Design and experimental evaluation of a multivariable self-tuning PID controller

T. Yamamoto and S.L. Shah

Abstract: In the paper, a new multivariable self-tuning PID controller design scheme is proposed which has a static matrix precompensator to reduce the interaction terms of the process transfer function matrix adjusted by an online estimator. The $p \times p$ precompensated multivariate system is then controlled via 'p' univariate self-tuning PID controllers whose parameters are adjusted by a second identifier placed around the precompensated plant. The PID parameters are calculated online based on the relationship between the PID and generalised minimum variance control laws. The scheme is experimentally evaluated on a 2×2 level plus temperature control system. Experimental results illustrate the effectiveness of this scheme.

1 Introduction

Self-tuning control schemes [1, 2] are useful for systems with unknown or slowly time-varying parameters and represent a class of advanced control algorithms. On the other hand, PID [3–5] control algorithms still continue to be widely used for most industrial control systems, particularly in the chemical process industry. This is mainly because PID controllers have simple control structures, and are simple to maintain and tune. Therefore it is still attractive to design discrete-time control systems with PID control structures. One may not be able to get good control performance in the case of time-varying processes. Many studies with autotuning [6–10] and self-tuning PID control [11–18] have been proposed. However, to the best of our knowledge, there are few studies of self-tuning PID control schemes for multivariable systems. The study of Yusof, *et al.* [19] has directly extended the self-tuning PID control schemes [11] considered for single-input and single-output (SISO) systems to multivariable cases. According to this scheme it is necessary to have $(p \times p)$-PID controllers for multivariable systems with p-inputs and p-outputs.

The main motivation in this study is to extend the univariate PID control scheme [18] to multivariate control systems. Many industrial processes are inherently multivariate in nature and yet are presently controlled by multiloop PID control schemes, where the interaction among the loops is essentially ignored. The results of such control schemes are highly detuned loops and consequently offer poor regulation and control. Many of the advanced model-based control (MBC) schemes can adequately deal with multivariable systems. The disadvantage of such schemes is that if the process conditions change then the model-plant mismatch can degrade control performance. To enhance the robustness of MBC schemes, the controller parameters are often detuned, consequently resulting in less than satisfactory performance.

Nyquist array techniques were introduce in the early 1970s to extend the paradigms of SISO design to MIMO systems. The basic idea was to decouple the MIMO system through the design of dynamic or static precompensators and then treat the precompensated system as a diagonal system for which separate PID loops can be designed. This study attempts to precisely implement this idea but in an adaptive manner, i.e. a static precompensator is estimated and implemented in an online manner followed by a separate adaptive outer loop for the design of 'p' SISO PID control loops based on the generalised minimum variance criterion [2, 18]. The advantage is that the proposed technique can deal with the time-variant nature of the process and also retain the designer's intuition and insight through the relatively simple design scheme that is proposed.

This paper is based on the classical Nyquist array paradigm. The Nyquist array techniques that were proposed in the late 1960s and early 1970s have not had the impact that their developers had sought primarily because of the difficulty in getting 'accurate' models of the process. Even if accurate models are available, processes inevitably change with time, for example due to scaling in the heat exchangers or catalyst deactivation and/or fouling in the reactors and so on. Furthermore, if processes drift or change then the initial controller design will not necessarily be optimal. The objective in this study is to show that the proposed adaptive design paradigms can indeed work without much CAD-based design effort if one can obtain reasonable estimates of the process models and if the models can be updated 'online' for on-going improved performance. To the best of our knowledge, this is one of the first online or adaptive multivariable design methods based on the simplified Nyquist array methods that has been proposed and successfully demonstrated to work on a computer-interfaced highly interactive two-input, two-output pilot-scale process.

The classical Nyquist array methods are model-based. This study offers a data-based approach to multivariate controller design where an initial data-based estimate of the static precompensator is used to obtain a Bode plot of the precompensated system, for example using functions 'SPA' or 'ETFE' in the Matlab System Identification toolbox [20].

IEE Proceedings online no. 20040853

doi: 10.1049/ip-cta:20040853

Paper first received 12th September 2003 and in revised form 17th June 2004

T. Yamamoto is with the Department of Technology and Information Education, Graduate School of Education, Hiroshima University, 1-1-1 Kagamiyama, Higashi-Hiroshima, Hiroshima, 739-8524, Japan

S.L. Shah is with the Department of Chemical and Materials Engineering, University of Alberta, Edmonton, Alberta, T6G 2G6, Canada

This is followed by simultaneous online estimation of the static precompensator and the univariate PI controllers to continuously 'fine-tune' the performance of the multivariate closed-loop system in the presence of process changes, disturbances, etc.

2 Multivariable self-tuning PID controller

2.1 Mathematical model

Let z^{-1} be the backward shift operator, then the following discrete-time p-input and p-output multivariable description of ARIMAX system describes the process

$$\boldsymbol{A}(z^{-1})\boldsymbol{y}(t) = D\boldsymbol{B}(z^{-1})\boldsymbol{u}(t-1) + \boldsymbol{x}(t)/\Delta \qquad (1)$$

where $\boldsymbol{u}(t)$ and $\boldsymbol{y}(t)$ are the input and output vector with p-elements, i.e.

$$\left.\begin{aligned}\boldsymbol{u}(t) &= [u_1(t), u_2(t), \ldots, u_p(t)]^T\\ \boldsymbol{y}(t) &= [y_1(t), y_2(t), \ldots, y_p(t)]^T\end{aligned}\right\} \qquad (2)$$

and $\boldsymbol{x}(t)$ denotes the output of a white gaussian noise vector through a disturbance transfer function vector; Δ is the differencing operator defined as $\Delta := 1 - z^{-1}$. Furthermore, $\boldsymbol{A}(z^{-1})$ is a diagonal polynomial matrix given by

$$\boldsymbol{A}(z^{-1}) = \mathrm{diag}\{A_1(z^{-1}), A_2(z^{-1}), \ldots, A_p(z^{-1})\} \qquad (3)$$

where each $A_i(z^{-1})$ is at most a second-order polynomial [Note 1] of the form

$$A_i(z^{-1}) = 1 + a_{i,1}z^{-1} + a_{i,2}z^{-2} \qquad (4)$$

$\boldsymbol{B}(z^{-1})$ is the following full polynomial matrix with elements:

$$\boldsymbol{B}(z^{-1}) = \begin{bmatrix} B_{1,1}(z^{-1}) & B_{1,2}(z^{-1}) & \ldots & B_{1,p}(z^{-1}) \\ B_{2,1}(z^{-1}) & B_{2,2}(z^{-1}) & \ldots & B_{2,p}(z^{-1}) \\ \vdots & \vdots & \ddots & \vdots \\ B_{p,1}(z^{-1}) & B_{p,2}(z^{-1}) & \ldots & B_{p,p}(z^{-1}) \end{bmatrix}$$
$$= B_0 + B_1 z^{-1} + \cdots + B_m z^{-m} \qquad (5)$$

with the coefficient matrix B_k defined as

$$B_k = \begin{bmatrix} b_{k,1,1} & b_{k,1,2} & \ldots & b_{k,1,p} \\ b_{k,2,1} & b_{k,2,2} & \ldots & b_{k,2,p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{k,p,1} & b_{k,p,2} & \ldots & b_{k,p,p} \end{bmatrix} \quad (k = 1, 2, \ldots, m) \qquad (6)$$

