RELAY CONTROL

The block diagram of a simple feedback control loop with a reference input or setpoint y_{sp}, process output y and control signal u is shown in Fig. 1. The main function of the controller is to automatically compute or generate the correct control signal to null the control error e rapidly and with good damping either when the setpoint is changed or when an unknown load disturbance l is introduced.

The simplest form of the controller is the relay or on–off control. An ideal relay has the characteristics shown in Fig.

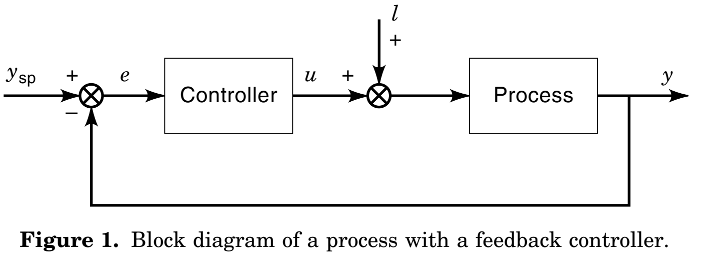

Figure 1. Block diagram of a process with a feedback controller.

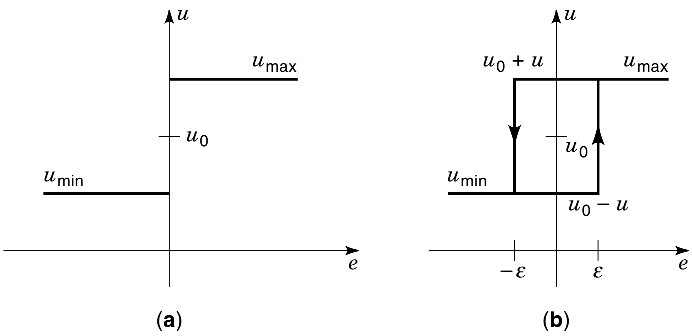

Figure 2. Simple control characteristics. (a) Ideal relay. (b) Relay with hysteresis.

2(a). Its input is the control error e, and its output is u_{max} when $e > 0$ and u_{min} when $e < 0$. The most important advantages of relay control are (1) its wide applicability to most industrial processes and dynamic systems, and (2) the simplicity of its design. The relay constants u_{max}, u_{min}, and u_0 in Fig. 2 can be simply set by steady-state analysis or by trial and error. It can be made robust to the influence of noise by incorporating a small hysteresis as shown in Fig. 2(b). Owing to their simplicity and low cost, relay controllers are found in many of our domestic appliances such as irons, refrigerators, and air-conditioners, as well as in some industrial processes that do not need precise control. The greatest disadvantage of relay control is that it gives rise to steady-state oscillations or limit cycles which are not acceptable in industrial applications which require precise control.

Another simple and widely used control is the PID controller. It has a proportional or P term (which is proportional to the control error), an integral or I term (which is proportional to the integral of the error), and a derivative or D term (which is proportional to the derivative of the error). The proportional term with a gain of k_c varies the output of the controller to speed up the transient response when there is a change in setpoint or load disturbance. It will, however, produce a steady-state offset or error which can be automatically eliminated only if integral action (I term) is added. The relative contribution of the I term is fixed by the integral time T_i. A shorter integral time will speed up the time taken to null the offset; however, it can destabilize the system or make the transient response much more oscillatory. This can be compensated by adding the D term, which has a predictive or damping action that is proportional to the derivative time T_d. The properly combined actions of the P, I, and D terms as given by Eq. (1) can produce fast, well-damped, and accurate control performance:

$$u_c = k_c\left(e + \frac{1}{T_i}\int e\,dt + T_d\,\frac{de}{dt}\right) \tag{1}$$

The operational properties of the PID controller are well understood by both experienced plant operators and instrument engineers who can select (or tune) the controller parameters (k_c, T_i, T_d) by trial and error or by a systematic procedure such as the well-known Ziegler–Nichols method. It is the simplicity, wide applicability, and familiarity to plant personnel that have made PID controllers the most widely used feedback controllers in industry for a long time and will remain dominant in the foreseeable future.

PID controllers have been routinely used when precise closed-loop control is demanded in practice. A large industrial plant may have hundreds of PID controllers. The controllers will perform extremely well if the three PID controller parameters (k_c, T_i, T_d) are properly selected or tuned to match the process dynamics (e.g., process gain, dead time, major time constant, etc). Nevertheless, in spite of its popularity and long history, it was common experience that many PID controllers were in practice poorly tuned prior to the advent of simple and yet reliable methods of automatic tuning (or autotuning in short) in the late 1980s and 1990s. The main reason was that any manual method of tuning was tedious and time-consuming. In a typical process plant where the major time constant is of the order of tens of minutes, it would take several hours or longer to tune just one loop. If a major process disturbance occurs during this process of tuning, the tuning procedure would have to be stopped and the whole procedure repeated. Faced with tens or hundreds of PID controllers in a large plant, it would not be very practical to manually tune all the controllers. On the other hand, for a smaller plant where there are only a few PID controllers, it would be feasible to manually tune all the controllers during the start-up phase where expert instrument engineers from the vendor were present. During normal operation, however, a need for retuning may arise when the process dynamics drift significantly owing to changes in process operating point, wear and tear of control valves, and other influencing variables such as throughput and disturbances. Experienced personnel who are competent to do manual tuning are rarely available in the small plants, and hence a practical solution prior to the advent of autotuning was to tune the controllers conservatively to achieve robustness at the expense of optimal performance.

The introduction of autotuning capabilities to PID controllers has enabled the control system commissioning time to be shortened and has facilitated control optimization through regular retuning. This success has led to the subsequent development work to extend autotuning methods to advanced controllers such as the cascade controller, the Smith predictor, and multivariable controllers. Some autotuning methods have also been extended to tune fuzzy controllers and gain-scheduling controllers.

There are many different methods of autotuning which have been successfully developed and commercialized. The simplest and yet reliable method of autotuning, which has received wide acceptance in practice since the late 1980s, is the relay feedback autotuning introduced by Astrom and Hagglund (1). This technique has several attractive features. First, it facilitates simple pushbutton tuning since it automatically extracts information needed for controller tuning without requiring a separate open-loop pretuning step which is essential in all other autotuning methods. Second, it is carried out under closed-loop control and the process can be kept close to the setpoint. This helps to keep the process in the linear region where the frequency response is of interest, which is precisely why the method works well on highly nonlinear processes when combined with gain-scheduling control. The process parameter estimation is also more accurate for the purpose of controller design as the relay transients and oscillations help to focus attention on frequencies near the process crossover frequency. Third, unlike other autotuning

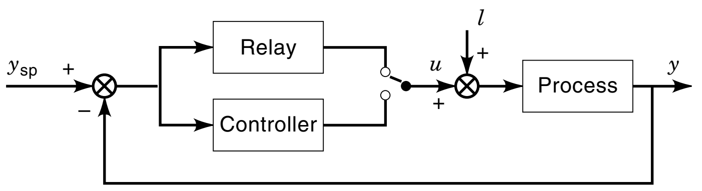

Figure 3. Block diagram of a relay feedback autotuning system.

methods, it does not require a careful choice of the sampling rate from the a priori knowledge of the process or from the pretuning step.

