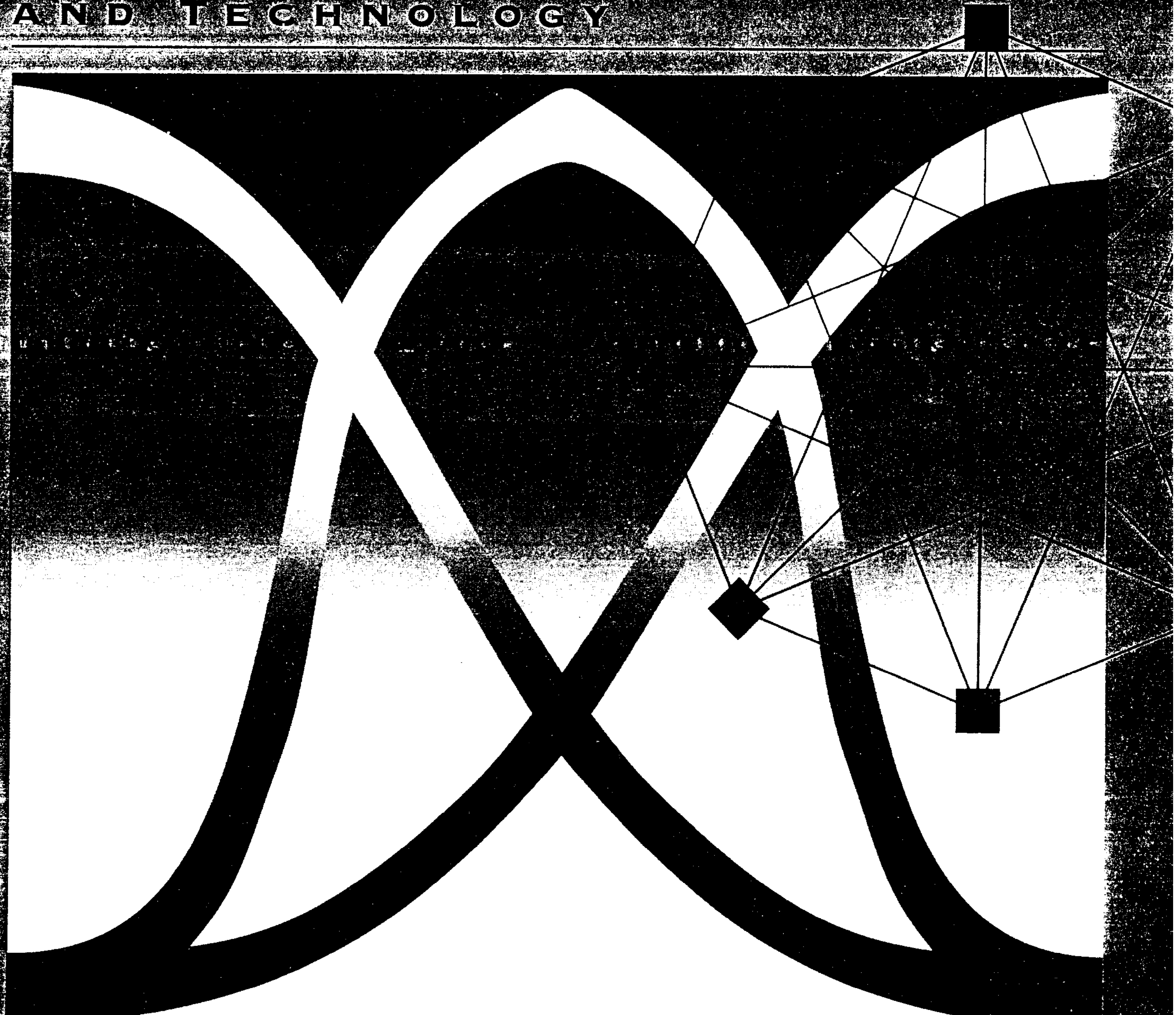
VOLUME 3 ISSUE 1 1995

JOURNAL OF INTELLIGENT & FUZZY SYSTEMS

APPLICATIONS IN ENGINEERING AND TECHNOLOGY



A Wiley-Interscience Publication
John Wiley & Sons, Inc.
New York • Chichester • Brisbane • Toronto • Singapore

SPECIAL ISSUE ON
FUZZY LOGIC AND NEURAL NETWORKS:
THEORY AND APPLICATIONS

MOHAMMAD JAMSHIDI & TIMOTHY J. ROSS, EDITORS
LOTFI A. ZADEH, HONORARY EDITOR

John S. Baras*
Nital S. Patel
Institute for Systems Research and Department of Electrical Engineering University of Maryland at College Park College Park, MD 20742

Derivation of Fuzzy Rules for Model-Free Tuning of PID Controllers

Abstract

In this article we derive the rules for a fuzzy logic based PID tuner (expert) for dominant pole systems with large rise times. We present applications of the expert to a third-order plant, separator temperature control, and pH control. It is observed that the expert can successfully tune the PID gains without requiring any process identification. © *1995 John Wiley and Sons, Inc.*

Introduction

The recent stimulus for the application of intelligent control to the chemical process industry has resulted in the proposal of a variety of schemes. However, most of them are targeted toward specific applications. The need is to develop schemes for intelligent control that are applicable to a broader class of problems (Åström and McAvoy, 1992).

It is often seen that control experts tune the parameters of a controller according to error versus time curves based on their knowledge and experience, rather than on complicated control algorithms. In fact, their tuning actions seem to be based on relations between the shape of the response curve and the parameters of the controller, rather than on explicit process models. This kind of tuning method, if realizable, is captivating because of its independence from a process model. Hence, an *expert* developed with such a principle in mind will be intelligent and universal to various controlled processes. The fact that the expert tunes the controller parameters, without actually knowing the process parameters, guarantees robustness with respect to process parameter variation.

Following Tolle and Ersü (1992), we identify two performance criteria that the expert needs to satisfy. (1) As with humans, satisfactory learning requires frequent repetition of the same effort, so the system is improved by being restarted from the same initial conditions again and again. (2) An important factor for technical control problems is the ability to stabilize the control loop in the first trial, however, with relatively bad performance in general.