D is the time-delay matrix of the form

$$D = \mathrm{diag}\{z^{-k_{m_1}}, z^{-k_{m_2}}, \ldots, z^{-k_{m_p}}\} \qquad (7)$$

where k_{m_i} denotes the minimum value of estimated time-delays for the ith row. If the true time delays k_i are known exactly in advance, it is better to set

$$k_{m_i} = \min_{j=1,\ldots,p}\{k_{i,j}\}$$

where $k_{i,j}$ denotes the time-delay between the ith output and the jth input signals. On the other hand, where the information about time-delays is not available, then k_{m_i} is set to 0. While the time-delay description in (7) may sound restrictive, in reality it is not because most real process have delay or interacter matrices of the diagonal form, or delays can be added to the actuators so that the delay matrix is of this diagonal form [21]. For the system (1) we make the following assumptions:

(i) The polynomial matrix $\boldsymbol{A}(z^{-1})$ is stable.
(ii) The degree of $\boldsymbol{B}(z^{-1})$, m is known, and the following relationship is satisfied:

$$k_{m_i} \le k_i \le m \quad (i = 1, 2, \ldots, p) \qquad (8)$$

(iii) $\boldsymbol{B}(1)$ is assumed to be nonsingular.
(iv) Reference input $w_i(t)$ consists of piecewise constant signals.

2.2 Precompensator

In designing multiloop controllers for multivariable control systems it is important to first remove or compensate for interactions. The simplest precompensator that can be designed is a static parameter, H is with the form [22, 23]

$$H := \boldsymbol{B}^{-1}(1)\boldsymbol{A}(1) = \begin{bmatrix} h_{1,1} & h_{1,2} & \ldots & h_{1,p} \\ h_{2,1} & h_{2,2} & \ldots & h_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ h_{p,1} & h_{p,2} & \ldots & h_{p,p} \end{bmatrix} \qquad (9)$$

Such a precompensator essentially looks after the low-frequency interaction. The augmented system constructed by (1) and (9) can then be described as

$$\boldsymbol{A}(z^{-1})\boldsymbol{y}(t) = D\boldsymbol{B}(z^{-1})H\boldsymbol{v}(t-1) + \boldsymbol{x}(t)/\Delta \qquad (10)$$

where $\boldsymbol{v}(t)$ denotes the input signal vector to the augmented or precompensated system. In fact, the precompensator could be designed as $\boldsymbol{B}^{-1}(z^{-1})\boldsymbol{A}(z^{-1})$ to decouple the system exactly. However, this would be needlessly cumbersome for realisation purposes and require the assumption that $\boldsymbol{B}(z^{-1})$ is asymptotically stable. Therefore a simple static precompensator as given by (9) is used in this paper. This is the static approximation of $\boldsymbol{B}^{-1}(z^{-1})\boldsymbol{A}(z^{-1})$. The idea behind static decoupling is that diagonal dominance is a suitably weaker requirement in comparison with complete or strong decoupling, via dynamic precompensators, in the design of controllers for multivariate systems. In general, careful design of a static decoupler can achieve diagonal dominance. Many industrial processes are controlled in a multiloop manner without any precompensation whatsoever.

By regarding the augmented system (10) as the approximately or almost decoupled system, the following model can be obtained for each diagonal element:

$$A_i(z^{-1})y_i(t) = z^{-k_{m_i}}\bar{B}_i(z^{-1})v_i(t-1) + x_i(t)/\Delta \quad (i = 1, 2, \ldots, p) \qquad (11)$$

where

$$\left.\begin{aligned}\bar{B}_i(z^{-1}) &:= \sum_{k=1}^{p} B_{i,k}(z^{-1})h_{k,i}\\ &= \bar{b}_{i,0} + \bar{b}_{i,1}z^{-1} + \cdots + \bar{b}_{i,m}z^{-m}\end{aligned}\right\} \qquad (12)$$

and $A_i(z^{-1})$ is given by (3).

Note 1: A very practical and standard assumption in process control is that most stable processes can be represented or approximated by at most a second-order system. In fact according to Ljung, identification or empirical modelling is an exercise in model approximation. In particular, we are interested in practical, reduced-complexity controllers. Thus the assumption that $A_i(z^{-1})$ are stable and of order ≤ 2 is reasonable.

2.3 Multiloop PID controller design

Next consider the design of PID controllers [18] for the augmented system given by (11). The digital PID control law to be considered in this paper is described as

$$\Delta v_i(t) = k_{c_i}\Big[\{e_i(t)-e_i(t-1)\}+\frac{T_s}{T_{I_i}}e_i(t) + \frac{T_{D_i}}{T_s}\{e_i(t)-2e_i(t-1)+e_i(t-2)\}\Big] \quad (13)$$

where $e_i(t)$ denotes the control error signal given by

$$e_i(t) := w_i(t) - y_i(t) \quad (14)$$

and k_{c_i}, T_{I_i} and T_{D_i} are the proportional gain, the reset time and the derivative time, respectively. Furthermore, T_s denotes the sampling interval. For convenience, let $L_i(z^{-1})$ be

$$L_i(z^{-1}) := k_{c_i}\left(1+\frac{T_s}{T_{I_i}}+\frac{T_{D_i}}{T_s}\right) - k_{c_i}\left(1+\frac{2T_{D_i}}{T_s}\right)z^{-1} + \frac{k_{c_i}T_{D_i}}{T_s}z^{-2} \quad (15)$$

then (13) can be rewritten by

$$L_i(z^{-1})y_i(t) + \Delta v_i(t) - L_i(z^{-1})w_i(t) = 0 \quad (16)$$

The tuning of the control constants in PID control laws (13) or (16) is important, since the performance of the control system strongly depends on them. For systems with unknown parameters and unknown time delays, however, it is difficult to easily find the 'optimal' PID parameters. Therefore a self-tuning PID control algorithm based on the relationship between PID control and generalised minimum variance control (GMVC) laws is derived as follows.

