Pretuning and adaptation of PI controllers

D.W. Clarke

Abstract: A method is proposed for the closed-loop tuning of a PI controller in cascade with a plant having an unknown transfer function, based on adding a variable-frequency sinewave to the loop's set-point. It applies design-point methodology in the Nyquist diagram to a phase–frequency adaptor coupled with gain estimators that monitor the closed-loop signals. The approach is shown to give good results over a wide range of cases, using test transfer functions taken from the literature that represent typical process dynamics. By comparing its step responses with those of loops tuned by standard Ziegler–Nichols and Cohen–Coon rules (as applied by many autotuners), the approach is seen to give consistent transients that are characterised by a user-specified damping factor. Adaptive tuning can be attained, 'quietly in the background' during normal PI control of a slowly varying plant, as convergence and tracking are obtained even if the output signal:noise ratio is less than one.

1 Introduction

This paper introduces and explores a method for the tuning (and for sustaining the tune) of a proportional + integral (PI) regulator in a closed loop. For a loop to be 'tuned', we mean that the controller gain K should be as high as possible (good for disturbance rejection) together with the closed-loop damping factor ζ being fixed at a user-specified value (for consistent overshoot to set-point steps), irrespective of changes in the plant dynamics. The approach involves injecting a probing sinewave at the loop's set-point: normally considered undesirable, but here its amplitude can be such that the signal:noise ratio (SNR) of the plant output is less than one, and hence the effect of the perturbation is negligible. Moreover, the loop can be subjected to the normal sequence of set-point changes without seriously affecting the tuning process.

An injected sinewave provides only two pieces of information, whereas controller design usually involves knowing the full plant transfer-function $G_p(s)$ or, equivalently, its frequency response $G_p(j\omega)$ over the full range $\omega = 0 \ldots \infty$. Hence it is important to demonstrate that the new approach is able to cope with the variety of transfer functions associated with typical process plant: this is indeed the case. However, the associated parameters differ according to whether the plant is type 0 or type 1 (because a type 1 plant with a PI controller has a Nyquist diagram which at low frequencies lies in the second rather than third quadrant), and hence prior knowledge of the plant's type number is necessary. The tuning algorithm, being adaptive, requires time-scale and other initialisation parameters. These are obtainable by a step-test. It is shown that this simple prior knowledge is then sufficient to give effective online tuning with no further user intervention.

© IEE, 2003

IEE Proceedings online no. 20030853

doi: 10.1049/ip-cta:20030853

Paper first received 9th December 2002 and in revised form 24th June 2003

The author is with Department of Engineering Science, Parks Road, Oxford OX1 3PJ, UK

The PID regulator is ubiquitous in single-loop process control. Despite decades of study, new ideas for PID analysis and design are still being developed, as shown in [1–3]. For the case where the plant's model $G_p(s)$ is known, a common approach is to optimise some performance index with respect to the PID parameters K, T_i, T_d (with perhaps a set-point prefilter): a good example is in [4]. For unknown G_p, information needs to be obtained experimentally from the plant's input/output behaviour, so interest focuses on the method's generality, its ability to cope with loop disturbances, its computational simplicity, and on user convenience. The autotuning approach [1, 5] has many of these attributes and is effective for single-shot pretuning. However, as it obtains information by replacing the PI regulator by a relay, it cannot sustain tuning without periodically reasserting a relay into the loop [1].

Ideally we would like to maintain a good state of tune by using just the normal operating signals circulating within the loop, and many self-tuning and adaptive algorithms have been proposed to achieve this objective. These are nearly all model-based, i.e. impose some parametric structure to $G_p(s)$, and often suffer from lack of generality, performing badly when the structure is wrong (the 'unmodelled-dynamics' problem). Experience shows that these adaptive controllers can work well when designed for specific applications [6] or when there is effective supervision of the parameter estimator [7]. The present method models the plant only at a carefully chosen frequency point, involves minimal supervision, but at the expense of requiring the injection of a perturbing sinusoidal signal.

The considered plants are formed by composing a set of elementary transfer functions, designed to conform with examples taken from the literature and also to validate the procedure over a wide range of test cases. The plant transfer function $G_p(s) = B(s)/A(s)$ is taken to be of the form '*NIU* G_k^n', where

multiple poles: k is the number of coincident poles, i.e. $A(s)$ includes $(s+1)^k$;
underdamped: the letter 'U' indicates a complex pole-pair $(s^2 + 0.5s + 1)$, having a damping factor $\zeta_p = 0.25$: such a system is rare in process control, but included for completeness;

integrator: '*I*' indicates that G_p is type 1;
nonminimum phase: '*N*' implies the presence of a term $(-2s+1)$ in $B(s)$: in other cases $B(s)$ is simply $B=1$;
noise: the superscript n indicates that the test includes additive output noise, typically with a signal:noise ratio of less than 1.

Examples of the notation are

$$G_2 \Rightarrow \frac{1}{(1+s)^2}; \quad NIU\, G_{10} \Rightarrow \frac{(-2s+1)}{s(s^2+0.5s+1)(s+1)^{10}}$$

The controller is a standard PI

$$G_C(s) = KC(s) = K\left(1+\frac{1}{sT_i}\right) = K\frac{(s+z_C)}{s} \qquad (1)$$

and the second form was adopted in the design and simulations, so the problem reduces to finding 'good' values of the gain K and the position of the PI zero z_C. To complete the notation, define the open-loop transfer function to be $G_o = KCG_p$, the closed-loop transfer function to be $G_c = G_o/(1+G_o)$, and say IG_1/PI to denote the closed-loop having IG_1 in cascade with the controller G_C of (1).

2 Design point and zero-placement rule

The design approach involves a judicious mixture of classical Nyquist-diagram and root-locus ideas. The algorithm is inspired by that of [8] (regrettably ignored by most of the PI literature), as it provides a simple, easily taught, yet remarkably effective approach to the tuning of PI controllers. The key idea is to prescribe a design point D in the Nyquist diagram through which, by proper choice of $KC(s)$, the open-loop frequency response $G_o(j\omega)$ is forced to pass. The concept is illustrated by Fig. 1, in which the design point D is characterised by a required open-loop gain and phase $\{|G_o(j\omega_D)| = OD, \arg G_o(j\omega_D) = -\pi + \phi_D\}$ at the frequency ω_D. Also shown in Fig. 1 is the critical point $Q = -1 + j0$, so that the closed-loop satisifes

$$|G_c(j\omega_D)| = \left|\frac{G_o}{1+G_o}\right| = \frac{OD}{QD}, \quad \arg G_c(j\omega_D) = -\theta_D \quad (2)$$

Of course, there are an infinite number of controller designs that make $G_o(j\omega) = \overline{OD}$; trivially one design could choose

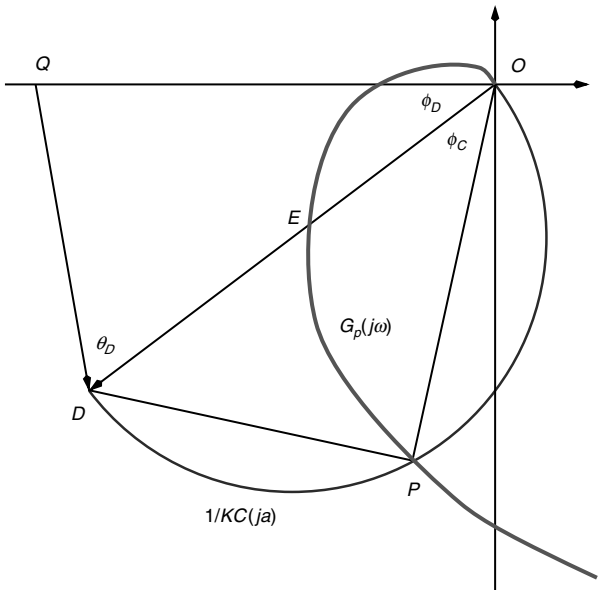

Fig. 1 *Design point D and 1/C(ja) semicircle*

$C=1$ and $K = OD/OE$, where E is the intersection of OD and the $G_p(j\omega)$ curve. To provide a unique solution requires adopting a 'zero-placement rule', as discussed subsequently.

The idea of a design point is old, indeed:

Phase-margin design: $\{OD=1, \phi_D = PM\}$: [8] advocates this variant of D;
gain-margin design: $\{OD = 1/GM; \phi_D = 0\}$.

However, it is argued that other choices of D are better and some thought about 'typical' Nyquist plots confirms this. For a multiple-pole (or dead-time) plant a PM design leads to low loop damping (as $|G_o| \to 0$ only slowly), whereas for type 1 plant a naïve GM design usually gives instability (e.g. IG_2/PI is conditionally stable).

If $C(s)$ is PI, then

$$\frac{1}{C(j\omega)} = \frac{1}{C(a)} = \frac{1}{1-ja} \qquad (3)$$

where $a = 1/\omega T_i$ is a design parameter. As a changes from 0 to ∞, $1/C$ moves along a semicircle above the real axis in the Argand plane, starting at $1+j0$ and finishing at O. This semicircle, scaled and rotated so that its diameter is OD, is shown in Fig. 1. Clearly, if $G_o = \overline{OD}$, then $G_p = \overline{OP}$ and $\phi_C = \arg 1/C = \tan a$. Now

$$\overline{OP} = G_p(j\omega_D) = \frac{1}{KC(ja)}G_o(j\omega_D) = \frac{1}{K}\frac{e^{j\phi_C}}{|C(ja)|}G_o(j\omega_D)$$

Consider this equation for a given ϕ_D: let D' be any point on the line OD and draw a semicircle with OD' as diameter. Supposing that the Nyquist plot of $G_p(j\omega)$ is precisely known, its intersection at P with the semicircle given $G_p(j\omega_D)$ (and hence the frequency ω_D) and $a = \arctan \phi_C \Rightarrow T_i = 1/a\omega_D$. Each possible semicircle has a corresponding $\{\omega_D, \phi_C\}$ and hence $\{a, T_i\}$. For example, when OD' is small there is no point of intersection; as OD' increases the point P is first close to D', so a is small and T_i large, and $a \to \infty$ as $OD' \to \infty$. The controller gain K is given by $K = OD/OD'$. Hence a 'large' semicircle is for a PI controller with a small $\{K, T_i\}$ (leading to a slow closed loop as ω_D is low) and *vice versa*.