The rest of the article is organized as follows. We state the problem. Then we derive the rule base. The rule base derivation is done in several steps. We first identify the required corrections in the position of the dominant closed-loop poles, to correct for the deviation in the observed response from that required. These are then linked to movements in the controller zeros. We then identify the required changes in the loop gain. Finally, we link variation of the PID gains to variations in the loop gain and the controller zeros. Hence, when we change the PID gains, we move the dominant closed-loop poles via movement of the controller zeros, and variation of the loop gain. The required changes in the PID gains are then used to generate the rule base. After deriving the rule base, we present some examples. This is followed by a section on overshoot control and conclusions.

*Martin Marietta Chair in Systems Engineering.

Journal of Intelligent and Fuzzy Systems, Vol. 3, 59–69 (1995)

CCC 1064-1246/95/010059-11

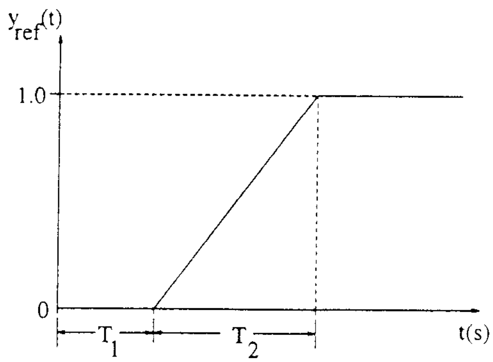


Figure 1. Reference response.

Statement of the Problem

The problem can be stated as follows. Given a stable plant

$$P(s) = \frac{Y(s)}{E(s)} = \frac{\bar{K}}{(\tau_1 s + 1)(\tau_2 s + 1)\cdots(\tau_n s + 1)} \tag{1}$$

where $\tau_1 \gg \max(1, \tau_i)$, $i = 2, 3, \ldots, n$, along with a PID controller

$$C(s) = \frac{R(s)}{Y(s)} = K_C + \frac{K_I}{s} + K_D s\,, \tag{2}$$

develop a strategy to instantaneously tune K_C, K_I, and K_D, based on observation of the plant output $y(t)$, and the set-point $s(t)$, to get a *good* response to setpoint changes. By good response, we mean that the closed-loop response approximate a given reference response. The latter is specified by two parameters, T_1 and T_2, and is illustrated in Figure 1. Here T_1 can be thought of as dead time and T_2 as the open-loop response time. The overall system architecture is illustrated in Figure 2.

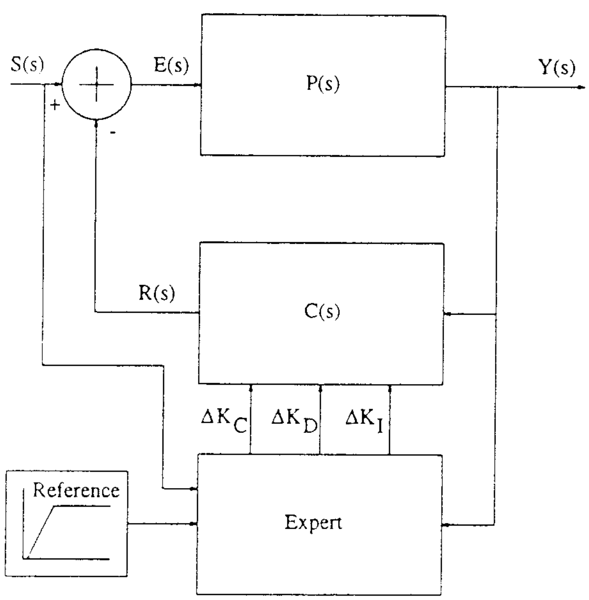


Figure 2. Overall system architecture.

This strategy deviates from the current trend where effort is made to obtain information about the plant by carrying out data analysis and modeling. A review of these techniques can be found in Koivo and Tanttu (1991). Note that gain scheduling is an immediate consequence of the proposed tuning strategy.

Derivation of the Rule Base

In the current context, we want to compensate for rise time and the settling time of the closed-loop response while ensuring that the system remains stable. Hence, it is natural to consider two sets of rules. The first set (R_1) deals with rise time compensation, whereas the second set (R_2) deals with reducing the settling time and stabilizing the closed-loop system. When a set-point change is detected, R_1 is activated. Once the response reaches the current set-point, R_1 is switched off, and R_2 is activated. The input to R_1 is the error $e_1(t)$, and the inputs to R_2 are (i) the error $e_2(t)$ and (ii) the rate of change of error $d/dt\, e_2(t)$. $e_1(t)$ and $e_2(t)$ are calculated as

$$e_1(t) = y_{ref}(t) - \frac{y(t) - s_{old}}{s_{new} - s_{old}} \tag{3}$$

$$e_2(t) = s_{new} - y(t) \tag{4}$$

where $y(t)$ is observed plant response, $y_{ref}(t)$ is the reference response, s_{old} is the previous set-point value, and s_{new} is the current set-point value. The transformation $(y(t) - s_{old})/(s_{new} - s_{old})$ is required in Eq. (3), because the reference response (Fig. 1) is always specified as going from 0 to 1. The above transformation maps the interval $[s_{old}, s_{new}]$ onto the interval $[0, 1]$. Hence, now we can directly compare the observed response (after transformation) with the reference response. The definition of $e_1(t)$ as in Eq. (3) has the further advantage that $-1 \leq e_1(t) \leq 1$ irrespective of the set-points chosen.

Based on classical control theory (see Kuo (1991)), the following rules are postulated for the manipulation of the dominant closed-loop poles.

R_1:

1. If $e_1(t)$ is positive, move the dominant closed-loop poles towards the imaginary axis, and away from the real axis.
2. If $e_1(t)$ is negative, move the dominant closed-loop poles away from the imaginary axis, and towards the real axis.

R_2:

1. If $e_2(t)$ is not *small*, or if $d/dt\, e_2(t)$ is not *zero*, move the dominant closed-loop poles away from the imaginary axis, and toward the real axis.
2. If $e_2(t)$ is *small*, and if $d/dt\, e_2(t)$ is *zero*, move the dominant closed-loop poles away from the real axis, and toward the imaginary axis. This rule is incorporated to prevent the response from getting overdamped.

These rules result in manipulation of the damping factor and the natural frequency of the dominant poles to push the response toward that desired. For example, R_2 rule 1 increases the damping factor and decreases the natural frequency to help reduce the settling time. It also ensures that, if the dominant closed-loop poles are unstable (i.e., lie in the right half-plane), they are drawn into the left half-plane. One can observe this by considering the root locus plot.