The block diagram of the relay feedback autotuning scheme (or relay autotuner in short) is shown in Fig. 3. There is a switch that selects either relay or PID control. When an operator demands autotuning, the controller is automatically disconnected and the relay is switched in as shown. Under relay control, the process output is maintained near the setpoint but will exhibit a limit cycle or sustained steady-state oscillation. The autotuner will adjust the relay magnitude such that the oscillation magnitude can be automatically regulated to a preset limit (e.g., 5% of the measurement span). Based on the steady-state or transient analysis of the relay oscillation response, information on one or more points of the process frequency response will be obtained which will in turn be used to compute the optimal values of the PID controller parameters. The relay is then switched out and the controller with the new PID parameters resumes its operation. The analysis and design of the relay autotuner will be presented later under the headings of "relay autotuning" and "PID controller design."

The basic idea of relay autotuning was motivated by the observation that the classic Ziegler–Nichols rule (2) for tuning PID controllers only made use of the knowledge of one point on the Nyquist curve of the process to be controlled (i.e., one point on the open-loop frequency response). This point is the intersection of the Nyquist curve with the negative real axis, which can be described in terms of the ultimate gain k_u and the ultimate period t_u. In the manual tuning procedure, the I and D terms are switched off and the proportional gain is gradually increased until steady oscillation is obtained; the proportional gain when this occurs is the ultimate gain, and the period of the oscillation is the ultimate period. The controller settings can then be computed according to Ziegler–Nichols formula of Eqs. (2), which aims to yield quarter amplitude damping:

$$\begin{aligned} k_c &= 0.6k_u \\ T_i &= 0.5t_u \\ T_d &= 0.125t_u \end{aligned} \tag{2}$$

It is difficult to automate the above manual procedure and perform it in such a way that the amplitude of oscillation is kept under control. It is also very time-consuming to complete the trial-and-error procedure which also demands undue attention of the plant operator. The relay control is an indirect but simple way to quickly generate sustained oscillation which can thus be used to automate the tuning procedure.

RELAY AUTOTUNING

A typical response when relay control is switched in is shown in Fig. 4. It is evident that we can obtain a first approximation of the ultimate gain as

$$k_u = \frac{d}{a}$$

where d and a are the amplitudes of the relay oscillation and the process output oscillation, respectively. By considering the first harmonic in the relay oscillation and assuming the process output to be near-sinusoidal, a more accurate estimate of the ultimate gain (1) can be obtained:

$$k_u = \frac{4d}{\pi a} \tag{3}$$

The ultimate period t_u, which is equal to the period of the sustained oscillation, can be easily measured from the times between zero-crossings. With the estimated k_u and t_u, the PID controller parameters can be computed using the Ziegler–Nichols formula of Eqs. (2).

In the presence of noise, a relay with hysteresis is used. The hysteresis width, ϵ, is selected on the basis of the noise level—for instance, two times larger than the noise amplitude. The approximate formula for computing the ultimate

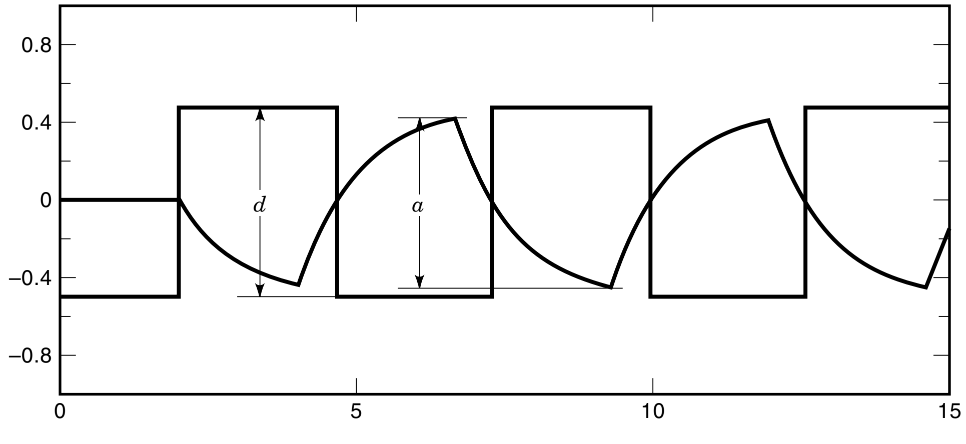

Figure 4. Relay oscillations.

gain should then be changed to

$$k'_{\rm u} = \frac{4d}{\pi\sqrt{a^2 - \epsilon^2}} \tag{4}$$

In addition, in order to obtain a reasonable signal-to-noise ratio, the relay magnitude d should be automatically adjusted so that the oscillation at the process output is acceptable, e.g., about three times the amplitude of the noise.

The formula of Eq. (4) is derived based on a more detailed harmonic analysis. The complex gain and phase of the relay can be represented by its describing function (1):

$$N(a) = \frac{4d}{\pi a}\left(\sqrt{1 - \left(\frac{\epsilon}{a}\right)^2} - j\frac{\epsilon}{a}\right) \tag{5}$$

The oscillation amplitude a and frequency ω should satisfy

$$G_{\rm p}(j\omega)N({\rm a}) = -1$$

Hence, we obtain

$$G_{\rm p}(j\omega) = -1/N(a) \tag{6}$$

where $G_{\rm p}(j\omega)$ is the frequency response or Nyquist curve of the open-loop process. Hence by changing the values of the relay amplitude and hysteresis, more points on the Nyquist curve can be identified using Eq. (6). A filter with known characteristics can also be introduced in cascade with the relay to identify other points on the Nyquist curve.