Hence the design procedure is simple:

- choose $\{\phi_D, OD, a\} \Rightarrow \phi_C = \arctan a$;
- draw a semicircle, whose centre lies on the line OD, having a radius such that it intersects $G_p(j\omega)$ at P where $D\hat{O}P = \phi_C$. Frequency $\omega_D = \omega_p$ is taken from $G_p(j\omega)$ at the point P, and hence $T_i = 1/a\omega_P$ or $z_C = a\omega_P$;
- use the gain K to scale OD to the required length.

The critical question now is the right choice of a or, roughly, how far around the 1/C semicircle should P lie? This is equivalent to the appropriate placement of the PI zero, which has a significant effect on the closed-loop dynamic response, as discussed in [8] and explained best via root-locus plots. To aid intuition, Fig. 2 shows the near-O sections of root loci for G_3/PI and IG_3/PI for arbitrary zero-placements of $z_C = [-0.5, -0.05]$, respectively. In each case there are three dominant poles, so the time response will be characterised by the damping factor ζ and natural frequency ω_n of the complex pole-pair, plus an exponential term arising from the real pole that is approaching the PI zero. For G_3/PI the zero is clearly too near the origin, and its step response will include an undesirable slow mode. Of course, for G_3/PI one can place the zero much further to the left, but in this case the corresponding controller gain K is found to be small for the desired damping ζ (the 'large semicircle' case). Hence it is found that for good transient

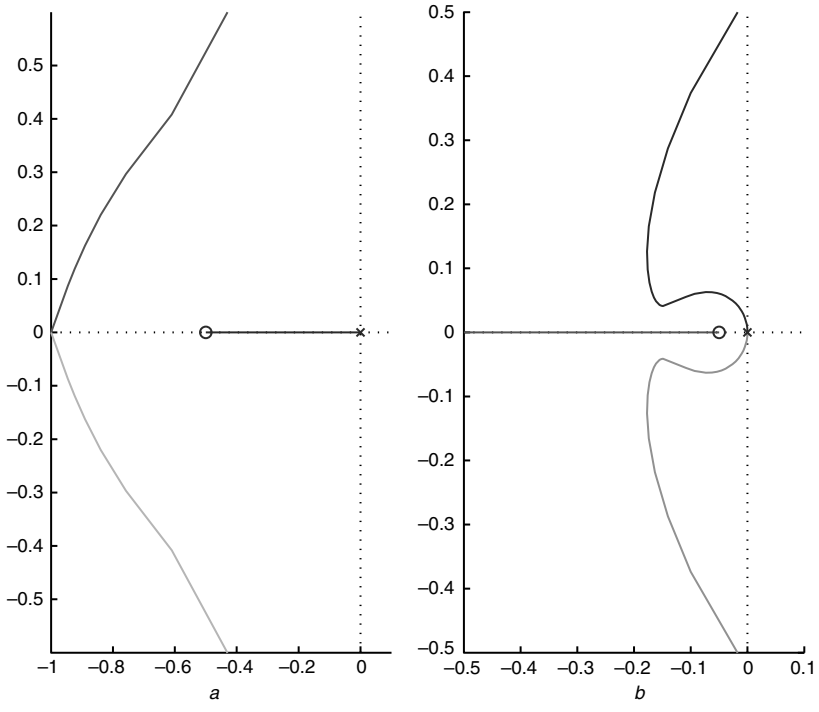

Fig. 2 *Dominant root-locus branches for G_3/PI (left) and IG_3/PI*
a G_3
b IG_3

response and a high loop gain, the three dominant poles should be in a Butterworth-like configuration.

To decide a value for *a exact* designs for $G_2/PI, IG_1/PI$ are performed with the objective that the closed loop has a desired damping ζ and the design point D is such that ω_D is the closed-loop natural frequency $\omega_n \Rightarrow z_C = 1/T_i = a\omega_n$. Start with G_2/PI

$$G_2(s) = \frac{1}{(1+s)^2} \quad \Rightarrow \quad G_o = \frac{K(s+a\omega_n)}{s(s+1)^2}$$

The closed-loop transfer function is then

$$G_c(s) = \frac{K(s+a\omega_n)}{s^3 + 2s^2 + (1+K)s + Ka\omega_n} \tag{4}$$

whose three closed-loop poles are to lie in a Butterworth-like configuration

$$A_c(s) = (s+\omega_n)(s^2 + 2\zeta\omega_n s + \omega_n^2),$$

so that by comparing coefficients,

$$\omega_n = \frac{2}{1+2\zeta} \tag{5}$$

$$1 + K = \frac{4}{1+2\zeta} \rightarrow K = \frac{3-2\zeta}{1+2\zeta} \tag{6}$$

$$a = \frac{\omega_n^2}{K} \rightarrow a = \frac{4}{(1+2\zeta)(3-2\zeta)} \tag{7}$$

In particular, for $\zeta = 0.5$ (which turns out to be probably the most effective choice) all of $\{\omega_n, K, a\} = 1$. Note for this choice of ζ that the PI's zero cancels the real closed-loop pole, giving

$$G_c(s) = \frac{1}{s^2 + s + 1}$$

thereby ensuring that the overshoot to a step input is not excessive but entirely predictable (16%). Repeating the exercise for IG_1 finds that all of

$$\{\omega_n, K, a\} = \frac{1}{1+2\zeta} \tag{8}$$

For $\zeta = 0.5$ the PI zero is at $z_C = a\omega_n = 0.5\omega_n$, so now there is no pole–zero cancellation and the zero in G_c augments the overshoot to a step. A set-point filter of the form $(1 + s/\omega_n)/(1 + s/a\omega_n)$ could however be used to minimise this additional overshoot.

The values of closed-loop natural frequency ζ, the controller gain K and the 'right' value of a are plotted in Fig. 3 as a function of required closed-loop damping ζ.

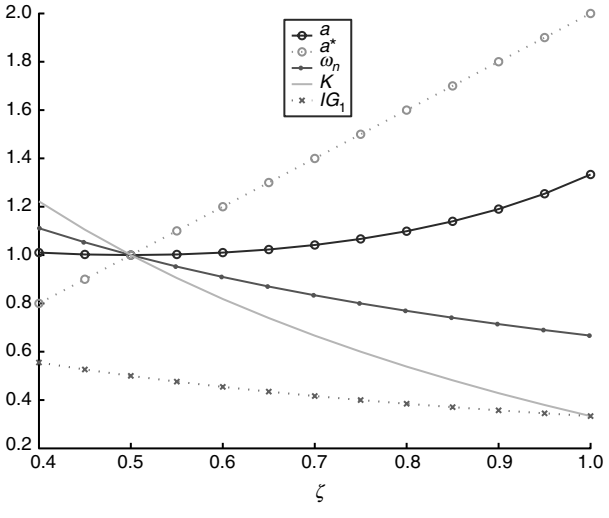

Fig. 3 *Design parameters for G_2, IG_1*

Note in particular that for G_2/PI, $a \approx 1$ throughout, i.e. the point P should lie half-way round the $1/C$ semicircle.

The design values are exact for these second-order systems, but it is desired to apply the same simple rules to an arbitrary plant G_p and hope that the subsequent closed-loop response is approximately like the nominal. In particular, it is important to see what happens when the plant is $(I)G_k$, when k is large. In fact this will also cover the case of plants with significant dead-time.

Figure 4 shows the near-O root-locus behaviour of G_k/PI for large k. The integrator pole leaves O and moves towards the PI zero at Z, arriving at R for the same K that gives the dominant complex pole C. Two of the k real poles leave Q at angles $\pm\pi/k$ and move along QC; the root locus along QC is approximately a straight line, especially for large k. Hence from the dominant complex poles, the closed-loop satisfies $\omega_n = OC, \zeta = \cos\theta$. Suppose that Z is placed such that $OZ = 2\zeta\omega_n$ (i.e. $a = 2\zeta$); then $CZ = \omega_n$. Now use the root-locus gain rule to show that R lies to the left of $Re(C)$, i.e. $OR > \zeta\omega_n$. The gain K to give a root at C is

$$K_C = \frac{CO.CQ^k}{CZ} = CQ^k = K_R$$

Suppose that R is at the point R', exactly below C: the corresponding gain is

$$K_{R'} = \frac{R'O.R'Q^k}{R'Z} = R'Q^k < K_C$$

Hence R lies between R' and Z: in the limit as $m \to \infty$ then R approaches R', as $R'Q = CQ\cos\pi/k$. This approximate analysis has shown that the closed-loop pole is in about the 'right' place (perhaps a little slow for large k) if the design a is chosen to be

$$a^* = 2\zeta \tag{9}$$

This value is plotted against ζ in Fig. 3. In summary, for a type 0 plant two reasonable choices for the design value a_D are

based on the prototype G_2: from (7), the recommended a_D is $4/(1+2\zeta)(3-2\zeta)$. This value should ensure that, given $G_p = G_2$, the exact ζ is attained.
avoiding a 'slow mode': the same argument for G_k gives $a_D = a^* = 2\zeta$.

Figure 3 shows how the suggested values of a_D vary with ζ. With chosen ζ typical for process loops (0.4 to 0.7, say) they are close, and indeed simply putting $a_D = 1$ is acceptable. It is found that the tuned behaviour is relatively insensitive to the choice, though larger values of a_D (e.g. using $a_D = a^* = 2\zeta$, $\zeta > 0.5$) leads to a smaller tuned value of K, which is undesirable.

