tuned [7].

Accounting for Windup in Design Simulations

Another solution to antiwindup is to reduce the possibilities for saturation by reducing the control signal, as in linear quadratic optimal control schemes that minimize the tracking error and control signal through a weighted objective function. However, preemptive minimization of the control signal can impede performance due to minimal control amplitude. Therefore, during the design evaluation and optimization process, the control signal should not be minimized but rather capped at the actuator limits when the input to the plant saturates since a simulation-based optimization process can automatically account for windup that might occur.

THE DERIVATIVE TERM

Stabilizing and Destabilizing Effect of the Derivative Term

Derivative action is useful for providing phase lead, which offsets phase lag caused by integration. This action is also helpful in hastening loop recovery from disturbances. Derivative action can have a more dramatic effect on second-order plants than first-order plants [9].

However, the derivative term is often misunderstood and misused. For example, it has been widely perceived in the control community that the derivative term improves transient performance and stability. But this perception is not always valid. To see this, note that adding a derivative term to a pure proportional term reduces the phase lag by

$$\angle\left(1 + j\omega T_D\right) = \tan^{-1}\frac{\omega T_D}{1} \in [0, \pi/2] \text{ for all } \omega, \tag{8}$$

which tends to increase the PM. In the meantime, however, the gain increases by a factor of

$$\left|1 + j\omega T_D\right| = \sqrt{1 + \omega^2 T_D^2} > 1, \text{ for all } \omega, \tag{9}$$

and hence the overall stability may be improved or degraded.

To demonstrate that adding a differentiator can destabilize some systems, consider the typical first-order delayed plant

$$G(s) = \frac{K}{1 + Ts}e^{-Ls}, \tag{10}$$

where K is the process gain, T is the time constant, and L is the dead time or transport delay. Suppose that this plant is controlled by a proportional controller with gain K_P and that a derivative term is added. The resulting PD controller

$$G_{PD}(s) = K_P(1 + T_D s) \tag{11}$$

leads to an open-loop feedforward path transfer function with frequency response

$$G(j\omega)G_{PD}(j\omega) = KK_P\frac{1 + jT_D\omega}{1 + jT\omega}e^{-jL\omega}. \tag{12}$$

For all ω, the gain satisfies

$$KK_P\sqrt{\frac{1 + T_D^2\omega^2}{1 + T^2\omega^2}} \geqslant KK_P \min\left(1, \frac{T_D}{T}\right), \tag{13}$$

where inequality (13) holds since $((1 + T_D^2\omega^2)/(1 + T^2\omega^2))^{1/2}$ is monotonic in ω.

Hence, if $K_P > 1/K$ and $T_D > T/KK_P$, then, for all ω,

$$\left|G(j\omega)G_{PD}(j\omega)\right| > 1. \tag{14}$$

Inequality (14) implies that the 0-dB gain crossover frequency is at infinity. Furthermore, due to the transport delay, the phase is

$$\angle G(j\omega)G_{PD}(j\omega) = \tan^{-1}\frac{\omega T_D}{1} - \tan^{-1}\frac{T\omega}{1} - L\omega.$$

Therefore, when ω approaches infinity,

$$\angle G(j\omega)G_{PD}(j\omega) < -180^\circ. \tag{15}$$

Hence, if $T_D > T/KK_P$ and $K_P > 1/K$, then by the Nyquist criterion, the closed-loop system is unstable. This analysis also confirms that some PID mapping formulas, such as the Ziegler-Nichol (Z-N) formula obtained from the step-response method, in which $K_P = (1.2(T/L))\,(1/K)$ and T_D is proportional to L, are valid for only a limited range of values of the T/L ratio.

As an example, consider plant (10) with $K = 10$, $T = 1$ s, and $L = 0.1$ s [7]. Control by means of a PI controller with $K_P = 0.644 > 1/K$ and $T_I = 1.03$ s yields reasonable stability margins and time-domain performance, as seen in Figures 2 and 3 (Set 1, red curves). However, when a differentiator is added, gradually

increasing T_D from zero improves both GM and PM. The GM peaks when T_D approaches 0.03 s; this value of T_D maximizes the speed of the transient response without oscillation. However, if T_D is increased further to 0.1 s, the GM deteriorates and the transient exhibits oscillation. In fact, the closed-loop system can be destabilized if T_D increases to 0.2 with $T/KK_P = 0.155$. Hence, care needs to be taken to tune and use the derivative term properly when the plant is subject to delay.

This destabilizing phenomenon can contribute to difficulties in designing PID controllers. These difficulties help explain why 80% of PID controllers in use have the derivative part switched off or omitted completely [5]. Thus, the functionality and potential of a PID controller is not fully exploited, while proper use of a derivative term can increase stability and help maximize the integral gain for better performance [11].

Remedies for Derivative Action

Differentiation increases the high-frequency gain, as shown in (9) and demonstrated by the four sets of frequency responses in Figure 2. A pure differentiator is not proper or causal. When a step change of the setpoint or disturbance occurs, differentiation results in a theoretically infinite control signal. To prevent this impulse control signal, most PID software packages and hardware modules add a filter to the differentiator. Filtering is particularly useful in a noisy environment.

Linear Lowpass Filter

The filtering remedy most commonly adopted is to cascade the differentiator with a first-order, lowpass filter, a technique often used in preprocessing for data acquisition. Hence, the derivative term becomes

$$\tilde{G}_D(s) = K_P \frac{T_D s}{1 + \frac{T_D}{\beta} s}, \tag{16}$$

where β is a constant factor. Most industrial PID hardware provides a value ranging from 1 to 33, with the majority falling between 8 and 16 [12]. A second-order Butterworth filter is recommended in [13] if further attenuation of high-frequency gains is required. Sometimes, the lowpass filter is cascaded to the entire G_{PID} in internal-model-control (IMC)-based design, which therefore leads to more sluggish transients.