PID Controller Design

The estimates of ultimate gain and period could be used in the Ziegler–Nichols formula of Eq. (2) to compute the PID controller parameters, and this yields the simplest controller design. Other PID controller design methods may be preferred if the quarter amplitude damping performance criterion as specified by the Ziegler–Nichols design is found to provide insufficient damping or robustness against parameter variations. Even for maintaining the quarter amplitude damping performance, the range of applicability of the Ziegler–Nichols formula is known to be limited to a small class of process dynamics. For instance, when the process dead time is very small, the output response to a step change in setpoint will have high overshoot and is very oscillatory. When the process dead time is large compared with the major time constant, the closed-loop response becomes more sluggish and a significant undershoot is developed. Another simple controller design is the phase margin design based on the analysis of the gain and phase modification of the process Nyquist curve by the controller (1). If the desired phase margin, as determined on the basis of desired damping or robustness, is $\phi_{\rm m}$, the tuning formula is

$$\begin{aligned} k_{\rm c} &= k_{\rm u}\cos\phi_{\rm m} \\ T_{\rm i} &= 4T_{\rm d} \\ T_d &= \frac{1+\sin\phi_m}{4\pi\cos\phi_m}t_u \end{aligned} \tag{7}$$

This formula works well when the process dead time is small. It gives sluggish response when the dead time is large. In order to cater to a wide range of process dynamics, the following refined Ziegler–Nichols formula has been introduced (3):

$$\begin{aligned} k_{\rm c} &= 0.6k_{\rm u} \\ T_{\rm i} &= 0.5\mu t_{\rm u} \\ T_{\rm d} &= T_{\rm i}/4 \\ \beta &= \frac{15 - k_u k_p}{15 + k_u k_p}; \qquad \mu = 1 \quad \text{for } 0.16 < \theta < 0.57 \\ &\text{or} \\ \beta &= \frac{8}{17}\left(\frac{4}{9}k_u k_p + 1\right); \qquad \mu = \frac{4}{9}k_u k_p \quad \text{for } 0.57 < \theta < 0.96 \end{aligned} \tag{8}$$

where θ is the normalized dead time computed as the ratio of the process dead time (or apparent dead time) and major time constant; β is the setpoint weighting factor which is used to reduce the overshoot of the setpoint response without affecting the load disturbance response; the integral time is reduced by the factor μ to prevent a large undershoot when the θ is large. The practical form of a PID controller which incorporates setpoint weighting and also performs the derivative action on the filtered output only is

$$\begin{aligned} u_{\rm c} &= k_{\rm c}\left[(\beta y_{\rm sp} - y) + \frac{1}{T_{\rm i}}\int e\,dt - T_{\rm d}\frac{dy_{\rm f}}{dt}\right] \\ \frac{dy_{\rm f}}{dt} &= \frac{N_{\rm f}}{T_{\rm d}}(y - y_{\rm f}) \end{aligned} \tag{9}$$

where the filter time constant is $T_{\rm d}/N_{\rm f}$; $N_{\rm f}$ is chosen in the range of 3 to 10 depending on noise level with a default value set at 10.

The computation of θ requires further knowledge of the process model, the simplest of which is a first-order plus dead-time model:

$$G_{\rm p}(s) = k_{\rm p}\frac{e^{-SL_1}}{1+sT_1} \tag{10}$$

Most industrial processes with open-loop dynamics which are well-damped can be adequately represented by this model. It is straightforward to relate these model parameters to the ultimate gain and period obtained from relay control (3). We thus have the following equations:

$$T_1 = \frac{t_u}{2\pi}\sqrt{(k_u k_p)^2 - 1} \tag{11}$$

$$L_1 = \frac{t_{\rm u}}{2\pi}\left(\pi - \tan^{-1}\frac{2\pi T_1}{t_{\rm u}}\right) \tag{12}$$

The static process gain $k_{\rm p}$ can be easily estimated on-line from the steady-state input–output data following any step change in setpoint. Together with the estimated values of $k_{\rm u}$ and $t_{\rm u}$, T_1 and L_1 can be computed from Eqs. (11) and (12). The normalized dead time θ, which is simply computed as the ratio of L_1 and T_1, can then be used to implement on-line tuning or controller design given by the refined Ziegler–Nichols formula of Eq. (8).

With the model of Eq. (10), other tuning formulae such as the Internal Model Control (IMC) tuning formula which aims to yield well-damped responses may be used:

$$\begin{aligned} k_c &= \frac{1}{k_p}\frac{2T_1 + L_1}{2\lambda + L_1} \\ T_i &= T_1 + \frac{L_1}{2} \\ T_d &= \frac{T_1 L_1}{2T_1 + L_1} \end{aligned} \tag{13}$$

The constant λ is equivalent to the desired closed-loop time constant. The controller gain can thus be chosen to be aggressive or conservative by varying λ (with a lower recommended bound of $0.2T_1$ or $0.25L_1$).

Autotuner

The combined relay control and on-line controller design to form an autotuner will be demonstrated in the following through an example. The process has a transfer function of $e^{-0.4s}/(1 + s)^2$. In the simulation result of Fig. 5, the first part of the response shows relay control with the resultant sustained oscillation. The Ziegler–Nichols formula was used to tune the PID controller and the relay was switched out. The next setpoint change shows that the process output was reasonably good except that the overshoot was excessive. Meanwhile, the static process gain could be measured from this setpoint response, and the refined Ziegler–Nichols formula with setpoint weighting factor were used to retune the PID controller. The subsequent setpoint change shows a much improved response with an acceptable overshoot.

In some applications, a P or PI controller rather than a full PID controller would be adequate and the corresponding tuning formulae are well-documented (4). With additional computation to estimate the process model of Eq. (10) using Eqs. (11) and (12)), relay autotuning can be employed to provide on-line autotuning of model-based advanced controllers such as the pole-placement control, generalized predictive control, and the Smith predictor (4,5). Using Eq. (6) and following a similar analysis and development as in the case of the PID controller, the relay autotuning method has been applied to phase-lead and phase-lag compensators (6) which are widely used in servomechanisms and to the increasingly popular fuzzy logic controllers (7). Owing to the simplicity of its operation, which merely requires the operator to push a button to start it, the relay autotuner has been combined with gain-scheduling for the control of a wide class of highly nonlinear processes. The instrument or control engineer would first need to specify a gain-scheduling variable, such as throughput, control valve output, level, and so on, which must be measurable. At different regions of operations associated with a specific value of gain-scheduling variable, the PID controller parameters are obtained using the relay autotuner. With a few more settings of PID controller parameters obtained at different regions, the gain schedule is automatically set. The simplicity of the relay autotuner has then facilitated the automatic generation of the gain schedule and made this nonlinear control strategy much easier to apply in practice.

PRACTICAL CONSIDERATIONS

As in other control applications, signal filtering and averaging should be used wherever possible to reduce the effect of measurement noise. The relay magnitude and hysteresis should also be adjusted either manually or automatically as discussed earlier. The need for relay bias adjustment in the presence of load disturbances and specific arrangement to facilitate autotuning of cascade controllers will be discussed in the following.

Effect of Load Disturbances

In Figs. 4 and 7 the relay oscillations are symmetrical, a basic condition to be satisfied for good accuracy of process modeling. In order to operate the process output y near the setpoint and in the presence of static load disturbance, the control signal u is normally biased to a suitable steady-state value u_o. During the normal operation when the setpoint or the load has changed significantly, u_o should also be changed accordingly.