2.4 PID tuning

Consider the following cost function to derive a GMVC control law

$$J_i = E\big[\phi_i^2(t+k_{m_i}+1)\big] \quad (17)$$

$\phi_i(t+k_{m_i}+1)$ in (17) denotes the generalised output of the form

$$\phi_i(t+k_{m_i}+1) := P_i(z^{-1})y_i(t+k_{m_i}+1) + \lambda_i\Delta v_i(t) - R_i(z^{-1})w_i(t) \quad (18)$$

where λ_i in (18) is the weighting factor with respect to the control input, $P_i(z^{-1})$ is the user-specified design polynomial of the form

$$P_i(z^{-1}) = 1 + p_{i,1}z^{-1} + p_{i,2}z^{-2} \quad (19)$$

and $R_i(z^{-1})$ is determined based on the relationship between PID control and GMVC laws. The control input minimising the cost function (17) is given by the following equation [2]:

$$F_i(z^{-1})y_i(t) + \{E_i(z^{-1})\bar{B}_i(z^{-1})+\lambda_i\}\Delta v_i(t) - P_i(1)w_i(t) = 0 \quad (20)$$

where $E_i(z^{-1})$ and $F_i(z^{-1})$ are obtained by solving the following Diophantine equation:

$$P_i(z^{-1}) = \Delta A_i(z^{-1})E_i(z^{-1}) + z^{-(k_{m_i}+1)}F_i(z^{-1}) \quad (21)$$

$$\left.\begin{aligned} E_i(z^{-1}) &= 1 + e_{i,1}z^{-1} + \cdots + e_{i,k_{m_i}}z^{-k_{m_i}} \\ F_i(z^{-1}) &= f_{i,0} + f_{i,1}z^{-1} + f_{i,2}z^{-2} \end{aligned}\right\} \quad (22)$$

Next, based on the relationship between PID control and GMVC laws, a tuning method of PID parameters is derived. Usually the dynamics of the system to be controlled, for example the time-delays or the time-constants, are rarely known precisely in advance. In particular, knowledge of the delay is important. Here we adopt the strategy that k_{m_i} be underestimated i.e. initially use the upperbound estimate of the delay, or assume that the order of $\bar{B}_i(z^{-1})$ is large enough, to cope with the problem. Therefore the estimates k_{m_i} and $\bar{B}_i(z^{-1})$, i.e. the second term $E_i(z^{-1})\bar{B}_i(z^{-1})$ in (20), includes some uncertainties. To obtain a control law with a PID structure, consider the following equation with $E_i(z^{-1})\bar{B}_i(z^{-1})$ replaced by the static gain $E_i(1)\bar{B}_i(1)$:

$$F_i(z^{-1})y_i(t) + \{E_i(1)\bar{B}_i(1)+\lambda_i\}\Delta v_i(t) - R_i(z^{-1})w_i(t) = 0 \quad (23)$$

Here v_i is defined as

$$v_i := E_i(1)\bar{B}_i(1) + \lambda_i \quad (24)$$

then (23) can be rewritten as

$$\frac{F_i(z^{-1})}{v_i}y_i(t) + \Delta v_i(t) - \frac{R_i(z^{-1})}{v_i}w_i(t) = 0 \quad (25)$$

Furthermore if the following relations are satisfied:

$$\left.\begin{aligned} R_i(z^{-1}) &= F_i(z^{-1}) \\ L_i(z^{-1}) &= \frac{F_i(z^{-1})}{v_i} \end{aligned}\right\} \quad (26)$$

as in (25), then (25) becomes identical to (16). Therefore based on (15) and (26), the PID parameters can be calculated as follows:

$$\left.\begin{aligned} k_{c_i} &= -\frac{(f_{i,1}+2f_{i,2})}{v_i} \\ T_{I_i} &= -\frac{f_{i,1}+2f_{i,2}}{f_{i,0}+f_{i,1}+f_{i,2}}T_s \\ T_{D_i} &= -\frac{f_{i,2}}{f_{i,1}+2f_{i,2}}T_s \end{aligned}\right\} \quad (27)$$

Note that the parameters λ_i are related to only k_{c_i}. In other words, T_{I_i} and T_{D_i} are independent of λ_i. Thus the proposed scheme has a feature such that after selecting $P_i(z^{-1})$ in the generalised output (18), λ_i can be determined or chosen independently by considering the stability of the control system based on *a priori* information. For example, the Bode diagram can be utilised to determine λ_i. Such a design method for selecting λ_i is discussed in detail in Section 3.

The design method of $P_i(z^{-1})$ is discussed next. $P_i(z^{-1})$ is designed based on the two features which allow one to select the response shape, overshoot, settling time, etc., i.e. rise-time and the damping property.

In continuous-time systems, Shigemasa *et al.* [24] presented a practical method to design the reference model based on these feature. By transforming these features into discrete time, the following coefficients can be obtained for the first and second-order models, respectively:

$$p_{i,1} = -e^{-\rho_i} \quad (28)$$

$$p_{i,1} = -2e^{-\frac{\rho_i}{2\mu_i}}\cos\left(\frac{\sqrt{4\mu_i-1}}{2\mu_i}\rho_i\right) \quad (29)$$

$$p_{i,2} = e^{-\frac{\rho_i}{\mu_i}} \quad (30)$$

where ρ_i and μ_i are defined by

$$\rho_i := T_s/\sigma_i \tag{31}$$

$$\mu_i := 0.25(1-\delta_i) + 0.51\delta_i \tag{32}$$

and σ_i and μ_i denote the rise-time and the damping index, respectively. The step responses of $z^{-1}P_i(1)/P_i(z^{-1})$ are shown in Fig. 1, where $\sigma_i = 1.0$, $T_s = 1.0$ and δ_i is changed from 0.0 to 2.0 in increments of 0.5.

From Fig. 1 one can see that the response shape can be chosen to one's liking by changing the parameter δ_i. On the other hand σ_i, corresponding to the rise-time, can be set to between 1/3 and 1/2 of the time constant depending on the practical needs of the particular loop.

Based on the control scheme discussed, a multivariable self-tuning PID control is designed in this Section. A block diagram of the multivariable self-tuning PID control system is shown in Fig. 2. This control system has two estimators named estimator 1 and estimator 2, as shown in Fig. 2. Each estimator plays the designated role of online tuning of the precompensator and PID parameters, respectively.

