Phase-locked loops for plant tuning and monitoring

D.W. Clarke and J.W. Park

Abstract: A loop's phase-margin ϕ_m provides helpful information: often it predictably affects damping, and the corresponding frequency ω_m is related to the closed-loop bandwidth. However, its use in controller design is frequently hampered by lack of knowledge of the plant's transfer-function $G(s)$. Hence, as with the popular autotuning methods, the authors consider a simple algorithm that automatically determines the frequency ω_d and gain $|G(j\omega_d)|$ corresponding to a prescribed plant phase-lag ϕ_d as this information can be used simply to tune a PI regulator This adaptive approach is shown to have the structure of a phased-locked loop (PLL). The key ingredient in a PLL is its phase-sensitive detector (PSD), as phase cannot be estimated instantaneously and the deduced value is affected by harmonics and noise. Four common PSDs are compared and it is found that the Hilbert-transform PSD has uniformly effective behaviour. Predictions of the convergence and noise-rejection properties of the algorithm are presented and confirmed via simulation. The same algorithm can be used for loop monitoring, in which plant variations are tracked via on-line estimation of $\omega_d(t)$. A benefit of the algorithm is that it enables active probing of a plant by low-amplitude sinewaves, as good tracking is possible even with signal-to-noise ratios much less than one.

1 Introduction

A central approach in classical control design via the Nyquist diagram is to determine the controller gain such that the closed-loop has a prescribed phase or gain margin. When the plant's transfer-function $G(s)$ is unknown, often the case in process control, a common solution is autotuning [1, 2], based on a relay experiment that reveals the critical frequency ω_d particular plant phase ϕ_d and deduces the corresponding plant gain K_c. Relay-based methods are deservedly popular as they lead automatically to simple yet effective tuning of PID regulators via Ziegler–Nichols [3] and related rules (see [2] for a full set of references). On the other hand, the method relies on the approximations inherent in describing-function analysis, leading to errors if the plant fails to filter adequately the higher-order harmonics of the injected square-wave. Our first objective is to provide a tuner that avoids these problems, yet is still simple to implement and apply.

Plant dynamics can change with time: for example heat-exchangers foul and pipes become blocked. A loop monitor is an on-line algorithm designed to detect this event such that plant repair or controller re-tuning can be scheduled. Most monitors are passive and are based on the assumption of a minimum-variance control law, e.g. [4], although [2] describes an active monitor that periodically activates a relay experiment. Our contention is that active monitors are much better at detecting anomalies and changes in dynamics, and propose a monitor that determines any drift in the critical frequency ω_d. The important point about such a monitor in applications is that the effect of the probing signal should be minimal in order to sustain output quality. Hence, the design has to be effective when the output response to the test signal is buried in the noise: this is not the case with relay-based autotuning designs.

Consider the steady-state response of a stable plant $G(s)$ to a sinusoidal input $u(t) = a \sin \omega t$:

$$y(t) = A \sin(\omega t + \phi) = a|G(j\omega)| \sin(\omega t + \arg G(j\omega)) \quad (1)$$

where $|G|$ and arg G are the plant's gain and phase at the frequency ω, respectively. For (1) to be valid, the input frequency has to be constant and t large enough such that effect of initial conditions has decayed to zero. To fix ideas and provide a running example, the 'reference' transfer-function:

$$G_r(s) = \frac{50(1 + s/10)}{s(1 + s)(1 + s/100 + (s/100)^2)} \quad (2)$$

is adopted, whose frequency response is shown in Fig. 1. Fig. 1*a* is the phase $\phi = \arg G_r$ in degrees plotted on a logarithmic frequency scale; Fig. 1*b* is the Nyquist diagram whilst Fig. 1*c* is an expanded Nyquist diagram that explores the high-frequency behaviour of G_r. In all cases the dashed line denotes a desired phase $\phi_d = -3\pi/4$ (i.e. corresponding to a design phase-margin of $\pi/4$). The points P1, P2, P3 and Q1, Q2, Q3 indicate solutions to $\phi = \phi_d$ thus determining three possible phase-margin frequencies $\omega_d = \omega_{1,2,3}$.

The problems dealt with by this paper are:

tuning: to derive an algorithm that, starting from a initial 'guessed' frequency $\omega_0(0)$, is such that $\omega_0(t) \rightarrow \omega_d$; to determine which of $\omega_{1,2,3}$ are stable convergence points, and to provide a theory determining the speed of response that offers guidelines to the choice of 'adaptive gain';

© IEE, 2003
IEE Proceedings online no. 20030140
DOI: 10.1049/ip-cta:20030140
Paper first received 30th May and in revised form 7th October 2002

The authors are with the Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, UK

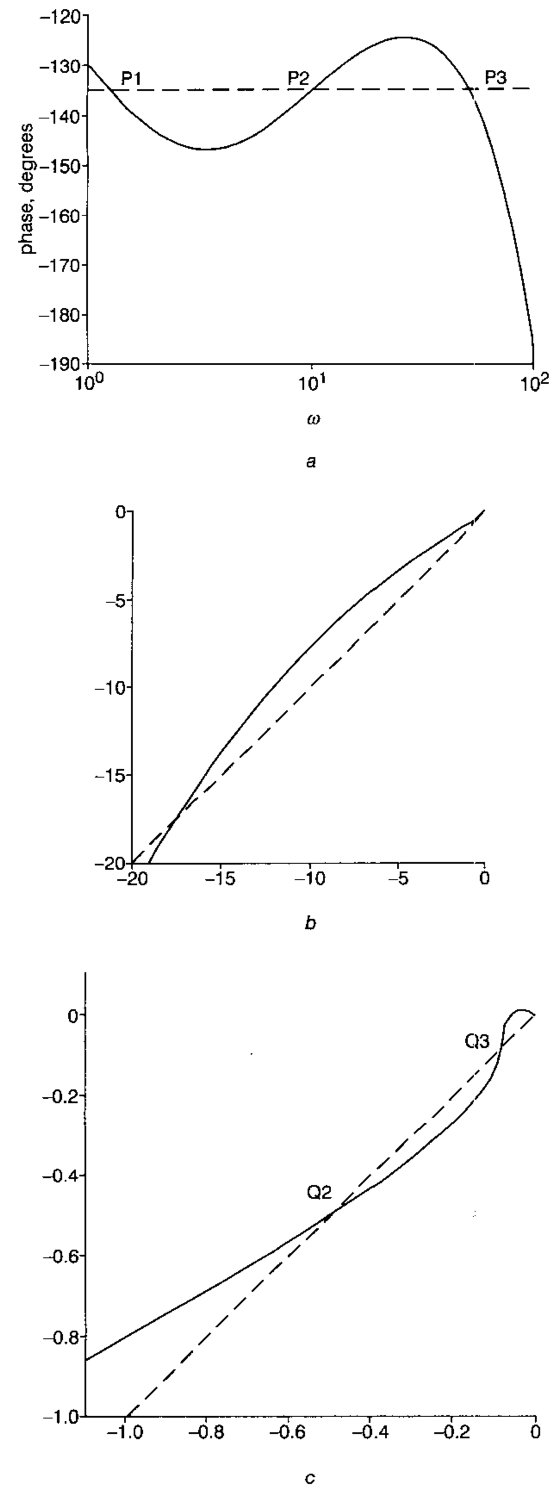


Fig. 1 *The behaviour of $G_r(s)$*
a Phase plot
b Nyquist diagram
c Expanded Nyquist diagram

monitoring: given $\omega_0(0) = \omega_d(0)$ (correct initial tuning), to determine how the signal-to-noise ratio (SNR) on $y(t)$ affects the estimated $\omega_d(t)$ such that significant changes requiring loop re-tuning can still be detected reliably.

The basic idea is depicted by the block diagram of Fig. 2. A sinewave generator injects $u(t) = a\sin\omega_0 t$ and an estimator determines the plant's output phase ϕ, leading to a phase error:

$$e(t) = \phi_d - \phi(t) \tag{3}$$

$$= \phi_d - \arg G(j\omega_0(t)) \tag{4}$$

Of course there is the fundamental 'frozen parameter' assumption common in much analysis of adaptive algorithms: the frequency $\omega_0(t)$ is taken to be so slowly varying that transients can be ignored and the idea of a constant phase is meaningful. In the limit, given $\omega_0 \to \omega_d$, there will indeed be a valid steady-state response, but we expect that the transient behaviour of the algorithm would be anomalous if rapid convergence were demanded. The simple adaptor we adopt asserts:

$$\frac{d\omega_0}{dt} = -Ke(t) = K(\arg G(j\omega_0) - \phi_d) \tag{5}$$

where $K > 0$ is the adaptive gain. Consider Fig. 1*a*: to the left of P1 we have $e = -3\pi/4 - \phi < 0$ thus $d\omega_0/dt > 0$ and our frequency estimate $\omega_0(t) \to \omega_{P1}$. Between P1 and P2 we have $e > 0$, thus $d\omega_0/dt < 0$ and ω_0 again tends to the frequency for P1. Hence P1 is clearly a stable convergence point. Similar arguments show that P3 is also a stable convergence point but that P2 is not. Hence for an initial $\omega_0(0)$ lying between zero and ω_{P2} we expect $\omega_0 \to \omega_{P1}$ whilst for $\omega_0(0)$ between ω_{P2} and ∞ we expect $\omega_0 \to \omega_{P3}$.