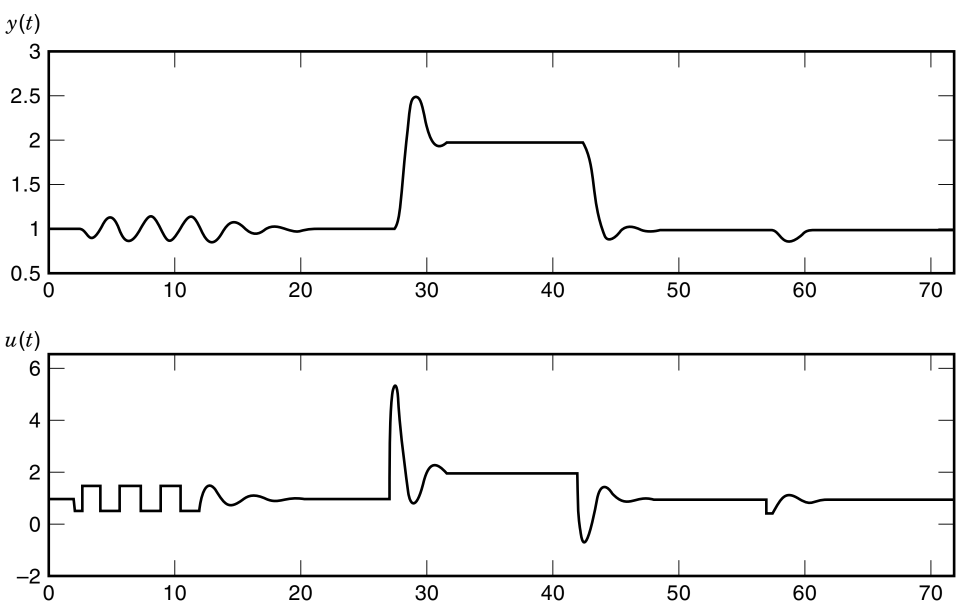

Figure 5. Autotuning performance (k_c = 3.43, T_i = 1.44, T_d = 0.36; β = 0.45 at $t > 35$).

This can be easily accomplished by allowing it to track the average value of the integrator output of the PID controller. If a load change of Δl occurs during relay autotuning, two possible cases will be encountered. The first is that Δl is so large that the relay oscillations will be quenched. In this case, an additional bias component will have to be added successively until the oscillation resumes. An alternative is to cascade the relay with an integrator to automatically generate the additional bias needed to restore oscillations. The integrator will then be switched off with the integrator output added to u_o, and relay autotuning will be continued. The second and more usual case is that oscillations will be maintained but they become asymmetrical (8). The ultimate gain and period estimated from the asymmetrical oscillations may then contain significant error. A simple way to correct this situation is to add an additional bias signal to the relay, u_b, that is equivalent to the negative of the estimated value of Δl. If d is the relay amplitude and t_1 and t_2 are the positive and negative relay output intervals, respectively, u_b can be computed as

$$u_b = \frac{t_1 - t_2}{t_1 + t_2} d + \frac{1}{k_p(t_1 + t_2)} \int_{\tau}^{\tau + t_1 + t_\tau} e\, dt \tag{14}$$

where τ is chosen such that the integration is performed over one period of the steady-state oscillation.

Cascade Control

The performance of single-loop controllers in the presence of load disturbance can be greatly improved if suitable intermediate (secondary) variables are available for measurement and are used to facilitate cascade control. In its simplest form, it consists of an inner loop by feeding back the intermediate variable so that the effect of load disturbance or certain nonlinearities can be largely reduced by an inner controller before it has a chance to upset the operation of the outer loop controlling the primary variable. The block diagram of a cascade control system is shown in Fig. 6, where y_2 is the intermediate variable which is highly affected by the load disturbance l_2; y_1 is the primary variable to be controlled, and u_2 is the control variable. The effectiveness of cascade control depends on the relative speed of the inner and outer loops, a rule of thumb being that the inner loop should be at least three times faster than the outer loop (9).

Figure 6 also shows how the relay autotuner could be connected. In most applications, the inner loop needs to be autotuned only once at the commissioning stage and there is little need for retuning. With the inner loop closed, the outer loop is then autotuned. The typical sequence and performance of the relay autotuner is demonstrated in Fig. 7 using the following example:

$$G_1(s) = \frac{e^{-s}}{(1+s)^2}$$
$$G_2(s) = \frac{e^{-0.1s}}{1+0.1s}$$

With autotuning of the inner loop, a PI controller with parameters (k_c = 1.03, T_I = 0.26) is commissioned. With autotuning of the outer loop, a PID controller with parameters (k_c = 1.45, T_i = 2.73, T_d = 0.68) is obtained. Note that the transients during the autotuning of the inner loop has little effect on the primary variable y_1. It is thus safe to even retune the inner loop if necessary without having to open the outer loop (9). The load disturbance response also clearly demonstrates the effectiveness of cascade control in that the load disturbance in the inner loop which occurs at t = 50 is well-regulated before it has a chance to upset the outer loop.

EXTENSIONS

Owing to the simplicity and robustness of relay autotuning, many extensions of tuning formula and applications have been made. One important class of tuning formula is the gain and phase margin method. It requires a second-order plus dead time model which approximates a high-order process better than the first-order plus dead time model:

$$G_p = k_p \frac{e^{-sL_2}}{(1+sT_2)^2} \tag{15}$$

It is straightforward to estimate these model parameters from the ultimate gain and period obtained from relay control (3):

$$T_2 = \frac{t_u}{2\pi}\sqrt{k_u k_p - 1} \tag{16}$$

$$L_2 = \frac{t_u}{2\pi}\left(\pi - 2\tan^{-1}\frac{2\pi T_2}{T_u}\right) \tag{17}$$

The details of the gain and phase margin design method can be found in Ref. 10. The tuning formula is given by

$$\begin{aligned} k_c' &= \frac{\omega_p T_2}{A_m k_p} \\ T_i' &= \left(2\omega_p - \frac{4\omega_p^2 L_2}{\pi} + \frac{1}{T_2}\right)^{-1} \\ T_d' &= T_2 \\ \omega_p &= \frac{A_m \phi_m + \frac{1}{2}\pi A_m (A_m - 1)}{(A_m^2 - 1)L_2} \end{aligned} \tag{18}$$

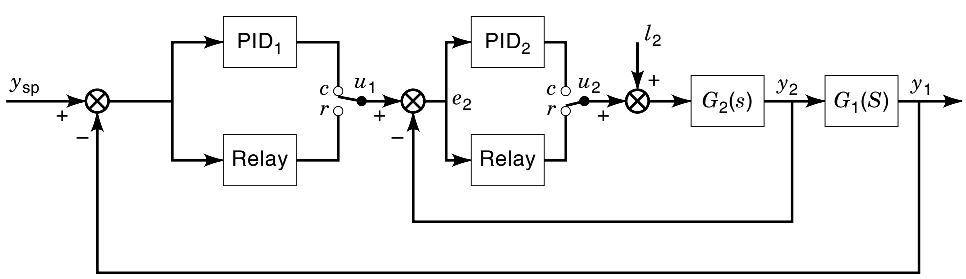

Figure 6. Block diagram of a cascade control system with relay autotuner.