In estimator 1, unknown parameters included in (1) are estimated by using the following recursive least squares (RLS) algorithm:

$$\hat{\theta}_{1,i}(t) = \hat{\theta}_{1,i}(t-1) + \frac{\Gamma_{1,i}(t-1)\psi_{1,i}(t-1)}{1+\psi_{1,i}^T(t-1)\Gamma_{1,i}(t-1)\psi_{1,i}(t-1)}\varepsilon_{1,i}(t) \tag{33}$$

$$\Gamma_{1,i}(t) = \frac{1}{\omega_1}\left[\Gamma_{1,i}(t-1) - \frac{\Gamma_{1,i}(t-1)\psi_{1,i}(t-1)\psi_{1,i}^T(t-1)\Gamma_{1,i}(t-1)}{\omega_1 + \psi_{1,i}^T(t-1)\Gamma_{1,i}(t-1)\psi_{1,i}(t-1)}\right] \tag{34}$$

$$\varepsilon_{1,i}(t) = \Delta y_{f_i}(t) - \hat{\theta}_{1,i}^T(t-1)\psi_{1,i}(t-1) \quad (i=1,2,\ldots,p) \tag{35}$$

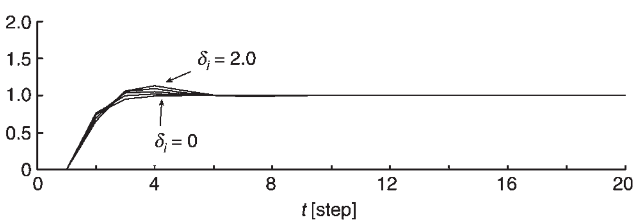

Fig. 1 *Step responses of $z^{-1}P_i(1)/P_i(z^{-1})$ to indicate effect of δ_i*

where ω_1 is a forgetting factor given by $0 < \omega_1 \leq 1$, and $\varepsilon_{1,i}(t)$ is a prediction error; $\hat{\theta}_{1,i}(t)$ and $\psi_{1,i}(t-1)$ are the unknown parameter and data vector, respectively, i.e.

$$\hat{\theta}_{1,i}(t) = [\hat{a}_{i,1}(t), \hat{a}_{i,2}(t), \hat{b}_{0,i,1}(t), \hat{b}_{0,i,2}(t), \ldots, \hat{b}_{0,i,p}(t), \ldots, \hat{b}_{m,i,1}(t), \hat{b}_{m,i,2}(t), \ldots, \hat{b}_{m,i,p}(t)]^T \tag{36}$$

$$\psi_{1,i}(t-1) = [-\Delta y_{f_i}(t-1), -\Delta y_{f_i}(t-2), \Delta \boldsymbol{u}_f^T(t-k_{m_i}-1), \ldots, \Delta \boldsymbol{u}_f^T(t-k_{m_i}-m-1)]^T \tag{37}$$

In (37) the subscript 'f' denotes filtered value of the inputs and outputs. Since the precompensator H given by (9) is supposed to capture the static dynamics and gain of the system, the following low-pass data prefilter $C_1(z^{-1})$ is used for the purpose of improving the reliability of the static gain calculation:

$$C_1(z^{-1}) := \frac{1-c_1}{1-c_1 z^{-1}} \tag{38}$$

It is well known that the filter plays an important role in the identification of real systems. The filtered values $y_{f_i}(t)$ and $u_{f_i}(t)$ included in (35) and (37) are given by

$$\left.\begin{array}{l} y_{f_i}(t) := C_1(z^{-1})y_i(t) \\ u_{f_i}(t) := C_1(z^{-1})u_i(t) \end{array}\right\} \tag{39}$$

Based on the online estimates included in $\hat{\theta}_{1,i}(t)$, the precompensator H is calculated.

In estimator 2, the unknown parameters included in the augmented system (11) are estimated via the following RLS algorithm:

$$\hat{\theta}_{2,i}(t) = \hat{\theta}_{2,i}(t-1) + \frac{\Gamma_{2,i}(t-1)\psi_{2,i}(t-1)}{1+\psi_{2,i}^T(t-1)\Gamma_{2,i}(t-1)\psi_{2,i}(t-1)}\varepsilon_{2,i}(t) \tag{40}$$

$$\Gamma_{2,i}(t) = \frac{1}{\omega_2}\left[\Gamma_{2,i}(t-1) - \frac{\Gamma_{2,i}(t-1)\psi_{2,i}(t-1)\psi_{2,i}^T(t-1)\Gamma_{2,i}(t-1)}{\omega_2 + \psi_{2,i}^T(t-1)\Gamma_{2,i}(t-1)\psi_{2,i}(t-1)}\right] \tag{41}$$

$$\varepsilon_{2,i}(t) = \Delta y_{g_i}(t) - \hat{\theta}_{2,i}^T(t-1)\psi_{2,i}(t-1) \; (i=1,2,\ldots,p) \tag{42}$$

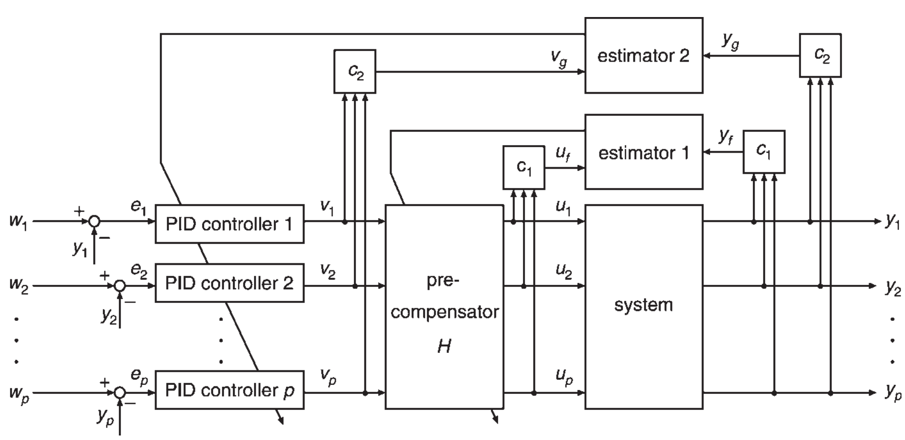

Fig. 2 *Block diagram of proposed multivariable self-tuning PID control system*

where ω_2 is a forgetting factor with limits $0 < \omega_2 \leq 1$, and $\varepsilon_{2,i}(t)$ is a prediction error; $\hat{\theta}_{2,i}(t)$ and $\psi_{2,i}(t-1)$ are given by

$$\hat{\theta}_{2,i}(t) = [\hat{a}_{i,1}(t), \hat{a}_{i,2}(t), \hat{\bar{b}}_{i,0}(t), \hat{\bar{b}}_{i,1}(t), \ldots, \hat{\bar{b}}_{i,m}(t)]^T \quad (43)$$

$$\psi_{2,i}(t-1) = [-\Delta y_{g_i}(t-1), -\Delta y_{g_i}(t-2), \Delta v_{g_i}(t-k_{m_i}-1), \ldots, \Delta v_{g_i}(t-k_{m_i}-m-1)]^T \quad (44)$$

