Self-Tuning PID Controllers: Algorithms and Implementation

PETER J. GAWTHROP, MEMBER, IEEE

Abstract—**A class of controllers with integral action is shown to arise directly from appropriate system models. Via the zero-gain predictor approach, a corresponding class of hybrid self-tuning controllers is shown to have both integral action in the controller and offset removal in the tuning algorithm. Implementation details and some experimental results are given.**

I. Introduction

THE large number of PI (proportional + integral) and PID (proportional + integral + derivative) controllers used routinely for process control applications may be regarded as experimental evidence for their usefulness. However, the large number of such loops on a typical process precludes regular retuning to account for process changes and aging. Therefore, there seems to be a place for an algorithm which can be used to tune PI and PID controllers either on a continuous basis or when required. A number of semiautomatic (e.g., [34]) and automatic methods (e.g., [26]) have been evolved for initial tuning which involve special process perturbations to identify the system. This paper considers a self-tuning method which directly tunes the controller parameters (the controller *not* the system is identified) and does not require an initial identification phase.

As PI and PID controllers are so common, there must be something about the dynamics of many systems which makes such control appropriate. It follows that it should not be necessary to force an adaptive controller to have a PI or PID structure, but rather this structure should arise naturally from reasonable assumptions about the dynamics of the controlled process. It is shown in this paper that this is indeed so: suitable modeling of nonzero mean disturbances leads to an algorithm with integral action, and the additional assumption of a first- (second)-order system gives rise to a PI (PID) controller.

This approach of letting the integral action arise naturally from the specification of a suitable disturbance model rather than forcing integral action into the controller distinguishes the algorithms of this paper from previous methods [32], [35]. As will be shown, this approach automatically removes offsets from both the controller and the estimator.

Along with the closely related subject of parameter identification, the recent history of self-tuning control [1]–[7], [21], [23], [27], [29]–[32] has largely been concerned with discrete-time system models. As pointed out by Young [33] in the context of parameter estimation, this has been largely due to the rise of the digital computer. It is argued in this paper and elsewhere [8], [10], [13], [17] that in some ways, a continuous-time approach embedded in a discrete-time implementation has advantages, not the least of which is that controller coefficients appear in the

Manuscript received April 17, 1984; revised August 30, 1985. This paper is based on a prior submission of August 2, 1983. Paper recommended by Associate Editor, R. R. Bitmead. This work was supported by the U.K. Science and Engineering Research Council.

The author is with the School of Engineering and Applied Sciences, University of Sussex, Falmer, Brighton, England.

IEEE Log Number 8406584.

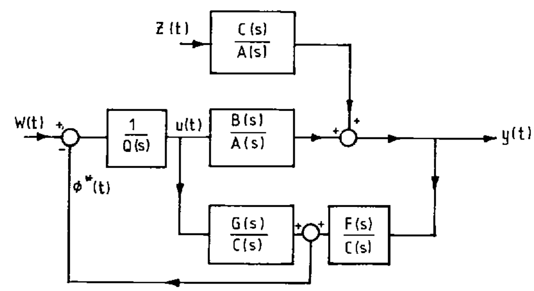


Fig. 1. The system with nonadaptive controller.

continuous-time form familiar to industrial process engineers. This is in contrast to previous nonadaptive and adaptive methods [1], [3], [35], [36] which involve estimation of discrete-time models. This paper concentrates on the details arising from applying this technique to controllers with integral action in general, and PI and PID controllers in particular.

Self-tuning methods may be divided into two groups [2]: implicit methods where the estimator directly produces controller coefficients, and explicit methods where the estimator generates system coefficients which are processed to give controller coefficients. The dividing line is somewhat blurred [12], but this paper concentrates on the implicit methods. Nevertheless, much of the paper is concerned with a nonadaptive design method (on which the implicit method is based) and this could be given an explicit adaptive implementation.

Much of this paper is concerned with the derivation and properties of the control algorithm. Although the paper focuses on tuning these control algorithms via a particular hybrid technique [10], [13], other methods of tuning continuous-time controller coefficients (using either discrete-time [8] or continuous-time methods [33]) could be used.

Some material in this paper is based on earlier work [15], [18], [19]; in particular, some precursors to the algorithms given are discussed in [18].

A. System Models

The system considered here (see Fig. 1) is a continuous-time version of the discrete-time single-input single-output system introduced, in the context of self-tuning, by Astrom and others [1], [3].

$$A(s)y(t)=B(s)u(t)+C(s)z(t). \tag{1.1.1}$$

The symbols y, u, and z are the system output, input, and disturbance, respectively. The somewhat imprecise notation that s is the Laplace operator or differential operator according to context is used. This rather obscures the role of initial conditions, but this may be included as discussed elsewhere [20]. Time delays are not considered in this paper, but small delays can be accommodated [28].

A, B, and C are polynomials in s. A and B define the system dynamics, but C is a chosen design polynomial in the algorithm to be presented; it is included in the system description for algebraic conveniences and to give compatibility with earlier work. The polynomial C is *defined* throughout to have the following properties:

$$\text{Either } \deg(C)=\deg(A)-1 \tag{1.1.2}$$

$$\text{or } \deg(C)=\deg(A) \tag{1.1.3}$$

($\deg(A)$ means the degree of the polynomial A).

There are two classes of disturbances considered in this paper: one including all bounded disturbances, and one including processes with a particular stochastic model. The system equation (1.1.1) with the former interpretation of z will be termed the *nonstochastic* framework and the latter the *stochastic* framework.

To avoid repetition, the nonstochastic framework will be treated in detail, and the appropriate modifications required for the stochastic framework will be indicated at the end of the appropriate section.

1) Stochastic Framework: The latter class of disturbances contains those stationary stochastic processes with a rational spectral density modeled by the equation:

$$z(t)=\frac{D(s)}{C(s)}\psi(t) \tag{1.1.4}$$

where $\psi(t)$ is a "white" noise stochastic process and

$$\deg(D)=\deg(C). \tag{1.1.5}$$

Thus, either the disturbance is bounded, but differentiation produces a white noise component, or the disturbance has a white noise component. Either assumption is practically reasonable and leads to controllers which do not require (nonphysically realizable) differentiators for their implementation. This stochastic framework corresponds to that usually used in discrete-time self-tuning control [1], [4], [6].

From (1.1.5), it follows that the the stochastic framework leads to a special case of the nonstochastic framework with $D(s)$ replacing $C(s)$ and ψ replacing z.