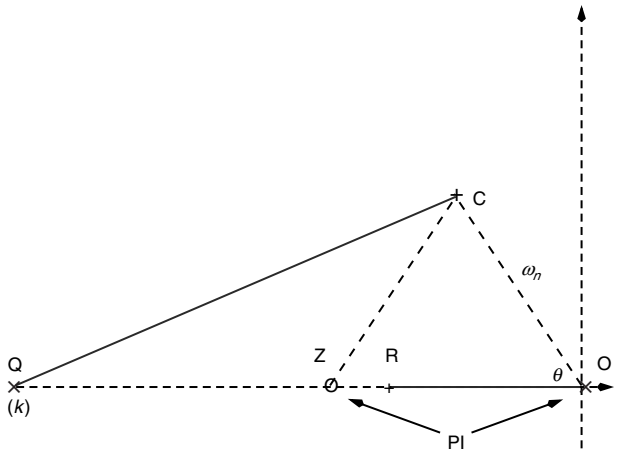

Fig. 4 *Dominant portion of root locus with G_k/PI*

One can now complete the design procedure, i.e. to find where the design point D should be consistent with the desired closed-loop damping factor ζ and that ω_D is to be the natural frequency ω_n of the dominant closed-loop poles, with the assurance that the associated real pole will be in a 'good' position by the adopted value of a. The closed-loop transfer function for both G_2, IG_1/PI is

$$G_c(s) = \frac{\omega_n^2(s + a\omega_n)/a}{(s+\omega_n)(s^2 + 2\zeta\omega_n s + \omega_n^2)} \Rightarrow$$
$$G_c(j\omega_n) = \frac{a+j}{2j\zeta a(1+j)}.$$

Hence, referring to Fig. 1

$$-\theta_D = \arg\, G_c(j\omega_n) = -\frac{3\pi}{4} + \cot^{-1} a \Rightarrow$$
$$\theta_D = \frac{3\pi}{4} - \cot^{-1} a \tag{10}$$

$$|G_c(j\omega_n)| = \frac{OD}{QD} = \frac{1}{2\zeta a}\sqrt{\frac{1+a^2}{2}} = \frac{1}{\alpha}, \text{ say.} \tag{11}$$

Defining $\{p = OD, q = QD\} \to q = \alpha p$, elementary trigonometry gives for the design point (using of course that $OQ = 1$)

$$OD = p = \frac{1}{\sqrt{1 + \alpha^2 - 2\alpha\cos\theta_D}} \tag{12}$$

$$\sin\phi_D = \frac{2}{p}\sqrt{s(s-1)(s-p)(s-q)} \tag{13}$$

where $s = 0.5(1+p+q)$ is the semiperimeter of the triangle QOD. But $p = OP = |G_o(j\omega_n)|$ and so the PI's gain is given by

$$p = K|G_C(j\omega_n)||G_p(j\omega_n)| = K|G_C(ja)||G_p| = K|K_p|\sqrt{1+a^2}$$
$$\text{so:}\quad K = \frac{p\cos\phi_D}{|G_p(j\omega_n)|} = \frac{p}{|G_p(j\omega_n)|\sqrt{1+a^2}}$$

These equations have a simple outcome for the central case $\{\zeta = 0.5, a = 1\}$

$$\text{design point:}\quad OD = \frac{1}{\sqrt{2}},\quad \phi_D = \frac{\pi}{4}(45^\circ)$$
$$\text{controller:}\quad \phi_C = \arctan a = \frac{\pi}{4},\quad K = \frac{1}{2|G_p(j\omega_n)|}$$
$$\text{CLTF:}\quad |G_c|_D = 1,\quad \arg G_c|_D = -\theta_D = -\frac{\pi}{2}$$

For a type 1 plant for $\zeta = 0.5$ we have $a = 0.5$, so the equations lead to

$$\text{design point:}\quad OD = 1,\quad \phi_D = 0.644\,(37^\circ)$$
$$\text{controller:}\quad \phi_C = \arctan a = 0.464\,(27^\circ),\quad K = \frac{0.894}{|G_p(j\omega_n)|}$$
$$\text{CLTF:}\quad |G_c|_D = 1.58,\quad \arg G_c|_D = -1.25\ (\theta_D = 72^\circ)$$

Note that for IG_1/PI this approach reduces to a classical phase-margin design ($OD = 1$), but for G_2/PI the point D is closer to O. There is of course an intuitive explanation of the difference in the position of D for the two cases. For IG_k/PI,

the Nyquist diagram typically starts in the second quadrant, so must cross the real axis to the *left* of Q. Hence P needs to be relatively near to Q so that $G_o(j\omega)$ is pulled round below Q accordingly. On the other hand, for G_k/PI the locus has a relatively small roll-off in gain compared with the increase in phase (seen especially in plants with dead time). Hence D should be nearer to O to avoid a bad value of the associated gain margin.

This discussion shows that the design-point method should give exact results for G_2, IG_1 and should be 'reasonable' for higher-order systems with real poles. However, it requires knowledge of $G_p(j\omega)$ so that its point of intersection P with the design semicircle can be determined. To apply the method in cases where G_p is at best only approximately known, the approach is: start with the point $P(0)$ (determined via appropriate initialisation) and the corresponding PI. During closed-loop operation the PI parameters are adjusted such that $P(t)$ converges to the correct P for the actual G_p. The procedure for automatic PI tuning is therefore

- choose the desired closed-loop damping factor ζ_D; decide whether the plant is Type {0, 1}. Choose a accordingly as $\{a_D, 1/(1+2\zeta_D)\}$ and find θ_D from (10)
- close the loop using an initial PI regulator;
- use an adaptive algorithm, based on injecting sinewaves of an adjustable frequency ω_o into the closed-loop set-point, such that $\arg G_c \rightarrow \arg G_c|_D = -\theta_D$, while ensuring that the controller settings obey the design-point rules: $T_i = 1/a\omega_o \Rightarrow z_C = a\omega_o$ and the controller gain K given by (14). From $T_i = a\omega_p$, the initial frequency should be $\omega_o(0) = 1/aT_i$.

The method can be compared with that of [9]. That approach attempts to fit classical PM and GM specifications via choice of PI gains k_p, k_i. This involves an 'inner' algorithm that uses two frequencies ω_1, ω_2 in turn: one corresponding to $\arg G_o(j\omega_1) = PM - \pi$ and the other to $\arg G_o(j\omega_2) = -\pi$, each requiring a time T_c (say) to converge. The 'outer', iterative, algorithm then adjusts k_p, k_i and the process is repeated m times until the required specifications are met. The overall tuning time of $2mT_c$ can be lengthy. Our algorithm requires only a single frequency ω_o that can be continuously varied (albeit slowly) and no further iteration is involved, so the tuning time is correspondingly shorter. While there is an attraction in achieving both a required PM and GM, perhaps the objective of the largest K consistent with a desired loop-damping ζ is more direct, as shown in the following examples.

3 Adaptive algorithm and its initialisation

An algorithm particularly suited for this application is a phase–frequency estimator (PFE), whose structure resembles a phase-locked loop [10, 11]. In essence the PFE generates a sinewave $A\sin\omega_o t$ of amplitude A and, using an appropriate phase-sensitive detector (PSD), adjusts the frequency ω_o until the phase difference between the generated and measured sinewaves is the desired value ϕ_d. Its application to closed-loop PI tuning is illustrated in Fig. 5: the sinewave is applied to the set-point of the loop and ω_o adapted until the closed-loop phase shift attains the required θ_D. The closed loop is itself time-varying, as the controller needs to be adjusted according to the current values of $\{\omega_o(t), \hat{K}_p(t) = |\hat{G}_p(j\omega_o)|(t)\}$ such that (for example taking a type 0 plant with $a = 1$)

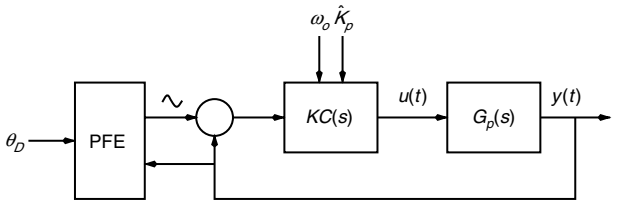

Fig. 5 *Closed-loop PI tuner*

$$K(t) = \frac{p\cos\phi_D}{|\hat{G}_p(j\omega_o(t))|} \Rightarrow \frac{1}{2\hat{K}_p(t)},$$

$$T_i(t) = \frac{1}{a\omega_o(t)} \Rightarrow z_C = \omega_o(t)$$

The frequency ω_o is available from the PFE, so to complete the tuner an estimator for K_p is required. There are of course many suitable methods, using for example RLS or similar approaches, but it is useful in this application to have a continuous (i.e. per-sample and not just once-per-cycle), ripple-free estimator of the amplitude A of a sinewave. Noting that the PFE of [11] uses a Hilbert-transform PSD, a similar approach is adopted here. Suppose that the measured sinewave is $u(t) = A\sin\omega_o t$; the 90° phase-shifted version is then $v(t) = A\cos\omega_o t$, and so

$$\hat{A} = \sqrt{u^2 + v^2}$$

is the required continuous estimator. The corresponding Simulink diagram is shown in Fig. 6. A Hilbert transformer (designed using the same rules as for the PFE) and its associated delay operator of $m/2$ samples provides two signals that have a relative phase shift of 90°. Preceding the estimator is a variable bandpass filter (VBPF)

$$F_v(s) = \frac{\beta\omega_o s}{s^2 + \beta\omega_o s + \omega_o^2}$$

whose centre frequency is the current value of ω_o (as provided by the PFE) and whose bandwidth is determined by the design parameter β. The objectives of the VBPF are

(i) elimination of *DC* values;
(ii) attenuation of harmonics, arising for example if there are nonlinearities in the plant or actuator;
(iii) reduction in the effect of additive noise.

Of course, one expects that a narrow passband would be accompanied by a slow transient response.

To test the method a unit-amplitude 1 Hz sinewave with/without added noise was applied to the estimator, using a sample rate of $f_s = 5$ Hz (a sampling interval of $h = 0.2$ s) and $m = 20$ the number of Hilbert transform parameters. Hence the delay element of Fig. 6 implies a time shift of $mh/2 = 2$ s and the time by which a change has rippled through all m terms of the HT is $mh = 4$ s. Figure 7 shows some simulation results. The top-left plots show the noise-free output, verifying that for $\beta = 1.5$ the output is correct and ripple-free after the corresponding delay. With the smaller value of $\beta = 0.15$ there is an additional effect: an exponential rise with time constant $\approx 1/\beta\omega_o$, thus increasing the settling-time.

The power in the sinewave is 0.5, so to generate noise of the same power (giving unity signal:noise ratio) the band-limited noise had spectral density $S_n = h/2$ for the sampling

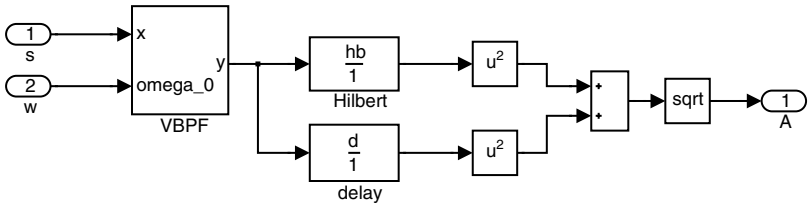

Fig. 6 *Amplitude estimator*

interval h. A section of the noise-free and noisy sinewave is shown in the top-right of the Figure. The lower plots show the estimator's output for $\beta = 0.15, 0.015$ respectively. As expected the reduction of β by 10 reduces $\mathrm{Var}\{\hat{A}\}$ also by 10 (SD by $\sqrt{10}$) - the usual time/variance trade-off. *Note*. The estimator is biased, clearly apparent for $\beta = 1.5$, but for the values of β in the Figure the bias is small. As the SNR is unlikely to be much less than 1 in applications, it is clear that the estimate's variance is more important: indeed it is useful to smooth the estimate further (say by a simple lag with time constant T_f). Hence in the simulations reported below

$$|\hat{G}_p(j\omega_o)| = \frac{\hat{A}_y^f}{\hat{A}_u^f}$$

is used, where the superscript f denotes a *posterior* filtering action: typically $T_f = 10/\omega_o(0)$.