For the purpose of improving reliability of the parameter estimation and once again emphasising estimation accuracy at lower frequencies, the following estimator filter $C_2(z^{-1})$ is also utilised:

$$C_2(z^{-1}) := \frac{1-c_2}{1-c_2 z^{-1}} \quad (45)$$

Therefore y_{g_i} and v_{g_i} included in (42) and (44) are given by

$$\left.\begin{aligned} y_{g_i}(t) &:= C_2(z^{-1})y_i(t) \\ v_{g_i}(t) &:= C_2(z^{-1})v_i(t) \end{aligned}\right\} \quad (46)$$

By solving the diophantine equation (21) based on estimates included in $\hat{\theta}_{2,i}(t)$, and calculating (24) and (27), PID parameters can be obtained.

The proposed multivariate self-tuning PID control algorithm is then realised via the following steps:

(i) Choose $P_i(z^{-1})$ and λ_i.
(ii) Design the estimator filters $C_1(z^{-1})$ and $C_2(z^{-1})$.
(iii) Estimate $\hat{\theta}_{1,i}(t)$ by using the RLS algorithm in (33)–(37).
(iv) Calculate the precompensator H based on (9).
(v) Estimate $\hat{\theta}_{2,i}(t)$ by using the RLS algorithm in (40)–(44).
(vi) Solve the diophantine equation (21).
(vii) Calculate v_i based on (24).
(viii) Calculate PID parameters based on (27).
(ix) Calculate the control input vector $\boldsymbol{u}(t)$ based on (13).
(x) Update t and return to (iii).

3 Experimental results

The proposed self-tuning controller is experimentally evaluated on a pilot-scale level plus temperature control system. A schematic diagram of the equipment is shown in Fig. 3.

This process system consists of a double-walled glass tank 50 cm high with an inside diameter of 14.5 cm. Cold and hot water enter the tank, in which the facility to provide additional heating via a steam coil is possible. The dynamics of the system can be changed by manipulating the exit valve position v. In this process system the control objective is to regulate the temperature of the water in the tank y_1 and the water level y_2 by manipulating the control valves u_1 and u_2. The steam coil and the exit valve positions are used for the purpose of evaluating the closed-loop system in the presence of disturbances and changing system parameters, respectively. The system is highly coupled.

The transfer function elements of the process can be approximated as first order with time delays. The multivariable PI control system is then constructed with $a_{i,2} = 0$ in (4). The sampling interval is set to 5.

For comparison, the effectiveness of the control system with a fixed-gain precompensator and the fixed PI control law was also evaluated. The system parameters are estimated offline in advance. Based on these estimates the precompensator is designed as follows:

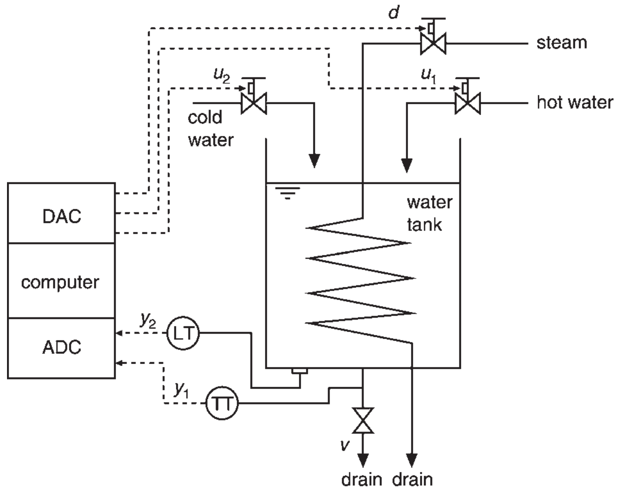

Fig. 3 *Schematic diagram of level plus temperature control system*

$$H = \begin{bmatrix} 2.120 & 0.167 \\ -1.552 & 0.181 \end{bmatrix} \quad (47)$$

Parameters of the augmented system constructed by inserting the precompensator (47) are estimated offline. This gives the following description of the 'weakly' decoupled system:

$$\left.\begin{aligned} y_1(t) &= 0.904y_1(t-1) + 0.062v_1(t-1) + 0.040v_1(t-2) \\ y_2(t) &= 0.945y_2(t-1) + 0.031v_2(t-1) + 0.025v_2(t-2) \end{aligned}\right\} \quad (48)$$

From (48) the time constants are calculated as $\tau_1 = 49.8$ s and $\tau_2 = 88.4$ s, respectively. Based on these time constants the rise-times are determined as $\sigma_1 = 30$ s and $\sigma_2 = 45$ s, and from (28) and (31) $P_1(z^{-1})$ and $P_2(z^{-1})$ are designed as

$$\left.\begin{aligned} P_1(z^{-1}) &= 1 - 0.847z^{-1} \\ P_2(z^{-1}) &= 1 - 0.895z^{-1} \end{aligned}\right\} \quad (49)$$

To ensure stability one can adjust λ_i by using a simple Bode analysis based on *a priori* or approximate knowledge of the process. In the Bode diagram only the gain curve is affected by λ_i and the phase curve is invariant to λ_i. Assuming (48) as the true system, the Bode diagrams of the control system are shown in Figs. 4 and 5, which is constructed by using (48) and the PI controllers whose PI parameters are calculated by (21), (24) and (27).