Variation of the Controller Zeros

We now consider the relationship between the controller zeros and the dominant closed-loop poles. This relationship is important, because what we want to manipulate are the PID gains, which in turn influence the controller zeros and the overall loop gain. The relationships derived below enable us to link the variations in the controller zeros to variations in the dominant closed-loop poles, and hence to R_1 and R_2 above. Note that the characteristic equation of the closed-loop system is given by

$$1 + \frac{N(s+z_1)(s+z_2)}{s(s+\lambda_1)(s+\lambda_2)\cdots(s+\lambda_n)} = 0\,, \qquad (5)$$

where $-z_1$, $-z_2$ are the controller zeros; $-\lambda_i$, $i = 1 \ldots n$ are the open-loop plant poles; and $N = KK_D$ is the loop gain, where $K = \bar{K}/(\tau_1\tau_2 \ldots \tau_n)$, with $\bar{K}$ the plant gain, and K_D is the derivative mode gain.

First Order Plant:

$$P(s) = \frac{K}{s+\lambda} \qquad (6)$$

$\Rightarrow$

$$1 + C(s)P(s) = 1 + \frac{N(s+z_1)(s+z_2)}{s(s+\lambda)} = 0\,. \qquad (7)$$

Let $z_1, z_2 = \alpha \pm j\beta$, and the roots of Eq. (7) be $s_1, s_2 = \nu \pm j\mu$. Then,

$$\frac{d\nu}{d\alpha} = -\frac{N}{1+N}\,, \qquad (8)$$

$$\frac{d\mu}{d\alpha} = \frac{N(2\alpha-\lambda)}{(1+N)\sqrt{4N(\alpha^2+\beta^2)(1+N)-(\lambda+2N\alpha)^2}}\,, \qquad (9)$$

$$\frac{d\nu}{d\beta} = 0\,, \qquad (10)$$

$$\frac{d\mu}{d\beta} = \frac{2N\beta}{\sqrt{4N(\alpha^2+\beta^2)(1+N)-(\lambda+2N\alpha)^2}}\,. \qquad (11)$$

Hence, if $\alpha \le \lambda/2$ then

$$\frac{d\nu}{d\alpha} \le 0,\ \frac{d\mu}{d\alpha} \le 0,\ \frac{d\nu}{d\beta} = 0,\ \frac{d\mu}{d\beta} \ge 0\,. \qquad (12)$$

Second Order Plant:

$$P(s) = \frac{K}{(s+\lambda_1)(s+\lambda_2)} \qquad (13)$$

$\Rightarrow$

$$1 + C(s)P(s) = 1 + \frac{N(s+z_1)(s+z_2)}{s(s+\lambda_1)(s+\lambda_2)} = 0\,. \qquad (14)$$

Let $z_1, z_2 = \alpha \pm j\beta$, and the closed-loop poles be $s_1, s_2 = \nu \pm j\mu$ and $s_3 = \gamma$ with $\gamma < \nu < 0$.
$\Rightarrow$

$$\frac{d\nu}{d\alpha} = \frac{N(\gamma+\alpha)}{(\gamma-\nu)^2+\mu^2}\,, \qquad (15)$$

$$\frac{d\mu}{d\alpha} = \frac{N(\mu^2-(\gamma-\nu)(\nu+\alpha))}{\mu((\gamma-\nu)^2+\mu^2)}\,. \qquad (16)$$

$$\frac{d\nu}{d\beta} = \frac{\beta N}{(\gamma - \nu)^2 + \mu^2}. \quad (17)$$

$$\frac{d\mu}{d\beta} = \frac{-\beta N(\gamma - \nu)}{\mu((\gamma - \nu)^2 + \mu^2)}. \quad (18)$$

Hence, if $\alpha \approx 0$, and $\mu^2 + \nu^2 < \nu\gamma$, we obtain

$$\frac{d\nu}{d\alpha} < 0, \frac{d\mu}{d\alpha} < 0, \frac{d\nu}{d\beta} > 0, \frac{d\mu}{d\beta} > 0. \quad (19)$$

A sufficient condition for the above is $\alpha \approx 0$, and $\mu^2 + \nu^2 < -\nu\lambda_2$.

Higher Order Plants:
Similar relations hold for higher-order plants. For example, in the case of a third-order plant we get

$$\frac{d\nu_1}{d\alpha} < 0, \frac{d\mu_1}{d\alpha} < 0, \frac{d\nu_1}{d\beta} > 0, \frac{d\mu_1}{d\beta} > 0. \quad (20)$$

provided $\nu_1^2 + \mu_1^2 < -\nu_1\lambda_2/2$, and $\alpha \approx 0$: where, $\nu_1 \pm j\mu_1$ are the dominant closed-loop poles and $-\lambda_2$ is the middle plant pole.

Because we are dealing with dominant pole plants having large rise times, we expect these relationships in Eqs. (12), (19), and (20) to hold. Based on these, we now deduce the rules for manipulating the controller zeros (z_i) to obtain the desired manipulation of the dominant closed-loop poles.

R_1:
1. If $e_1(t)$ is positive, then increase $\text{Re}(z_i)$, and increase $\text{Im}(z_i)$ in magnitude.
2. If $e_1(t)$ is negative, then decrease $\text{Re}(z_i)$, and decrease $\text{Im}(z_i)$ in magnitude.

R_2:
1. If $e_2(t)$ is not *small* or *zero*, or $d/dt\, e_2(t)$ is not *zero*, then decrease $\text{Re}(z_i)$.
2. If $e_2(t)$ is *small*, and $d/dt\, e_2(t)$ is *zero*, then increase $\text{Re}(z_i)$, and increase $\text{Im}(z_i)$ in magnitude.

Variation of the Loop Gain (N)

As observed above, R_2 stabilizes the system with respect to unstable dominant poles by decreasing $\text{Re}(z_i)$. Another cause of instability could be dead-time. To stabilize the system with respect to dead-time, we need to reduce the loop gain (N). Also, from root locus theory, we know that a larger gain (N) forces the dominant closed-loop poles towards the controller zeros. Based on these observations, the following rules are proposed for manipulating the loop gain.

R_1:
1. If $e_1(t)$ is *positive large*, increase N. This helps speed up very slow responses.
2. If $e_1(t)$ is *negative large*, decrease N. This aids in slowing down a very fast response.