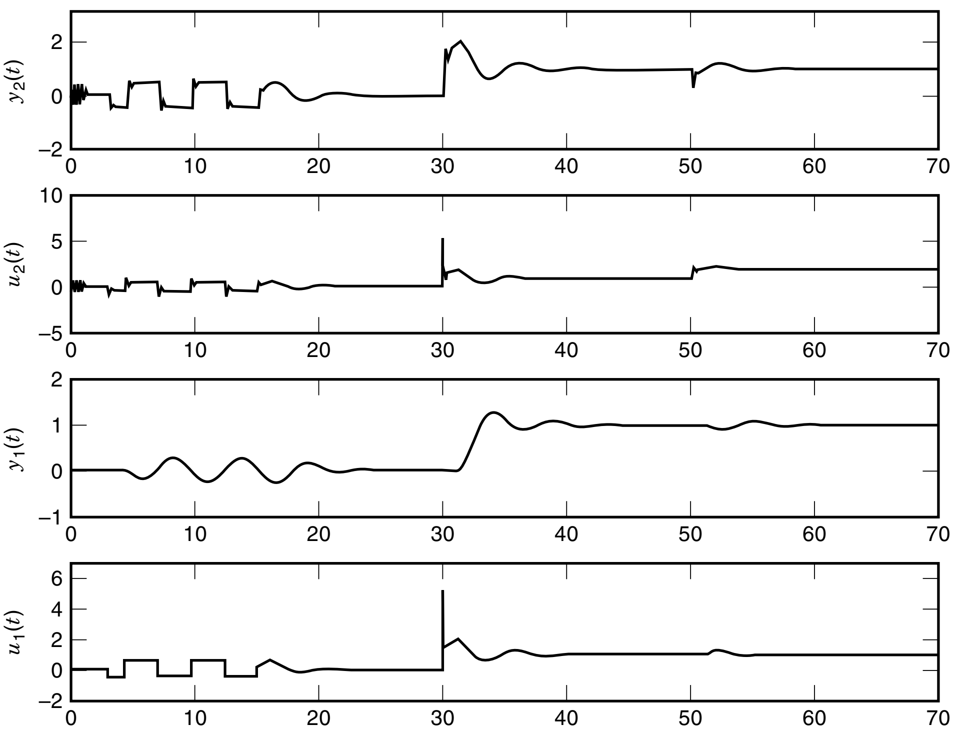

Figure 7. Autotuning performance of cascade controllers. (Autotuning of inner loop starts at $t = 3$; subsequent setpoint change at $t = 30$ and load change at $t = 50$).

where A_m and ϕ_m are the desired gain and phase margins, respectively. Note that the above PID parameters (k_c', T_i', T_d') are those corresponding to the interacting form (4). To convert them into the standard noninteracting form of Eq. (1), the following formula can be used (4):

$$k_c = k_c' \frac{T_i' + T_d'}{T_i'}$$
$$T_i = T_i' + T_d' \tag{19}$$
$$T_d = \frac{T_i' T_d'}{T_i' + T_d'}$$

The optimal PID controller parameters can hence be computed from Eqs. (16)–(19) once the desired gain and phase margins are specified based on practical requirements of speed and robustness. Their default values can be set as (3, 60°). The performance of this gain and phase margin design is also much better than the simpler phase margin design of Eq. (7) over a wide range of process dynamics. Compared to the refined Ziegler–Nichols method, it has the advantage that the robustness properties could be specified.

We shall present two more sophisticated extensions in the following: the frequency response approach and the relay control of multivariable systems.

Frequency Response Approach

In the basic relay autotuner employing Ziegler–Nichols or other tuning formula, the ultimate gain and phase (or one point on the Nyquist curve near the critical point) are identified and used in the controller design. It has been mentioned earlier that by varying the relay amplitude and the hysteresis width and repeating the relay test, other points on the Nyquist curve can be identified. However, this is more time-consuming. It would thus be attractive to estimate more points on the Nyquist curve with only one relay test. Wang et al. (11) have proposed such an approach by further analyzing the transient and steady-state response of the relay oscillations. If the process dynamics is described by $Y(s) = G_p(s)U(s)$, the frequency response is given by $G_p(j\omega)$ which can be numerically computed as the ratio of $Y(j\omega)$ and $U(j\omega)$. $Y(j\omega)$ and $U(j\omega)$ could be obtained by taking the fast Fourier transform (FFT) of the signals $y(t)$ and $u(t)$. This, however, requires that both $y(t)$ and $u(t)$ be decaying to zero in finite time, which is not the case because the relay oscillations contain periodic components. The decay method of Wang et al. (11) overcomes this difficulty by a further numerical processing of the process input and output signals using an exponential window $e^{-\alpha t}(\alpha > 0)$. We thus have in the time domain the following exponentially weighted signals:

$$\tilde{y}(t) = y(t)e^{-\alpha t}$$
$$\tilde{u}(t) = u(t)e^{-\alpha t}$$

and in the frequency domain:

$$\tilde{Y}(j\omega) = \int_0^{t_f} \tilde{y}(\tau)e^{-j\omega t}\,dt \tag{20}$$

$$\tilde{U}(j\omega) = \int_0^{t_f} \tilde{y}(\tau)e^{-j\omega t}\,dt \tag{21}$$

α can be suitably chosen based on the ultimate period and the noise level such that $y(t)$ and $u(t)$ decay to zero exponentially at $t > t_f$ and both the transient and steady-state components have been well-utilized. It is straightforward to show that $\tilde{Y}(j\omega)$ and $\tilde{U}(j\omega)$ are equivalent to $Y(j\omega + \alpha)$ and $U(j\omega + \alpha)$. We can thus compute the shifted frequency response:

$$G_p(j\omega + \alpha) = \frac{\tilde{Y}(j\omega)}{\tilde{U}(j\omega)} \tag{22}$$

The relevant frequency response for a number of specified points can thus be computed at discrete frequencies (11). This is sufficient for the purpose of controller design in the frequency domain to complete the relay autotuning design. To demonstrate the accuracy of the process frequency response thus estimated, an inverse FFT of $G_p(j\omega + \alpha)$ can be computed which produces $g(kT)e^{-\alpha kT}$, where $g(t)$ is the impulse response of the process and T is the corresponding discrete time interval. The unshifted frequency response $G_p(j\omega)$ can then be computed by applying FFT to the computed $g(kT)$. Figure 8 shows the results of frequency response estimation for four different processes. It is evident that accurate estimation at frequencies up to the ultimate frequency can be obtained.