The algorithm is simple, indeed trivial, and the interesting problems of analysis and design lie not in (5) but in the phase estimator. There are many approaches to the design of such a component and their properties can be characterised by:

linearity: is the signal $S_e(t)$ provided by the estimator a linear function of the phase [phase error] $\phi(t)$ [$e(t)$]?
amplitude independence: is S_e independent of A, or does A have to be independently estimated for consistent adaptive behaviour?
harmonics: does S_e include harmonic (second or higher) components that will affect the estimated frequency ω_0?
noise rejection: is the rejection of noise optimal and sufficient for practical application of the algorithm?
dynamics: does the phase estimator have inherent dynamics that affect the algorithm's convergence?

Two important classes of estimator use synchronous demodulation (which can be used as the underlying frequency $\omega_0(t)$ is known) or zero crossing (cheerful and cheap to implement). Synchronous demodulation, as will be seen, has excellent noise-rejection properties, although necessary filtering introduces dynamics into the adaptive loop. Zero-crossing algorithms are inherently discrete-time in nature and their noise-rejection is poor.

A long-standing approach to experimental gain/phase estimation is the so-called 'transfer-function analyser', based on synchronous demodulation and depicted in Fig. 3. With an input sinewave $u(t) = a\sin\omega t$ the plant output is $y(t) = aK_p\sin(\omega t + \phi)$. The estimator also generates $a\cos\omega t$, and the outputs of the product terms in the Figure are:

$$s_x(t) = y(t)a\sin\omega t = a^2K_p(\sin^2\omega t\cos\phi + \cos\omega t\sin\omega t\sin\phi)$$

$$s_y(t) = y(t)a\cos\omega t = a^2K_p(\sin\omega t\cos\omega t\cos\phi + \cos^2\omega t\sin\phi)$$

An exact approach would integrate s_x, s_y, over an integral number of sinewave periods, giving an intermittent output and so is appropriate for discrete-time adaptation of ω_0.

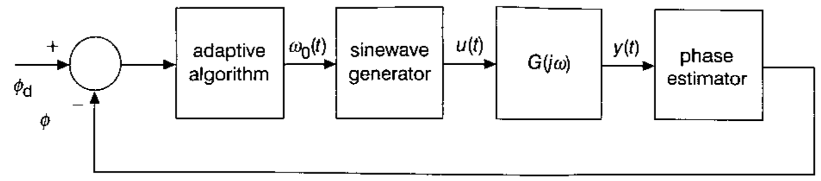


Fig. 2 *Conceptual diagram of a phase/frequency estimator*

The idea of the low-pass filtering in the Figure is to produce continuous signals X, Y by noting that $\langle\sin^2\rangle = 0.5 = \langle\cos^2\rangle$ and $\langle\sin\cos\rangle = 0$, i.e. the double-frequency components are filtered to zero. Hence:

$$X \approx 0.5a^2K_p\cos\phi \quad Y \approx 0.5a^2K_p\sin\phi$$
$$\to K_p = \frac{2\sqrt{X^2+Y^2}}{a^2} \, \phi = \text{arc}\tan(Y, X)$$

Note that the estimate of ϕ is independent of the input amplitude a.

A phase estimator based on Fig. 3 would provide a perfectly adequate module for the adaptive algorithm of Fig. 2, but we can do better. The problem is that the filter bandwidths need to be narrow in order to remove adequately the second-harmonic component (which will affect the estimated ω_0): this will slow down convergence. On the other hand 'heavy' filtering is useful to eliminate also the effect of noise, and for plant monitoring convergence speed is less important. Nevertheless an alternative approach, based on the phase-locked loop (PLL) gives a more elegant solution.

In the following Sections the use of a PLL in this application is outlined, showing the central role of its phase-sensitive detector (PSD), and four representative PSDs are described. Local-linearisation of phase/frequency behaviour leads to convergence-rate predictions, confirmed by simulation. The phase/frequency estimator is applied to tracking the Nyquist plot of a third-order system G_3 by asserting a phase ramp and then to finding the phase-margin frequencies of G_r. A noise analysis provides an experimentally-verified prediction of variance of the loop's frequency estimate. The four described PSDs are compared for transient and noise-rejection response. To assess their viability for plant monitoring, the PLLs (afforced by a narrow band-pass filter having a variable central frequency) are applied to G_3, when it is subjected to adverse output signal:noise ratios down to 1:100.

As there are many different PSDs [5, 6], the corresponding loops are named according to the acronyms of Table 1. Except for {PSD$_1$, PLL$_1$} (the first PLL to be developed in the literature), a method is denoted by, say, {HTPSD, HTPLL} according to whether it is the detector or the overall loop that is under discussion.

Table 1: Acronyms

HT	Hilbert transform
PLL	Phase-locked loop
PLL$_1$	PLL using PSD$_1$
PSD	Phase-sensitive detector
PSD$_1$	Analogue-multiplier PSD
SDM	Synchronous demodulation
VBPF	Variable band-pass filter
ZC	Zero-crossing

2 The PLL approach

The inspiration for our work is the neat idea of [7–9], independently proposed by [10, 11], that the structure of Fig. 2 is similar to that of a PLL, as depicted by Fig. 4. PLLs have widespread and effective (if sometimes temperamental) application in communications, and have been extensively analysed, e.g. [5, 6, 12]. To fix the standard nomenclature, the idea of a PLL is to 'lock' the phase $\phi_d(t)$ of an internally-generated signal $\cos\phi_0(t)$ (produced by a 'voltage-controlled oscillator' VCO) to that of the incoming signal $A\sin\phi_i(t)$. The PSD provides an error signal $S_0(t)$ that depends on $\phi_i - \phi_0$ and the 'loop filter' is designed to ensure that $S_e \to 0 \Rightarrow \phi_0 \to \phi_i$. The integrator within the VCO arises from the instantaneous frequency/phase property $\omega_0(t) = d\phi_0(t)/dt$: a constant ω_0 induces a ramp in ϕ_0 and hence a pure sinusoid with frequency ω_0. The loop filter can take many forms, but an effective practical $C(s)$ is just a PI regulator. In some PLL designs a lead-lag network is used instead of a PI. However in our algorithm we want the steady-state error to be precisely zero, so an integral term in $C(s)$ is necessary.

Note that in standard PLL applications $C(s)$ cannot be only an integrator (there has to be a zero else loop stability is lost) and that major questions in PLL design involve lock-in time (when $\omega_i \neq \omega_0(0)$) and 'cycle-skipping' under high-noise conditions. These points reflect the fact that normally the input frequency is unknown but in our application the input frequency is just that injected by the VCO into the plant and hence always available. This simplifies the analysis considerably, so that emphasis is again placed on appropriate PSD design.

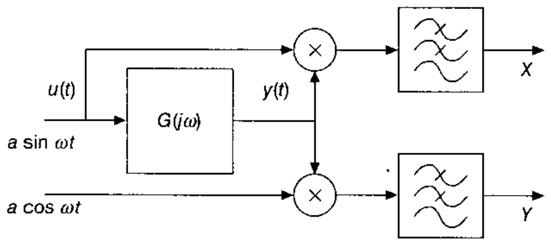


Fig. 3 *Synchronous demodulation for gain/phase estimation*

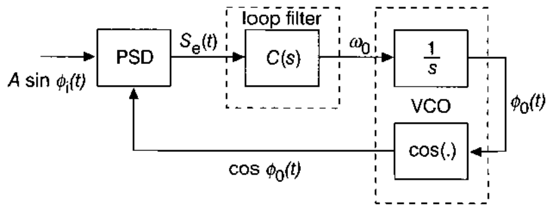


Fig. 4 *Conceptual diagram of a PLL*

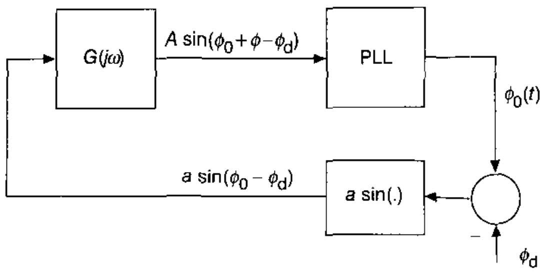


Fig. 5 *Using a PLL for phase-margin tracking*

The simplest PSD (which we call PSD_1 and the corresponding phase-locked loop PLL_1) found in the literature is an analogue multiplier that gives:

$$\begin{aligned} S_e(t) &= A \sin\phi_i(t)\cos\phi_0(t) \\ &= 0.5A[\sin(\phi_i - \phi_0) + \sin(\phi_i + \phi_0)] \\ &\approx 0.5Ae(t) + DFS \end{aligned}$$

provided the error $\phi_i - \phi_0 = e$ is small. When PLL_1 is locked, the integrator in $C(s)$ ensures that the average error $\bar{e} = 0$, so $\omega_0 = \omega_i$ but the double-frequency sinewave (DFS) due to PSD_1 remains in the loop and causes $e(t)$ to vary sinusoidally around its mean value of zero. In typical PLL applications this is unimportant (the DFS frequency is well outside the bandwidth of the PLL), but for our adaptor we would need to add a narrow-band low-pass filter to remove its effect. Hence PLL_1 is characterised by: nonlinearity amplitude dependence, harmonic content, instantaneous (but only if no filter is used and A is precisely known). It has been superseded in communications applications by all-digital methods, but retained here as its properties have been extensively analysed and its noise-rejection is effective.