Having described the components of the closed-loop PI tuner, it remains to define how its parameters (such as the adaptive gain K_a) can be chosen in a semiautomatic way. The idea is to use a procedure that initialises the algorithm without the need for user intervention. Guidance for this is in [12], which shows how all the parameters of a PFE can be chosen from a step-test and by fitting the experimental response to that of a first-order/dead-time (FODT) model

$$G_a(s) = \exp(-sT_d)\frac{K_p}{1+sT}$$

The model is time-normalised using $t' = t/T \rightarrow p = sT$ to give the NFODT model

$$G_A(p) = \exp(-p\lambda)\frac{K_p}{1+p} \tag{15}$$

and initialisation depends on the dimensionless parameter $\lambda = T_d/T$. Through an approximate theory and simulation over a range of test cases, a value of adaptive gain K_a was deduced in [12] that is near-optimal for a fixed transfer-function. However, in this application the closed-loop is changing during adaptation, as the PI parameters are continually being updated. Hence the adaptive gain for the results shown subsequently was 'detuned' by (say) $K_a = K_o/4$, where K_o is the 'optimal' value suggested by [12]. While this arbitrary detuning was effective for the reported simulations, inspection of the results shows that a more comprehensive theory is required.

The prior step-test provides a FODT model for G_p, not G_o, G_c. Hence full initialisation requires an estimate for the starting PFE frequency $\omega_o(0)$ and loop-controller $KC(s)_{t=0}$. To achieve this we proceed from the design point D and back-calculate the corresponding point P on the FODT's

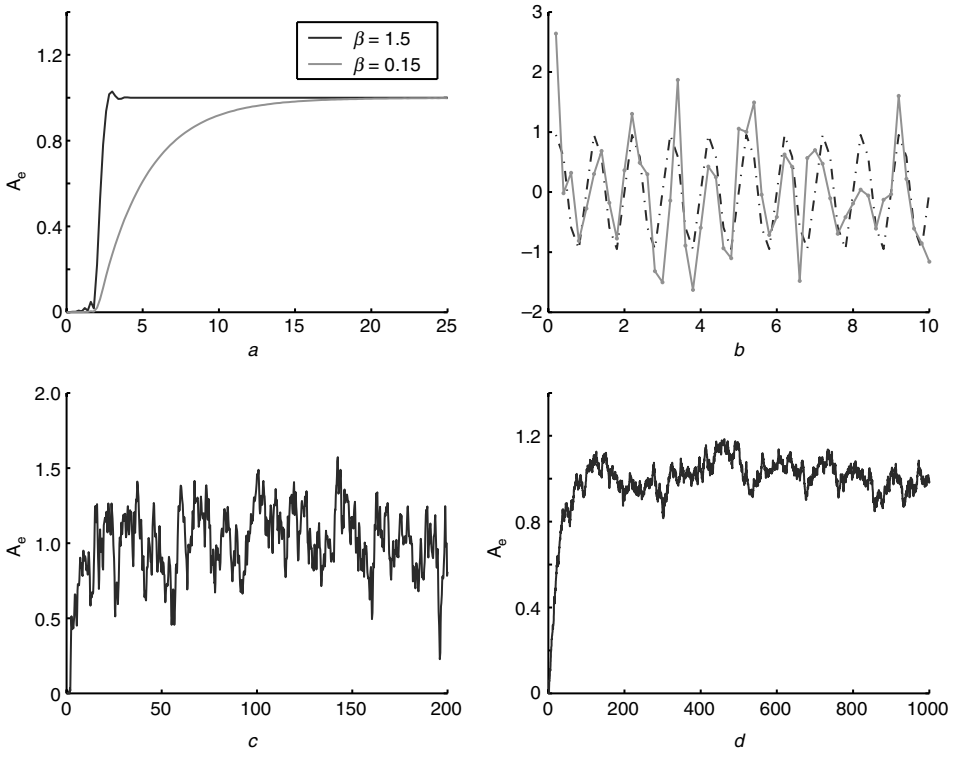

Fig. 7 *Applying amplitude estimator*

a Noise-free output, $m = 20$, $fr = [0.2 \ldots 0.8]$
b Noise-free and noisy sinewave, $h = 0.2$, $Sn = 0.1$
c Estimator output for $\beta = 0.15$
d Estimator output for $\beta = 0.015$

Nyquist diagram. From the design point, ϕ_D, ϕ_C are known and hence can assert from the NFODT of (15)

$$\arg\, G_A(j\Omega_o) = -\pi + \phi_D + \phi_C = -\lambda\Omega_o - \arctan\, \Omega_o$$

where $\Omega_o = \omega_o T$ is the normalised frequency. This equation is solved for Ω_o (perhaps using the approximations in [12]), and then the initial controller has

$$K(0) = \frac{p\cos\phi_D\sqrt{1+\Omega_o^2(0)}}{K_p}, \quad T_i(0) = \frac{T}{a\Omega_o(0)} \qquad (16)$$

The adaptive gain K_a and associated adaptor parameters need to be chosen. This is a complex question, for as discussed the closed-loop is time-varying during the tuning process. The initial, probably unsatisfactory, approach is to base the parameters on the open-loop FODT model, and to reduce the adaptive gain by some factor to avoid instability in the adaptation. From [12], the local slope of the NFODT's phase-frequency response is

$$G_A^*(\Omega_o) = \frac{d\arg\, G_A}{d\Omega}\bigg|_{\Omega=\Omega_o} \Rightarrow G_A^*(\Omega_o(0)) = -\lambda - \frac{1}{1+\Omega_o^2(0)},$$

and the recommended adaptive gain is

$$K_a = \frac{0.75\pi}{T^2 G_A^*(0)\lambda(3\pi + 2m)}, \qquad (17)$$

where m is the number of Hilbert-transformer parameters. As stated, this gain is then detuned by a factor of four: an experimentally-proven value that worked for all the type 0 plants. For the integrating type 1 examples the NFODT is clearly a poor predictor of the plant's Nyquist diagram. In particular it is found that the estimated K_p, T are large and the dimensionless dead-time λ is small. Hence for the results shown, K_a was detuned by a factor of eight, but even this was found to be insufficient for IG_1/PI (bounded oscillations in the estimator), and so here a factor of 16 was used.

Recalled that the objective of the Hilbert transformer is to produce a transfer function with precisely unit gain and phase shift of 90° over the expected frequency range of $\omega_o(t)$ as the tuning proceeds. The finite number of parameters m only provides an approximation which is better for large m at the expense of slow adaptation. For a type 0 plant the NFODT model gives a good first-guess of the tuned frequency, so fewer m designed to fit over a small frequency range fr are required, i.e. $m = 10, fr = [0.2 \ldots 0.8]$ was used. When using the tuner with type 1 plants the frequency ω_o can move substantially (by a factor of up to 3), so here $m = 20$, $fr = [0.1 \ldots 0.9]$ was adopted. In all cases the sample interval h associated with the Hilbert transformers was fixed at $h = 1/\omega_o(0)$.

The parameters of the passband prefilters and simple-lag postfilters were chosen by simple (and perhaps unjustifiable) rules-of-thumb. For the VBPF preceding the HTPLL, β= 1.5, whereas those associated with the amplitude estimators a value of 0.15 was adopted. The postfilters for $\hat{\boldsymbol{K}}_p$ were given the value $T_f = 10/\omega_o(0) = 10h$ for noise-free simulations, but this was increased to $100h$ for the noisy case.

Finally the adaptors need to be 'switched on'. The PFE starts by injecting sinewaves of user-chosen amplitude A (with $A = 0.1$ for the simulations reported subsequently) into the loop's set-point. These sinewaves then excite the plant (after a delay T_d) and its output then passes through several VPBFs and Hilbert transformers. To allow their responses to settle, the updating of ω_o and $|\hat{G}_p|$ was inhibited for an arbitrary period $2(T_d + mh)$.

4 Noise-free simulations

The objective of this set of simulations are to:

- verify that the adaptive algorithm works as predicted;
- validate the design-point idea and show that it produces well-tuned closed-loop responses for a wide variety of plants;
- demonstrate that the initialisation of the algorithm via a step-test/FODT-fit is effective;
- to see how much further tuning the algorithm needs beyond that deduced from the approximating FODT model.

The simulations followed the following sequence. A step was asserted over a prescribed time T_m to the open-loop G_p. The amplitude of the step was 1 for a type 0 and $1/T_m$ for a type 1 plant. From the FODT fit and the design choices ζ_D, a_D the adaptor was initialised. The PFE then operated for a period that depended on some simple function of the plant's complexity. Step-tests were then applied to the closed-loop with the initial and the final, tuned, controller. Half-way through the step-test the loop was subjected to a load disturbance, being a further step added to the plant's control input $u(t)$. For type {0, 1} plant the step had a value of {0.5,0.1}: the PI's control action is in the opposite sense so as to remove its effect from the plant output $y(t)$.

The example plots shown have been designed to provide sufficient information so that an interested reader can reproduce the results. They all have four subplots:

Top-left: This shows the step response of $G_p(s)$ (full line) and that of the FODT-fit G_a (dotted), together with the estimated parameters K_p, T, T_d and the deduced $\lambda = T_d/T$. In all cases there was equal weighting of squared errors over the defined response time. The experiment time T_m was a simple and arbitrary function of the model complexity.
Bottom-left: this show the progress of the adaptation using normalised parameters $\omega_o(t)/\omega_o(0)$ (full line) and $|\hat{G}_p|(t)/|\hat{G}_p|(0)$ (dotted line), together with the number of HT parameters m and the adaptive gain K_a (other parameters, such as the sample interval h were based on the results of [12]).
Top-right: this plots the closed-loop step response using the initial controller (dotted) and the final controller (full). The design parameters ζ_D, a_D and the initial tuning $K(0), T_i(0)$ are given.
Bottom-right: the corresponding control signal $u(t)$ and the final, tuned controller parameters.