B. Model-Reference Controllers

The aim of a model reference controller is to give a closed-loop response to setpoint changes which corresponds to a given model, which in this paper is taken to be an all-pole model of the form $1/P(s)$ where $P(s)$ is a polynomial in s and $P(0) = 1$ for unit steady-state gain. (P can also be a transfer function, but this is not pursued here.) For this to be possible using proper (differentiator-free) feedback, the least possible degree of P is given by

$$\deg(P)=\deg(A)-\deg(B). \tag{1.2.1}$$

As is noted elsewhere [11], [17], model reference controllers may be obtained by replacing the unrealizable quantity

$$\phi(t)=P(s)y(t) \tag{1.2.2}$$

by a corresponding prediction $\phi^*(t)$ with a feedback loop. (Here, the term *predictor* is used in the generalized sense of [17] to imply the estimation of unrealizable quantities using proper transfer functions. In this paper these unrealizable quantities are derivatives.) Least-squares predictors combined with least-squares estimation give self-tuning predictors which, when put into a feedback loop, give self-tuning model reference controllers. The basic equations of the model reference control laws required in the sequel are now listed; further details may be found elsewhere [10], [13].

With the nonstochastic framework, the polynomials F and E are defined by the polynomial identity

$$\frac{P(s)C(s)}{A(s)}=E(s)+\frac{F(s)}{A(s)} \tag{1.2.3}$$

$$\deg(F)<\deg(A)$$

thus $F(s)/A(s)$ is the proper part of $P(s)C(s)/A(s)$ and $E(s)$ is the remainder. A predictor may now be defined as

$$\phi^*(t)=\frac{G(s)}{C(s)}u(t)+\frac{F(s)}{C(s)}y(t) \tag{1.2.4}$$

where

$$G(s)=E(s)B(s).$$

Because of the way F is defined, F/C is *proper* (finite gain at infinite frequency); because of the choice of the degree of P (1.2.1)

$$\deg(G)=\deg(C)=\deg(A)-1 \tag{1.2.5}$$

and so G/C is also *proper*. Hence, ϕ^* may be regarded as a realizable approximation to the result of passing y through the nonrealizable transfer function $P(s)$. The prediction error $e(t)$ is defined as

$$e(t)=\phi(t)-\phi^*(t)(=Ez(t)). \tag{1.2.6}$$

A model-reference controller is obtained [13], [17] by setting

$$\phi^*(t)=w(t) \tag{1.2.7}$$

where w is a desired value, or setpoint. Using (1.2.4) an explicit expression for the control signal is

$$u(t)=\frac{1}{EB}(Cw(t)-Fy(t)). \tag{1.2.8}$$

This gives the closed-loop system

$$y(t)=\frac{1}{P(s)}w(t)+\frac{E(s)}{P(s)}z(t). \tag{1.2.9}$$

This closed-loop system responds to the setpoint according to the chosen model $1/P(s)$ and to the disturbance according to a transfer function whose poles are the roots of P. Hence, the name "model-reference control." The signal $w(t)$ may also be regarded as the output of a filter with transfer function $R(s)$ driven by a setpoint w'

$$w(t)=R(s)w'(t). \tag{1.2.10}$$

The transfer function R thus allows separate specification of setpoint and disturbance response.

1) Stochastic Framework: As discussed elsewhere [13], [17], the corresponding result for the stochastic framework is obtained by setting $C(s) = D(s)$ in the preceding equations to give the "optimal" predictor.

The distinction between nonstochastic and stochastic results is directly related to that between the steady-state Kalman filter with poles prescribed by optimality considerations, and observers with poles chosen by the designer.

C. Detuned Model-Reference Controllers

As is well known, the model-reference method is not applicable to systems with zeros with nonnegative real part. In addition, the model-reference method is sensitive to errors in the chosen model order (1.2.1). These problems may be overcome to some extent by using a *detuned* version of this control given by

$$\phi^*(t)+Q(s)u(t)=w(t) \tag{1.3.1}$$

where Q is a transfer function representing a detuning factor. This gives a closed-loop system

$$y(t)=\frac{B(s)}{P(s)B(s)+Q(s)A(s)}w(t)+\frac{E(s)B(s)+Q(s)C(s)}{P(s)B(s)+Q(s)A(s)}z(t). \tag{1.3.2}$$

Typically, $Q(s)$ is chosen to be small at low frequencies (i.e., for small ω when $s = j\omega$) and large at high frequencies; thus the the model reference objective is satisfied at low frequencies, but not at high frequencies.

The detuned model reference approach avoids the unnecessary and dangerous propensity of the model-reference adaptive controller to try to match the reference model at high frequencies.

The use of detuning in the context of PID control is discussed further in Section IV-C.

D. The Offset Problem

An important property of feedback controllers, adaptive or nonadaptive, is that the controller should eventually force the system output to follow a constant reference level with zero average error. This average steady-state error is termed the offset; the elimination of this quantity constitutes the offset problem.

The offset problem may conveniently be divided into two main subproblems: offsets due to the choice of controller design parameters (P, Q), and offset due to a nonzero mean disturbance $z(t)$.

The latter problem is overlooked in many papers: see [7] for a critique of some unsuccessful approaches to this problem. The solution of this problem via the *zero-gain* predictor is the main purpose of this paper.

The first problem is readily overcome. Constraining the design parameters by

$$P(0)=1;\; Q(0)=0 \tag{1.4.1}$$

has the desired effect [18]. Thus, for example, P and Q could be of the form

$$P(s)=1+sP_0(s) \tag{1.4.2}$$

$$Q(s)=\frac{s}{(s+q)}. \tag{1.4.3}$$

An alternative method, due to Allidina and Hughes [8], involves tuning a setpoint gain factor.

The second subproblem, concerning nonzero mean disturbances, is more challenging that the first. It may be divided into two subproblems: the controller to which the adaptive algorithm eventually tunes must be capable of eliminating offset, and the estimation algorithm must be able to cope with the nonzero mean disturbance. Any proposed solution to the offset problem should be examined from both these aspects.

There are at least four main approaches to the elimination of offsets due to nonzero mean disturbances to be found in the literature: estimate the value of the mean of the disturbance and use this in the control law; allow the estimator to adjust the steady-state controller gain to eliminate the offset at a particular setpoint; embed the adaptive controller in a loop involving an integrator; or endow the controller with integral action. All these methods have been discussed previously by Wittenmark and Astrom [31], [32] and by Clarke [4].

This paper presents new methods based on adaptive controllers incorporating integral action in both the controller and tuning components.

II. Parameter Estimation

This section reviews some prediction error estimation schemes with particular reference to the offset problem. Other classes of estimation schemes could also be used here; but the main point is that zero-mean data should be used for good estimation performance.