The PLL is converted to the required phase/frequency estimator by the simple configuration of Fig. 5: a sinewave of amplitude a and frequency ω_0 is generated, lagging the PLL's internal phase ϕ_0 by a prescribed phase-shift ϕ_d. Hence, in the steady-state, $\phi_i = \phi_0 + \phi - \phi_d$ is locked by the PLL to ϕ_0 and thus $\phi = \phi_d$ as required. Comparing with the conceptual diagram of Fig. 2, it is seen that the adaptive algorithm of (5) corresponds to the loop filter $C(s)$. Unlike standard PLLs, where $C(s)$ requires a zero to ensure loop stability, in this application a simple integrator of gain K is sufficient. Hence we turn to properties of representative PSDs appropriate for the adaptive algorithm.

3 PSDs

Simulink diagrams of the four considered PSDs are shown in Fig. 6. The input nodes correspond to signals $s_i(t)$ (the plant's response, being the upper input to the algorithm) and $\phi_0(t)$ (the PLL's current phase), whilst the output is the (possibly filtered) error signal $S_e(t)$ that drives the updating integrator. Excluded from the Figure, though necessary in applications, are the corresponding estimators of the signal amplitude $\hat{A}$ (leading to plant-gain estimates $|\hat{G}| = \hat{A}/a$). Fig. 6*a* is the analogue multiplier PSD_1, discussed above, with the addition of a Butterworth low-pass filter $F_B(S)$ that can optionally attenuate the DFS inherent in the S_e of PSD_1. For the simulations described below, a fourth-order filter with breakpoint frequency ω_f was adopted: in practice the design obviously depends on the unknown ω_d. The other PSDs of the Figure are now discussed in turn.

3.1 Phase-frequency detector using ZCs

The PLL algorithm of [7] is based on a 'digital identifier module' that evaluates phase using 'peaks and troughs', i.e. a once-per-cycle measurement; in [8] the authors comment that a ZC approach is a viable alternative. Hence the simpler ZC method is proposed here: rather than the trivial XOR method (which replaces the analogue multiplier of PSD_1 by relays and an exclusive-OR gate) the useful phase-frequency detector [5] is suggested, as shown in Fig. 6*b*.

Suppose that initially both D-type flip-flops hold logic 0 and that the input $s_i(t)$ passes through zero before $\sin\phi_0(t)$. Then *UP-ff* is set to one and its output ensures that $S_e(t) \to 1$. Similarly if $\sin\phi_0$ passes through zero first, *DOWN-ff* is set to one and $S_e(t) \to -1$. When the other signal makes the transition, the corresponding flip-flop is set and, via the NAND gate, both flip-flops are then reset to zero. Hence $S_e(t) = \pm 1$ depending on which of the PSD inputs makes the first transition through zero, and the duration of non-zero S_e depends on the time-shift between the input sinewaves, for after the second transition $S_e \to 0$. The output signal is a pulse of correct sign whose width is directly proportional to phase error. At lock-in with zero phase-error, $S_e = 0$ and hence there is no double-frequency or other harmonic component. On the other hand, ZC methods are affected by noise (spurious crossings): methods to alleviate the problem (e.g. by using hysteresis levels) require knowledge of the amplitude A.

3.2 SDM

This is the approach of [10]. The SDMPLL, based on the transfer-function-analyser theory described above, is depicted in Fig. 6*c*. As the amplitude A is reflected in both s_x, s_y, the signal $S_e = \arctan(s_y, s_x)$ is independent of A (though not of any plant noise): useful in that the convergence property for a linear plant will be independent of test amplitude a. The mean value of the error signal $\overline{S_e}$ is linear in the phase-error e, but the unfiltered $S_e(t)$ follows a sawtooth waveform of amplitude π and whose fundamental frequency is $2\omega_0$. Hence, as with PSD_1, the low-pass $F_B(s)$ is included in SDMPSD for optional filtering of these fluctuations. Only the single filter of Fig. 6*c* is really necessary, though [10] and Fig. 3 use two.

3.3 The HT PSD

SDMPSD derives its advantages over PSD_1 by employing $\cos\phi_0(t)$ as well as $\sin\phi_0(t)$, easily done using a numerically-controlled oscillator as ϕ_0 is known. The HTPSD goes further by trying to provide $A\cos\phi_i(t)$ as well as $A\sin\phi_0(t)$ (the raw data). This cannot be achieved by an oscillator (as A, ϕ_i are unknown), but can be approximated by the output of a transfer-function having unity gain and phase-shift $\pm\pi/2$: a Hilbert transformer [13]. In our case a design phase-shift of $-\pi/2$ is used, so that for an input of $A\sin\phi_i(t)$ the output is $-A\cos\phi_i(t)$. Hence, based on Fig. 6*d*, the following signals are computed:

$$\begin{aligned} s_x &= A[\cos\phi_i\cos\phi_0 + \sin\phi_i\sin\phi_0] = A\cos(\phi_i - \phi_0) \\ s_y &= A[\sin\phi_i\cos\phi_0 - \cos\phi_i\sin\phi_0] = A\sin(\phi_i - \phi_0) \\ S_e(t) &= \arctan(s_y, s_x) = \phi_i - \phi_e = e(t) \\ A &= \sqrt{s_x^2 + s_y^2} \end{aligned}$$

Hence this PSD provides a linear, instantaneous, and harmonic-free error signal that is independent of A: an ideal combination.

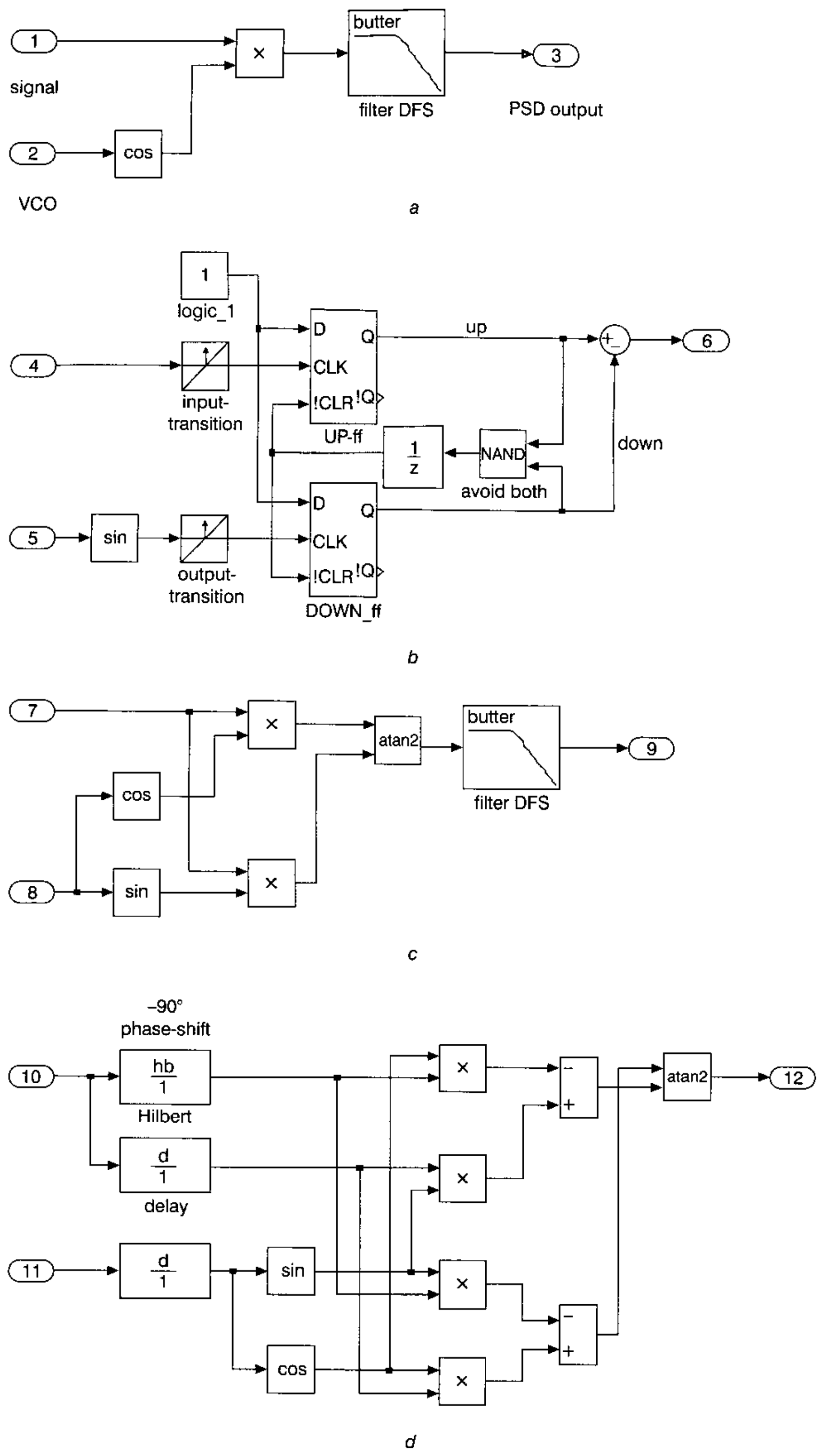


Fig. 6 *PSDs*

a Simple analogue multiplier (PSD$_1$)
b Phase-frequency (ZC)
c Synchronous demodulation (SDM)
d Hilbert transform (HT)