An important advantage of the frequency response approach is that accurate information on the structure of the process model, such as the order of the system and whether the dynamics has oscillation modes, is not required. The controller design after the relay test should therefore be selected accordingly. Following the direct controller design approach from frequency response data proposed by Goberdhansingh et al. (12), the following controller design using the shifted frequency response data computed from Eq. (22) has been developed (11).

First the desired closed-loop frequency response data points are generated. This can be obtained from a specified general closed-loop transfer function model of the following form:

$$H(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n^2 + \omega_n^2} e^{-sL} \tag{23}$$

The estimate of the apparent dead time L is obtained from the relay test results using Eq. (12), or more accurately using Eq. (17). The apparent dead time which accounts for the pure dead time and any nonminimum phase term represents the noncontrollable part of the process dynamics and is hence retained in the closed-loop dynamics. The natural frequency ω_n and the damping factor ξ can be easily set depending on the speed of response and the robustness required (11). They can also be easily computed if the specifications are given as the gain and phase margins.

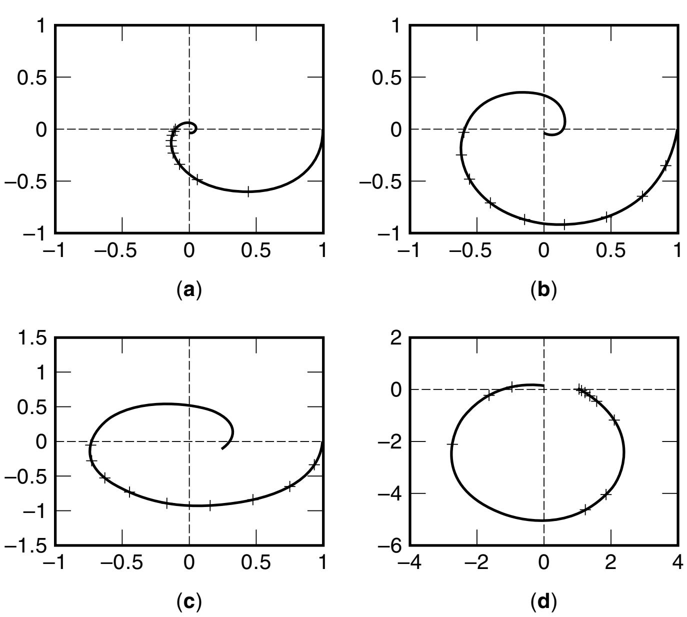

Figure 8. Nyquist plots of different processes. (a) $e^{-2s}/(1 + 10s)$; (b) $1/(1 + s)^{10}$; (c) $(1 - s)e^{-2s}/(1 + s)^5$; (d) $e^{-0.2s}/(1 + 0.25 + 1)$ (——— actual, --- estimated).

With the desired $H(s)$ specified, the desired open-loop frequency response can be numerically computed as

$$G_0(j\omega) = \frac{H(j\omega)}{1 - H(j\omega)} \tag{24}$$

The PID controller has a frequency response given by

$$G_c(j\omega) = k_c\left(1 + \frac{1}{j\omega T_i} + j\omega T_d\right)$$

The open-loop frequency response of the combined controller and process is

$$G_c(j\omega)G_p(j\omega) = \phi(j\omega)x$$

where

$$\phi(j\omega) = \left[G_p(j\omega)\ \frac{G_p(j\omega)}{j\omega}\ j\omega G_p(j\omega)\right] \tag{25}$$

$$x = \left[k_c\ \frac{k_c}{T_i}\ k_c T_d\right]^T \tag{26}$$

The controller design problem can now be formulated as a typical minimization problem of selecting x to minimize the loss function

$$J = \sum_{i=1}^{m} |\phi(j\omega_i)x - G_o(j\omega_i)|^2 \tag{27}$$

where m is the total number of frequency points selected. With $\phi(j\omega_i)$ and $G_o(j\omega_i)$ computed at each discrete frequency, the standard least squares solution can be used to solve for x and hence the PID controller parameters recovered from Eq. (26). The least-squares solution of x is given by

$$x = (\Phi_2^T\Phi_2)^{-1}\Phi_2^T\Omega_2 \tag{28}$$

where

$$\Phi_2 = \begin{bmatrix} R_e(\Phi_1) \\ I_m(\Phi_1) \end{bmatrix}, \qquad \Omega_2 = \begin{bmatrix} R_e(\Omega_1) \\ I_m(\Omega_1) \end{bmatrix}$$

$$\Phi_1 = [\phi(j\omega_1)\ldots\phi(j\omega_m)]^T$$

$$\Omega_1 = [G_o(j\omega_1)\ldots G_o(j\omega_m)]^T$$

The complete frequency-response-based relay autotuner is given by the combination of the relay control, the FFT computation of Eq. (22), and the least-squares estimate of Eq. (28). Since Eq. (22) is obtained in the shifted frequency domain, the development of Eq. (24)–(28) should be modified likewise. Another simple modification (11) is to allow the closed-loop transfer function from the load disturbance to the output to be specified, and hence the controller is tuned for optimal load disturbance. Simulation studies of a number of different processes have shown that this method yields better results than

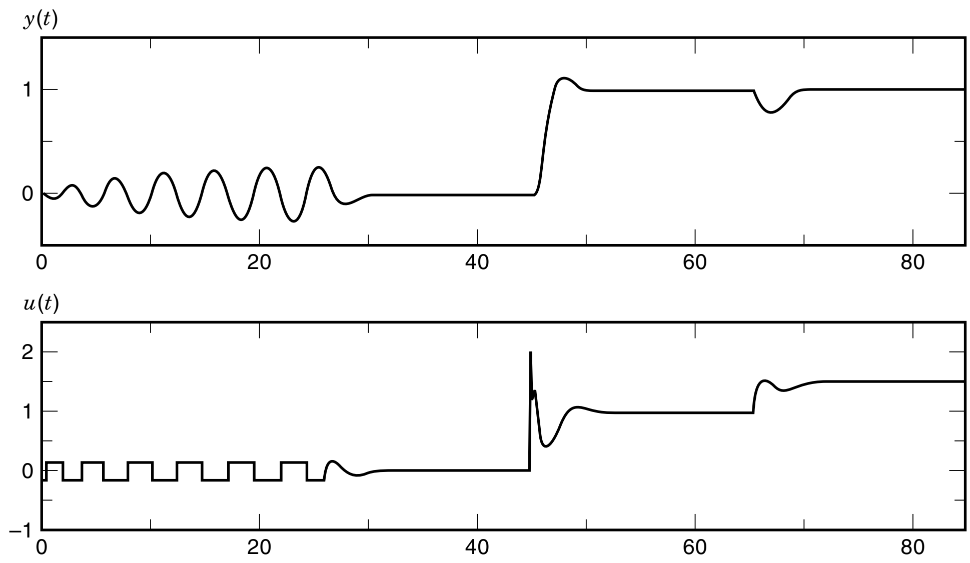

Figure 9. Autotuning performance of an oscillatory process.