R_2:
1. If $e_2(t)$ is greater than *small*, or $d/dt\, e_2(t)$ is not *zero*, then decrease N. This rule stabilizes the system against dead-time.
2. If $e_2(t)$ is *small*, and $d/dt\, e_2(t)$ is *zero*, then increase N. This rule aids in decreasing the damping.

We now consider manipulation of the PID gains to obtain the proposed variations in the controller zeros and the loop gain.

PID Gain Variation

The variation in the PID gains is derived under the following assumptions:

1. We can express the PID gain variation ΔK_C, ΔK_D, and ΔK_I as

$$\Delta K_C = \rho_{K_C}[K_{C_{max}} - K_{C_{min}}]\delta_{K_C} = \rho_{K_C} R_{K_C} \delta_{K_C},$$
$$\Delta K_D = \rho_{K_D}[K_{D_{max}} - K_{D_{min}}]\delta_{K_D} = \rho_{K_D} R_{K_D} \delta_{K_D},$$
$$\Delta K_I = \rho_{K_I}[K_{I_{max}} - K_{I_{min}}]\delta_{K_I} = \rho_{K_I} R_{K_I} \delta_{K_I},$$

   where δ_{K_C}, δ_{K_D}, δ_{K_I} are the defuzzified output from the fuzzy logic controller.
2. The output scaling factors are equal, i.e., $\rho_{K_C} = \rho_{K_D} = \rho_{K_I} = \rho$.
3. The PID gains are of the same order of magnitude as their ranges ($R_{K_C}, R_{K_D}, R_{K_I}$).

Influence on N

$$N = KK_D \quad (21)$$

$\Rightarrow$

$$\Delta N \approx K\,\Delta K_D = \rho K R_{K_D} \delta_{K_D}. \quad (22)$$

Influence on the Controller Zeros

We present an order of magnitude analysis. Let $z_i = \alpha + j\beta$.
$\Rightarrow$

$$\alpha = \frac{-K_c}{2K_D}, \tag{23}$$

$$\beta = \frac{K_C}{2K_D}\sqrt{\frac{4K_D K_I}{K_C^2} - 1} \tag{24}$$

$\Rightarrow$

$$\Delta\alpha \approx \frac{1}{2K_D} \times \left(\frac{K_C}{K_D}\rho_{K_D} R_{K_D}\delta_{K_D} - \rho_{K_C} R_{K_C}\delta_{K_C}\right). \tag{25}$$

By assumption 3, we have $K_C R_{K_D}/K_D \approx R_{K_C}$. Hence

$$\Delta\alpha \approx \frac{\rho R_{K_C}(\delta_{K_D} - \delta_{K_C})}{2K_D}. \tag{26}$$

Similarly

$$\Delta\beta \approx \frac{1}{\sqrt{\frac{4K_D K_I}{K_C^2} - 1}} \times \left\{\frac{\rho R_{K_C}}{2K_D}(\delta_{K_D} - \delta_{K_C}) + \frac{\rho R_{K_I}}{K_C}(\delta_{K_I} - \delta_{K_D})\right\}. \tag{27}$$

Because the rules for manipulating N, α, and β have been determined, we can relate δ_{K_D}, δ_{K_C}, and δ_{K_I} from Eqs. (22), (26), and (27).

We divide the inputs and the outputs into seven fuzzy classes. Namely, (1) Positive Large (PL), (2) Positive Medium (PM), (3) Positive Small (PS), (4) Zero (Z), (5) Negative Small (NS), (6) Negative Medium (NM), and (7) Negative Large (NL). Furthermore, we let the outputs share a common fuzzy membership function Δ, and denote by Δ_{K_C}, Δ_{K_D}, and Δ_{K_I} their individual output classes obtained after rule evaluation from which we obtain δ_{K_C}, δ_{K_D}, and δ_{K_I} after defuzzification. We adopted the centroid strategy for defuzzification of the output sets, the details of which can be found in any standard source on fuzzy logic, e.g., Kosko (1990). The centroids of the PL and NL classes were placed at -1 and $+1$, respectively.

Before we derive the final rule bases, we place two additional requirements on the tuner: (i) the tuning actions are monotone with respect to the inputs to the rule bases, i.e., larger inputs result in larger tuning actions, and (ii) we try to minimize the derivative mode gain, because doing so makes the PID controller more robust against measurement noise. One would perhaps suspect that the rules will, in general, be nonunique. We will observe that this is indeed the case. Furthermore, note that because the outputs share a common membership function Δ, this ensures an ordered relation, i.e., if $\Delta_{K_D} = \text{NM}$ and $\Delta_{K_C} = \text{NL}$, then $\delta_{K_D} > \delta_{K_C}$.

To illustrate how the rules are derived, we present the derivation of four rules for R_1. These correspond to $e_1(t) = \text{Z}$, $e_1(t) = \text{PS}$, $e_1(t) = \text{PM}$, and $e_1(t) = \text{PL}$.

$\mathbf{e_1(t) = Z}$:

Because the error is *zero*, clearly we need not carry out any compensation, and hence we immediately obtain $\Delta_{K_C} = \text{Z}$, $\Delta_{K_D} = \text{Z}$, and $\Delta_{K_I} = \text{Z}$.

$\mathbf{e_1(t) = PS}$:

In order to compensate for this, we require that the dominant closed-loop poles move towards the imaginary axis, and away from the real axis. We also require no change in the loop gain (N). Based on the analysis relating the controller zeros to the dominant closed-loop poles (Eqs. (12), (19), (20)), we need to move the controller zeros ($\alpha \pm j\beta$) toward the imaginary axis, and away from the real axis, i.e., we need $\Delta\alpha > 0$ and $\Delta\beta > 0$. From Eq. (22), we observe that to keep the loop gain constant, we require that δ_{K_D} equal 0. Furthermore, from Eqs. (26) and (27), we observe that to obtain $\Delta\alpha > 0$ and $\Delta\beta > 0$, we need $\delta_{K_C} < 0$ and $\delta_{K_I} > 0$. One possible choice that results in this is $\Delta_{K_D} = 0$, $\Delta_{K_C} = \text{NS}$, and $\Delta_{K_I} = \text{PS}$. Clearly, this choice is nonunique. However, we will see later on that the requirement for monotonicity will automatically preclude certain rule sets from occurring.

$\mathbf{e_1(t) = PM}$:

Here again, we require $\Delta N \approx 0$, $\Delta\alpha > 0$, and $\Delta\beta > 0$. Moreover, we need to move the controller zeros by at least as much as in the $e_1(t) = \text{PS}$ case. Hence, one choice of the tuning actions will be $\Delta_{K_D} = 0$, $\Delta_{K_C} = \text{NM}$, and $\Delta_{K_I} = \text{PM}$.