A Hilbert transformer must be implemented in discrete-time, and for a sample interval of h sec the Nyquist frequency is $f_N = 1/2h$ Hz, or $\omega_N = \pi/h$ rad/s. The filter has the non-causal pulse response with coefficients:

$$b[n] = \frac{2}{\pi}\frac{\sin^2(\pi n/2)}{n} \quad n = \pm 1, \pm 2, \ldots; \; = 0 \quad n = 0 \quad (6)$$

and a frequency response of $-j$ for $0 \le \omega < \pi$ and j for $-\pi \le \omega < 0$. Note that the phase-shift is always $\pm\pi/2$ for any non-causal filter whose pulse response is an odd function, as each element $b[n]z^{-n}$ has a corresponding $-b[n]z^n$, so giving a frequency response with terms:

$$b[n]e^{-jn\omega} - b[n]e^{jn\omega} = -j2b[n]\sin n\omega \quad (7)$$

To produce a practical filter, the theoretically infinite number of terms is approximated by a finite number m, and causality is imposed by passing the sampled input signal $s_i(t)$ both through the filter and through a simple delay amounting to half the length of the finite pulse sequence (i.e. the delay element 'd' in Fig. 6*d* corresponds to a dead-time of $(m/2+1)h \approx mh/2$ s). There will then be the desired phase difference of $\pi/2$ between the two output signals, but on the other hand there will be an error in the output amplitude (i.e. the gain of the filter is not unity for all ω) because of necessary truncation of the pulse sequence $b[n]$.

The design can use the remez algorithm of Matlab's signal processing toolbox. A best-fit (equi-ripple) to the desired gain of unity is specified for a given normalised frequency range $[\beta_1 \cdots \beta_2]$ (i.e. real frequencies $[\beta_1 f_N \cdots \beta_2 f_N]$ Hz) and the appropriate parameters $b[n]$ are computed. Inspecting the equations leading to S_e for HTPSD indicates that a small sinewave leakage arises if the gain of the Hilbert transformer is not precisely one. An example of the (acceptable) behaviour of a HT for $m = 20$, $\beta_i = 0.1$, 0.9 is seen in Fig. 7. There are two important choices to make: (i) the number of terms m in $b[n]$ (large m gives a better fit, yet leads to more computation and a greater imposed delay); and (ii) what coverage of the

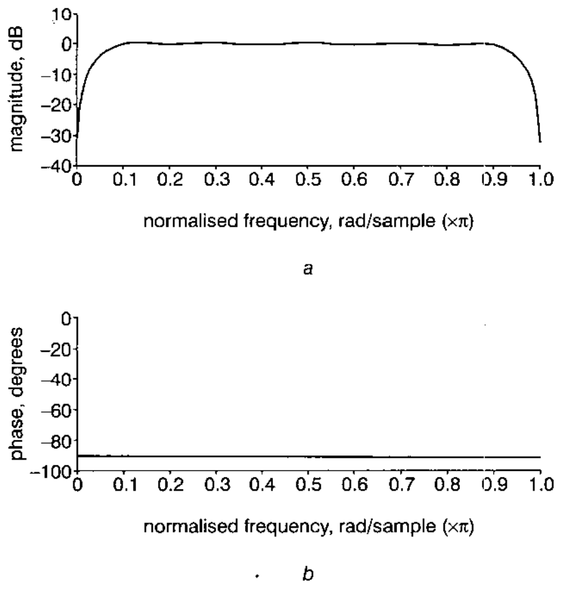


Fig. 7
a Hilbert transformer gain for $f = [0.1 \cdots 0.9]f_N$ and $m = 20$
b Hilbert transformer phase for $f = [0.1 \cdots 0.9]f_N$ and $m = 20$

Table 2: Properties of representative PSDs

Method	PSD$_1$	ZC	SDM	HT
Linear?	×	√	√	√
Amplitude independent?	×	√	√	√
No harmonics?	×	√	×	√
Noise rejection?	√	×	√	√
Dynamics?	$F_B(s)$	Sampled	$F_B(s)$	exp $(-smh/2)$

frequency range of $0 \cdots f_N$ to attempt when choosing β_i. It is found best to make the 'fitting zone' symmetric around the mid-frequency $f_N/2$ Hz and not to cover an excessive range (e.g. $[0.01 \cdots 0.99]$ would need a much larger m than $[0.1 \cdots 0.9]$ for the same maximum gain error). In this application the parameters of Fig. 7 are satisfactory.

The main choice in using the HTPSD is the sample-interval h. For best effect ω_0 must lie in the middle zone, or there will be ripples (harmonic leakage) in $S_e(t)$: this implies:

$$\beta_1\omega_N < \omega_0 < \beta_2\omega_N \to \frac{0.3}{\omega_0} < h < \frac{3}{\omega_0} \quad (8)$$

approximately, for our choice of β_i where h is in seconds and ω_0 is in radius per second. Of course, as the algorithm progresses ω_0 will change: in practice it is safest to adopt h half-way in this range and based on the initial 'guessed' frequency $\omega_0(0)$. Then there will be a factor of three in frequency on each side before h needs to be re-chosen. Note that in most applications a frequency ratio of $\bar{\omega}:\underline{\omega}$ of 9:1 is acceptable. If a greater range is required, one approach is to use heterodyning. Space precludes a full discussion: [12] shows the effective use of heterodyning over a 100:1 frequency range.

Table 2 summarises the properties of the PSDs described above. In practice all will be entirely adequate, but PLL$_1$/SDM require a post-filter F_B to remove harmonics whilst ZC methods require a pre-filter to for reliable operation with noisy data. HT is the more computationally complex, though not excessively so, and seems to work best.

4 Predicting convergence speed and the effect of noise

4.1 Convergence

The convergence property of the updating algorithm of (5) depends on the adaptive gain K, the behaviour of arg $G(j\omega)$, a nonlinear function of frequency, and on:

- the inherent dynamics of the PSD;
- the dynamics associated with the plant's phase response to a sinewave of slowly-varying frequency.

As a first approximation the last two components can be ignored, and attention is concentrated on the simple adaptive law assuming that both the PSD and the plant have instantaneous responses to phase changes.

It is useful to consider local behaviour, when $\omega_0 \approx \omega_d$ via appropriate linearisation. Suppose that the plant's transfer function is characterised by a dead-time T_d, n poles p_i and m zeros z_i, where $n > m$, so that:

$$G(s) = K_p^* e^{-sT_d}\frac{\Pi^m(s - z_i)}{\Pi^n(s - p_i)} \quad (9)$$

giving the plant's phase $\phi = \arg G$ at the loop frequency ω_0 to be:

$$\phi = -\omega_0 T_d + \Sigma^n \arctan\frac{\omega_0}{p_i} - \Sigma^m \arctan\frac{\omega_0}{z_i} \qquad (10)$$

We take a small perturbation around the frequency ω_d that gives the desired phase ϕ_d:

$$\omega_0 = \omega_d + \Delta\omega \rightarrow \phi = \phi_d + \Delta\phi \qquad (11)$$

and from the Taylor series for ϕ and ignoring quadratic and higher terms in $\Delta\omega$ we find that:

$$\Delta\phi = G^*(\omega_d)\Delta\omega \qquad (12)$$

where

$$G^*(\omega_d) = \left.\frac{d \arg G}{d\omega_0}\right|_{\omega_0=\omega_d} = -T_d + \Sigma^n \frac{p_i}{\omega_d^2 + p_i^2} - \Sigma^m \frac{z_i}{\omega_d^2 + z_i^2} \qquad (13)$$

Note that by appropriately combining complex-conjugate terms, it is easily shown that this formula applies also to a G having complex (underdamped) singularities.

Now as $\omega_0 = \omega_d + \Delta\omega$ we see that $d\omega_0/dt = d\Delta\omega/dt$. Hence, inserting into the adaptive law of (5) and invoking (12) we find:

$$\frac{\Delta\omega}{dt} = KG^*(\omega_d)\Delta\omega \qquad (14)$$

giving the key convergence-rate result:

$$\Delta\omega(t) = \Delta\omega(0)\exp\{KG^*(\omega_d)t\} \qquad (15)$$

This is an exponentially-decaying response when the local slope $G^*(\omega_d) < 0$, e.g. around the points P1 and P3 in Fig. 1*a*, though not round P2.