4.1 Nominal G_2/PI and IG_1/PI

For G_2 and IG_1 the design-point procedure for a known model should lead to exact results, with the closed-loop poles in a Butterworth-like configuration. On the other hand, a FODT model is at best only an approximation to the open-loop dynamics, so that the adaptor is indeed required for final tuning. Figures 8 and 9 show that the algorithm worked well in both cases and the controller parameters K, T_i converge close to the optimal values, as shown in Table 1. Despite the expected poor initial FODT model of IG_1, Fig. 9 shows that the adaptor is adequately initialised. The excessive overshoot of IG_1/PI for set-point changes is clearly shown by the Figure (the suggested prefilter would reduce this); however, the load-disturbance response is good and comparable with that of G_2/PI in Fig. 8.

4.2 Other noise-free plants

The algorithm was successfully tested for $G_p = (NIU)G_k/PI$ over the range $k = 1 \ldots 20$ and for all combinations of extra dynamics NIU; further simulations included additional

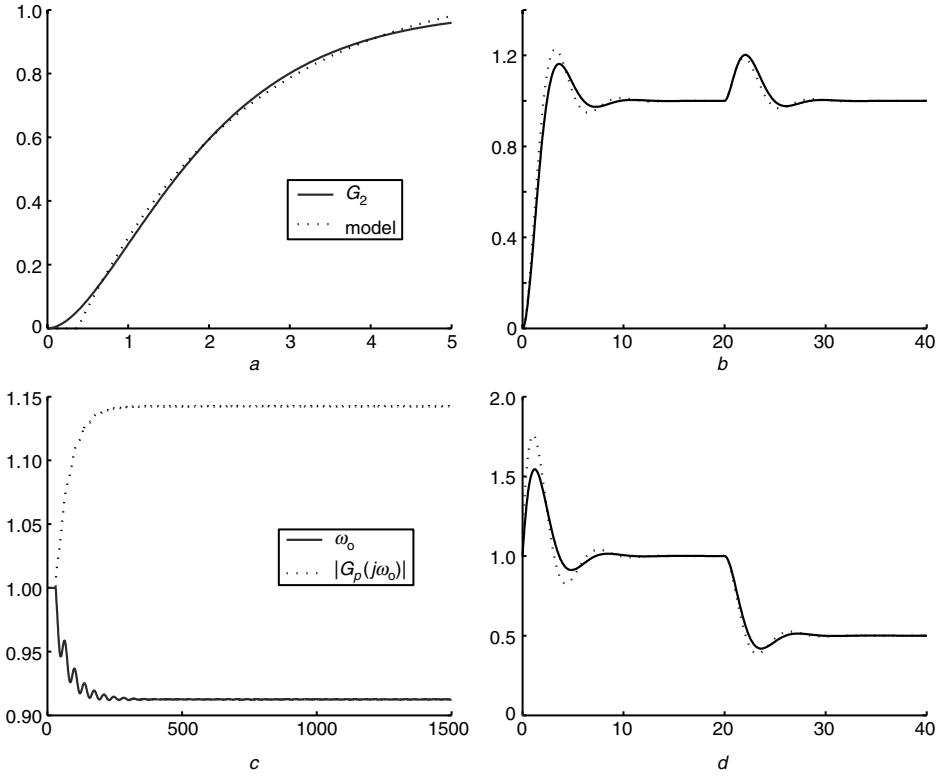

Fig. 8 *Results for G_2/PI*

a LS fit by FODT, $K_p = 1.1038$, $T = 2.107$, $T_d = 0.37273$, $\lambda = 0.1769$
b Response $\zeta_D = 0.5, a_D = 1$, initial $K = 1.1402$, $T_i = 0.91223$
c Adaptation (normalised) $K_a = 0.076148$, $m = 10$, $fr = [0.2 \ldots 0.8]$
d Control signal, final $K = 0.99789$, $T_i = 0.99975$

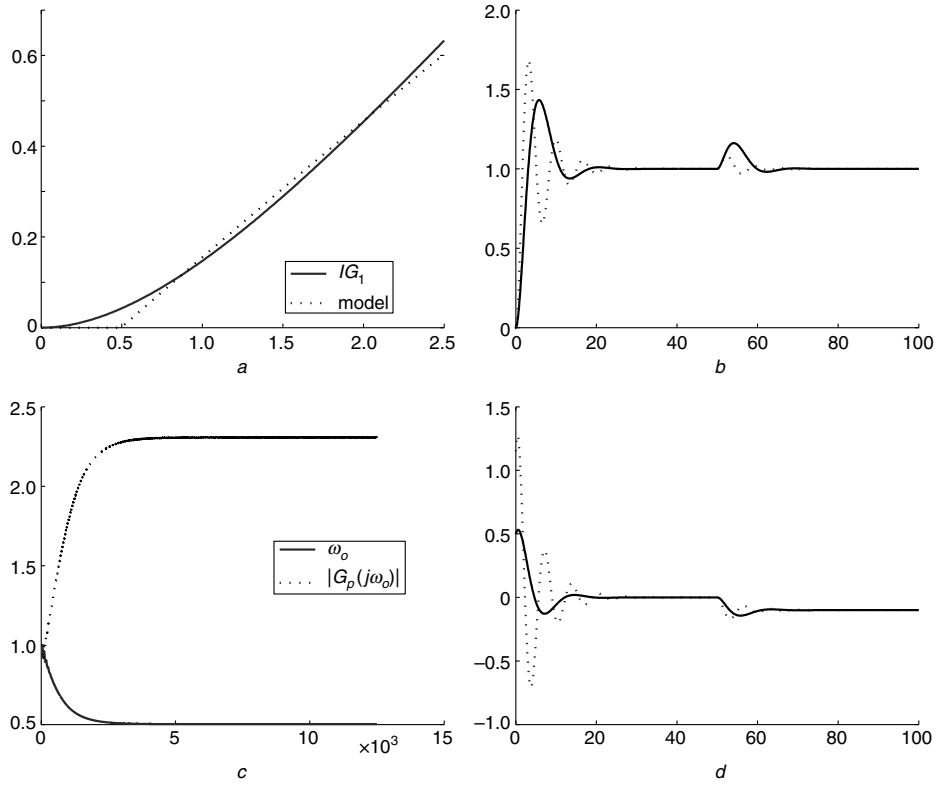

Fig. 9 *Results for IG_1/PI*

a LS fit by FODT, $K_p = 26.8669$, $T = 34.7091$, $T_d = 0.49526$, $\lambda = 0.014269$
b Response $\zeta_D = 0.5$, $a_D = 0.5$, initial $K = 1.1498$, $T_i = 2.0108$
c Adaptation (normalised) $K_a = 0.011473$, $m = 20$, $fr = [0.1 \ldots 0.9]$
d Control signal, final $K = 0.49825$, $T_i = 4.0029$

Table 1: Predicted and tuned controller parameters

	K	T_i
G_2/PI optimal	1.0	1.0
G_2/PI as tuned	0.99789	0.99975
IG_1/PI optimal	0.5	4.0
IG_1/PI as tuned	0.49825	4.0029

plant dead-time $\exp(-sT_d)$. Space precludes a full presentation here, and a suitable subset is shown. As a benchmark the results of [4] are chosen, as these are representative of best practice. Note however that [4] requires full prior knowledge of G_p and involves optimising over four parameters k, k_i, k_d, b in the control law

$$u(t) = k(by_{sp}(t) - y(t)) + k_i \int_0^t (y_{sp}(\tau) - y(\tau))d\tau + k_d\left(-\frac{dy(t)}{dt}\right)$$

and includes provision for set-point and output filtering. Given the number of degrees-of-freedom in their design approach, the resulting PI is likely to be 'as good as it gets'. Hence if the simple autotuning approach described here gives comparable performance, its effectiveness would be validated.

The first example is G_7/PI, giving the results shown in Fig. 10. As predicted by the discussion of expected root-locus behaviour for large k, there is evidence that the real mode is a little slow (the overshoot is not the expected 16%). The simulated plant corresponds to G_7 of [4]: their results show slightly more damping, yet have comparable overshoots and settling times.

It is of interest that in this case the initial tuning based on the FODT model is near-optimal. This behaviour is consistently seen with G_k/PI for large k and when the plant has additional dead-time, as then FODT gives an excellent fit. Hence the simple design-point results of (16) giving $K(0), T_i(0)$ can be recommended.

The second example of Fig. 11 is IG_3, which corresponds to $G_1$of [4]. Again because of the added integrator the FODT initialisation is poor and further tuning is required. The final set-point response is similar to that of Fig. 9, with the prescribed damping factor of 0.5. The corresponding plots in [4] show a little less overshoot to the set-point step (because of their use of a set-point prefilter), though the settling-times are comparable. For this example the results of [4] are better, but only marginally.

The noise-free simulations suggest that:

- the design-point approach gives good tuned responses having the desired damping factor and about the correct speed for the near-O real zero;
- the initial FODT model provides acceptable initial tuning, particularly for G_k/PI: only for $(NI)G_k/PI$ is further retuning necessary;
- the adaptive gain K_a and associated parameters appear at least to be reasonable. The tuning process takes about $10\times$ the system's settling time for type 0 plant, though is longer for type 1 G_p.