the simpler methods based on only one point on the Nyquist curve. Its performance will be demonstrated in the following example of a highly oscillatory system in which case all the previous methods based on the assumption that the process has well-damped dynamics would produce poor results:

$$G_p(s) = \frac{e^{-0.2s}}{1 + 0.2s + s^2}$$

The accurate frequency response estimated from the relay control and the FFT computation has already been demonstrated in Fig. 8(d). The frequency response approach optimized for load response would yield a PID controller of K_c = 1.71, T_i = 1.26, and T_d = 1.37. The autotuning transients and resultant step and load responses in Fig. 9 clearly show the excellent performance of this method.

One additional feature of relay control is that it can stabilize a process rapidly. In other autotuning or self-tuning controllers, when the closed-loop response becomes unstable due to large parameter changes, the supervisory software will reduce the PID controller gain successively until the system is stable before subsequent retuning could take place. This usually results in very long recovery time. Relay control could quickly stabilize the system and simultaneously provide the controller oscillation for autotuning. Figure 10 shows a typical case of instability when the process dead time is drastically increased. The subsequent relay control and retuning demonstrate the capability of rapid recovery of this method.

Multivariable Control

As in the case of single-variable systems, the majority of multivariable systems are controlled in practice by multiloop PID controllers without decouplers (5). The tuning of a multiloop PID controller for a multivariable process is naturally much more complex than the single-variable case. The extension of the simple-to-use and robust autotuning technique to a multivariable controller has recently been ad-

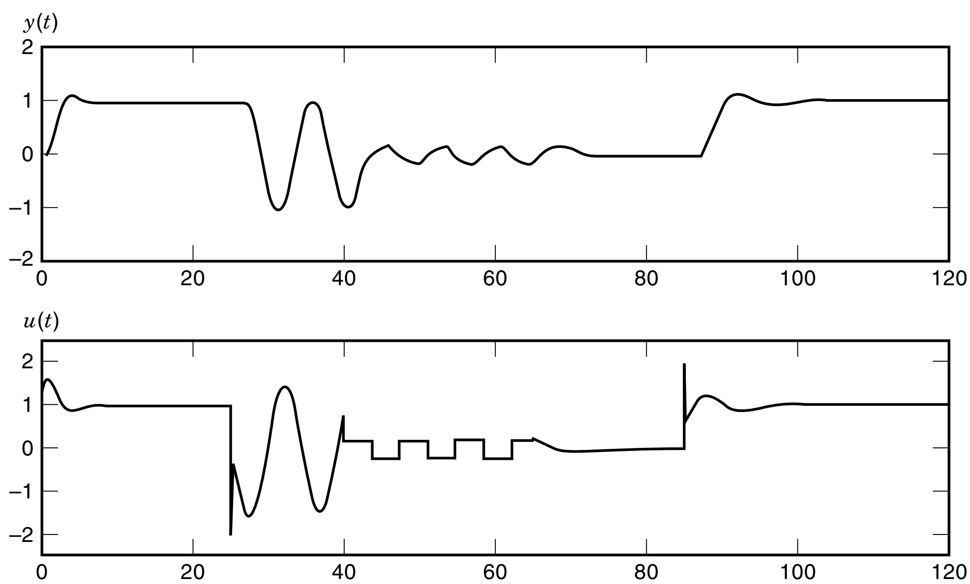

Figure 10. Relay stabilization and autotuning when the system becomes unstable. (At t = 0, $G_p(s) = [1/(1+s)^2]e^{-0.5s}$; at t = 20, $G_p(s) = [1/(1+s)^2]e^{-2s}$.)

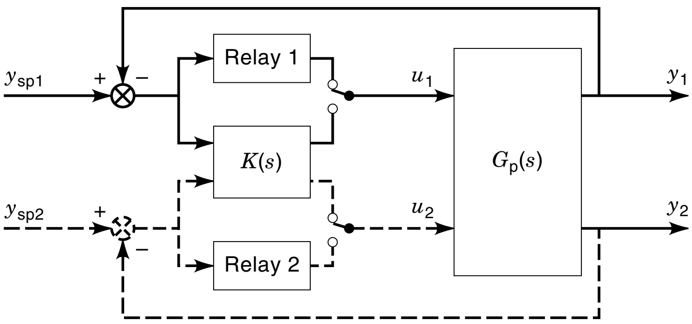

Figure 11. Block diagram of sequential relay autotuning of a multivariable system.

dressed (13–16). It has been found that it is not worthwhile to tune all the loops together because the relay oscillations would interact and create complications. A sequential procedure has therefore been recommended, and that for a 2 × 2 system as shown in Fig. 11 is outlined as follows:

1. The faster loop is first autotuned using the relay autotuner, with the other loop being left opened.
2. The slower loop is then autotuned using the relay autotuner, with the inner loop closed after step 1.

Step 2 may be repeated for the other loop if the prior information on the relative speed of the loops is wrongly given. Finally the multivariable PID controller $K(s)$ can be computed using the multivariable extension of the frequency response design approach (16). We shall illustrate the procedure by the auto-tuning of a well studied multivariable process model of a distillation column (17):

$$G_p(s) = \begin{bmatrix} \dfrac{12.8e^{-s}}{1+16.7s} & \dfrac{-18.9e^{-3s}}{1+21s} \\ \dfrac{6.6e^{-s}}{1+10.9s} & \dfrac{-19.4e^{-3s}}{1+14.4s} \end{bmatrix} \quad (29)$$

The sequential relay autotuning is shown in Fig. 12. Loop 1, which is the fast loop, is first autotuned. With loop 1 closed, loop 2 is then autotuned and the multiloop PID controller $K(s)$ is found to be

$$\begin{bmatrix} 0.375\left(1+\dfrac{1}{8.29s}\right) & 0 \\ 0 & -0.075\left(1+\dfrac{1}{23.6s}\right) \end{bmatrix}$$

The subsequent set point and load responses demonstrate that the controllers are reasonably tuned. The performance is indeed better than that tuned manually using conventional techniques which require a tedious trial-and-error procedure (5).