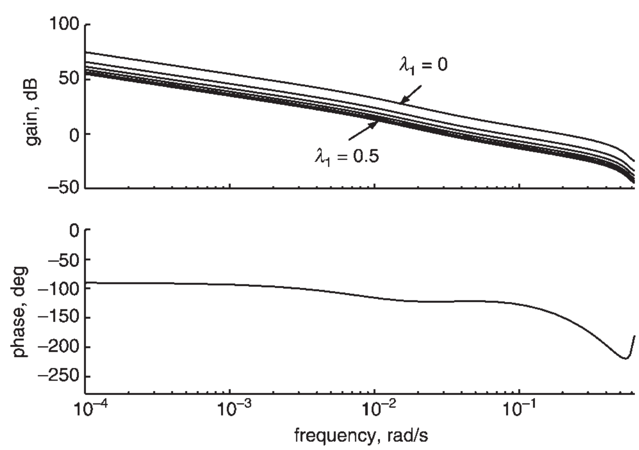

Fig. 4 *Bode diagram of temperature control system to indicate effect of λ_1*

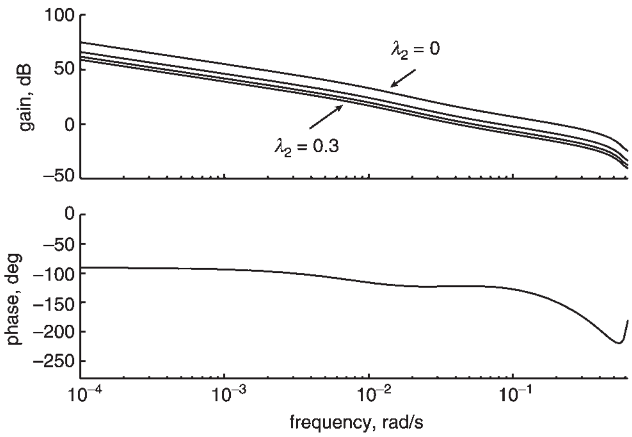

Fig. 5 *Bode diagram of level control system to indicate effect of* λ_2

Figures 4 and 5 show the Bode diagrams of the temperature and level control systems, where λ_1 in Fig. 4 and λ_2 in Fig. 5 are changed from 0.0 to 0.5 and from 0.0 to 0.3 in increments 0.1, respectively. From Figs. 4 and 5 λ_1 and λ_2 are, respectively, set to 0.3 and 0.1 to obtain a gain margin of approximately 20 dBs. This yields the following PI parameters:

$$\left.\begin{array}{l} k_{c_1} = 2.249, k_{c_2} = 6.058 \\ T_{I_1} = 29.443, T_{I_2} = 44.931 \end{array}\right\} \quad (50)$$

The results of the fixed PI control scheme with the fixed precompensator (47) employed are shown in Fig. 6, where the exit valve position v is changed at 500, 600, and 700 steps to investigate the system behaviour due to a disturbance and a change of the system parameters, i.e. the time-constant changes.

The utility of the static precompensator is illustrated by applying the fixed PI control scheme without the precompensator as shown in Fig. 7, with the PI parameters designed as

$$\left.\begin{array}{l} k_{c_1} = 2.307, k_{c_2} = 5.677 \\ T_{I_1} = 29.923, T_{I_2} = 39.611 \end{array}\right\} \quad (51)$$

From Figs. 6 and 7, the effectiveness of the precompensator in removing almost all the interactions is clearly demonstrated. However, there is a some oscillation in the level control result of Fig. 6. This oscillation occurs owing to some mismatch between the true system and the identified model in the transient and static modes. Notice that both results are inferior when the system parameters change.

Next, the self-tuning PID control scheme without the precompensator is employed, i.e. H is set to identity matrix, and the results are shown in Fig. 8. In this case $P_1(z^{-1})$ and $P_2(z^{-1})$ are designed via (49) and λ_1 and λ_2 are also set to 0.3 and 0.1, respectively. The forgetting factor in the RLS algorithm ω_2 is set to 1.0, and the estimator filter parameter c_2 is set to 0.4.

In comparing Fig. 7 with Fig. 8, the self-tuning PI control scheme works adaptively to compensate for the change of system parameters, and its effectiveness is clearly evident. However, the influence of the interaction still appears in the control results. The results of the complete control scheme with the self-tuning precompensator and the self-tuning PI controllers are shown in Fig. 9, where $P_i(z^{-1})$ and λ_i are the same as the case of Fig. 8, ω_1 and ω_2 are all set to 1.0, and the estimator filter parameters c_1 and c_2, are set to 0.85 and 0.4, respectively. The results of the estimator to compute the control parameters during the interval 450 to 750 steps are shown in Figs. 10 and 11, respectively. By using the proposed

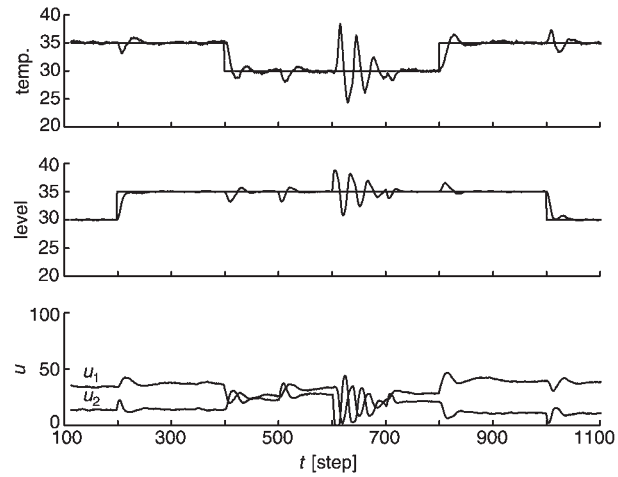

Fig. 7 *Control result by using fixed PI controller without a precompensator*

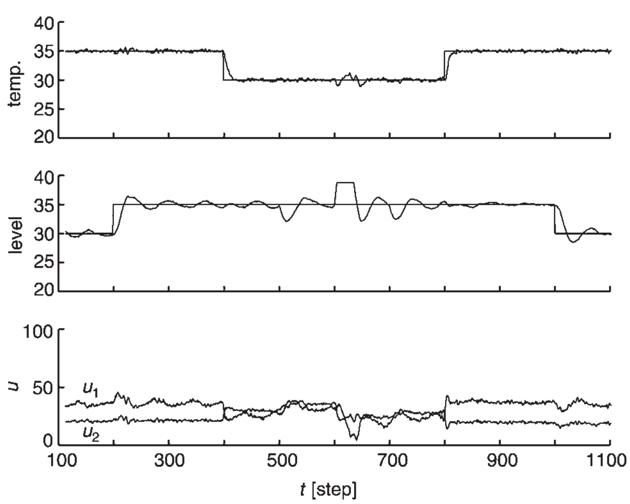

Fig. 6 *Control result by using fixed PI controller with a fixed precompensator*

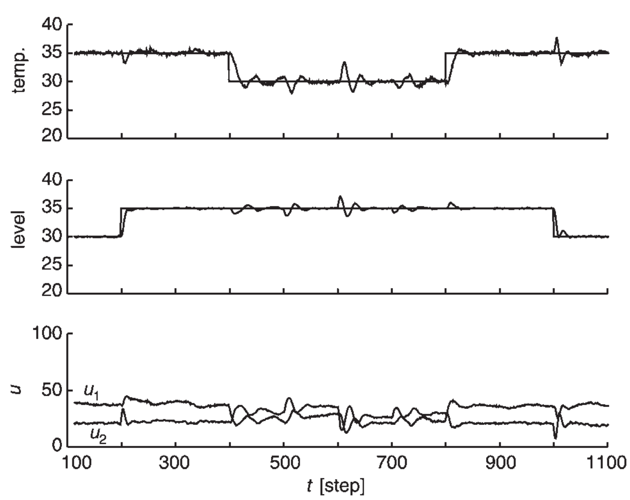

Fig. 8 *Control result by using self-tuning PI controller without a precompensator*