There are two main classes of estimation schemes generating coefficients of predictors which will be considered: those which estimate the predictor denominator polynomial, and thus vary the location of the predictor poles, and those which do not. In this paper, the former will be called *moving poles* methods, and the latter *fixed pole* methods.

The latter are appropriate to the nonstochastic framework and will be considered first. In each case, the crucial step is to obtain a linear-in-the-parameters representation of the predictor.

A. Estimation with Fixed Poles (Nonstochastic Framework)

The use of the fixed pole estimators does not imply any stochastic assumption on the noise structure: the predictor denominator C is rather regarded as a design parameter (in a similar way to P) which determines the response of the closed-loop control system to the disturbance z. Clearly, due to the absence of stochastic assumptions about the disturbance, no stochastic convergence results are applicable. Nevertheless, it would seem that robustness and stability results are applicable (this is the subject of current research), thus, this class of schemes may be a more realistic option.

As C is a chosen design polynomial which is not estimated, (1.2.4) is a linear-in-the-parameters (the coefficients of F and G). In particular, this equation may be written in the linear-in-parameters form

$$\phi^*(t)=X^T(t)\boldsymbol{\theta} \tag{2.1.1}$$

where

$$X^y(t)=\frac{1}{C(s)}[1,\; s,\; \cdots,\; s^n]^T y(t) \tag{2.1.2}$$

$$X^u(t)=\frac{1}{C(s)}[1,\; s,\; \cdots,\; s^n]^T u(t)$$

$$X=\begin{bmatrix} X^u \\ X^y \end{bmatrix};\; \boldsymbol{\theta}=\begin{bmatrix} \boldsymbol{\theta}^u \\ \boldsymbol{\theta}^y \end{bmatrix} \tag{2.1.3}$$

and $\boldsymbol{\theta}^y$ and $\boldsymbol{\theta}^u$ contain the polynomial coefficients corresponding to X^y and X^u.

X^y and X^u may be generated using a state-variable filter in the phase-variable canonical form, normally using a discrete-time approximation. See Section V for implementation details.

A problem here is that ϕ involves derivatives due to the polynomial P and so, cannot be used directly in an estimation scheme. As discussed elsewhere [13], [17] this can be overcome using a *hybrid* approach where ϕ is replaced by a (realizable) filtered approximation.

$$\bar{\phi}(t)=L(s)\phi(t) \tag{2.1.4}$$

where $L(s)$ is a low-pass filter such that $L(s)P(s)$ is proper. The details of how this approximation affects the estimator properties is considered elsewhere [10], [13]. Implementation aspects are covered in Section V.

Equations (2.1.1)–(2.1.8) form the basis of a prediction error estimation scheme which estimates the elements of $\boldsymbol{\theta}$ using the measurements X and $\bar{\phi}$. For example, the well-known recursive least-squares algorithm [3], [4] can be used. This is given by

$$\hat{e}(i)=\bar{\phi}(ih)-X(ih-h)^T\hat{\boldsymbol{\theta}}(i-1) \tag{2.1.5}$$

$$\hat{\boldsymbol{\theta}}(i)=\hat{\boldsymbol{\theta}}(i-1)+S^{-1}(i-1)X(ih-h)e(i) \tag{2.1.6}$$

$$S(i)=S(i)+X(ih)X^T(ih) \tag{2.1.7}$$

where h is the estimator sample interval. Usually, S^{-1} rather than S is updated; there are a number of implementation details such as the use of numerically stable methods such as UDU factorization and exponential forgetting which can be found in, for example, [4].

B. Estimation with Moving Poles (Stochastic Framework)

The use of moving pole methods involves the assumption that the disturbance z has a rational spectral density implying the existence of an optimal predictor denominator polynomial $D(s)$ whose coefficients may be determined by the estimator. Extended least-squares [3], [4] is an example of such a method. If this assumption is true, then such methods potentially have stochastic convergence properties. It is, however, probably true to say that such models are used more for analytical convenience rather than realism. The corresponding prediction error algorithm must be capable of estimating the noise numerator (and optimal predictor denominator) D.

Choosing a polynomial C, the optimal predictor obtained from (1.2.4) by replacing C and D can be rewritten as

$$\frac{D(s)}{C(s)}\phi^* = \frac{F(s)}{C(s)}y + \frac{G(s)}{C(s)}u \tag{2.2.1}$$

or

$$\phi^* = \frac{F(s)}{C(s)}y + \frac{G(s)}{C(s)}u + \left(1 - \frac{D(s)}{C(s)}\right)\phi^*. \tag{2.2.2}$$

Without loss of generality, the disturbance may be rescaled such that

$$d_n = c_n \tag{2.2.3}$$

where d_n and c_n are the coefficients of the highest degree terms of D and C, respectively. The vector X_ϕ can now be defined by

$$X_\phi = \frac{1}{C(s)}[1, \cdots, s^{n-1}]^T\phi^*; \ \boldsymbol{\theta}_\phi = [c_0 - d_0, \cdots, c_{n-1} - d_{n-1}]^T. \tag{2.2.4}$$

Hence,

$$\phi^* = X'^T\boldsymbol{\theta}' \tag{2.2.5}$$

where

$$X' = \begin{bmatrix} X \\ X_\phi \end{bmatrix}; \ \boldsymbol{\theta}' = \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{\theta}_\phi \end{bmatrix}. \tag{2.2.6}$$

This model is linear in the parameters, but ϕ^* is unknown. Extended least-squares may be applied by replacing the unknown ϕ^* by the corresponding estimate generated by the estimated predictor.

C. Why Zero-Mean Data are Important

If the estimation problem is given a stochastic framework, it is well known that satisfactory estimation (e.g., unbiased estimation) by such methods requires that X and e are uncorrelated

$$E\{Xe\} = 0. \tag{2.3.1}$$

This condition may be rewritten as

$$E\{X\}E\{e\} + E\{(X - E\{X\})(e - E\{e\})\} = 0. \tag{2.3.2}$$

The two components of this expression represent correlation due to nonzero means, and correlation due to cross covariance. The fixed pole algorithm cannot eliminate the second term as $z(t)$ is not necessarily white, but the first term can be a far more important factor if not properly dealt with.

The first term disappears if either $E\{e\} = 0$ or $E\{X\} = 0$ (or both). It is usual to concentrate on setting $E\{e\} = 0$, possibly by estimating a bias.