To show the validity of (15) the third-order transfer function $G_3 = 4/(1+s)^3$ was connected to the HTPLL for a desired phase of $\phi_d = -3\pi/4 \rightarrow \omega_d = 1$. The initial frequency of the PLL was chosen as $\omega_0 = 0.5$, 1.5 (i.e. below and above ω_d). G_3 has three poles $p_i = -1$, so $G_3^*(1) = -3/2$ from (13). Hence (15) predicts that the phase/frequency estimator produces:

$$\omega_0(t) = 1 - (1 - \omega_0(0))\exp(-1.5Kt)$$

The response of the adaptive algorithm from the two initial frequencies $\omega_0(0)$ is shown in Fig. 8, together with the above predicted response (dashed lines). The correspondence is close: for $\omega_0 < \omega_d$ the PLL is slightly faster than the prediction, as the slope $G_3^*(\omega_0)$ is greater than the local slope at the converged point $G_3^*(1)$. The effect of the intrinsic delay in the HTPSD is clearly seen in the initial response.

For larger adaptive gains K the omitted dynamics become significant. Suppose that the plant's contribution is ignored but the effect of the delay in the HTPSD is now included, so that:

$$\frac{\Delta\omega}{dt} = KG^*(\omega_d)\Delta\omega(t - \tau)$$

where $\tau \approx mh/2 = 1.5$ s in this example. A simple Nyquist argument shows that this delay/differential equation becomes unstable at a frequency where the phase shift of the delay exceeds $-\pi/2$, i.e. $\omega\tau = \pi/2$. The critical gain is given by $KG^*/\omega = 1 \rightarrow KG^* = \pi/2\tau = \pi/mh = \omega_N/m$ (a nice result implying that small m and fast sampling are useful for rapid adaptation). For our example this indicates $K \ll (\pi/20 \times 0.3) \times 2/3 \approx 0.3$. Fig. 9 shows the result of a simulation of HTPSD with $K = 0.1$, together with the predicted response based on solving just the above delay/differential equation (dashed line). It is seen that the response is rather more oscillatory than predicted, so the plant's contribution to the loop behaviour is not negligible. Nevertheless, the PSD's inherent dynamics are relevant in that the predicted response is already showing overshoot.

The settling-time of G_3 to a unit step is roughly 10 s, so the HTPLL completes its estimate in about five plant settling-times: reasonably fast. The initial response of G_3 can be approximated by a dead-time of 1.5 s. It is conjectured that simply adding this to the HTPSD dead-time (doubling it and so bounding $K < 0.15$) gives a fair guide to the adaptive loop's stability.

4.2 *Tracking a phase ramp*

Given that fairly rapid convergence to a fixed ϕ_d is possible, attention turns to the ability to track a range of phases, thus in effect providing an estimate of the plant's Nyquist diagram. To achieve this the algorithm needs to estimate the plant's gain from $|\hat{G}| = \hat{A}/a$: for HTPLL use is made of the SDM results. It was found that different estimates are produced according to the direction of the phase-sweep, so the results shown here involve sweeping in both directions. The difference in the estimates were found also to depend on the demanded speed of tracking (a

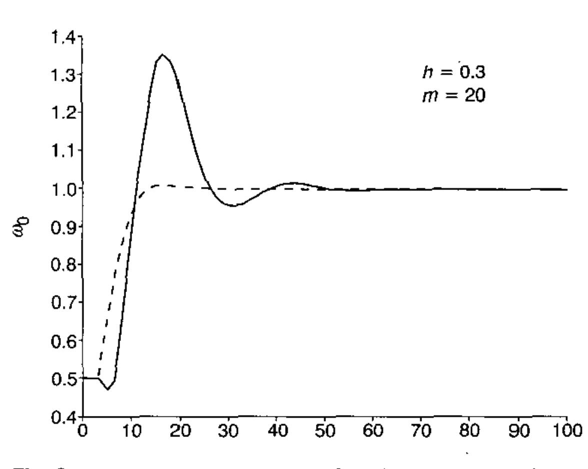


Fig. 8 *Convergence with 'reasonable' adaptive gain $K = 0.01$*

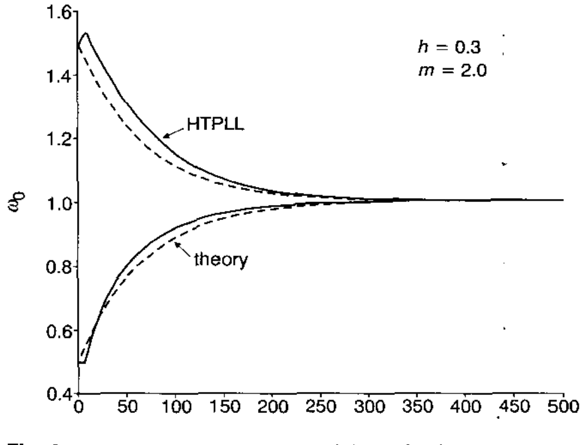


Fig. 9 *Excessive gain $K = 0.1$ invalidates the frozen-parameter assumption*

consequence of the violation of the frozen-parameter assumption).

The range of phases adopted for the test of G_3 was $\phi_d = [-2\pi/3 \cdots -4\pi/3]$, the important set in most applications. The experiments were based on a specified total time T_{max}. For the first quarter of the experiment ϕ_d was fixed in order to let HTPLL converge. For the final part of the experiment, ϕ_d followed a ramp whose slope enabled the phase precisely to attain the other desired bound. Fig. 10 shows the results for the case $T_{max} = 1000$ s; the circles correspond to the exact frequency response. If greater accuracy were required, $d\phi_d/dt$ would need to be smaller. This is confirmed by Fig. 11, where now $T_{max} = 10\,000$. It is nice to see that in both examples the true $G(j\omega)$ lies about half-way between the two swept estimates: would this be the case in general?

4.3 Application to G_r

We now turn to the rather more challenging example of $G_r(s)$, noting two possible sources of difficulty:

- the plant's integrator invalidates any assumption that its mean output level is zero (indeed the output mean was found to meander). This causes difficulty with ZC methods;
- the small variations of $\arg G_r$ over a wide frequency range, as seen in particular in the Nyquist diagrams of Figs. 1*a* and 1*b*.

The first problem is of course seen in other applications. For example ZC is a popular approach to frequency estimation in vortex-shedding flow meters [14], and noise reduction for that application is achieved by the variable band-pass filter:

$$F_v(s) = \frac{y(s)}{x(s)} = \frac{\beta\omega_0 s}{s^2 + \beta\omega_0 s + \omega_0^2} \tag{16}$$

whose Simulink diagram is shown in Fig. 12. At the centre-frequency ω_0 the gain is unity and the phase-shift is zero. A common choice of the coefficient β is 1.5, as this gives half-power frequencies of $0.5\omega_0$, $2\omega_0$. The filter has no effect on the final converged frequency, and apparently minimal effect on convergence rate, but is extremely useful in cleaning up the data by removing noise and drift (including DC levels).

The full HTPLL algorithm is depicted in Fig. 13. The VBPF of Fig. 12 is fed with noisy plant data, and its centre frequency is the current PLL frequency $\omega_0(t)$. Its output is transferred as input $s_i(t)$ to the HTPSD of Fig. 6*d* which provides $S_e(t)$ for the algorithm's integrator and an estimate $\hat{A} = \sqrt{(s_x^2 + s_y^2)}$ of the input signal's amplitude. As noise (and sinewave leakage) can affect $\hat{A}$, a Butterworth low-pass filter is optionally inserted. The desired phase ϕ_d is the initial condition of the lower integrator, whose input step optionally enables the phase-sweeping experiments described above to be performed.

The second difficulty influences the choice of adaptive gain K. In transfer functions dominated by dead-time, the phase-lag increases linearly with frequency (so K should be independent of frequency), but when $G(s)$ is rational the phase-lag follows log-frequency (as in the phase plot of Fig. 1*a*). Hence K should be roughly proportional to frequency: this approach is adopted here. For the results shown in Fig. 14 four initial frequencies: $\omega_0(0) = [0.5, 3, 20, 100]$ were chosen, and the following adaptive parameters:

gain: $K = \omega_0(0)K_0$, with $K_0 = 0.03$ to achieve the results of the Figure;
sample interval: as it would not be known in which direction the algorithm would move the estimate $\omega_0(t)$, h s was placed half-way in the suggested range for the Hilbert transformer design, i.e. $h = 1/\omega_0(0)$ s.

The other parameters were the default values of $m = 20$, $\beta = 1.5$, and to clarify the results a logarithmic y-axis is used in the Figure. The results confirm the heuristic predictions: the algorithm indeed converges reliably to points P1, P3 from both directions, though convergence is noticeably slower (as expected) for initial frequencies between P1 and P3.