Velocity Feedback

Because a lowpass filter does not completely remove, but rather averages, impulse derivative signals caused by sudden changes of the setpoint or disturbance, modifications of the unity negative feedback PID structure are of interest [8]. To block the effect of sudden changes of the setpoint, we consider a variant of the standard feedback. This variant uses

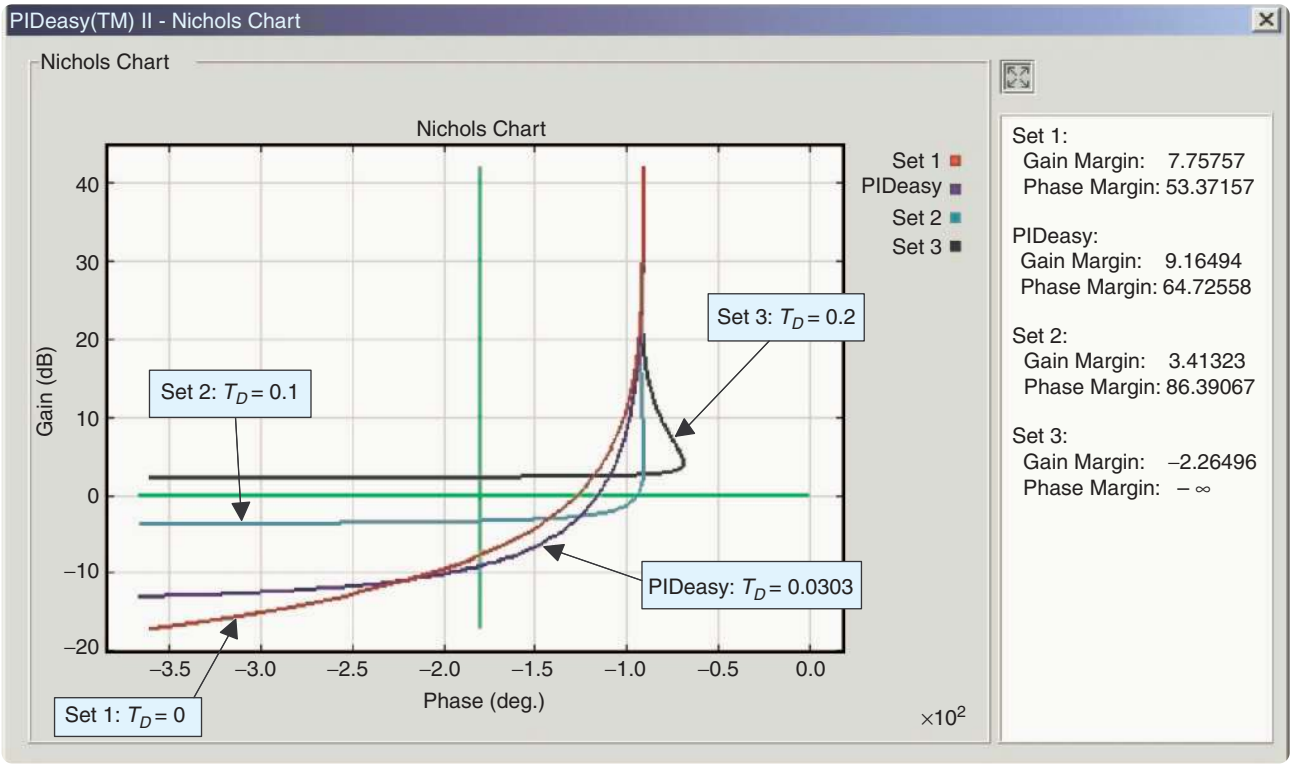

FIGURE 2 Destabilizing effect of the derivative term, measured in the frequency domain by GM and PM. Adding a derivative term increases both the GM and PM, although raising the derivative gain further tends to reverse the GM and destabilize the closed-loop system. For example, if the derivative gain is increased to 20% of the proportional gain ($T_D = 0.2$ s), the overall open-loop gain becomes greater than 2.2 dB for all ω. At $\omega = 30$ rad/s, the phase decreases to $-\pi$ while the gain remains above 2.2 dB. Hence, by the Nyquist criterion, the closed-loop system is unstable. It is interesting to note that MATLAB does not compute the frequency response as shown here, since MATLAB handles the transport delay factor $e^{-j\omega L}$ in state space through a Padé approximation.

the process variable instead of the error signal for the derivative action [14], as in

$$u(t) = K_P\, e(t) + K_I \int_0^t e(\tau)\, d\tau - K_D \frac{d}{dt} y(t), \qquad (17)$$

where $y(t)$ is the process variable, $e(t) = r(t) - y(t)$ is the error signal, and $r(t)$ is the setpoint or reference signal. The last term of (17) forms velocity feedback and, hence, an extra loop that is not directly affected by a sudden change in the setpoint. However, sudden changes in disturbance or noise at the plant output can cause the differentiator to produce a theoretically infinite control signal.

Setpoint Filter

To further reduce sensitivity to setpoint changes and avoid overshoot, a setpoint filter may be adopted. To calculate the proportional action, the setpoint signal is weighted by a factor $b < 1$, as in [8] and [14]

$$u(t) = K_P \left(b\, r(t) - y(t)\right) + K_I \int_0^t e(\tau)\, d\tau - K_D \frac{d}{dt} y(t). \qquad (18)$$

This modification results in a bumpless control signal and improved transients if the value of b is carefully chosen [2]. However, modification (18) is difficult to analyze quantitatively using standard techniques of stability and robustness analysis.