Without imposing a stochastic framework, it is not possible to use these ideas directly; indeed, in the adaptive control context, X is not stationary and so the operations of expectation and averaging are not equivalent. However, they may be used as the basis of a heuristic argument to derive a criterion for judging how appropriate a given predictor structure will be when combined with a prediction error algorithm in the presence of nonzero mean noise. Suppose that, over some time period, e had a small, but nonzero, average value; as the product of X and e is important, this would not, by itself, imply satisfactory estimation. However, if the underlying predictor is designed so that both e and X have small (with respect to deviations) average values, then the product of these averages will also be small.

Thus, it is proposed that a requirement for satisfactory estimation is that the predictor structure should be such that *both* e and X have small average values. This criterion will be used in the sequel.

These ideas have been experimentally verified by Hodgson [22].

III. Integral Action

Controllers with integral action are commonly used in industry; so a number of authors [7], [31] have introduced self-tuning controllers with integral action. The usual approach taken is to force an integrator into the controller and so eliminate offsets. The approach taken here is the reverse: the system is modeled to include an offset term; and the corresponding controller is found to have integral action. Moreover, the corresponding predictor is found to have the desirable zero-mean data properties mentioned in Section II-C.

A. Modified System Model

Disturbances affecting real systems do not usually have a zero mean; the disturbance may drift with time. A simple way of modeling such a process is by the equations

$$z_1(t) = \frac{1+cs}{s}z(t) \tag{3.1.1}$$

$$A_1(s)y(t) = B_1(s)u(t) + C_1(s)z_1(t) \tag{3.1.2}$$

where $z(t)$ is a *zero-mean* disturbance. This corresponds to the system model (1.1.1) if

$$A(s) = sA_1(s); \ B(s) = sB_1(s); \ C(s) = (1+cs)C_1(s). \tag{3.1.3}$$

In other words, the presence of nonzero mean disturbances may be modeled by postulating a system integrator which is controllable with respect to the disturbance, but uncontrollable with respect to the control input u.

In some ways, this approach is similar to that used in discrete time by Palmor and Shinnar [36].

A more complex disturbance model could be used, but the crucial point is that the integrator in the disturbance model implies that

$$A(0) = B(0) = 0; \ C(0) \neq 0. \tag{3.1.4}$$

B. The Zero Gain Predictor

The idea behind using the predictor developed in Section I-B is to replace the physically unrealizable derivatives implicit in $P(s)y$ by a derivative-free predictor. As, by assumption, $P(0) = 1$, $P(s)$ can be rewritten (1.4.2) as

$$P(s) = 1 + sP_0(s) \tag{3.2.1}$$

where

$$P_0(s) = p_1 + p_2 s + \cdots + p_n s^{n-1}; \ n = \deg(P). \tag{3.2.2}$$

Hence, the unrealizable quantity ϕ becomes

$$\phi(t) = y(t) + \phi_0(t) \tag{3.2.3}$$

where

$$\phi_0 = sP_0(s)y(t). \tag{3.2.4}$$

It is the term ϕ_0 that is unrealizable and needs to be replaced by a predictor, $y(t)$ can be used directly in the control law.

Following Section I-B, but replacing $P(s)$ by $sP_0(s)$, the predictor polynomial identity becomes

$$\frac{(P(s)-1)C(s)}{A(s)} = \frac{sP_0(s)C(s)}{A(s)} = E'(s) + \frac{F'(s)}{A(s)}. \tag{3.2.5}$$

As $A(0) = 0$ (3.1.4), it follows that $F'(0) = 0$ and thus

$$F'(s) = sF_0(s)$$

where

$$F_0(s) = f_1 + f_2 s + \cdots + f_m s^{m-1}; \ m = \deg(C). \tag{3.2.6}$$

In addition, defining

$$G' = B(s)E'(s) \tag{3.2.7}$$

and noting that $B(0) = 0$ (3.1.4) it follows that $G'(s)$ is of the form

$$G'(s) = sG_0(s) \tag{3.2.8}$$

where

$$G_0(s) = g_1 + g_2 s + \cdots + g_m s^{m-1}.$$

Hence, the predictor for ϕ_0 is given by

$$\phi_0^* = \frac{sF_0(s)}{C(s)} y(t) + \frac{sG_0(s)}{C(s)} u(t). \tag{3.2.9}$$

This is the equation of the *zero-gain predictor*; each of the two filters in the predictor has zero steady-state gain. The corresponding prediction error is

$$e(t) = E'(s)z(t). \tag{3.2.10}$$

As, by definition (3.1.1) $z(t)$ is zero mean, e is also zero mean.

C. Integral Action

The control law equation (1.2.7) may be rewritten in terms of the zero-gain predictor as

$$\phi_0^*(t) = w(t) - y(t). \tag{3.3.1}$$

Using the explicit expression for the zero-gain predictor, the control law becomes

$$u(t) = \frac{C(s)}{G_0(s)} \left[\frac{w(t) - y(t)}{s} - \frac{F_0(s)}{C(s)} y(t) \right]. \tag{3.3.2}$$

This controller has integral action $(w - y)/s$ and so ensures elimination of offset whether or not the controller is being tuned. It will thus be used as the basis for the PI and PID controllers examined in the sequel.

1) Stochastic Framework: The corresponding results within the stochastic framework are obtained by replacing $C(s)$ by the optimal value $D(s)$.

D. Parameter Estimation

As has been shown in the previous subsection, the zero-gain predictor leads to controllers with integral action. In this section, the equally important fact that the zero-gain predictor leads to estimation based on zero-mean data is demonstrated.

As in Section II-A, the zero-gain predictor can be rewritten as

$$\phi_0^* = X_0^T \boldsymbol{\theta}_0 \tag{3.4.1}$$

where X_0 is the same as X, but with the elements $y/C(s)$ and $u/C(s)$ deleted

$$X_0 = \begin{bmatrix} X_0^u \\ X_0^y \end{bmatrix} \tag{3.4.2}$$

$$X_0^u(t) = \frac{1}{C(s)} [s, \cdots, s^m]^T u(t)$$

$$X_0^y(t) = \frac{1}{C(s)} [s, \cdots, s^m]^T y(t) \tag{3.4.3}$$

and $\boldsymbol{\theta}$ is given by

$$\boldsymbol{\theta}_0 = \begin{vmatrix} \boldsymbol{\theta}_0^u \\ \boldsymbol{\theta}_0^y \end{vmatrix} \tag{3.4.4}$$

where

$$\boldsymbol{\theta}_0^u = [g_1, \cdots, g_n]^T$$

$$\boldsymbol{\theta}_0^y = [f_1, \cdots, f_n]^T. \tag{3.4.5}$$

Each element of X_0 is generated by passing y or u through a filter of the form $s^j/C(s)$ where $j \geq 1$; such a filter blocks any constant component in the signals y and u and so average values are removed. The zero-gain predictor thus satisfies the criteria for satisfactory estimation set forth in (2.3).