4.4 Variance of the frequency estimate

Space precludes a full discussion about the effect of noise on the estimated frequency: indeed for ZC methods

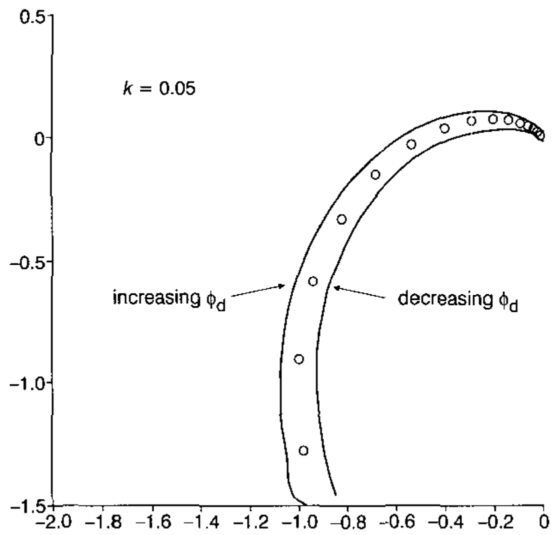


Fig. 10 *Estimated Nyquist diagram for tracking over a range of phases*

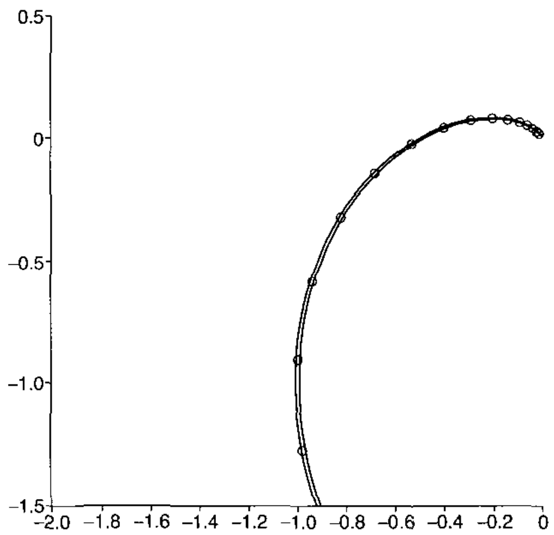


Fig. 11 *Estimated Nyquist diagram for tracking over a range of phases slower rate*

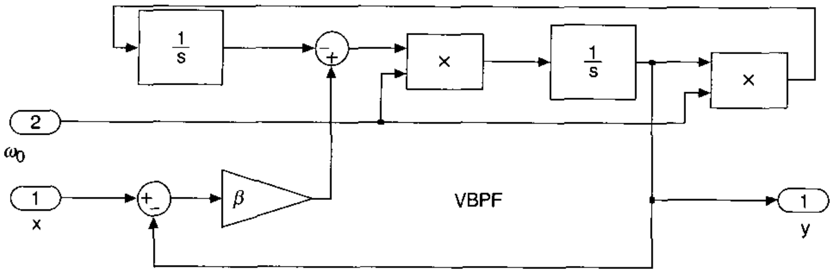


Fig. 12 *VBPF*

the analysis is fairly intricate. We concentrate here on the HT approach: more details about this and other methods are in [12]. Suppose that the plant output has additive noise so that:

$$y(t) = A\sin\omega_0 t + n(t) = E(t)\sin(\omega_0 t + \theta(t))$$

and assume that the noise $n(t)$ has a double-sided band-limited spectrum, with bands of width B Hz being centred around $\pm\omega_0$ and having constant spectral density S_n W/Hz, so that $\sigma_n^2 = \mathrm{Var}(n) = 2S_nB$. Note that it is the noise spectrum around ω_0 that matters, because of the modulation inherent in a PLL. The envelope $E(t)$ and phase $\theta(t)$ of $y(t)$ are random, following the Rician distribution, but for reasonable SNR these distributions can be approximated by the normal: $p_E(E) \sim N(A, \sigma_n^2)$, $p_\theta(\theta) \sim N(0, \sigma_n^2/A^2)$.

It is convenient to use the quadrature representation for the noise, i.e.:

$$n(t) = n_c(t)\cos\omega_0 t + n_s(t)\sin\omega_0 t$$

where the components n_c, n_s are baseband, having spectral densities $2S_n$ W/Hz over the range $\pm B/2$ Hz. A full analysis of the HTPSD shows that, approximately:

$$S_e(t) = e(t) + \frac{n_c(t)}{A}$$

where $A = a|G|$ is the plant's output amplitude. Hence (14) now reads:

$$\frac{\Delta\omega}{dt} = KG^*\Delta\omega + \frac{K}{a|G|}n_c(t)$$

This is the simple first-order transfer function:

$$H(s) = \frac{K/a|G|}{s - KG^*} \tag{17}$$

driven by band-limited noise of spectral density $2S_n$. We now take B large enough (e.g. $B \gg KG^*/\pi$) so that the standard integral (which has limits of $\pm\infty$) can be used to compute the variance of $\Delta\omega$ around the lock-in frequency ω_d, giving the key result:

$$\mathrm{Var}(\Delta\omega) = \frac{KS_n}{a^2|G(j\omega_d)|^2 G^*(\omega_d)} \tag{18}$$

Note that the evaluated variance is indeed positive, as a stable convergence point has $G^* < 0$. The variance can be made as small as desired, by choice of K, but (17) then shows that $H(s)$ would then have a low break-point frequency, and ω_0 could 'meander' about its correct mean value. An interesting design question, not explored here, is the use of post-filtering: large K is used for rapid convergence (but this increases $\mathrm{Var}(\Delta\omega)$) and the post-filter for presenting a smoothed ω_0 to the 'next-level up'.

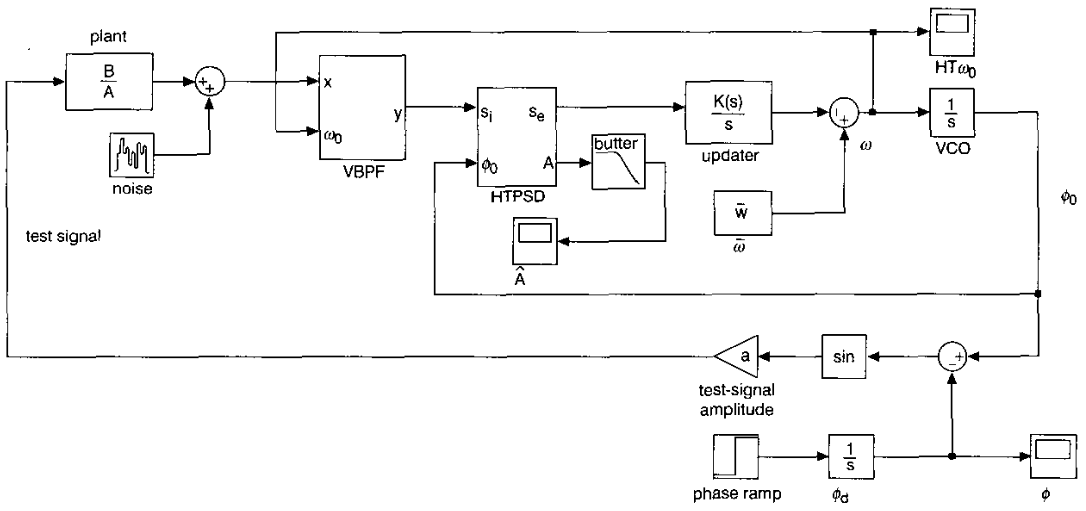


Fig. 13 *Simulink diagram of the full HTPLL algorithm*

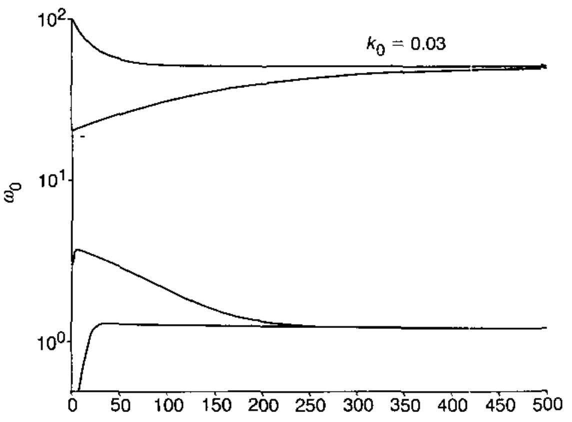


Fig. 14 *Using HTPLL with $G_r(s)$*

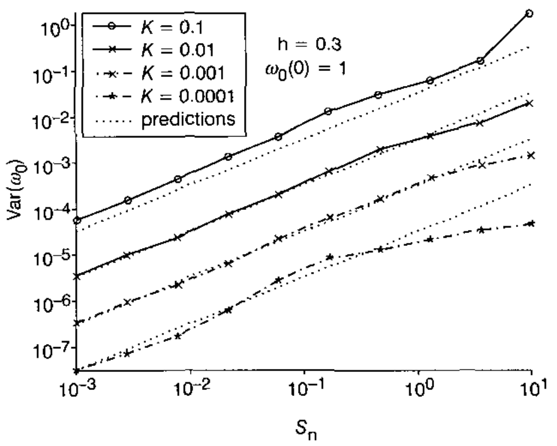


Fig. 15 *Verifying the predicted variance of the HTPLL for G_3*

To validate the above expression for variance the HTPLL plus the third-order G_3 was simulated, the plant output being subjected to additive output noise over the wide range $S_n = 10^{-3} \cdots 10$ W/Hz, the HTPLL having $h = 0.3$ s and starting from the lock-in frequency $\omega_d = 1$. The noise was generated by Simulink's 'band-limited noise' block, with a chosen sample interval equal to h: note that this implies a noise variance of S_n/h. At the upper S_n limit the output SNR was 1:30, as the input amplitude of $a = 1$ and $|G| = \sqrt{2}$ gives $A = \sqrt{2}$ and hence the signal power was unity. Adaptive gains $K = [10^{-4} \cdots 0.1]$ were used (the upper value being near to the predicted instability point of the algorithm).