$\mathbf{e_1(t) = PL}$:

Here, we need $\Delta N > 0$, $\Delta\alpha > 0$, and $\Delta\beta > 0$. From Eq. (22), we observe that to obtain $\Delta N > 0$, we need $\delta_{K_D} > 0$. We fix $\Delta_{K_D} = \text{PS}$. Now, because we need $\Delta\alpha$ and $\Delta\beta$ to be at least as large as those for the $e_1(t) = \text{PM}$ case, we choose

Δ_{K_C} = NL and Δ_{K_I} = PL. Note from Figure 3 that the output membership function Δ divides the interval [0, 1] equally, i.e., the centroids of the fuzzy classes are spaced equally apart. Hence, the values assigned to Δ_{K_C}, Δ_{K_I}, and δ_{K_D} ensures that $\Delta\alpha$ and $\Delta\beta$ for this case are equal to those obtained for $e_1(t)$ = PM.

This procedure for assigning rules may have to be carried out iteratively. For example, if we had assigned Δ_{K_C} = NL and Δ_{K_I} = PL for the case when $e_1(t)$ = PS, then we would not be able to assign any values for the $e_1(t)$ = PL case, because the monotonicity property would fail.

Carrying out a similar analysis as above, we obtain the remainder of the rule bases. These are presented in Table I for R_1, and in Table II for R_2. The number of rules on R_2 have been almost halved, by requiring that the expert take action only when the response is returning to its steady state value. The input membership functions were determined after simulation studies with a

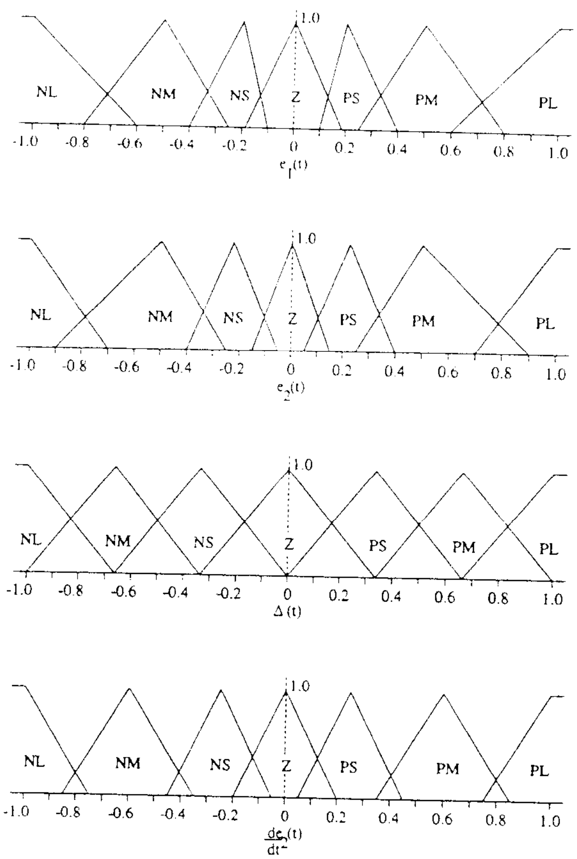


Figure 3. Fuzzy membership functions.

Table I. Rule Set R_1

$e_1(t)$	Δ_{K_C}	Δ_{K_D}	Δ_{K_I}
PL	NL	PS	PL
PM	NM	Z	PM
PS	NS	Z	PS
Z	Z	Z	Z
NS	PS	Z	NS
NM	PM	Z	NM
NL	PL	NS	NL

Table II. Rule Set R_2

Δ_{K_C}

$e_2(t)$ \ $\frac{de_2(t)}{dt}$	PL	PM	PS	Z	NS	NM	NL
PL	Z	Z	Z	Z	PS	PS	PM
PM	Z	Z	Z	Z	PS	PM	PM
PS	Z	Z	Z	NS	PS	PM	PM
Z	Z	Z	Z	Z	Z	Z	Z
NS	PM	PM	PS	NS	Z	Z	Z
NM	PM	PM	PS	Z	Z	Z	Z
NL	PM	PS	PS	Z	Z	Z	Z

Δ_{K_D}

$e_2(t)$ \ $\frac{de_2(t)}{dt}$	PL	PM	PS	Z	NS	NM	NL
PL	Z	Z	Z	Z	NM	NM	NL
PM	Z	Z	Z	Z	NM	NL	NL
PS	Z	Z	Z	PS	NM	NL	NL
Z	Z	Z	Z	Z	Z	Z	Z
NS	NL	NL	NM	PS	Z	Z	Z
NM	NL	NL	NM	Z	Z	Z	Z
NL	NL	NM	NM	Z	Z	Z	Z

Δ_{K_I}

$e_2(t)$ \ $\frac{de_2(t)}{dt}$	PL	PM	PS	Z	NS	NM	NL
PL	Z	Z	Z	Z	NS	NS	NM
PM	Z	Z	Z	Z	NS	NM	NM
PS	Z	Z	Z	PS	NS	NM	NM
Z	Z	Z	Z	Z	Z	Z	Z
NS	NM	NM	NS	PS	Z	Z	Z
NM	NM	NM	NS	Z	Z	Z	Z
NL	NM	NS	NS	Z	Z	Z	Z

second-order plant. These are shown in Figure 3. It should be noted that the abscissas of the membership functions have been normalized to lie between −1 and +1. Hence, the inputs to the fuzzy logic controller should be scaled accordingly. Suggested scaling factors are 1 for $e_1(t)$, $0.2/(s_{new} - s_{old})$ for $e_2(t)$, and $0.2/T_2$ for $d/dt\, e_2(t)$. However, these values are not binding.

Applications

To verify the validity of the rule base, we present three applications.