1) Stochastic Framework: However, because D is unknown in the self-tuning context, it must be estimated using the moving pole methods of Section II-B. The ϕ^* elements in X_0' are zero-mean by definition, so the effect of offset on estimation is removed in this case as well.

E. An Alternative Zero-Gain Predictor

Another form of zero-gain predictor is derived in this subsection. It has the advantage that its coefficients relate more closely to those of PI and PID controllers than those of the zero-gain predictor of Section III-B. The disadvantage is that it is only appropriate to the nonstochastic case.

In the nonstochastic case, $C(s)$ may be chosen so that $C(0) = 1$ and so

$$C(s) = 1 + sC_0(s). \tag{3.5.1}$$

Adding $sC_0(s)/Cy$ to each side of (3.2.9)

$$\phi_0^{c*} = \frac{sF_0^c(s)}{C(s)} y(t) + \frac{sG_0(s)}{C(s)} u(t) \tag{3.5.2}$$

where

$$\phi_0^{c*} = \phi_0 + \frac{sC_0(s)}{C(s)} y(t) \tag{3.5.3}$$

and

$$F_0^c(s) = F_0(s) + C_0(s) \tag{3.5.4}$$

$$G_0^c(s) = G_0(s).$$

ϕ_0^{c*} can be regarded as the prediction of

$$\phi^c(t) = \phi_0 + \frac{sC_0(s)}{C(s)} y(t) = \phi(t) - \frac{1}{C(s)} y(t). \tag{3.5.5}$$

This alternative predictor formulation enjoys the same zero-gain properties as the predictor of Section III-B. In particular, it gives an identical integral action controller reparameterized as

$$\phi_0^{c*}(t) = w(t) - \frac{1}{C(s)} y(t). \tag{3.5.6}$$

Using the explicit expression for the alternative zero-gain predictor, the control law becomes

$$u(t) = \frac{1}{G_0(s)} \left[\frac{w(t) - y(t)}{s} + C_0(s)w(t) - F_0^c(s)y(t) \right]. \tag{3.5.7}$$

For the purposes of estimation, the alternative zero-gain predictor can be rewritten as

$$\phi_0^{c*} = X_0^T \boldsymbol{\theta}_0^c \tag{3.5.8}$$

where the data vector X_0 is as before and the parameter vector $\boldsymbol{\theta}_0^c$ is as before except that f_i is replaced by $f_i^c = f_i + c_i$.

Using (3.5.5), $G_0^c(s)$ and $F_0^c(s)$ can be obtained from a polynomial identity similar to that in Section III-B (3.2.5)

$$\frac{(P(s) - 1/C(s))C(s)}{A(s)} = \frac{P(s)C(s) - 1}{A(s)} = E'^c(s) + \frac{F'^c(s)}{A(s)} \tag{3.5.9}$$

$$sG_0^c(s) = sG_0(s) = E'^c(s)B(s)$$

$$= E'(s)B(s);\ sF_0^c(s) = F'^c. \tag{3.5.10}$$

The alternative predictor cannot be used in the stochastic case as the optimal value (D) of C is unknown *a priori*.

IV. PI and PID Controllers

The controllers discussed in the previous section had the common characteristic of integral action, and thus provide a framework for generating PI and PID controllers. PI and PID controllers operate successfully on a restricted class of systems; it is thus not surprising that by restricting the system polynomials A and B to be of a certain form, PI and PID controllers naturally arise. In particular, the assumption (3.3) that $A(s) = sA_1(s)$; $B(s) = sB_1(s)$; $C(s) = (1 + cs)C_1(s)$ combined with

$$\deg(A_1) = 1;\ \deg(B_1) = 0 \tag{4.1}$$

gives a PI controller, and combined with

$$\deg(A_1) = 2;\ \deg(B_1) = 0 \tag{4.2}$$

gives a PID controller. These assumptions, coupled with the self-tuning algorithm based on the zero-gain predictor, give the PI and PID algorithms considered in the rest of this section. The nonstochastic framework will be used here, the formulas for the stochastic framework may be obtained as described at the end of Section III.

The alternative zero-gain predictor is used in this section, as its coefficients are more closely related to PI and PID controller coefficients. But the more general zero gain predictor could be equally well used with some change of notation.

A. PI Controllers

The assumptions of (4.1) combined with the predictor structure of Section I-B give $C(s)$; $E'(s)$ and $F_0^c(s)$ and $G_0(s)$ as

$$C(s) = 1 + cs;\ E'(s) = e;\ F_0^c(s) = f^c;\ G_0(s) = eb \tag{4.1.1}$$

where b, c, e, and f are real numbers. The control law based on the alternative zero-gain predictor (3.5.7) then becomes

$$u(t) = \frac{1}{eb}\left[\frac{w(t) - y(t)}{s} + cw(t) - f^c y(t)\right]. \tag{4.1.2}$$

This may be compared to a PI controller written as

$$u(t) = k\left[\frac{(w(t) - y(t))}{T_i s} + (r_p w(t) - y(t))\right] \tag{4.1.3}$$

where k is the proportional gain on the output, kr_p the proportional gain on the setpoint, and T_i the integral time constant. In typical industrial controllers, r_p is set either to zero or one, the former giving no "proportional kick." The controller derived here makes use of this term to give the desired setpoint response. The consequences of setting r_p to zero are discussed elsewhere [19]. The PI controller is identical to the predictive controller if

$$T_i = f^c;\ k = \frac{f^c}{eb};\ r_p = \frac{c}{f^c}. \tag{4.1.4}$$

The estimated coefficients ($g = eb$, and f^c) are closely related to the PI coefficients, (k, T_i), and are thus readily interpreted. As this is a model-reference based method, the adaptive algorithm attempts to tune the PI controller to give a closed-loop response to setpoint of the form $1/(1 + ps)$ where p is the desired closed-loop time constant specified by the user.