control scheme it is clear that the interaction of the system is removed, and that the scheme works well.

To investigate the effectiveness of the proposed control scheme in the presence of disturbances, the steam valve position d is also manipulated as shown in Fig. 12. The control result by using the fixed PI control scheme without the precompensator is shown in Fig. 13, where PI parameters are set to the same values as in (51). On the other hand, the result of using the proposed control scheme is shown in Fig. 14, where $P_i(z^{-1}), \lambda_i$ and $c_i (i = 1, 2)$ are designed as in the case of Fig. 9, but ω_1 and ω_2 are now set to 0.998.

It is clear that the self-tuning algorithms work well to provide satisfactory regulatory control in the presence of disturbances.

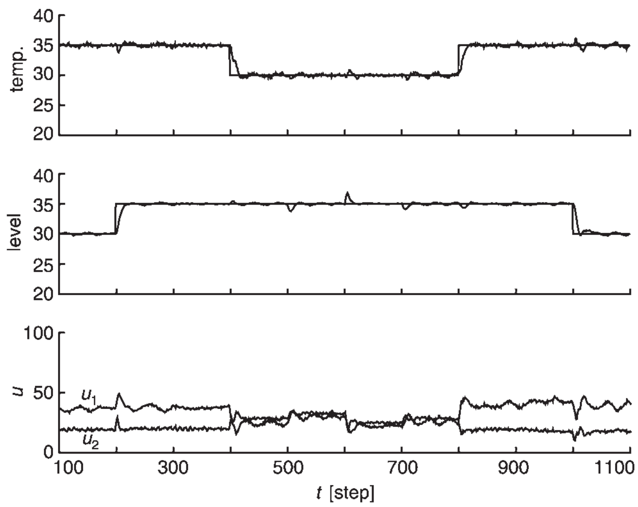

Fig. 9 *Control result by using newly proposed control scheme*

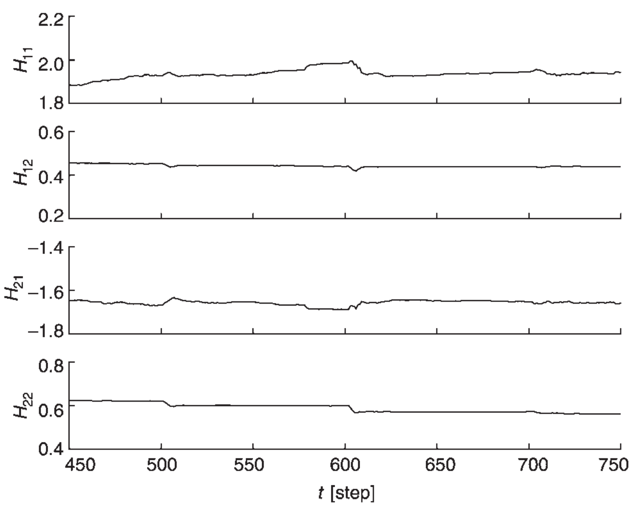

Fig. 10 *Estimated parameter trajectories of static gain matrix H as used in Fig. 9*

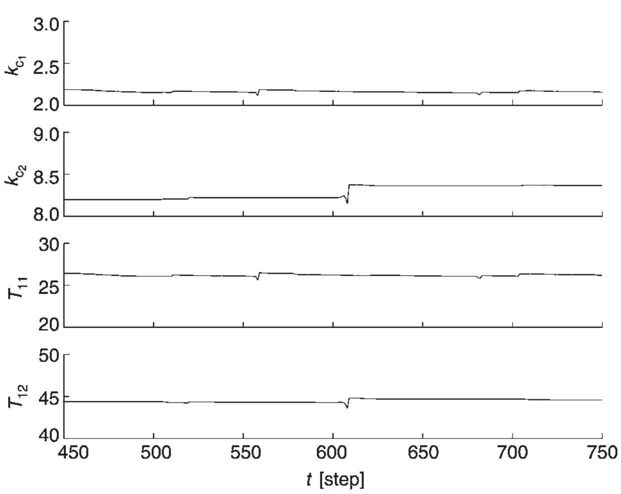

Fig. 11 *Controller parameter trajectories for control law used in Fig. 9*

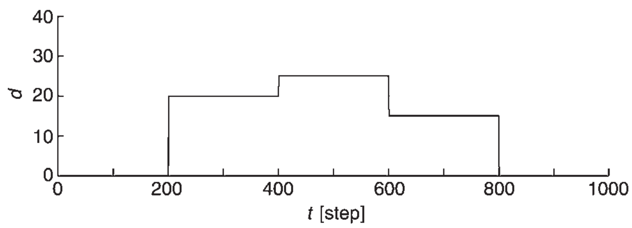

Fig. 12 *Graph to indicate valve position of steam coil*

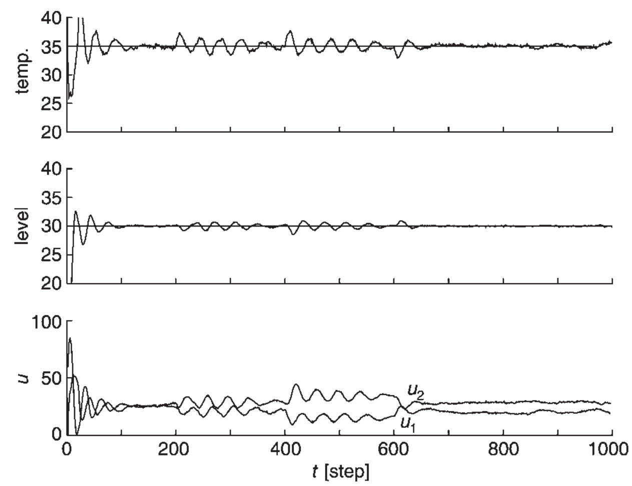

Fig. 13 *Regulatory control performance by using fixed PI controller without precompensator*