Structure (17) is referred to as Type B PID (or PI-D) control, structure (18) is known as Type C PID (or I-PD) control, and structures (1)–(3) constitute Type A PID control. Types B and C introduce more structures, and the need for preselection of, or switching between, suitable structures can pose a design challenge. To meet this need, PID hardware vendors have developed artificial intelligence techniques to suppress overshoots [15], [16].

Nevertheless, the ideal, parallel, series, and modified PID controller structures can be found in many software packages and hardware modules. Techmation's *Applications Manual* [12] documents the structures employed in many industrial PID controllers. Since vendors often recommend their own controller structures, tuning rules for a specific structure do not necessarily perform well with other structures. Readers may refer to [17] and [18] for detailed discussions on the use of various PID structures.

Prefilter

For setpoint tracking applications, an alternative to using a Type B or C structure is to cascade the setpoint with a prefilter that has critically damped dynamics. When a step change in the setpoint occurs, continuous output of the prefilter helps achieve soft start and bumpless control [8], [19]. However, a prefilter does not solve the problem caused by sudden changes in the disturbance since it is not embedded in the feedback loop.

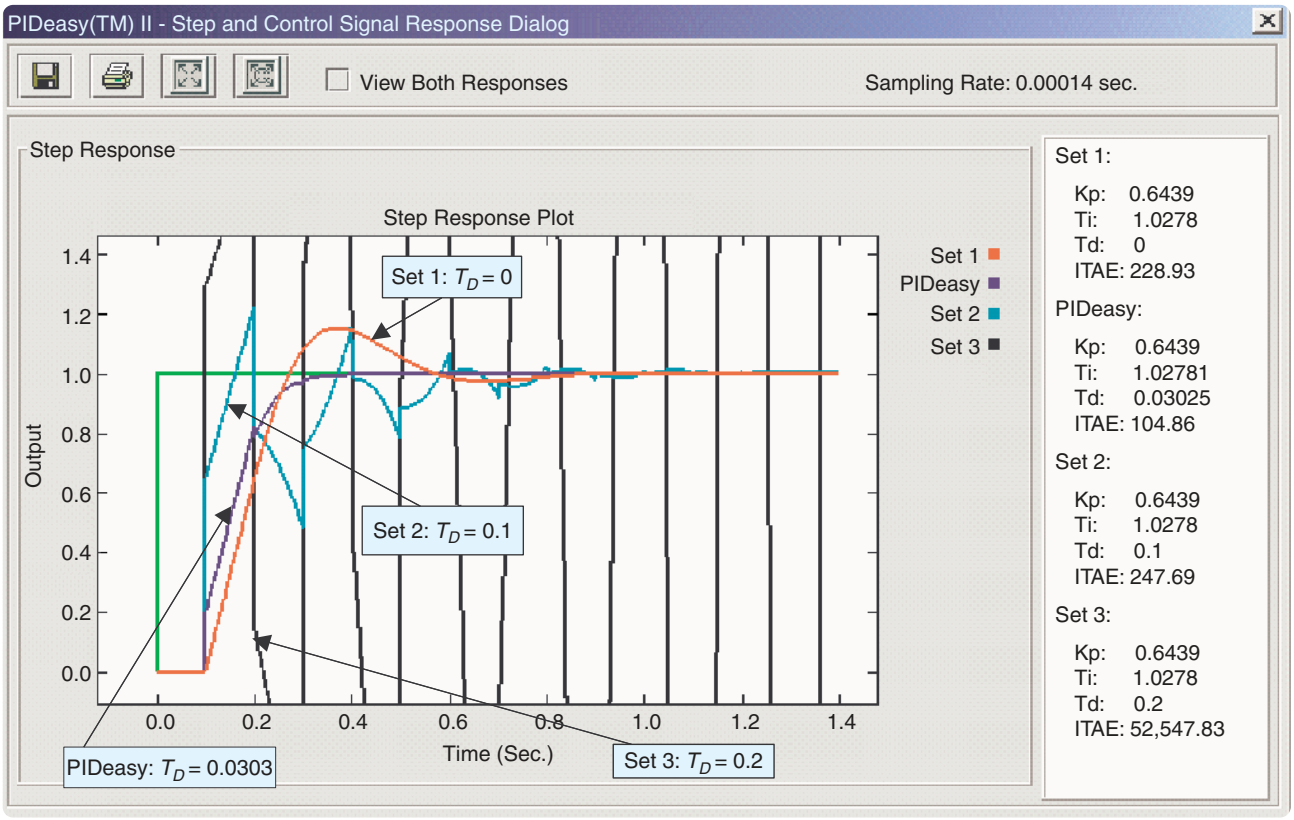

FIGURE 3 Destabilizing effect of the derivative term, confirmed in the time domain by the closed-loop step response. Although increasing the derivative gain initially decreases the oscillation, this trend soon reverses and the oscillation grows into instability.

Nonlinear Median Filter

Another method for smoothing the derivative action is to use a median filter [7], which is nonlinear and widely applied in image processing. Such a filter compares several data points around the current point and selects their median for the control action. Consequently, unusual or unwanted spikes resulting from a step command, noise, or disturbance are removed completely. Median filters are easily realized, as illustrated in Figure 4, since almost all PID controllers are now implemented in a digital processor. Another benefit of this method is that extra parameters are not needed to devise the filter. A median filter outperforms a prefilter as the median filter is embedded in the feedback loop and, hence, can deal with sudden changes in both the setpoint and the disturbance; a median filter may, however, overly smooth underdamped processes.