B. PID Controllers

Assuming that the system structure is given by (4.2), it follows that B is as for the PI controller and C, $E'(s)$, $F_0^c(s)$, and $G_0(s)$ are of the form

$$C(s) = 1 + c_1 s + c_2 s^2 \tag{4.2.1}$$

$$E'(s) = e_1 + e_2 s \tag{4.2.2}$$

$$F_0^c(s) = f_1^c + f_2^c s \tag{4.2.3}$$

$$G_0(s) = be_1 + be_2 s. \tag{4.2.4}$$

The control law (3.3.2) then becomes

$$u(t) = \frac{1}{(e_1 + e_2 s)b}\left[\frac{w(t) - y(t)}{s} + c_1 w(t) - f_1^c y(t) + c_2 w(t) - f_2^c y(t)\right]. \tag{4.2.5}$$

A PID controller may be written as

$$u(t) = \frac{k}{1 + T_f s}\left[\frac{(w(t) - y(t))}{T_i s} + (r_p w(t) - y(t))\right] + (r_d w - y)T_d s \tag{4.2.6}$$

where k is the proportional gain, T_f the time constant of a filter to avoid derivative action at high frequencies, and T_i and T_d the integral and derivative time constants, respectively. Once again, r_p and r_d are gain factors to control the setpoint proportional and derivative kick. These quantities are related to estimated parameters by

$$T_i = f_1^c;\ T_d = \frac{f_2^c}{f_1^c};\ T_f = \frac{e_1}{e_0} \tag{4.2.7}$$

$$g = \frac{f_1^c}{be_0};\ r_p = \frac{c_1}{f_1^c};\ r_d = \frac{c_2}{f_2^c}. \tag{4.2.8}$$

As in the PI controller, the PID coefficients are closely related to estimated parameters. Once again, the PID controller is a model-reference controller which attempts to give a closed-loop setpoint response corresponding to the chosen model $1/1 + p_1 s + p_2 s^2$. The two coefficients p_1 and p_2 allow the user to specify desired response time and damping.

C. Detuned PI and PID Controllers

The detuned version of the model-reference controller (Section I-C) may also give rise to PI and PID algorithms. The PID version is discussed here, the PI version may be deduced in a similar fashion.

Together with the system structure of Section IV-B, choose a detuning transfer function Q to be of the form

$$Q(s) = \frac{q_1 s + q_2 s^2}{C(s)}. \tag{4.3.1}$$

This choice satisfies the no offset criterion ($Q(0) = 0$) of (1.4.1). The detuned model reference control law (1.3.1) rewritten in the form of (3.5.7) then becomes

$$\frac{(be_1 + q_1)s + (be_2 + q_2)s^2}{C(s)}u(t) + \frac{f_{1s}^c + f_2^c s^2}{C(s)}y(t) + \frac{y(t)}{C(s)} = w(t). \tag{4.3.2}$$

This is still of the PID form of (4.2.4). The corresponding PID terms thus arise from (4.2.3)–(4.2.6) by substituting $(be_i + q_i)$ for be_i, $i = 1$ or $i = 2$.

Q is beneficial in that it improves robustness, but has the disadvantage that it is not clear how to choose it. Some guidance arises from the following analysis.

Suppose P is chosen to be unity, i.e., the chosen model has zero time constants (poles at infinity). The corresponding zero-gain predictor polynomials are, trivially

$$F_0(s) = G_0(s) = 0. \tag{4.3.3}$$

The corresponding control law is thus

$$u(t) = Q(s)^{-1}[w(t) - y(t)]. \tag{4.3.4}$$

The resultant PID control law is given by the inverse of Q and no adaptation is involved.

Two extremes are then defined by ($Q = 0$; $p(s)^{-1}$ = desired closed-loop system) and ($P = 1$; $Q(s)^{-1}$ = open-loop controller). These correspond to a completely prespecified closed-loop system requiring tuning to generate the corresponding open-loop compensator. and to a completely prespecified open-loop compensator requiring no tuning to derive its value. Suitable choice of P and Q thus gives a continuous range of self-tuning PID controllers from fully adaptive to nonadaptive.

D. Combined PI and PID Structures

The self-tuning PI and PID controllers described in Sections IV-A and IV-B have different structures in the sense that filters $1/C(s)$ have different orders (PI – first order, PID – second order). This has the practical disadvantage that it not possible to change from a PI algorithm to a PID algorithm without altering the internal database of the algorithm; this cannot be achieved on-line.

The reason for this difference in structure lies in the assumption (1.1.3) that

$$\deg(C) = \deg(A) - 1. \tag{4.4.1}$$

In other words, the disturbing process is bounded but not differentiable. PI regulators arise when $\deg(A_1) = 1$; PID regulators arise when $\deg(A_1 = 2)$. If, when considering PI regulators, the alternative assumption (1.1.3)

$$\deg(C) = \deg(A) \tag{4.4.2}$$

is used, the PI and PID regulators will have the same structure as far as the C filter is concerned. The consequence of this assumption is that a uniform structure is obtained for PI and PID regulators.

V. Implementation

The equations derived in previous sections are not in a form appropriate for actual software implementation. This section contains some of the details necessary for effective implementation. The description is restricted to those parts peculiar to the controllers of this paper; in particular, details of the well-known least-squares estimator are not considered as they may be found elsewhere [4].

A. Implementation of the Controller

The expression (1.2.7) for the model-reference controller defines the control signal $u(t)$ implicitly. For practical application, a discrete-time approximation yielding an explicit value for the control is necessary.

First, consider a continuous-time implementation of

$$u^* = \frac{G_0(s)}{C(s)}\, u(t) \tag{5.1.1}$$

where

$$\frac{G(s)}{C(s)} = \frac{g_1 s + \cdots + g_m s^m}{c_0 + c_1 s + \cdots + c_m s^m}. \tag{5.1.2}$$

Regarding s^{-1} as an integrator, a state-space implementation is given by

$$x^u_{i-1}(t) = s^{-1} x^u_i(t) \qquad i = 1 \cdots m \tag{5.1.3}$$

$$x^u_m(t) = \left(u - \sum_{i=0}^{m-1} c_i x_i(t)\right) \Big/ c_m \tag{5.1.4}$$

$$u^* = \sum_{i=1}^{m} g_i x^u_i. \tag{5.1.5}$$

The values of $x_i i = 0 \cdots m-1$ represent the "state" of the filter, however, it is convenient to store x_m as well.

A simple (approximate) discrete-time version arises by replacing the integrator by a discrete-time approximation

$$s^{-1} \text{ is repaced by } \frac{h}{2}\frac{1+z^{-1}}{1-z^{-1}} \tag{5.1.6}$$

where h is the sample interval. A similar filter representation is obtained for $y^*(t) = F(s)/C(s)y(t)$. These two filters are combined to give a one-step-ahead estimate of ϕ^*

$$\phi^* = u^* + y^*. \tag{5.1.7}$$

This implementation of $G/Cu(t)$ has the property that only x^u_0 depends on the current control signal. The control law may thus be implemented as follows.

Control Algorithm

A: Update the filter states $x^u x^y$ with $u(t) = 0$ and $y(t) = 0$.