4.3 Comparison with other tuning rules

A 'tuning rule' denotes some simple procedure by which the controller's parameters K, T_i can be chosen based on the

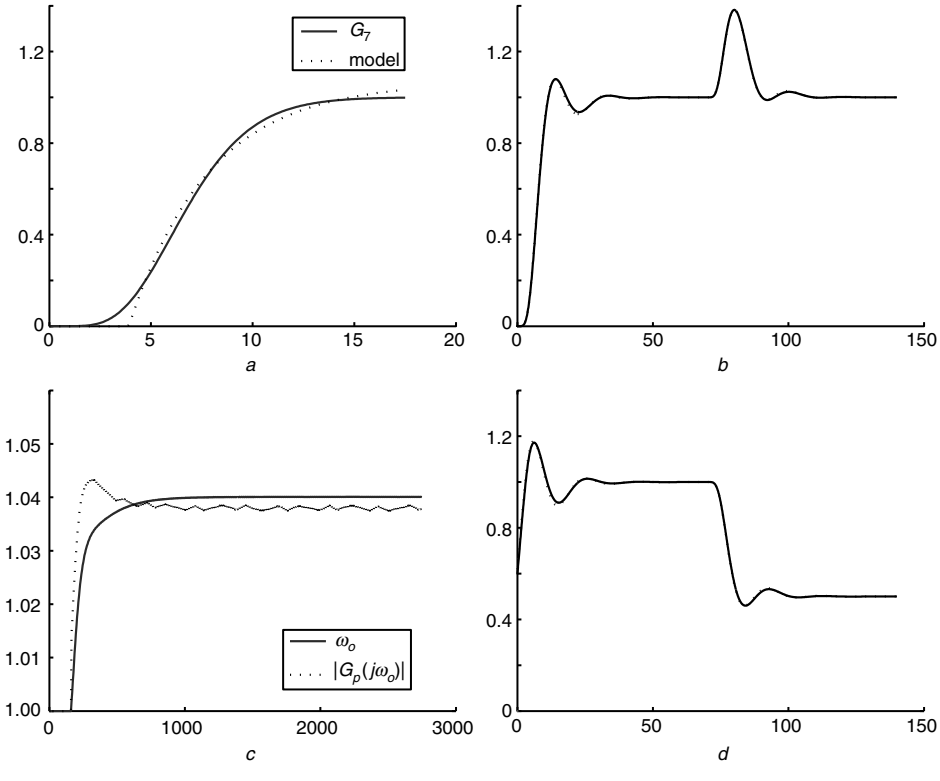

Fig. 10 *Results for G_7/PI*
a LS fit by FODT, $K_p = 1.0669$, $T = 3.943$, $T_d = 3.9083$, $\lambda = 0.99122$
b Response $\zeta_D = 0.5$, $a_D = 1$, initial $K = 0.61972$, $T_i = 4.5575$
c Adaptation (normalised) $K_a = 0.00083105$, $m = 10$, $fr = [0.2 \ldots 0.8]$
d Control signal, final $K = 0.59719$, $T_i = 4.3819$

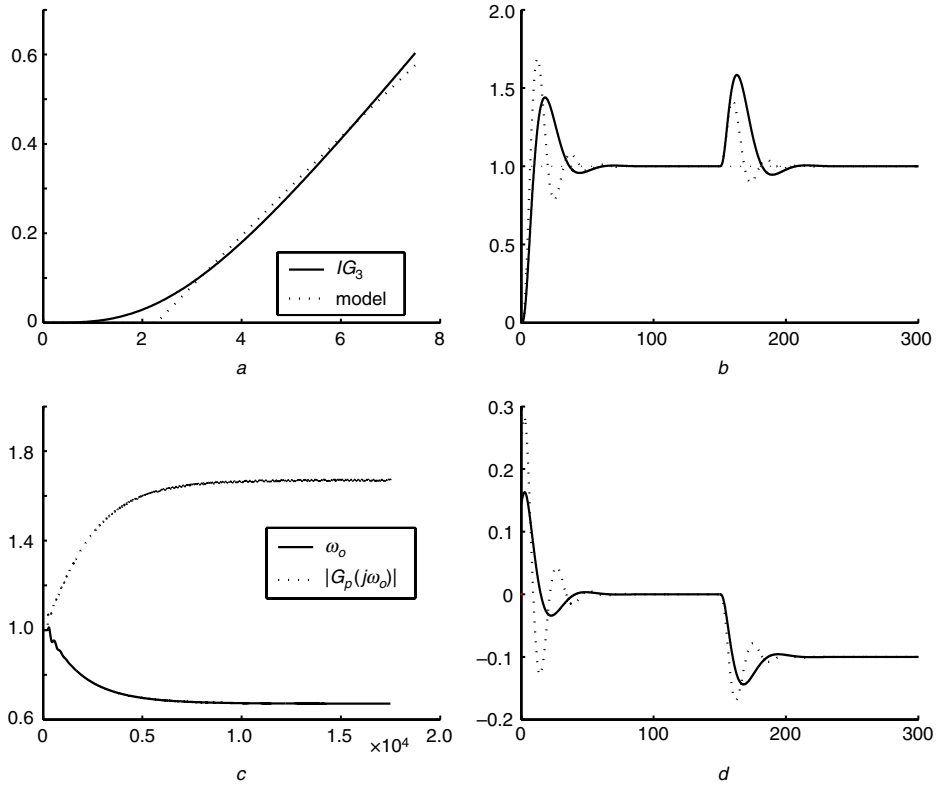

Fig. 11 *Results for IG_3/PI*

a LS fit by FODT, $K_p = 55.2572$, $T = 64.0132$, $T_d = 2.2865$, $\lambda = 0.03572$
b Response $\zeta_D = 0.5$, $a_D = 0.5$, initial $K = 0.24109$, $T_i = 8.6149$
c Adaptation (normalised) $K_a = 0.0010121$, $m = 20$, $fr = [0.1 \ldots 0.9]$
d Control signal, final $K = 0.1445$, $T_i = 12.8381$

results of a simple plant test. There are many such rules, but the classical benchmarks [2, 1] are

Ziegler–Nichols: if K_u, P_u are the plant gain and 'ultimate period' (corresponding to $\arg G(j\omega) = -\pi$ and obtainable by an autotuning relay test), then $K = K_u/2.2, T_i = P_u/1.2$. Tyreus–Luyben and other rules give a more conservative tuning from the same data.

Cohen–Coon: the plant is described by an FODT model, obtainable by a step-test, and the controller parameters are then:

$$K = \frac{1}{K_p}\left(0.083 + \frac{0.9}{\lambda}\right); \quad T_i = \frac{3.3 + 0.31\lambda}{1 + 2.2\lambda} T_d.$$

The following compares the closed-loop responses using the ZN and CC rules with those of the design-point method. For ZN, K_u, P_u are obtained numerically (i.e. exact values) from the given $G(s)$, while for CC the step-test that initialises the adaptor also provides the required CC tuning parameters K_p, λ, T_d. Note that the objective of the design-point method is to provide a closed-loop having a user-chosen damping-factor ζ. Of course the actual ζ will differ according to how much $G(s)$ differs from the assumed 'prototype': the interesting point is to see the size of the discrepancy. As a guide, the percentage overshoot to a step for $\zeta = 0.5, 0.4, 0.3$ should be about 16, 25, 37%, respectively.

Figures 12 and 13 give comparative results for G_3, G_{10}, with the DP tuned for a desired $\zeta = 0.5$. It is seen that for G_3, DP gives a response with damping very close to that specified, but for G_{10} the actual damping is a little lower. In contrast, both the ZN and CC tunings show a marked change of damping between the two cases, and note in particular that for G_{10} the ZN response clearly shows the 'slow mode' described. There is some trace of a slow mode in the DP response for G_{10}, but it is not nearly so marked. Repeating the exercise for a desired $\zeta = 0.4$ leads to a DP and CC tuning that are very close.

The next pair of Figures shown the results for $\exp(-0.1s)G_1$ (Fig. 14) and NG_2 (Fig. 15). The transfer function $\exp(-0.1s)G_1$ was chosen to correspond exactly to the assumed CC FODT model, and as both CC and ZN consistently show low damping, the desired ζ for DP was reduced to $\zeta = 0.3$. Figure 14 shows that in this case the tuned results are quite close: the excessively large initial

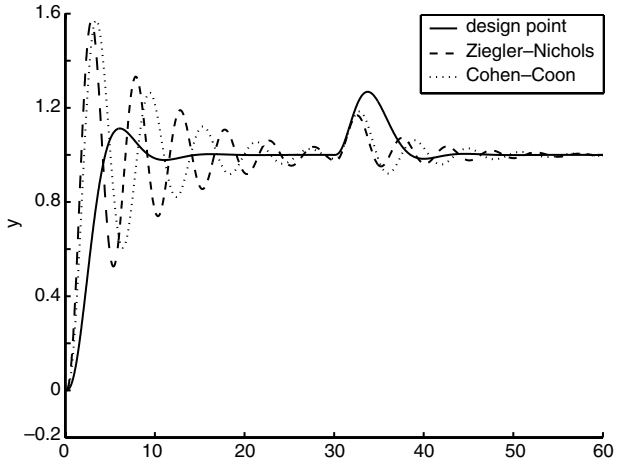

Fig. 12 *Comparing design-point tuning with Ziegler–Nichols and Cohen–Coon for $G_3(s)$*

DP($\zeta = 0.5$)$K, T_i = [0.76842, 1.7322]$ $Z-N = [3.6364, 3.023]$ $C-C = [2.3901, 1.841]$

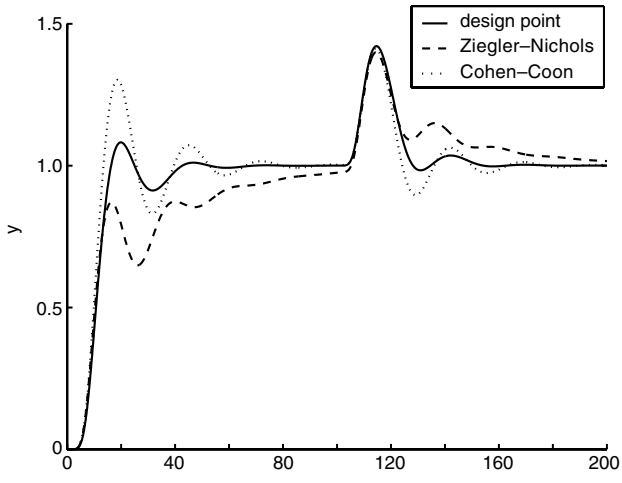

Fig. 13 *Comparing tunings for $G_{10}(s)$*
DP($\zeta = 0.5$) K, $T_i = [0.56587, 6.3146]$ Z–$N = [0.75078, 16.1147]$ C–$C = [0.68903, 5.7964]$

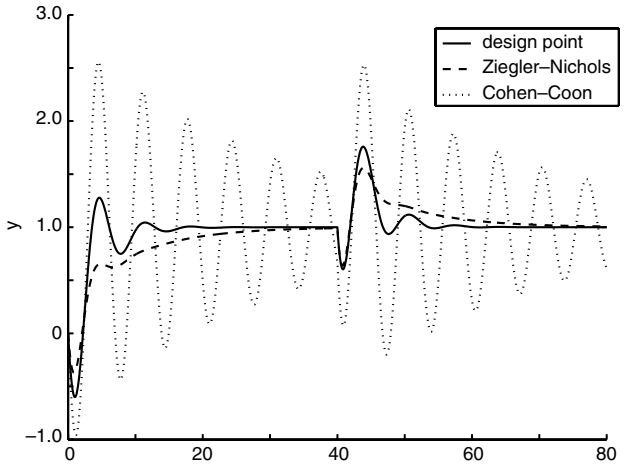

Fig. 15 *Comparing tunings for $NG_2(s)$*
DP($\zeta = 0.4$) K, $T_i = [0.59026, 2.3119]$ Z–$N = [0.45455, 3.7024]$ C–$C = [0.74278, 1.8412]$

overshoot is due in part to the position of the PI's zero. Figure 15 shows considerable variety in the tuned behaviour for the nonminimum-phase plant NG_2. CC exhibits very low damping, ZN has high damping and a slow mode: the DP response is good.