Design Objectives and Methods

Design Objectives and Existing Methods

Excellent summaries of PID design and tuning methods can be found in [4], [8], [20], and [21]. While matters concerning commissioning and maintenance (such as pre- and postprocessing as well as fault tolerance) also need to be considered in a complete PID design, controller parameters are usually tuned so that the closed-loop system meets the following five objectives:

1) stability and stability robustness, usually measured in the frequency domain
2) transient response, including rise time, overshoot, and settling time
3) steady-state accuracy
4) disturbance attenuation and robustness against environmental uncertainty, often at steady state
5) robustness against plant modeling uncertainty, usually measured in the frequency domain.

Most methods target one objective or a weighted composite of the objectives listed above. With a given objective, design methods can be grouped according to their underlying nature listed below [7], [8].

Heuristic Methods

Heuristic methods evolve from empirical tuning (such as the Z-N tuning rule), often with a tradeoff among design objectives. Heuristic search now involves expert systems, fuzzy logic, neural networks, and evolutionary computation [19], [22].

Frequency Response Methods

Frequency-domain constraints, such as GM, PM, and sensitivities, are used to synthesize PID controllers offline [2], [3]. For real-time applications, frequency-domain measurements require time-frequency, localization-based methods such as wavelets.

Analytical Methods

Because of the simplicity of PID control, parameters can be derived analytically using algebraic relations between a plant model and a targeted closed-loop transfer function with an indirect performance objective, such as pole placement, IMC, or lambda tuning. To derive a rational, closed-loop transfer function, this method requires that transport delays be replaced by Padé approximations.

```
derivative = (error -previous_error) / sampling_period;
if (derivative > max_d)
        new_derivative = max_d;              // median found
else if (derivative < min_d)
        new_derivative = min_d;              // median found
else
        new_derivative=derivative;           // median found

if (derivative > previous_derivative) {      // for next cycle
        max_d=derivative;
        min_d=previous_derivative;
} else {
        max_d=previous_derivative;
        min_d=derivative;
}
previous_derivative = derivative;            // for next cycle
```

FIGURE 4 Pseudocode for a three-point median filter to illustrate the mechanism of complete removal of impulse spikes. Median filters are widely adopted in image processing but not yet in control system design. This nonlinear filter completely removes extraordinary derivative values resulting from sudden changes in the error signal, unlike a lowpass filter, which averages past values.

TABLE 2 ABB's Easy-Tune PID formulas mapping the three parameters K, T, and L of the first-order delayed plant (10) to coefficients of P, PI, PID, and PD controllers, respectively [23]. The formulas are obtained by minimizing the integral of time-weighted error index, except the PD formula for which empirical estimates are used. Usually expressed in percentage, $PB = (U_{\max} - U_{\min})/K_P$ is the proportional band, where $U_{\max}$ and $U_{\min}$ are the upper and lower saturation levels of the control signal, respectively, and $|U_{\max} - U_{\min}|$ is usually normalized to one.

Mode	Action	Value
P	PB	$2.04K\left(\frac{L}{T}\right)^{1.084}$
PI	PB	$1.164K\left(\frac{L}{T}\right)^{0.977}$
	T_I	$\frac{T}{40.44}\left(\frac{L}{T}\right)^{0.68}$
	PB	$0.7369K\left(\frac{L}{T}\right)^{0.947}$
PID	T_I	$\frac{T}{51.02}\left(\frac{L}{T}\right)^{0.738}$
	T_D	$\frac{T}{157.5}\left(\frac{L}{T}\right)^{0.995}$
PD	PB	$0.5438K\left(\frac{L}{T}\right)^{0.947}$
	T_D	$\frac{T}{157.5}\left(\frac{L}{T}\right)^{0.995}$

Numerical Optimization Methods

Optimization-based methods can be regarded as a special type of optimal control. PID parameters are obtained by numerical optimization for a weighted objective in the time domain. Alternatively, a self-learning evolutionary algorithm (EA) can be used to search for both the parameters and their associated structure or to meet multiple design objectives in both the time and frequency domains [19], [22].

Some design methods can be computerized, so that designs are automatically performed online once the plant is identified; hence, these designs are suitable for adaptive tuning. While PID design has progressed from analysis-based methods to numerical optimization-based methods, there are few techniques that are as widely applicable as Z-N tuning [2], [3]. The most widely adopted initial tuning methods are based on the Z-N empirical formulas and their extensions, such as those shown in Table 2 [23]. These formulas offer a direct mapping from plant parameters to controller coefficients.

Over the past half century, researchers have sought the next key technology for PID tuning and modular realization [6]. With simulation packages widely available and heavily adopted, computerizing simulation-based designs is gaining momentum, enabling simulations to be carried out automatically so as to search for the best possible PID settings for the application at hand [22]. By using a computerized approach, multiple design methods can be combined within a single software or firmware package to support various plant types and PID structures.