Third-Order Plant

Here, we consider the application of the expert to a simple third-order plant. The transfer function of the system is given by

$$P(s) = \frac{Y(s)}{U(s)} = \frac{0.0301}{(s + 0.01)(s^2 + 10s + 41)}. \quad (28)$$

The reference response has $T_1 = 20$ sec, $T_2 = 250$ sec, and is shown in Figure 4 (top). A step

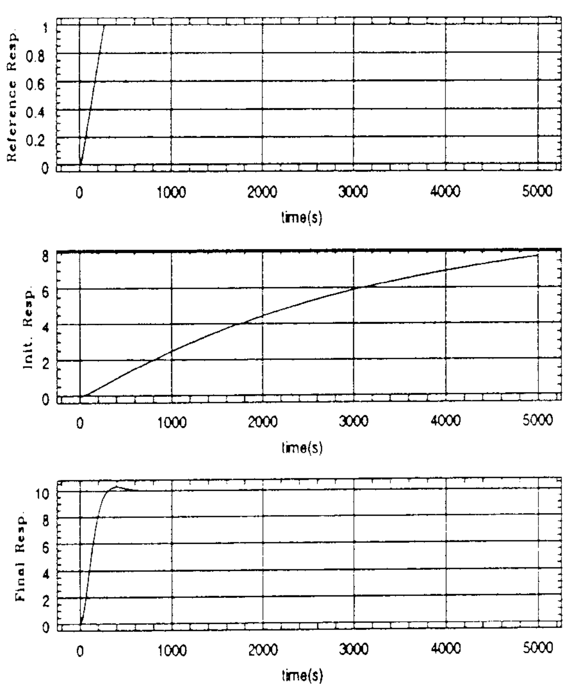


Figure 4. Third order plant: reference response (top), initial response (middle), final response (bottom).

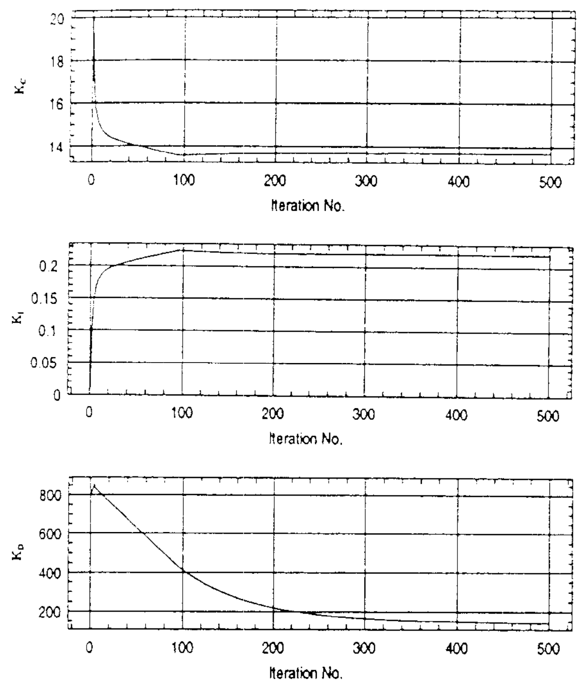


Figure 5. Third order plant: steady state gains versus iteration number.

change from 0 to 10 is chosen. The initial PID gains yield a very large rise time (Fig. 4 (middle)). Figure 4 (bottom) shows the response after the expert has tuned the gains. The variation in the steady state gains is illustrated in Figure 5.

Separator Temperature Control

The separator is part of a larger plant (Tennessee Eastman Test Problem (Vogel and Downs, 1990)) comprised of a reactor, condensor, separator, stripper, and recycle compressor. Fig-

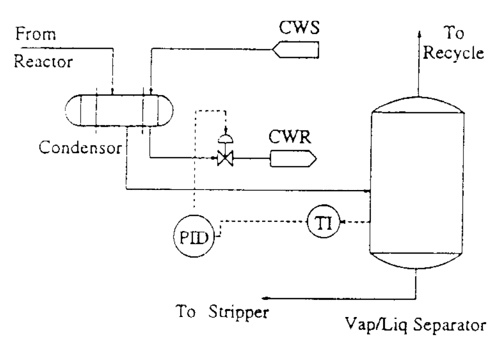


Figure 6. Separator: chosen control loop.

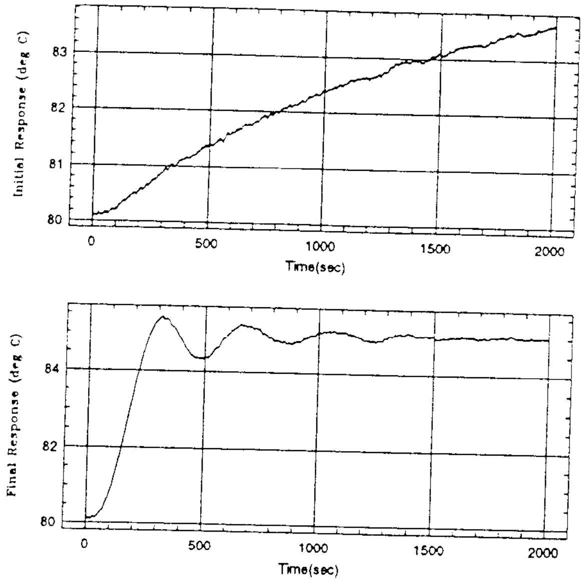


Figure 7. Separator: very slow initial response (top), final response (bottom).

ure 6 presents a simplified view of the portion of the plant of interest to us, illustrating the control loop. We note that (i) there is dead time present, (ii) the plant has an unknown number of poles, (iii) there is measurement noise, and (iv) there are restrictions on the manipulated variable, i.e., the condensor cooling water flow valve. The reference response has $T_1 = 80$ sec, and $T_2 =$

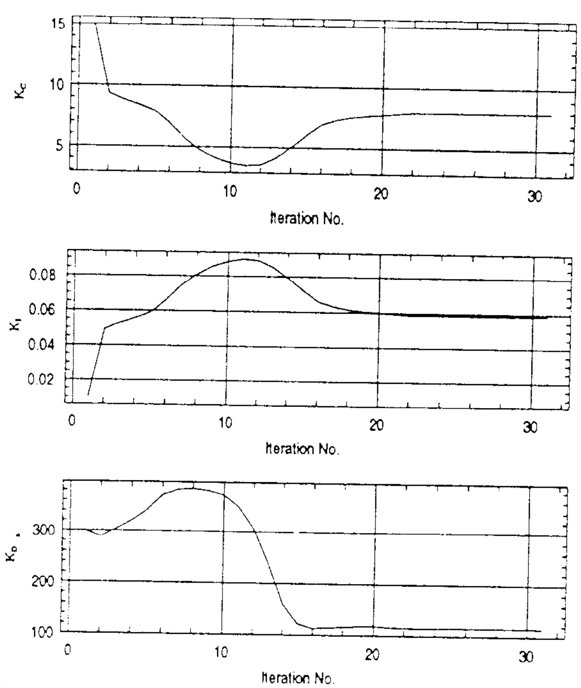


Figure 8. Separator (initially slow): steady state gains versus iteration number.