These results reflect those obtained by a series of trials on candidate $G(s)$: the ZC/CC methods show variability of tuning, but DP is relatively consistent, with responses determined by the chosen ζ. Of course, ZN in particular has a deliberately low damping so that high gains K (good for disturbance rejection) can be asserted. The GM/PM method of [13] leads to higher damping, but needs an FODT model. Low damping usually implies low loop robustness: indeed the DP adaptor can be unstable if the chosen damping is less than about 0.25.

The inconsistency of CC is perhaps due to the simple FODT plant model adopted. For NG_k, IG_k an FODT model is poor, as shown by the large frequency change in the DP adaptor during convergence to the design point. The inconsistency of ZN is because plant behaviour around the ultimate frequency (open-loop phase of $-\pi$) is not particularly representative of that around the closed-loop bandwidth (closed-loop phase $\approx -\pi/2$), which determines the closed-loop damping. The DP adaptor adjusts the working frequency until the closed-loop phase is 'just so':

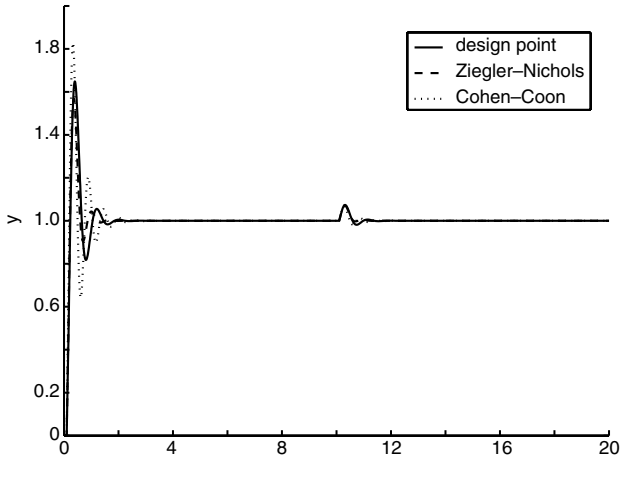

Fig. 14 *Comparing tunings for $\exp(-0.1s)G_1(s)$*
DP($\zeta = 0.3$) K, $T_i = [6.3479, 0.22993]$ Z–$N = [7.4321, 0.32083]$ C–$C = [9.1267, 0.27234]$

the cost of the approach is the time taken (typically $10\times$ the plant's settling time).

4.4 Tracking a closed-loop

As the method gives acceptable tuning for fixed G_p with a constant set-point, its behaviour for time-varying $G(s)$ and set-points becomes of interest. To assess the algorithm's tracking ability, the plant

$$VG_3(s) = \frac{\alpha^2}{(s+\alpha)^3}$$

was simulated. The parameter α was set to 1 for $t = 0 \ldots 1000$ and then ramped to three over the range $t = 1000 \ldots 5000$. Hence during this period the plant's gain K_p and its time-scale both fall by factors of three. For effective tuning, therefore, K should increase by $3\times$ and T_i reduce accordingly during this period.

Tracking under normal operating conditions stresses any algorithm because of

noise: degrades the estimator and hence slows the tracking ability;
excitation: the signal power of set-point variations might swamp the estimator and cause convergence to incorrect values;
actuator nonlinearity: often results in small-signal behaviour that is unrepresentative of the required large-signal model. This problem is discussed in [7].

The PFE method has a certain defence against these problems, i.e. its inherent filtering action, that is tested by the simulations reported as follows. The sinusoidal amplitude was maintained at $A = 0.1$ and (where appropriate) the noise power was chosen to give an SNR of 0.5. A squarewave set-point was imposed with amplitude 0.5, between levels 0.5 and 1.5, with a switching interval of 250 seconds. To simulate actuator nonlinearity, a backlash element was inserted into the loop having a user-chosen bandwidth b. To simplify comparisons, initialisation was chosen to depend on the FODT parameters from the noise- and backlash-free system.

The first trial of Fig. 16 had $\{b = 0; S_n = 0\}$ and shows how set-point changes affect the tuning process. It is seen from the Figure that each step has some small effect on ω_0 but not on the gain estimate. The algorithm is able to track the plant variations, and the final tuned loop (as shown by

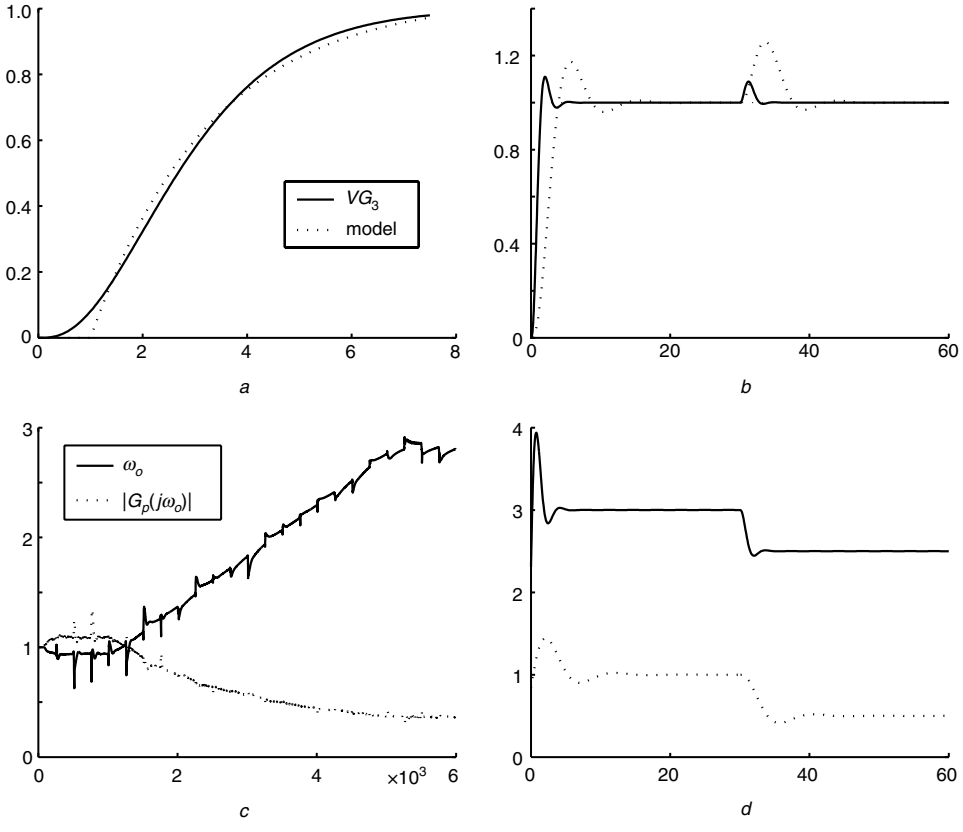

Fig. 16 *Results for VG_3/PI*

a LS fit by FODT, $K_p = 1.0341$, $T = 2.3026$, $T_d = 1.0005$, $\lambda = 0.43451$
b Response $\zeta_D = 0.5$, $a_D = 1$, initial $K = 0.83798$, $T_i = 1.6267$
c Adaptation (normalised) $K_a = 0.011323$, $A = 0.1$, $m = 10$, $fr = [0.1 \ldots 0.9]$
d Control signal: backlash = 0, final $K = 2.3093$, $T_i = 0.58057$

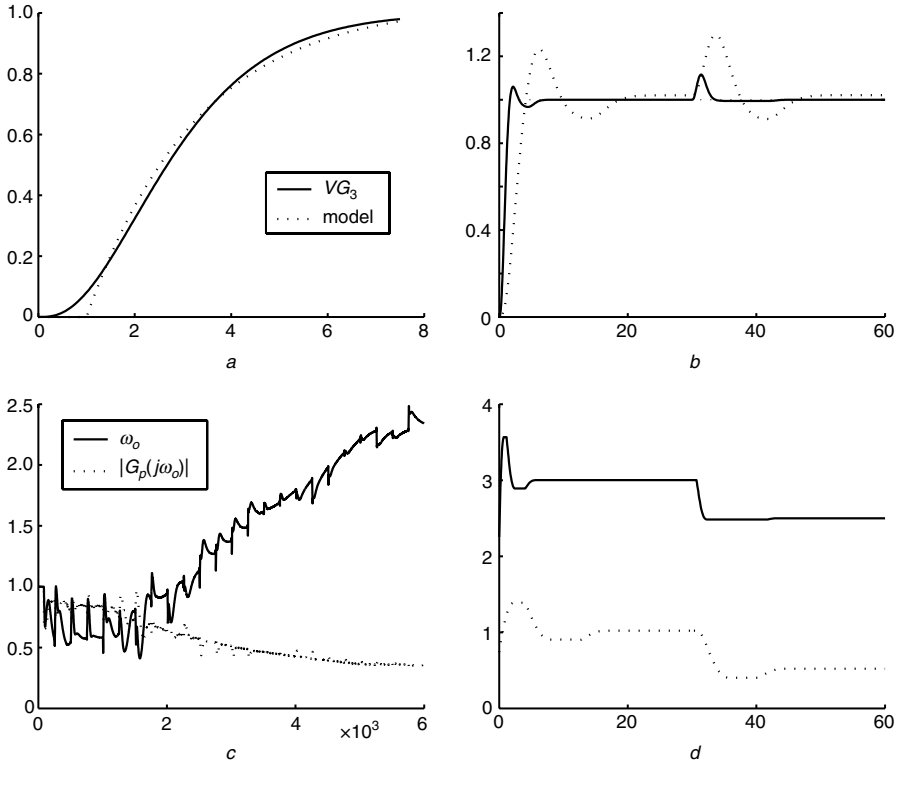

Fig. 17 *Tuning VG_3/PI + backlash ...*

a LS fit by FODT, $K_p = 1.0341$, $T = 2.3026$, $T_d = 1.0005$, $\lambda = 0.43451$
b Response $\zeta_D = 0.5$, $a_D = 1$, initial $K = 0.83798$, $T_i = 1.6267$
c Adaptation (normalised) $K_a = 0.011323$, $A = 0.1$, $m = 10$, $fr = [0.1 \ldots 0.9]$
d Control signal: backlash = 0.2, final $K = 2.3546$, $T_i = 0.6934$