B: Include the current system output y in the filter state

$$x^y_m := x^y_m + \frac{y}{c_m}. \tag{5.1.8}$$

(Here and hereafter ":=" means assignment in the algorithmic sense) Compute

$$\phi'(t) = u^* + y^*. \tag{5.1.9}$$

C: Compute the control form

$$u(t) = \frac{1}{g_m}(w - \phi'(t)). \tag{5.1.10}$$

If necessary (for example, to avoid saturation) replace $u(t)$ by a suitable modified value.

D: Update the element x^u depending on $u(t)$ and (if required) generate the prediction ϕ^*.

$$x^u_m := x^u_m + \frac{u(t)}{c_m} \tag{5.1.11}$$

$$\phi^*(t) = \phi'(t) + \frac{g_m}{c_m}\, u(t). \tag{5.1.12}$$

For practical real-time use, the delay between sampling and signal generation is minimized if these steps are implemented in the sequence:

Read in the system output and previous input.

B, C.

Send the control signal

Modify the control signal (if required) to form the system input D, A.

Reading the last system input during the first step allows the control signal to be externally modified, while ensuring that the controller database corresponds to the control actually used.

The filter states in (5.1.3) and (5.1.4) are used directly in the estimation algorithm, as from (2.1.2)

$$X^u = [x^u_1,\ x^u_2,\ \cdots\ x^u_n];\ X^y = [x^y_1,\ x^y_2,\ \cdots\ x^y_n]. \tag{5.1.13}$$

B. Approximating Phi

To give a realizable adaptive algorithm, ϕ is replaced by a realizable approximation ($\bar{\phi}$) decomposed as

$$\bar{\phi} = \bar{\phi}^* + \bar{e}. \tag{5.2.1}$$

If this model is to yield the required vector (θ) of continuous-time parameters, the approximation to ϕ must have the following properties:

A: The approximation must be realizable.

B: $\bar{e}$ should be uncorrelated with X.

C: $\bar{\phi}^*$ should be approximately equal to ϕ^*.

There are a number of ways of achieving this. The method proposed in an earlier paper [13] involved the use of a low-pass IIR (infinite impulse response) filter with time constants short with respect to the control sample interval h. This was shown to have the desired properties. This method has a number of implementation problems, so a simpler method using a finite impulse response (FIR) filter is described here.

In its simplest form the FIR filter involves replacing the unrealizable derivatives in the P polynomial (the s operator) by a backward difference approximation

$$s'(s)=\frac{1-e^{-sh/k}}{h/k} \tag{5.2.2}$$

where k is an integer larger than the degree of P

$$k>\deg\ (P). \tag{5.2.3}$$

An approximation to $\phi(t)$ is thus given by

$$\bar{\phi}(t+h)=e^{sh}P(s'). \tag{5.2.4}$$

The constraint on k ensures that the approximation depends on a finite number of *future* measurements of the system output y. Effectively, then, the approximation replaces unrealizability due to derivatives with unrealizability due to noncausality. By shifting the time arguments of the defining equations back by h, the unrealizable quantity $\bar{\phi}(t+h)$ becomes the realizable quantity $\bar{\phi}(t)$. It is thus possible to satisfy condition A.

Conditions B and C are investigated elsewhere [10] and shown to be valid.

Using this method, it is important to prefilter the analog signal prior to sampling using the usual antialias filter.

VI. Experimental Results

The PI algorithm with the fixed-pole estimation algorithm and the alternative zero-gain predictor of Section III-E has been applied to the level control of a lab scale water tank. Details are given in a laboratory report [15]; suffice it to say that the water level in the tank was measured using a pneumatic sensor and the inflow controlled with a diaphragm valve (Fig. 2). The outflow was via a hand valve to a sump from where the water was pumped up again to the inlet valve. In this experiment, the output valve was at a fixed position.

Fig. 3 shows graphs of the water level, setpoint, control signal, and estimated gain and integral time constants plotted against time. Initially, the PI coefficients are poorly set and the closed-loop response is oscillatory. The chosen design parameters were of the form $P = 1 + ps$, $C = 1 + cs$, and $c = 0.2p$. As can be seen from Fig. 3, when the tuning is switched on, the coefficients retune to give the desired nonoscillatory response. This is despite the noisy and nonlinear tank characteristics.

VII. Conclusion

A class of adaptive controllers relevant to the control of industrial processes has been introduced. In particular, it has the following features: the controllers have integral action; the zero-gain predictor structure eliminates nonzero mean quantities from the estimation process; the controllers reduce to PI and PID controllers for certain typical process models; the estimation is not dependent on the control law in use (apart from possible identifiability problems); and the controller parameters are in continuous-time form.

A laboratory experiment has illustrated the performance of the PI version of the algorithm, and a full-scale industrial trial of a related algorithm developed by Proudfoot has been reported [28].

The self-tuning PI and PID algorithms can be used to tune external PID regulators. Simulation results reported elsewhere have confirmed that this can be done [19].

Current research is involved with a number of extensions to the methods described here. These include switched control systems, interconnected self-tuning controllers in feedforward and cascade configurations, and methods to control systems with time delay or other nonminimum phase characteristics. The robustness of the self-tuning algorithms to neglected system dynamics are being studied at the moment.

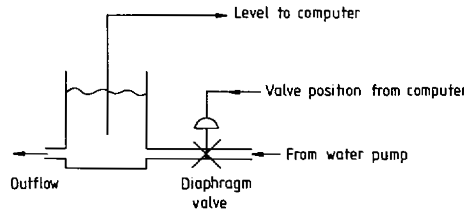


Fig. 2. The experimental system.

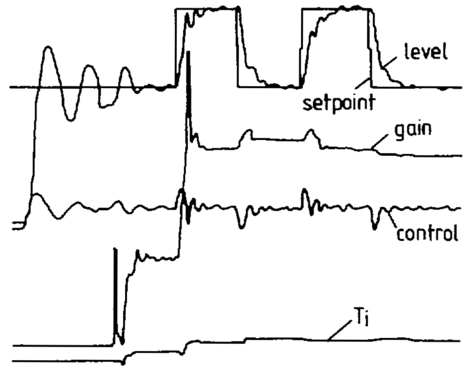


Fig. 3. Self-tuning PI controller.

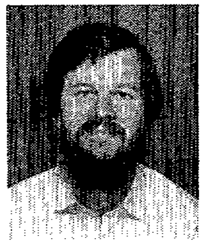
Acknowledgment

The author would like to acknowledge the many helpful comments of the associate editor and the referees.