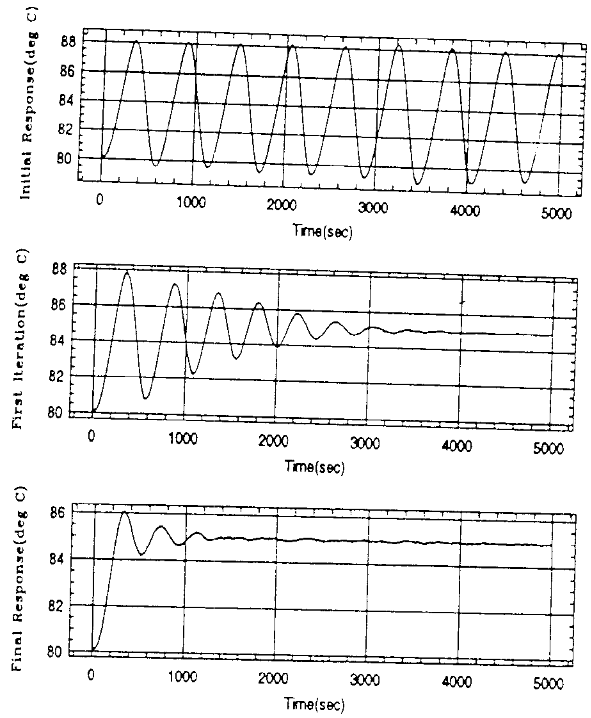


Figure 9. Separator: oscillatory initial response (top), first iteration response (middle), final response (bottom).

300 sec. A step change from 80.109 deg C to 85 deg C is chosen.

Two initial responses are considered: (i) The

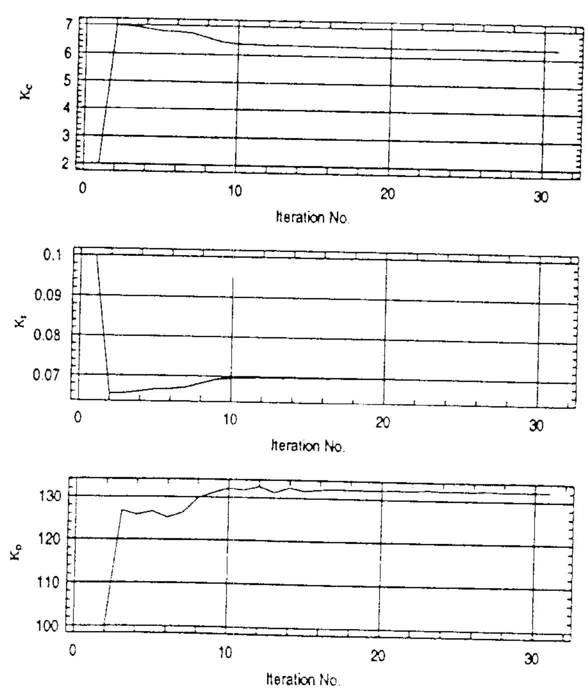


Figure 10. Separator (oscillatory): steady state gains versus iteration number.

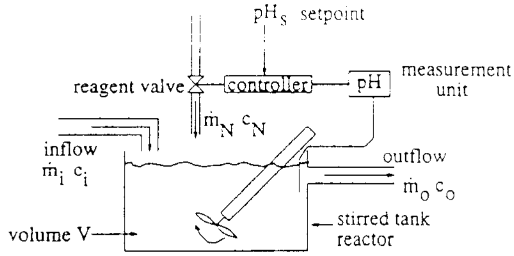


Figure 11. General layout of pH neutralization.

initial PID settings give a very large rise time, and (ii) the initial settings result in an oscillatory response. Figure 7 (top) illustrates the initial response for the case of large rise time. Figure 7 (bottom) illustrates the response after the expert has tuned the PID gains. Figure 8 shows the variation of the steady state gains from iteration to iteration. Similarly, Figure 9 (top) illustrates the initially oscillatory response. Figure 9 (middle) shows the response during the first application of the expert, and Figure 9 (bottom) shows the final response. Figure 10 shows the variation of the steady state gains from iteration to iteration.

Thus, the expert correctly tunes the gains of the PID controller. It also damps out oscillations quickly and gives rapid convergence of the gains.

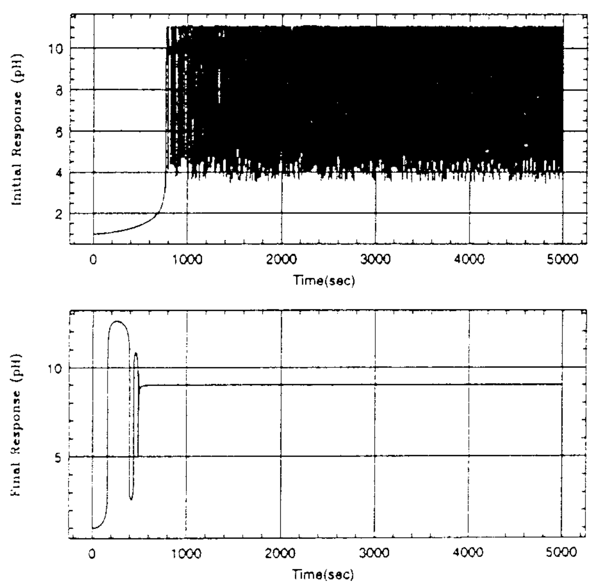


Figure 12. pH neutralization: initial response (top), final response (bottom).