Simulations, of course, only give an approximate verification of the predictions, as the experimental variance is a random variable whose own variance reduces for increasing experiment-time T_e. The complication is that the HTPLL bandwidth is proportional to $1/K$, so T_e needs to be increased for smaller K to achieve consistent estimates. Hence for $K = [10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$ the simulation time was chosen to be $T_e = [2 \times 10^5, 10^5, 2 \times 10^4, 2 \times 10^3]$, giving the results shown in Fig. 15. Note that both axes are logarithmic, in view of the wide range of S_n values adopted. Plotted (dotted lines) are the predictions of (18). There are some experimental fluctuations, particularly for small K as the corresponding T_e should really have been larger. Nevertheless, the predictions are amply fulfilled. For $K = 0.1$ the increased experimental variance arises from too tight an adaptive loop, as seen in the corresponding convergence analysis.

The value of (18) is the assurance that the proposed monitoring role is feasible and indeed suggests how parameters can be chosen. In practice, variance can be further reduced by band-pass filtering: this is particularly useful for larger values of K.

5 Comparing the different phase detectors

To demonstrate the relative behaviour of the four PSDs in this application, a set of simulations were performed on G_3, and results from a representative subset are discussed below. To provide a fair comparison, the adaptive gains K were chosen such that the local behaviour of the PLLs near $\omega_d = 1$ should be similar. Hence for each K used by HTPLL, the following were adopted:

PLL$_1$: as $|G_3(j1)| = \sqrt{2}$ and with the input amplitude $a = 1$, the output amplitude is $\sqrt{2}$ and so the amplitude of $S_e = 1/\sqrt{2}$. Hence $K_1 = \sqrt{2}K$ was adopted;
ZC: suppose, for example, that the phase error is $\pi/2$. Then the PSD outputs the value ± 1 for one-quarter the period, rather than the continuous $\pi/2$ of HTPLL. Hence $K_{ZC} = 2\pi K$;
SDM: the filtered S_e has the same mean as that delivered by the HTPSD, so $K_{SDM} = K$.

For the chosen examples, both the VBPF $F_v(s)$ and the Butterworth filters $F_B(s)$ were inserted in the loop. However, ω_f was ten in all cases, and so the response of filters F_B was flat over the useful frequency range. Integration for the simulations was Runge–Kutta with a fixed interval of 0.01 s, and the initial condition was $\omega_0(0) = 0.5$ rad/sec.

For the noise-free simulations that gave Fig. 16, the β of each VBPF was set to 50 so that the filter had wide bandwidth and minimal effect. Convergence was positive (compare with Fig. 8), and the following points are worth noting:

PLL$_1$: the response is faster initially, as the amplitude $A > \sqrt{2}$ for $\omega_0 < 1$. However, if say $\omega_0(0) = 2$, a much slower response would be seen. This confirms the amplitude-dependence of convergence for PLL$_1$ that, together with the observed DFS makes it harder to apply;
ZC: the sampled nature is apparent (see how the steps in ω_0 become closer together as ω_0 increases), and no DFS is seen;
SDM: positive convergence (independent of A), but with a DFS of slightly larger amplitude than for PLL$_1$;
HT: after the initial dead-time there is a small downward movement, though the subsequent continuous response is rapid and 'clean'.

To remove the DFS of PLL$_1$ and SDM the bandwidth of F_B has to be reduced: a possibly good choice is $\omega_f = \omega_0(0)$. This was found to be acceptable in other simulations, though the added dynamics affected the convergence speed.

To test the effect of disturbances, band-limited noise with a sample interval of $h_n = 0.1$ s and spectral density S_n was added to $y(t)$: the noise variance was $\sigma_n^2 = S_n/h_n = 10S_n$ giving a converged SNR of $0.1/S_n$ (the output signal power being unity). For quite small amounts of noise the ZC method failed, but improvements were seen when the β of

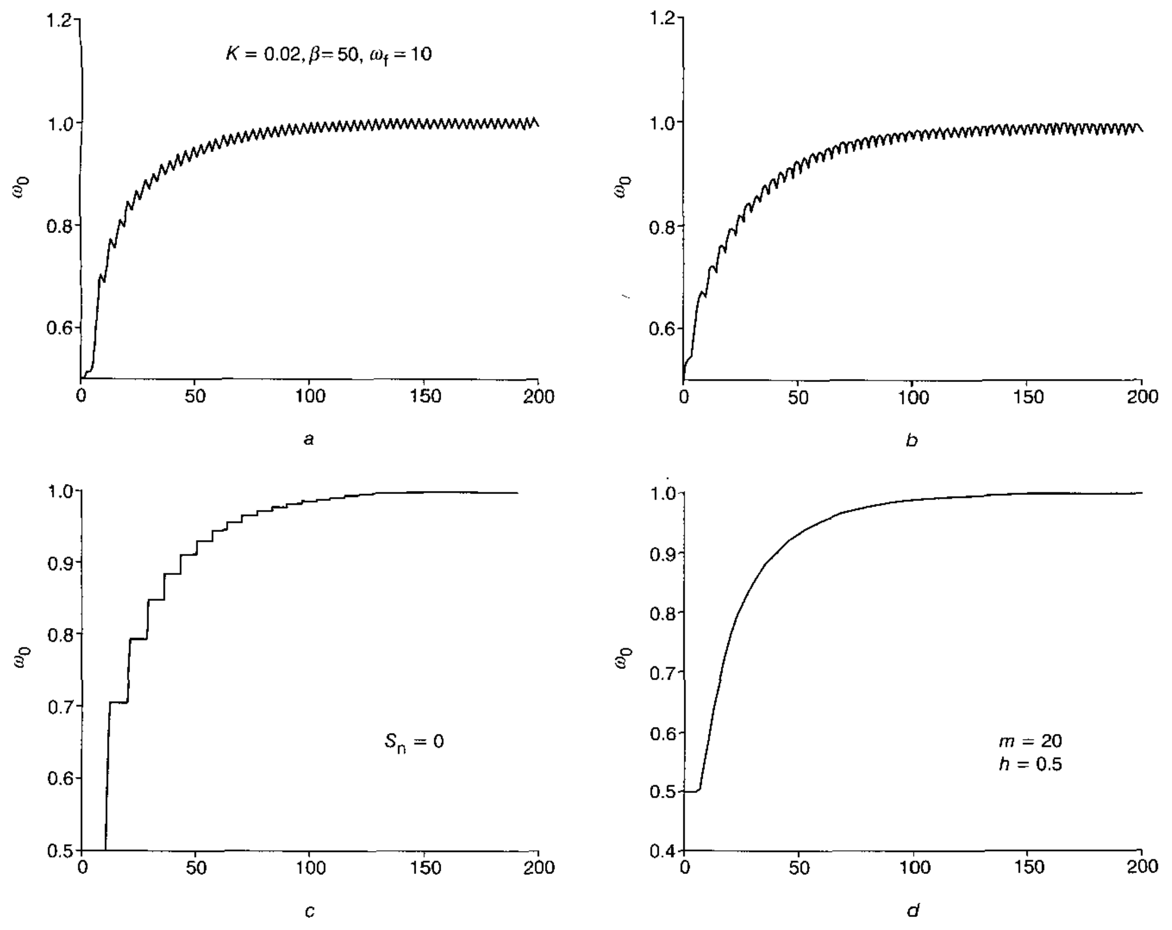


Fig. 16 *Transient response: no noise*
a PLL_1
b SDM
c ZC
d HT

F_v was reduced to 1.5, thereby limiting the effective noise bandwidth. Fig. 17 shows the responses for the case when the SNR was two. It is seen that ZC is badly affected, with large variations around the converged values of ω_0. This behaviour is consistent with the comments of [8], where several noise-mitigation methods (including Kalman filtering) are discussed. The other PLLs converge at about the same rate as for Fig. 16, and the noise has a comparable effect on their final estimates.