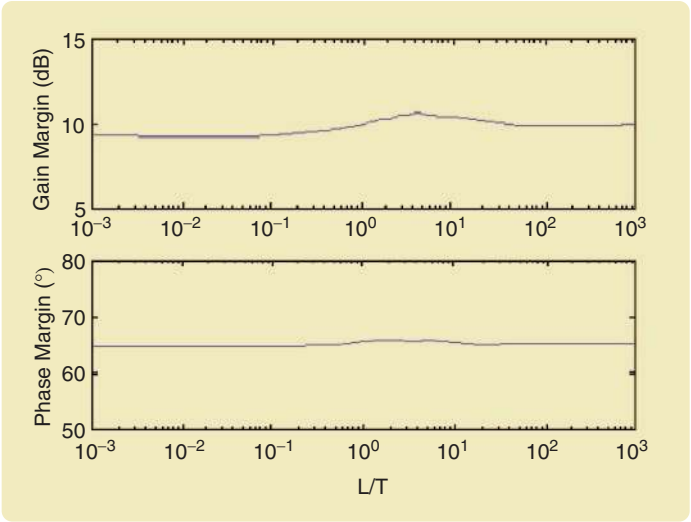

FIGURE 5 Gain and phase margins resulting from PIDeasy designs for first-order delayed plants with various L/T ratios. While requirements of fast transient response, no overshoot, and zero steady-state error are accommodated by time-domain criteria, multiobjective design goals provide frequency-domain margins in the range of 9–11 dB and 65–66°.

A Computerized Simulation Approach

PIDeasy [7] is a software package that uses automatic simulations to search globally for controllers that meet all five design objectives in both the time and frequency domains. The search is initially performed offline in a batch mode [19] using artificial evolution techniques that evolve both

TABLE 3 **Multioptimal PID settings for normalized typical high-order plants. Since PIDeasy's search priorities are time-domain tracking and regulation, the corresponding gain and phase margins are given to assess frequency-domain properties.**

Plants		PID Coefficients K_p	T_i (s)	T_d (s)	Resultant Margins GM (dB)	PM (°)
$G_1(s) = \frac{1}{(s+1)^\alpha}$	$\alpha = 1$	92.1	1.0	0.0022	∞	102
	$\alpha = 2$	1.95	1.61	0.14	∞	62.4
	$\alpha = 3$	1.12	2.13	0.28	26.8	60.7
	$\alpha = 4$	0.83	2.61	0.43	13.9	61
	$\alpha = 8$	0.50	4.31	1.01	9.05	58.9
$G_2(s) = \frac{1}{(s+1)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)}$	$\alpha = 0.1$	5.53	1.03	0.04	52.8	68.7
	$\alpha = 0.2$	2.87	1.08	0.07	38.6	66.3
	$\alpha = 0.5$	1.19	1.36	0.17	19.1	62.6
$G_3(s) = \frac{1-\alpha s}{(s+1)^3}$	$\alpha = 0.1$	1.03	2.15	0.31	19.4	61.2
	$\alpha = 0.2$	0.96	2.18	0.33	16.6	61.6
	$\alpha = 0.5$	0.79	2.23	0.39	13	62.4
	$\alpha = 1.0$	0.63	2.30	0.47	7.52	50.9
	$\alpha = 2.0$	0.48	2.39	0.57	7.45	58.6
	$\alpha = 5.0$	0.36	2.58	0.72	2.69	40.4
$G_4(s) = \frac{1}{(1+s\alpha)^2} e^{-s}$	$\alpha = 0.1$	0.23	0.43	0.12	10.4	66
	$\alpha = 0.2$	0.30	0.59	0.17	10.4	65.8
	$\alpha = 0.5$	0.49	1.07	0.26	10.5	65.6
	$\alpha = 2.0$	1.04	3.49	0.49	15	62.4
	$\alpha = 5.0$	1.42	8.32	0.92	24.2	62.1
	$\alpha = 10$	1.65	16.35	1.59	32.8	62.1

controller parameters and their associated structures. For practical simplicity and reliability, the standard PID structure is maintained as much as possible, while allowing augmentation with either lowpass or median filtering for the differentiator and with explicit antiwindup for the integrator. The resulting designs are then embedded in the PIDeasy package. Further specific tuning can be continued by local, fast numerical optimization if the plant differs from its model or data used in the initial design.

First-Order Delayed Plants

An example of PIDeasy for a first-order delayed plant is shown in Figures 2 and 3. To assess the robustness of design using PIDeasy, GMs and PMs resulting from designs for plants with various L/T ratios are shown in Figure 5 [19]. While requirements of fast transient response, no overshoot, and zero steady-state error are accommodated by time-domain criteria, PIDeasy's multiobjective goals provide frequency-domain margins in the range of 9–11 dB and 65–66°.

Higher Order Plants

For higher order plants, we obtain multioptimal designs for the 20 benchmark plants [24]

$$G_1(s) = \frac{1}{(s+1)^\alpha}, \quad \alpha = 1,\ 2,\ 3,\ 4,\ 8, \tag{19}$$

$$G_2(s) = \frac{1}{(s+1)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)}, \quad \alpha = 0.1,\ 0.2,\ 0.5, \tag{20}$$

$$G_3(s) = \frac{1-\alpha s}{(s+1)^3}, \qquad \alpha = 0.1,\ 0.2,\ 0.5,\ 1,\ 2,\ 5, \tag{21}$$

$$G_4(s) = \frac{1}{(1+s\alpha)^2} e^{-s}, \qquad \alpha = 0.1,\ 0.2,\ 0.5,\ 2,\ 5,\ 10. \tag{22}$$

The resulting designs and their corresponding gain and PMs are summarized in Table 3.

Setpoint-Scheduled PID Network

Consider the constant-temperature reaction process

$$\frac{dy(t)}{dt} = -Ky^2(t) + \frac{1}{V}\left[d - y(t)\right] u(t), \tag{23}$$

where

$y(t)$ = concentration in the outlet stream (mol/ℓ),
$u(t)$ = flow rate of the feed stream (ℓ/h),
K = rate of reaction (ℓ/mol-h),
V = reactor volume (ℓ),
d = concentration in the inlet stream (mol/ℓ).