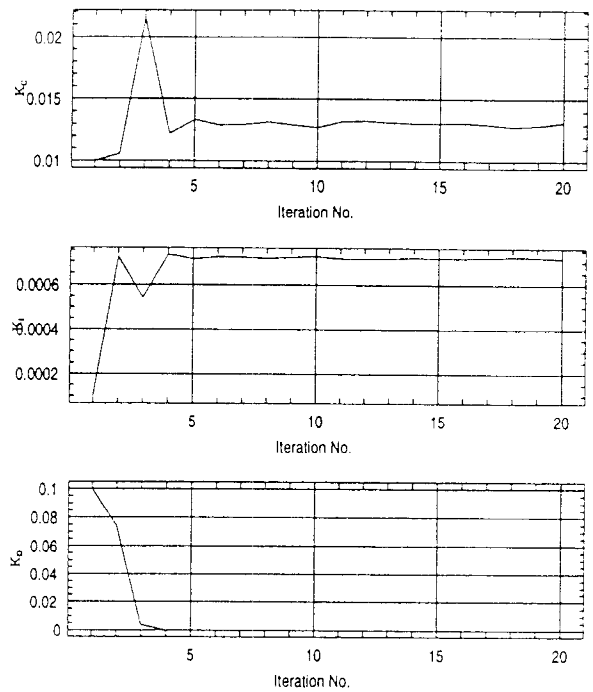


Figure 13. pH: steady state gains versus iteration number.

pH Control

We now consider an application of the expert to a plant with nonlinearity in its output. The system chosen is the one considered by Tolle and Ersü (1992). Figure 11 shows the plant and the control loop. The parameters chosen were as follows: $c_i = 10^{-3}$ mol/l; $c_N = 0.1$ mol/l; $\dot{m}_i = 50$ l/sec; $V = 5000$ l; and $\dot{m}_N = u$ is the manipulated variable, with $0 \leq u \leq 8$ l/sec.

The reference response has $T_1 = 20$ sec, and $T_2 = 200$ sec. A step change in the pH from 1 to 9 is considered. Figure 12 (top) shows the initial response, and Figure 12 (bottom) shows the response after the expert has tuned the PID gains. Figure 13 illustrates the variation in the steady state gains from iteration to iteration. Again we observe that the response is improved, and the gains converge rapidly.

Building on the Expert

The behavior of the expert may be modified to suit one's needs. We consider one such modification here.

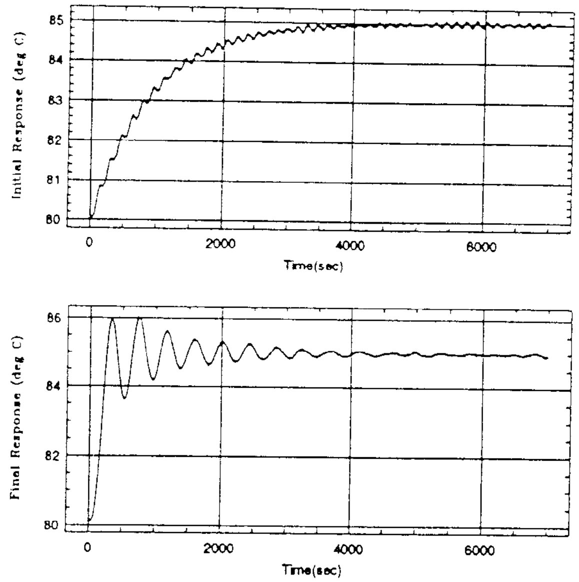


Figure 14. Separator (liquid level change): initial response (top), final response (bottom).

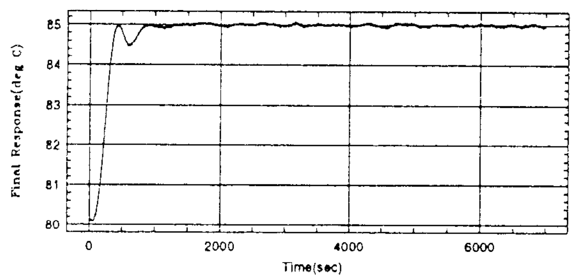


Figure 15. Separator (liquid level change): final response obtained with overshoot control.

Overshoot Control

So far, no attempt has been made to control the amount of overshoot. In fact, the rules have been developed so that there is a greater emphasis on rise time compensation. For some applications, the control of overshoot may be more important.

The overshoot control, in brief, is based upon adaptively changing the switching point between the two rule sets (R_1 and R_2) depending on the amount of overshoot. If the overshoot is large, switch to R_2 sooner, else if the overshoot is within acceptable limits, but the rise time is large, then switch to R_2 later. Furthermore, scale the inputs to R_2 so that their largest values in the previous iteration fall in the membership set *large*. This helps damp out oscillations about the steady state.

As an application, we consider separator temperature control with the liquid level in the separator increased by 20%. The reference response is not modified. Figure 14 (top) illustrates the initial response to a set-point change to 85 deg C, and Figure 14 (bottom) shows the final response obtained by the original expert. To meet the rise time requirement, the response is highly underdamped. Figure 15 illustrates the final response obtained when the expert was applied with overshoot control, with a maximum allowable overshoot of 5%. We observe that the response meets the overshoot and rise time specifications.

Conclusions

Although much work has been done in the area of tuning PID gains, most of it is based on the analysis of the plant response and on parameter estimation. In this article, we have shown that for the class of stable, dominant pole plants with large rise times, it is not necessary to carry out data analysis and parameter estimation. There are inherent properties of this class that can be exploited to design a learning controller that can learn on-line, and that displays rapid convergence. This could be thought of as being analogous to humans, who possess the ability to tune PID parameters without necessarily carrying out data analysis or identification.

The results show that the fuzzy logic rule base derived in this article performs as required. The resultant PID controllers have a small derivative mode gain. This is advantageous as it prevents measurement noise from influencing the control action. The example concerning pH control illustrates robustness of the rule base to monotone, continuous output nonlinearities. One could also consider constructing a *supervisor* to enhance the performance of the expert. Finally, results provide encouragement for the design of intelligent controllers to handle a class of systems.

This work was supported by the National Science Foundation Engineering Research Centers Program: NSFD CDR 8803012.

References

Åström KJ, McAvoy TJ (1992): "Intelligent control." *Journal of Process Control* **2**: 115–126.

Koivo HN, Tanttu JT (1991): "Tuning of PID controllers: Survey of SISO and MIMO techniques." *Proc. of IFAC Intelligent Tuning and Adaptive Control*, Singapore, pp. 75–80.

Kosko B (1990): *Neural Networks and Fuzzy Systems*, Englewood Cliffs, NJ: Prentice Hall.

Kuo BC (1991): *Automatic Control Systems, 6th ed.*, Englewood Cliffs, NJ: Prentice Hall.

Toile H, Ersü E (1992): *Neurocontrol*, New York: Springer-Verlag.

Vogel E, Downs J (1990): "A plant-wide industrial process control problem," A.I.Ch.E., Paper 24a, Chicago, IL.

Received July 1994
Accepted January 1995