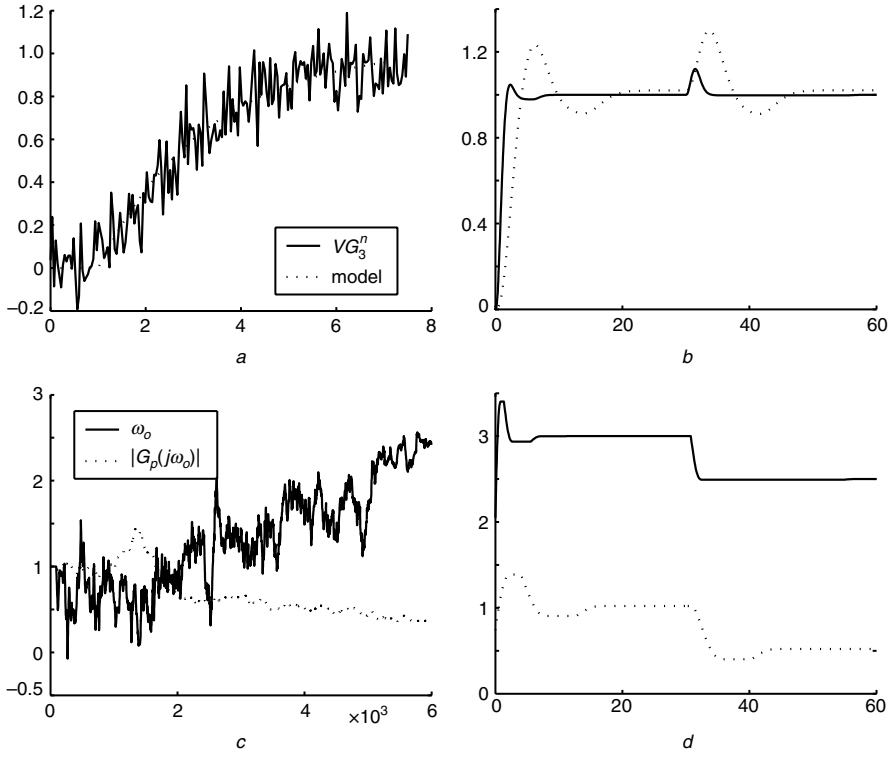

Fig. 18 *... and with noise*

a LS fit by FODT, $K_p = 1.0341$, $T = 2.3026$, $T_d = 1.0005$, $\lambda = 0.43451$
b Response $\zeta_D = 0.5$, $a_D = 1$, initial $K = 0.83798$, $T_i = 1.6267$
c Adaptation (normalised) $K_a = 0.011323$, $A = 0.1$, $m = 10$, $fr = [0.1 \ldots 0.9]$
d Control signal: backlash = 0.2, final $K = 2.1496$, $T_i = 0.67505$

the step-test) is as required, having a damping factor $\zeta \approx 0.5$ and being about $3\times$ faster than the original tuned loop (whose overshoot exceeds 16%, as it is based on the FODT initial model).

The next test investigates the effect of backlash, and for the results of Fig. 17 its bandwidth b was set to 0.2 (i.e. the backlash bandwidth is comparable with the control amplitude associated with the sinusoidal perturbations).

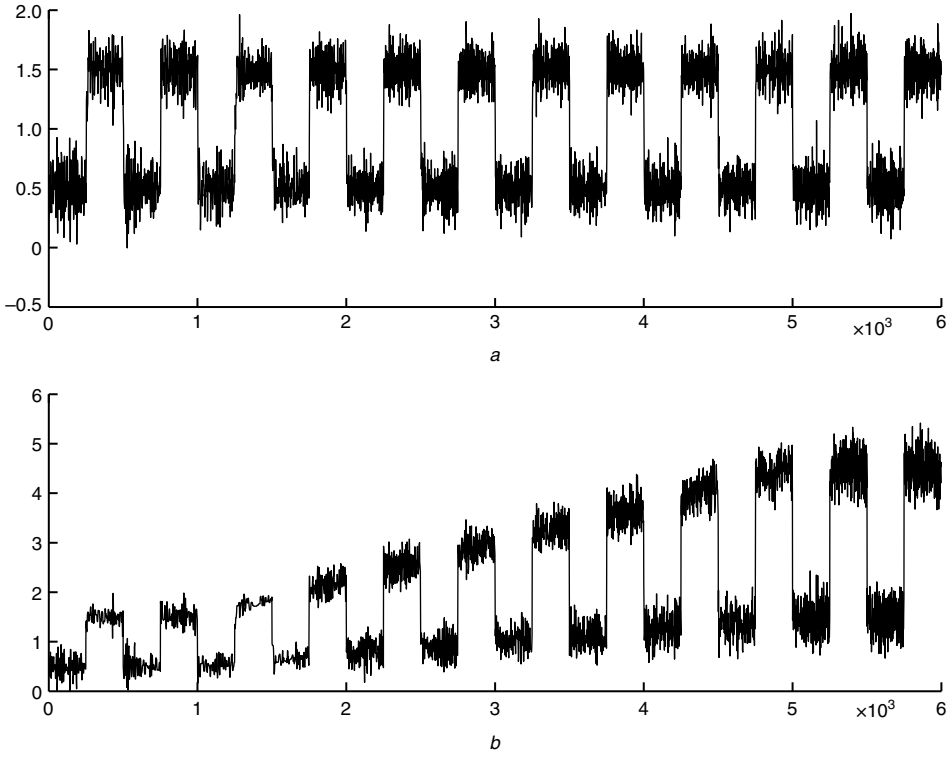

Fig. 19 *Plant input and output signals during tuning*

a Plant output during tuning
b Control input

This nonlinearity has degraded the tuning, though not excessively for the chosen value of b. However, increasing b indeed causes the loop to become unstable during tuning.

Finally, noise was added to the plant output. Figure 18 shows the tuning process (indicating a further small deterioration) and Fig. 19 presents the plant's input and output signals These results demonstrate that, despite the challenge of poor SNR and relatively large backlash, the tuning remained effective.

5 Conclusions

Combining the design-point method and a phase–frequency estimator has produced a PI-tuning algorithm that appears to work well for a range of test cases. The approach is simple computationally, and its associated parameters (such as adaptive gain) can be computed in a semiautomatic way based on the results of a prior step-test. This initialisation procedure leads to adaptation speeds that are predictable and rather faster than those of the PLL method in [9, 10]. Compared with the well-established relay-based or FODT autotuning methods, the tuned responses are more consistent, particularly for plants G_p where neither the ultimate gain–period (K_u, P_u) nor an FODT model is a good indicator of the plant's behaviour around the closed-loop bandwidth.

The algorithm can, as demonstrated, be used to keep a loop continuously in tune at the expense of adding a low-amplitude perturbing sinewave signal to its set-point. In practice it is more likely that a larger-amplitude signal would be injected when providing 'tuning on demand'. By retaining the existing PI regulator in the loop, our approach is simpler than the 'controller monitor' of [1] which (on demand) replaces the PI regulator by a relay with hysteresis.

In a typical process-control loop the main cause of discrepancy between small- and large-signal behaviour is the nonlinear actuator characteristic (e.g. a flow valve with stiction). The modem trend is to embed microelectronics for actuator 'validation' [14], one purpose of which is to model and thus to compensate for this nonlinearity. One possibility is to couple our tuning procedure with a modern validated actuator.

The method admits several extensions. Provided that the frequencies are not 'too close' [15], a phase-locked loop can accurately lock-on to a given sinewave despite the presence of sinusoidal interference. Hence it is possible to develop a MIMO loop-tuner or one that can tune all three PID terms using two simultaneously-injected sinewaves. A recent application (to be reported) on a flow rig whose dynamics vary with mean flow-rate, shows the DP adaptor to work well in practice, leading to well-tuned responses throughout the flow range.

6 References

1 Yu, C.-C.: 'Autotuning of PID controllers' (Springer-Verlag, London, UK, 1999)
2 Tan, K.K., Wang, Q-G., Hang, C.C., and Hägglund, T.J.: 'Advances in PID control' (Springer-Verlag, London, UK, 1999)
3 Special section on PID control, *IEE Proc., Control Theory Appl.*, 2002, **149**, (1), pp. 1–81
4 Panagopoulos, H., Åström, K.J., and Hägglund, T.: 'Design of PID controllers based on constrained optimisation', *IEE Proc., Control Theory Appl.*, 2002, **149**, (1), pp. 32–40
5 Åström, K.J., and Hägglund, T.: 'Automatic tuning of simple regulators with specifications on phase and amplitude margins', *Automatica*, 1984, **20**, (5), pp. 645–651
6 Clarke, D.W., and Hinton, C.E.: 'Adaptive control of a materials-testing machine', *Automatica*, 1997, **33**, (6), pp. 1119–1131
7 Hägglund, T., and Åström, K.J.: 'Supervision of adaptive control algorithms', *Automatica*, 2000, **36**, pp. 1171–1180
8 Hind, E.C.: 'Controller selection and tuning', *Trans. Inst. Meas. Control*, 1980, **2**, (1), pp. 46–56
9 Crowe, J., and Johnson, M.A.: 'Towards autonomous PI control satisfying classical robustness specifications', *IEE Proc., Control Theory Appl.*, 2002, **149**, (1), pp. 26–31
10 Johnson, M.A., and Crowe, J.: 'New methods of nonparametric methods for PID control'. Presented at the IFAC Workshop on Adaptive Control and Signal Processing, Glasgow, UK, 1998
11 Clarke, D.W., and Park, J.W.: 'Phase-locked loops for plant tuning and monitoring', *IEE Proc., Control Theory Appl.*, 2003, **150**, (2), pp. 155–169
12 Clarke, D.W., and Park, J.W.: 'Initialising an adaptive phase/frequency estimator'. Technical report, Department of Engineering Science, Oxford University, 2002
13 Ho, W.K., Hang, C.C., and Cao, L.S.: 'Tuning of PID controllers based on gain and phase margins', *Automatica*, 1995, **31**, (3), pp. 497–502
14 Yang, J.C.Y., and Clarke, D.W.: 'The self-validating actuator', *Control Eng. Pract.*, 1999, **7**, (3), pp. 249–260
15 Clarke, D.W.: 'Designing phase-locked loops for instrumentation applications', *Measurement*, 2002, **32**, (3), pp. 205–227