While all the single-loop modeling and controller designs can be used to design the multiloop PID controller, the frequency response approach has an additional advantage that it could be easily extended to a truly multivariable PID controller with cross-coupling terms to reduce interaction effect. Such a controller has been designed on-line using the same sequential relay test, and the resultant multivariable PID controller (16) is found to be

$$\begin{bmatrix} 0.184\left(1+\dfrac{1}{3.92s}\right) & -0.0102\left(1+\dfrac{1}{0.445s}-0.804s\right) \\ -0.0674\left(1+\dfrac{1}{4.23s}+0.796s\right) & -0.006\left(1+\dfrac{1}{4.25s}\right) \end{bmatrix}$$

The much improved set point and load responses are clearly shown in Fig. 12.

CONCLUDING REMARKS

It has been established from many real industrial applications that the relay control and autotuners are easy to apply and they require the minimum prior information. A survey on commercial products could be found in Ref. 18. The extension of relay autotuners to multivariable PID controllers and to other model-based controllers would be even more significant because they are very difficult to commission without autotuning. In addition, relay control generates information on the critical (crossover) frequency of the process which could

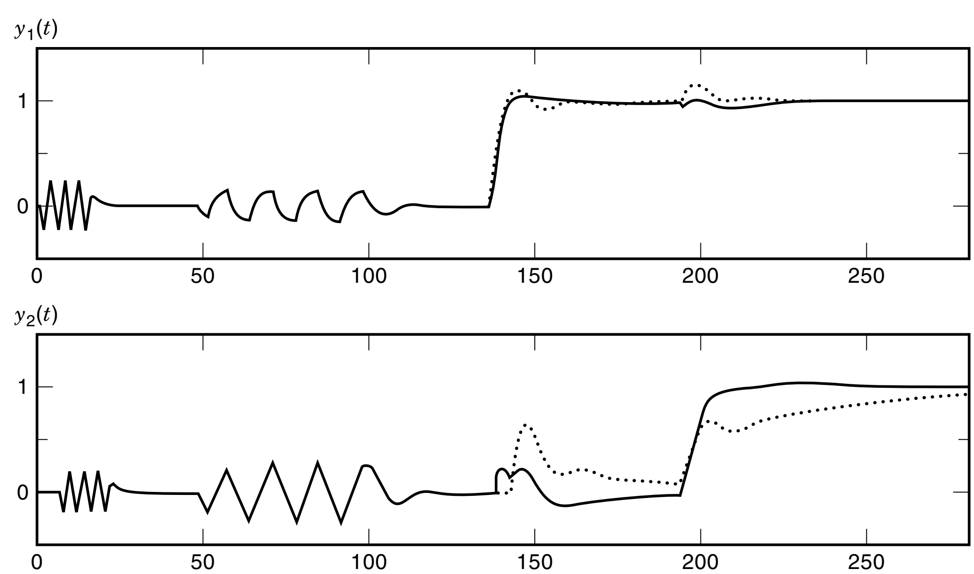

Figure 12. Autotuning performance of a multivariable system (——— controller with cross-coupling term; --- controller without cross-coupling terms).

be used to automate the selection of sampling rates for digital control. They have thus found increasing applications as automatic tools to initialize the more complicated self-tuning and adaptive controllers.

BIBLIOGRAPHY

1. K. J. Astrom and T. Hagglund, Automatic tuning of simple regulators with specification on phase and amplitude margins, *Automatica,* **20** (5): 645–651, 1984.
2. J. G. Ziegler and N. B. Nichols, Optimum settings for automatic controllers, *Trans. ASME,* **64**: 759–768, 1942.
3. C. C. Hang, K. J. Astrom, and W. K. Ho, Refinements of the Ziegler–Nichols tuning formula, *IEE Proc.,* **138**: Part D, 111–118, 1991.
4. K. J. Astrom and T. Hagglund, *PID Controllers: Theory, Design and Tuning,* Research Triangle Park, NC: Instrument Society of America, 1995.
5. C. C. Hang, T. H. Lee, and W. K. Ho, *Adaptive Control,* Instrument Society of America, 1993.
6. J. Yang, C. S. Chen, and Y. S. Xu, Automatic tuning of phase-lead and phase-lag compensators, *Int. J. Control,* **60** (4): 631–640, 1994.
7. J. X. Xu, L. Chen, and C. C. Hang, Modified tuning of a fuzzy logic controller, *Eng. Appl. Artif. Intell.,* **9** (1): 65–74, 1996.
8. C. C. Hang, K. J. Astrom, and W. K. Ho, Relay auto-tuning in the presence of static load disturbance, *Automatica,* **29** (2): 563–564, 1993.
9. C. C. Hang, A. P. Loh, and V. U. Vasnani, Relay feedback auto-tuning of cascade controllers, *IEEE Trans. Control Syst. Technol.,* **2**: 42–45, 1994.
10. W .K. Ho, C. C. Hang, and L. S. Cao, Tuning of PID controllers based on gain and phase margin specifications, *Automatica,* **31** (3): 497–502, 1995.
11. Q. G. Wang, C. C. Hang, and Q. Bi, A frequency domain controller design method, *Trans. I Chem. E,* **75**: Part A, 64–72, 1997.
12. E. Goberdhansingh, L. Wang, and W. R. Cluett, Robust control design using direct frequency response, *Proc. Autom. Control Conf.,* **4**: 3026–3028, 1992.
13. Z. J. Plamor, Y. Halevi, and N. Krasney, Automatic tuning of decentralized PID controllers for TITO processes, *Proc. IFAC World Congr.,* Australia, **2**: 311–314, 1993.
14. S. H. Shen and C. C. Yu, Use of relay feedback test for automatic tuning of multivariable systems, *AIChE J.,* **40** (4): 627–643, 1994.
15. S. Menani and H. Koivo, Relay tuning of multivariable PI controllers, *Proc. IFAC World Congr.,* USA, **K**: 139–144, 1996.
16. Q. G. Wang, C. C. Hang, and B. Zhou, A frequency response approach to auto-tuning of multivariable controllers, *Proc. IFAC World Congr.,* USA, **K**: 295–300, 1996.
17. W. L. Luyben, A simple method for tuning SISO controllers in a multivariable system, *Proc. Des. Dev.,* **25**: 654–660, 1986.
18. K. J. Astrom, T. Hagglund, C. C. Hang, and W. K. Ho, Automatic tuning and adaptation for PID controllers—A survey, *Control Eng. Pract.,* **1** (4): 699–714, 1993.

C. C. HANG
National University of Singapore

RELAY COORDINATION. See POWER SYSTEM RELAYING.