The setpoint, equilibrium, or steady-state operating trajectory of the plant is governed by

$$Ky^2 + \frac{1}{V} u\, y - \frac{d}{V} u = 0. \tag{24}$$

For setpoints ranging from 0 to 1 mol/ℓ, an initial PID controller can be placed effectively at $y = 0.49$ by using the maximum distance from the nonlinear trajectory to the linear projection linking the starting and ending points of the operating envelope, as illustrated by node 2 in Figure 6 [22]. Similarly, two more controllers can be added at nodes or setpoints 1 and 3, forming a pseudolinear controller network comprised of three PIDs to be interweighted by scheduling functions $S_1(y)$, $S_2(y)$, and $S_3(y)$, examples of which are shown in Figure 7.

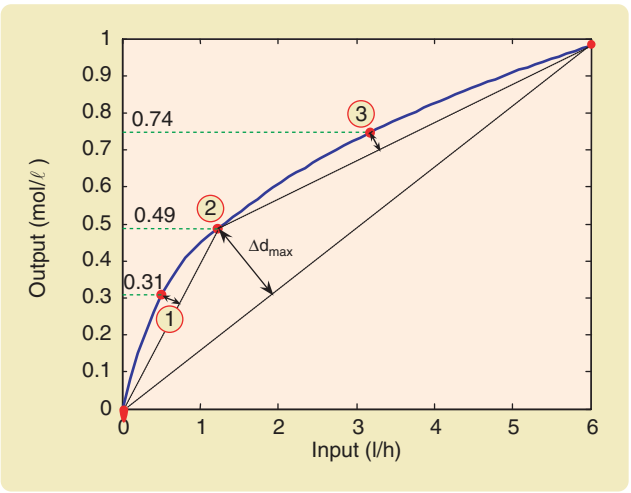

FIGURE 6 Operating trajectory (bold curve) of the nonlinear chemical process (23) for setpoints ranging from 0 to 1 mol/ℓ, as given by (24). A PID controller is first placed at $y = 0.49$ (node 2) by using the maximum distance from the nonlinear trajectory to the linear projection (thin dotted line) linking the starting and ending points of the operating envelope. Similarly, two more controllers can be added at nodes 1 and 3, forming a pseudo-linear controller network comprised of three PIDs. Without the need for linearization, these PID controllers can be obtained individually by PIDeasy or other PID software directly through step-response data, or obtained jointly by using an evolutionary algorithm [22].

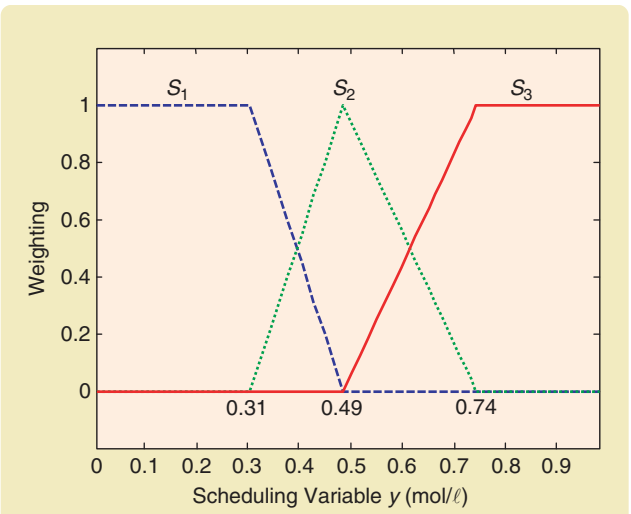

FIGURE 7 Fuzzy logic membership-like scheduling functions $S_1(y)$, $S_2(y)$, and $S_3(y)$ for individual PID controllers contributing to the PID network at nodes 1, 2, and 3, respectively. Due to nonlinearity, these functions are often asymmetric. Similar to gain scheduling, linear interpolation suffices for setpoint scheduling.

The PID controller centered at node 2 can be obtained by PIDeasy or other PID software directly through step-response data without the need for linearization at the current operating point [22]. The remaining PIDs can be obtained similarly. For simplicity, we obtain the controller centered at node x, where $x = 1$, 2, and 3, by using step-response data from the starting point to node x. The resulting PID network is given by

$$u(t) = [S_1 \quad S_2 \quad S_3] \begin{bmatrix} 9.82 & 1.22 & 0.0376 \\ 15.6 & 0.784 & 0.0241 \\ 28.6 & 0.481 & 0.0137 \end{bmatrix} \begin{bmatrix} 1 \\ p^{-1} \\ p \end{bmatrix} e(t), \tag{25}$$

where p denotes the derivative operator.

To validate tracking performance using a setpoint that is not originally used in the design process, the setpoint $r = 0.53$ mol/ℓ is used to test the control system. The response is shown in Figure 8, where a 10% disturbance occurs during [3, 3.5] h, confirming load disturbance rejection at steady state. Figure 9 shows the performance of the network at multiple operating levels not originally encountered in the design. If a more sophisticated PID network is desirable, the number of nodes, controller parameters for each node, and scheduling functions can be optimized globally in a single design process by using an EA [22].

It is known that gain scheduling provides advantages over continuous adaptation in most situations [8]. The setpoint-scheduled network utilizes these advantages of gain scheduling. Furthermore, by bumpless scheduling, the network does not require discontinuous switching between various controller structures.