References

[1] K. J. Astrom and B. Wittenmark, "On self-tuning regulators," *Automatica,* vol. 9, no. 2, pp. 185–199, 1973.
[2] ——, "Self-tuning controllers based on pole-zero placement," *Proc. IEE,* vol. 127, pt. D, no. 3, pp. 120–130, 1980.
[3] K. J. Astrom, U. Borisson, L. Ljung, and B. Wittenmark, "Theory and applications of self-tuning regulators," *Automatica.* vol. 13, no. 5, pp. 457–476, 1977.
[4] D. W. Clarke, "Introduction to self-tuning controllers," in *Self-Tuning and Adaptive Control,* Harris and Billings, Eds. New York: Perigrinus, 1981.
[5] D. W. Clarke and P. J. Gawthrop, "Self-tuning controller," *Proc. IEE,* vol. 122, no. 9, pp. 929–934, 1975.
[6] ——, "Self-tuning control," *Proc. IEE,* vol. 126, no. 6, pp. 633–640, 1979.
[7] D. W. Clarke, A. J. F. Hodgson, and P. S. Tuffs, "Offset problem and k-incremental predictors in self-tuning," *Proc. IEE,* vol. 130, pt. D, no. 5, 1983.
[8] H. Elliot, "Hybrid adaptive control of continuous-time systems," *IEEE Trans. Automat. Contr.,* vol. AC-72, no. 2, pp. 419–426, 1982.
[9] P. J. Gawthrop, "Input-output analysis of self-tuning controllers," in *Encyclopedia of Systems and Control,* M. G. Singh, Ed. 1984.
[10] ——, "Hybrid self-tuning control," in *Encyclopedia of Systems and Control,* M. G. Singh, Ed. 1984.
[11] ——, "Some interpretations of the self-tuning controller," *Proc. IEE,* vol. 124, no. 10, pp. 889–894, 1977.
[12] ——, "A comment on 'self-tuning pole/zero assignment regulators,'" *Int. J. Contr.,* vol. 31, pp. 999–1002, 1980.
[13] ——, "Hybrid self-tuning control," *Proc. IEE,* vol. 127, pt. D, no. 5, pp. 229–236, 1980.
[14] ——, "On the stability and convergence of a self-tuning controller," *Int. J. Contr.,* vol. 31, no. 5, pp. 973–998, 1980.
[15] ——, "Self-tuning PI control: Some experiments," *OUEL* Rep. 1385/81, 1981.
[16] ——, "Some properties of discrete adaptive controllers," in *Self-Tuning and Adaptive Control,* Harris and Billings, Eds. New York: Perigrinus, 1981.
[17] ——, "A continuous-time approach to discrete-time self-tuning control," *Opt. Contr. Appl. Methods,* vol. 3, no. 4, pp. 399–414, 1982.
[18] ——, "Self-tuning PI and PID controllers," presented at the IEEE Conference on Applications of Adaptive and Multivariable Control, Hull, England, 1982.
[19] ——, "Using the self-tuning controller to tune PID regulators," School of Appl. Sci., Univ. Sussex, Brighton, England, Rep. CE/T/2, 1982.

[20] P. J. Gawthrop and K. W. Lim, "Robustness of self-tuning controllers," *Proc. IEE,* vol. 129, pt. D, no. 1, pp. 21–29, 1982.
[21] G. C. Goodwin, P. J. Ramadge, and P. E. Caines, "Discrete-time stochastic adaptive control," *SIAM J. Contr. Optimiz.,* vol. 19, no. 6, pp. 829–853, 1981.
[22] A. J. F. Hodgson, "Problems of integrity in applications of adaptive controllers," Ph.D. dissertation, Univ. Oxford, Oxford, UK, 1982.
[23] R. E. Kalman, "Design of a self-optimising control system," *Trans. ASME,* vol. 80, pp. 468–478, 1958.
[24] L. Ljung, "Analysis of recursive stochastic algorithms," *IEEE Trans. Automat. Contr.,* vol. AC-22, pp. 551–575.
[25] K. W. Lim, "Robustness of self-tuning controllers," Ph.D. dissertation, Univ. Oxford, Oxford, UK, 1982.
[26] Y. Nishikawa and N. Sannomiya, "A method for auto-tuning of PID-control parameters," *IFAC Congress,* Kyoto, Japan, 1981.
[27] V. Peterka, "Adaptive digital regulation of noisy systems," presented at the IFAC Symposium on Identification and Process Parameter Estimation, Prague, 1970.
[28] C. G. Proudfoot, P. J. Gawthrop, and O. L. R. Jacobs, "Self-tuning PI control of a pH neutralisation process," *Proc. IEE,* pt. D, vol. 130, no. 5, 1983.
[29] P. E. Wellstead, D. Prager, and P. Zanker, "Pole assignment self-tuning regulator," *Proc. IEE,* vol. 126, no. 8, pp. 781–787, 1979.
[30] J. Wieslander and B. Wittenmark, "An approach to adaptive control using real-time identification," *Automatica,* vol. 7, no. 2, pp. 211–217, 1971.
[31] B. Wittenmark, "Self-tuning PID controllers based on pole-placement," *Dep. Automat. Contr.,* Lund Inst. Technol., Lund, Sweden, Tech. Rep., 1979.
[32] B. Wittenmark and K. J. Astrom, "Simple self-tuning controllers," presented at the Symposium on Adpative Systems, Bochum, 1980.
[33] P. C. Young, "Parameter estimation for continuous-time models," in *Proceedings of the 5th IFAC Symposium on 'Identification and System Parameter Identification,* Darmstadt, Isermann, Ed., 1979.
[34] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME,* vol. 64, pp. 759–768, 1942.
[35] F. Cameron and D. E. Seborg, "A self-tuning controller with a PID structure," *Int. J. Contr.,* vol. 38, no. 32, pp. 401–417, 1983.
[36] Z. J. Palmor and R. Shinnar, "Design of sampled data controllers," Ind. Eng. Chemical Process Des Dev. vol. 18, no. 1, 1979.



Peter J. Gawthrop (M'82) received the B.A. and D.Phil. degrees in engineering science from Queen's College, Oxford, England, in 1973 and 1976, respectively.

From 1976 to 1980 he was an SERC Post-doctoral Research Assistant in the Department of Engineering Science, Oxford University, Oxford, England. From 1980 to 1981 he was the W. W. Spooner Research Fellow at New College Oxford. Since 1981 he has been a Lecturer in control engineering at the University of Sussex, Brighton, England. His research interests are the theory and application of self-tuning control and the identification of ship dynamics.

Dr. Gawthrop is a member of the IEE and is a chartered engineer. He is an Associate Editor of *Automatica* and of the *IMA Journal of Mathematical Control and Information*.