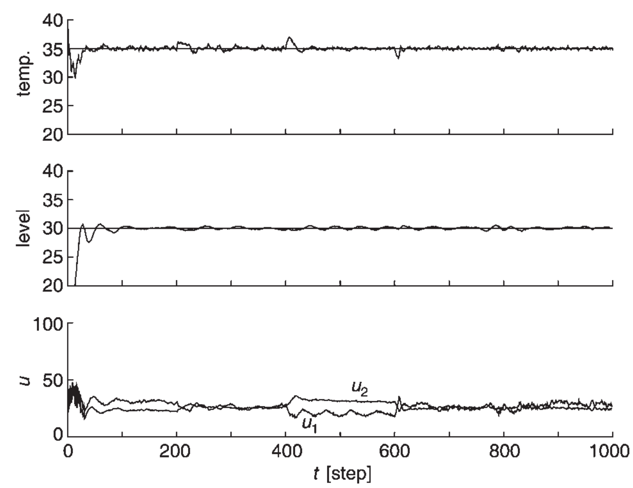

Fig. 14 *Regulatory control performance to demonstrate properties of newly proposed control scheme*

4 Conclusions

A design scheme for multivariable self-tuning PID controllers has been proposed. The main features of the proposed control scheme are summarised as follows.

- A practical self-tuning design strategy for multivariable systems based on the classical Nyquist array techniques has been presented.
- To cope with the interaction of the system a static precompensator is designed by the inverse of the estimated static gain matrix of the system.
- The control algorithm has two separate parameter estimators which work for the online tuning of the precompensator and the PID parameters, respectively.
- For the purpose of improving the reliability of the parameter estimator, simple exponential-type data prefilters condition the signals prior to estimation.
- PID parameters are calculated based on the relationship between PID control and GMVC laws.
- The user-specified design polynomial $P_i(z^{-1})$ and parameter λ_i are selected based on some *a priori* information about the controlled system.

The effectiveness of the proposed control scheme has been illustrated by experimental evaluation on a computer-interface pilot-scale level plus temperature process control system.

5 References

1 Åström, K.J.: 'Theory and applications of adaptive control - A survey', *Automatica*, 1983, **19**, (5), pp. 471–486
2 Clarke, D.W., and Gawthrop, P.J.: 'Self-tuning control', *Proc. Inst. Electr. Eng.*, 1979, **126D**, (6), pp. 633–640
3 Ziegler, J.G., and Nichols, N.B.: 'Optimum settings for automatic controllers', *Trans. ASME*, 1942, **64**, (8), pp. 759–768
4 Nims, P.T.: 'Some Design Criteria for Automatic Controls', *Trans. AIEE*, 1950, **70**, pp. 606–611
5 Chien, K.L., Hrones, J.A., and Reswick, J.B.: 'On the automatic control of generalized passive systems', *Trans. ASME*, 1972, **74**, pp. 175–185
6 Åström, K.J., and Hägglund, T.: 'Automatic tuning of PID controllers' (Instrument Society of America, Research Triangle Park, 1988)
7 Nishikawa, Y., Sannomiya, N., Ohta, T., and Tanaka, H.: 'A method for autotuning of PID control parameters', *Automatica*, 1984, **20**, (3), pp. 321–332
8 Schei, T.S.: 'Automatic tuning of PID controllers based on transfer function estimation', *Automatica*, 1994, **30**, (12), pp. 1983–1989
9 Voda, A.A., and Landau, I.D.: 'A method for the autocalibration of PID controllers', *Automatica*, 1995, **31**, (1), pp. 41–53
10 Menani, S., and Koivo, H.N.: 'Relay tuning of multivariable PI controllers'. Preprints of 13th World IFAC Congress, San Francisco, 1996, vol. K, pp. 139–144
11 Cameron, F., and Seborg, D.E.: 'A self-tuning controller with a PID structure', *Int. J. Control*, 1983, **38**, (2), pp. 401–417
12 Tanttu, J.T.: 'A self-tuning multivariable controller'. PhD Thesis, Tampere University of Technology, 1983
13 Proudfoot, C.G., Gawthrop, P.J., and Jacobs, O.L.R.: 'Self-tuning PI control of a PH neutralisation process', *IEE Proc. D. Control Theory, Appl.*, 1983, **130**, (5), pp. 267–272
14 Tjokro, S., and Shah, S.L.: 'Adaptive PID control'. Proc. American Control Conf., Boston, MA, pp. 1528–1534
15 Gawthrop, P.J.: 'Self-tuning PID controller: Algorithms and Implementation', *IEEE Trans. Autom. Control.*, 1986, **31**, (3), pp. 201–209
16 Radke, F., and Isermann, R.: 'A parameter-adaptive PID-controller with stepwise parameter optimization', *Automatica*, 1987, **23**, (4), pp. 449–457
17 Miler, R.M., Kwok, K.E., Shah, S.L., and Wood, R.K.: 'Development of a stochastic predictive PID controller'. Proc. American Control Conf., Seattle, WA, 1995, pp. 4204–4208
18 Fujii, K., Yamamoto, T., and Kaneda, M.: 'Self-tuning PID control of a polybutene process'. Preprints of IFAC Int. Symp. on Advance Control of Chemical Processes, Banff, 1997, pp. 107–112
19 Yusof, R., Omatu, S., and Khalid, M.: 'Self-tuning PID control: A multivariable derivation and application', *Automatica*, 1994, **30**, (12), pp. 1975–1981
20 Ljung, L.: 'System identification toolbox for use with Matlab'. (The Math Works Inc., Natio, MA, 1995)
21 Singh, R.P., and Narendra, K.S.: 'Prior information in the design of multivariable adaptive controllers', *IEEE Trans. Autom. Control*, 1984, **29**, (12), pp. 1108–1111
22 Rosenbrock, H.H.: 'Computer-aided control system design' (Academic Press, New York, 1974)
23 Shah, S.L.: 'Design studies of multivariable control systems'. MSc Thesis, University of Manchester Institute of Science and Technology, UK, 1972
24 Shigemasa, T., Takagi, Y., Ichikawa, Y., and Kitamori, T.: 'A practical reference model for control system design (in Japanese)', *Trans. Soc. Instrum. Control Eng.*, 1983, **19**, (7), pp. 592–594