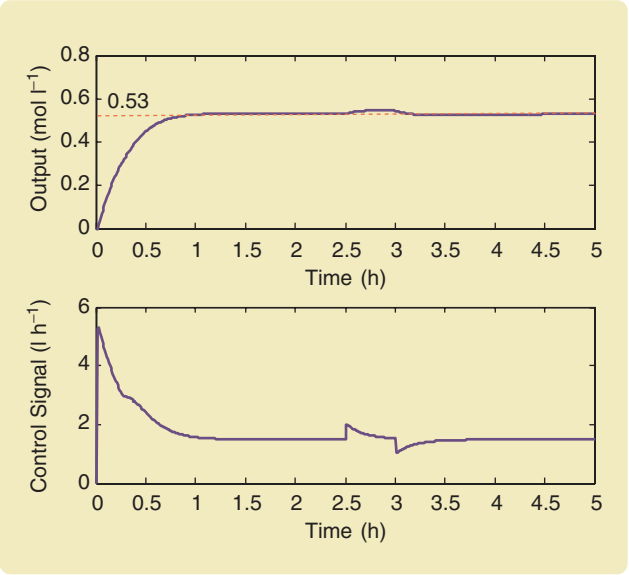

FIGURE 8 Performance of the pseudolinear PID network applied to the nonlinear process example (23). To validate tracking performance using a setpoint that is not originally used in the design process, the setpoint $r = 0.53$ mol/ℓ is used to test the control system. The controller network tracks this setpoint change accurately without oscillation and rejects a 10% load disturbance occurring during [3, 3.5] h.

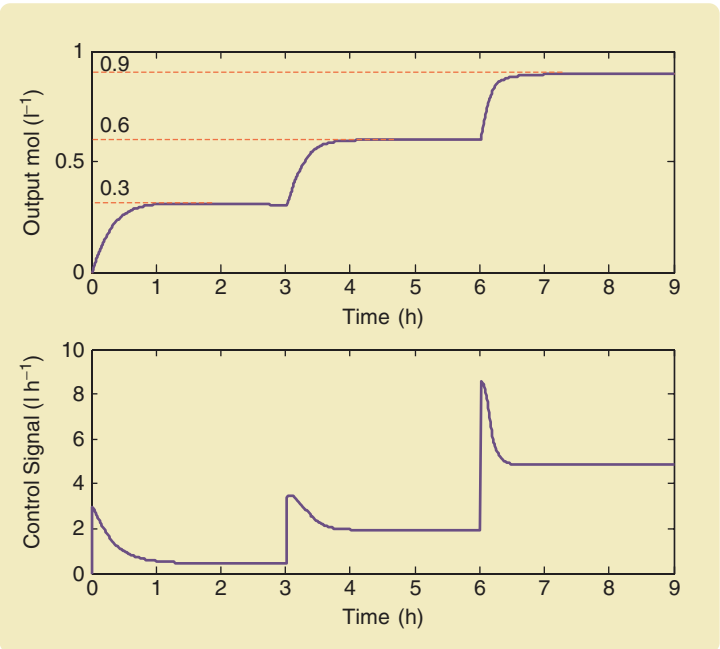

FIGURE 9 Performance of the pseudolinear PID network applied to the nonlinear chemical process (23) at multiple operating levels that are not originally used in the design process. The network tracks these setpoint changes accurately without oscillation. It can be seen that the control effort increases disproportionally to the setpoint change along the nonlinear trajectory, compensating for the decreasing gain of the plant when the operating level is raised.

Discussion and Conclusions

PID is a generally applicable control technique that derives its success from simple and easy-to-understand operation. However, because of limited information exchange and problem analysis, there remain misunderstandings between academia and industry concerning PID control. For example, the message that increasing the derivative gain leads to improved transient response and stability is often wrongly expounded. These misconceptions may explain why the argument exists that academically proposed PID tuning rules sometimes do not work well on industrial controllers. In practice, therefore, switching between different structures and functional modes is used to optimize transient response and meet multiple objectives.

Difficulties in setting optimal derivative action can be eased by complete understanding and careful tuning of the D term. Median filtering, which is widely adopted for preprocessing in image processing but yet to be adopted in controller design, is a convenient tool for solving problems that the PI-D and I-PD structures are intended to address. A median filter outperforms a lowpass filter in removing impulse spikes of derivative action resulting from a sudden change of setpoint or disturbance. Embedded

in the feedback loop, a median filter also outperforms a prefilter in dealing with disturbances.

Over the past half century, researchers have sought the next key technology for PID tuning and modular realization. Many design methods can be computerized and, with simulation packages widely used, the trend of computerizing simulation-based designs is gaining momentum. Computerizing enables simulations to be carried out automatically, which facilitates the search for the best possible PID settings for the application at hand. A simulation-based approach requires no artificial minimization of the control amplitude and helps improve sluggish transient response without windup.

In tackling PID problems, it is desirable to use standard PID structures for a reasonable range of plant types and operations. Modularization around standard PID structures should also help improve the cost effectiveness of PID control and maintenance. This way, robustly optimal design methods such as PIDeasy can be developed. By including system identification techniques, the entire PID design and tuning process can be automated, and modular code blocks can be made available for timely application and real-time adaptation.

ACKNOWLEDGMENTS

This article is based on [25]. Kiam Heong Ang and Gregory Chong are grateful to the University of Glasgow for a postgraduate research scholarship and to Universities UK for an Overseas Research Students Award. The authors thank Prof. Hiroshi Kashiwagi of Kumamoto University and Mitsubishi Chemical Corp., Japan, for the nonlinear reaction process model and data.

AUTHOR INFORMATION

Yun Li (Y.Li@elec.gla.ac.uk) is a senior lecturer at the University of Glasgow, United Kingdom, where he has taught and conducted research in evolutionary computation and control engineering since 1991. He worked in the U.K. National Engineering Laboratory and Industrial Systems and Control Ltd, Glasgow, in 1989 and 1990. In 1998, he established the IEEE CACSD Evolutionary Computation Working Group and the European Network of Excellence in Evolutionary Computing (EvoNet) Workgroup on Systems, Control, and Drives. He was a visiting professor at Kumamoto University, Japan. He is currently a visiting professor at the University of Electronic Science and Technology of China. His research interests are in parallel processing, design automation, and discovery of engineering systems using evolutionary learning and intelligent search techniques. He has advised 12 Ph.D. students and has 140 publications. He can be contacted at the Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow G12 8LT, U.K.

Kiam Heong Ang received a First-Class Honors B.Eng. and a Ph.D. degree in electronics and electrical engineering from the University of Glasgow, United Kingdom, in 1996 and 2005, respectively. From 1997–2000, he was a software engineer with Advanced Process Control Group, Yokogawa Engineering Asia Pte. Ltd., Singapore. Since 2005, within the same company, he has been working on process industry standardization and new technology development. His research interests include evolutionary, multiobjective learning, computational intelligence, control systems, and engineering design optimization.

Gregory Chong received a First-Class Honors B.Eng. degree in electronics and electrical engineering from the University of Glasgow, United Kingdom, in 1999. He is completing his Ph.D. at the same university in evolutionary, multiobjective modeling, and control for nonlinear systems.

REFERENCES

[1] J.G. Ziegler and N.B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, vol. 64, no. 8, pp. 759–768, 1942.
[2] W.S. Levine, Ed., *The Control Handbook*. Piscataway, NJ: CRC Press/IEEE Press, 1996.
[3] L. Wang, T.J.D. Barnes, and W.R. Cluett, "New frequency-domain design method for PID controllers," *Proc. Inst. Elec. Eng.*, pt. D, vol. 142, no. 4, pp. 265–271, 1995.
[4] J. Quevedo and T. Escobet, Eds., "Digital control: Past, present and future of PID control," in *Proc. IFAC Workshop*, Terrassa, Spain, Apr. 5, 2000.
[5] I.E.E. Digest, "Getting the best out of PID in machine control," in *Digest IEE PG16 Colloquium (96/287)*, London, UK, Oct. 24, 1996.
[6] P. Marsh, "Turn on, tune in—Where can the PID controller go next," *New Electron.*, vol. 31, no. 4, pp. 31–32, 1998.
[7] Y. Li, W. Feng, K.C. Tan, X.K. Zhu, X. Guan, and K.H. Ang, "PIDeasy and automated generation of optimal PID controllers," in *Proc. 3rd Asia-Pacific Conf. Control and Measurement*, Dunhuang, P.R. China, 1998, pp. 29–33.
[8] K.J. Åström and T. Hägglund, *PID Controllers: Theory, Design, and Tuning*. Research Triangle Park, NC: Instrum. Soc. Amer., 1995.
[9] F.G. Shinskey, *Feedback Controllers for the Process Industries*. New York: McGraw-Hill, 1994.
[10] C. Bohn and D.P. Atherton, "An analysis package comparing PID antiwindup strategies," *IEEE Contr. Syst. Mag.*, vol. 15, no. 2, pp. 34–40, Apr. 1995.
[11] K.J. Åström and T. Hägglund, "The future of PID control," *Contr. Eng. Pract.*, vol. 9, no. 11, pp. 1163–1175, 2001.
[12] Techmation Inc., *Techmation* [Online], May 2004. Available: http://protuner.com
[13] J.P. Gerry and F.G. Shinskey, "PID controller specification," white paper [Online]. May 2004. Available: http://www.expertune.com/PIDspec.htm
[14] BESTune, PID controller tuning [Online]. May 2004. Available: http://bestune.50megs.com
[15] Honeywell International Inc. [Online]. May 2004. Available: http://www.Acs.Honeywell.Com/Ichome/Rooms/DisplayPages/LayoutInitial
[16] Y. Li, K.H. Ang, and G. Chong, "Patents, software, and hardware for PID control," *IEEE Contr. Syst. Mag.*, vol. 26, no. 1, pp. 42–54, 2006.
[17] J.P. Gerry, "A comparison of PID control algorithms," *Contr. Eng.*, vol. 34, no. 3, pp. 102–105, Mar. 1987.
[18] A. Kaya and T.J. Scheib, "Tuning of PID controls of different structures," *Contr. Eng.*, vol. 35, no. 7, pp. 62–65, July 1988.
[19] W. Feng and Y. Li, "Performance indices in evolutionary CACSD automation with application to batch PID generation," in *Proc. 10th IEEE Int. Symp. Computer Aided Control System*, Hawaii, Aug. 1999, pp. 486–491.
[20] R. Gorez, "A survey of PID auto-tuning methods," *Journal A*, vol. 38, no. 1, pp. 3–10, 1997.
[21] A. O'Dwyer, *Handbook of PI and PID Controller Tuning Rules*. London: Imperial College Press, 2003.
[22] Y. Li, K.H. Ang, G. Chong, W. Feng, K.C. Tan, and H. Kashiwagi, "CAutoCSD—Evolutionary search and optimisation enabled computer-automated control system design," *Int. J. Automat. Comput.*, vol. 1, no. 1, pp. 76–88, 2004.
[23] *Specification Data File of Commander 355*, ABB, SS/C355, Issue 3, 2001.
[24] K.J. Åström and T. Hägglund, "Benchmark systems for PID control," in *Proc. IFAC Workshop*, Terrassa, Spain, 2000, pp. 165–166.
[25] K.H. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Trans. Contr. Syst. Tech.*, vol. 13, no. 4, pp. 559–576, 2005.