The noise was increased further to $S_n = 1$, leading to an SNR of 1:10, a large value and hence Var(ω_0) would become excessive for application. It was decided to reduce K by a factor of ten, so theory predicts a ten times slower convergence rate but also a ten times reduction in the variance of ω_0. The corresponding simulations are shown in Fig. 18. Except for ZC (which failed), the convergence and subsequent fluctuations of $\omega_0(t)$ are satisfactory. Note that the reduced K means that the DFS of PLL_1 and SDM have a much reduced amplitude and indeed cannot be seen within the noise band. Hence for monitoring purposes these simpler methods might be effective. To demonstrate the signal behaviour of HTPLL under these noise conditions, Fig. 19 shows the corrupted plant output together with the PLL's internal sinewave, suitably phase-shifted so that in theory the noise-free signals should be exactly in phase. The Figure verifies that effective convergence and tracking is possible with this algorithm, despite a very poor SNR.

5.1 Applying the VBPF

The VBPF by restricting the effective noise bandwidth, makes the ZC approach more reliable (though it is still inferior to the others), and allows plant (e.g. type 1) with drifting output data to be tracked. It was thought interesting to see the effect of a filter with a very small bandwidth, i.e. $\beta = 0.001$. The 'Q' of the VBPF is given by $Q = 1/\beta = \omega_0/\Delta\omega_0$, where $\Delta\omega_0$ is the filter's half-power bandwidth, so in this case $\Delta\omega_0 = \omega_0/1000$, small indeed.

The first simulation was to investigate how the filter affects the convergence of the algorithms, so the adopted plant was G_r with an initial frequency of $\omega_0(0) = 100$ rad/sec and with other experimental conditions corresponding to Fig. 14. Note that $|G_r|$ rolls off rapidly at these frequencies, so to make PLL_1 have comparable

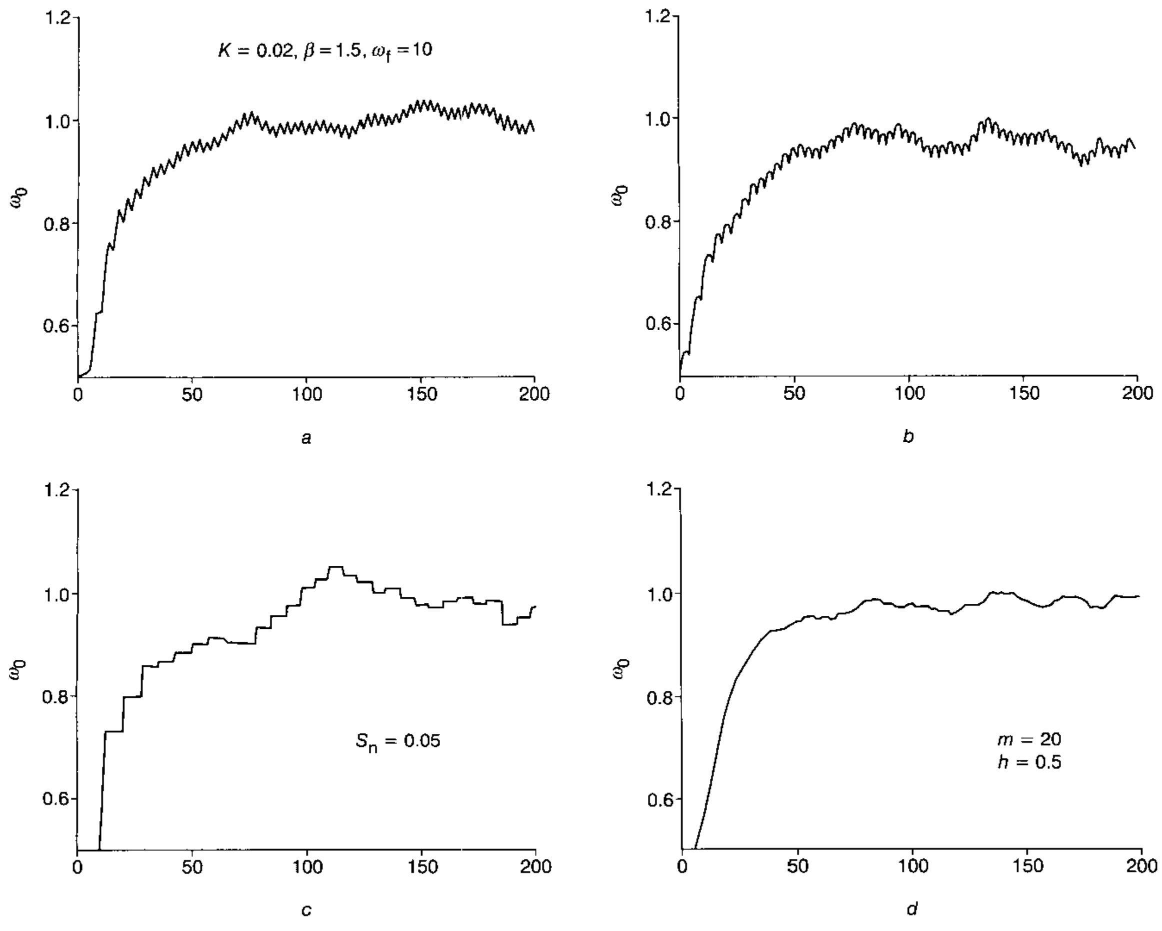


Fig. 17 *Additive plant noise with spectral density $S_n = 0.05$ W/Hz*
a PLL_1
b SDM
c ZC
d HT

performance, the input amplitude was chosen to be $a = 10$, ensuring that the output amplitude $A \approx 1$. The results of Fig. 20 show that the VBPF has some effect on the transient response, there now being overshoot in all cases, unlike the responses of Fig. 14.

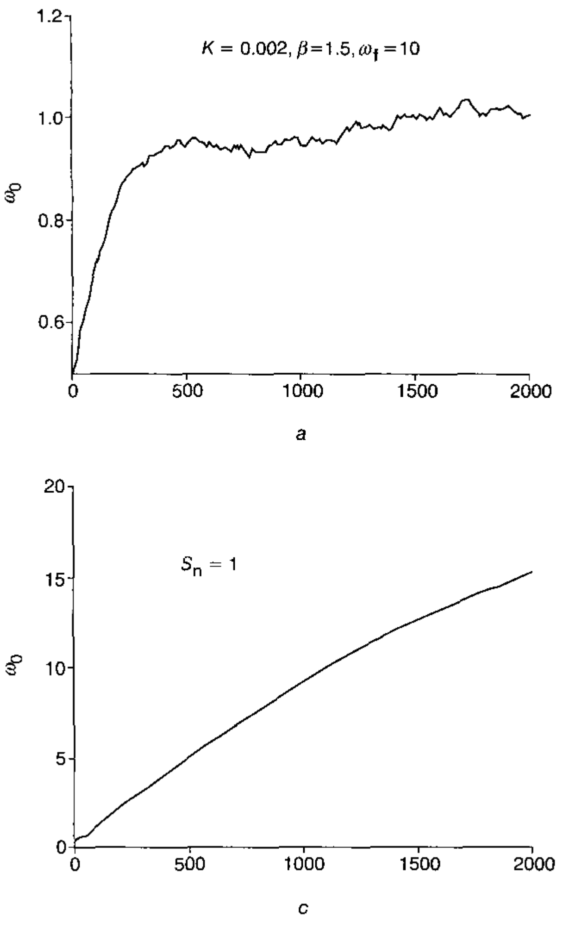
The simulation was repeated with noise parameters $S_n = 0.1$, $h_n = 0.001 \rightarrow f_n = 1$ kHz. Hence the noise variance was 100, implying that the SNR had the tiny value of 0.01. The results of Fig. 21 show that all the algorithms still converge. Of course, the narrow-band filtering leads to low-frequency 'meandering' of the estimates: for greater accuracy further averaging could be used.

The narrow-band filter ($\Delta\omega_0 = 1/20$ rad/sec when converged) is highly effective at rejecting noise and sinusoids outside the band, thus allowing the algorithms to work at surprisingly adverse SNRs. This opens up several intriguing and potentially useful applications. Suppose that there is more than one sinewave circulating round the loop: if their frequencies are separated, each will be rejected by the other VBPF. Hence we can track more than one ϕ_d at the same time or, perhaps, apply the method to multiple loops. The practical consequence of this idea would be an interesting topic of research.

6 Conclusions

Our goal was to generate and analyse effective algorithms that track the frequency ω_d corresponding to a given phase ϕ_d of an arbitrary transfer-function $G(s)$. The utility of such methods has been discussed by, say, [8] and related papers. The contribution here is in the precise and validated quantification of performance in terms of convergence rate and variance, and in the proposal that a HTPSD offers useful benefits in applications. Extensive simulations show that the designs can cope with poor SNRs and, using VBPFs, open up the possibility of simultaneous tests covering several specified phases.

Space limits preclude details appropriate for phase-margin design, where it also is necessary to know $|G(j\omega_d)|$and hence requires estimation of the output amplitude A. They also preclude the necessary noise analysis for PLL_1, ZC and SDM, together with predictions of the 'high-noise' behaviour of the HTPSD (as the Rician distribution cannot then be approximated by a normal distribution): [12] contains some discussion of these points.



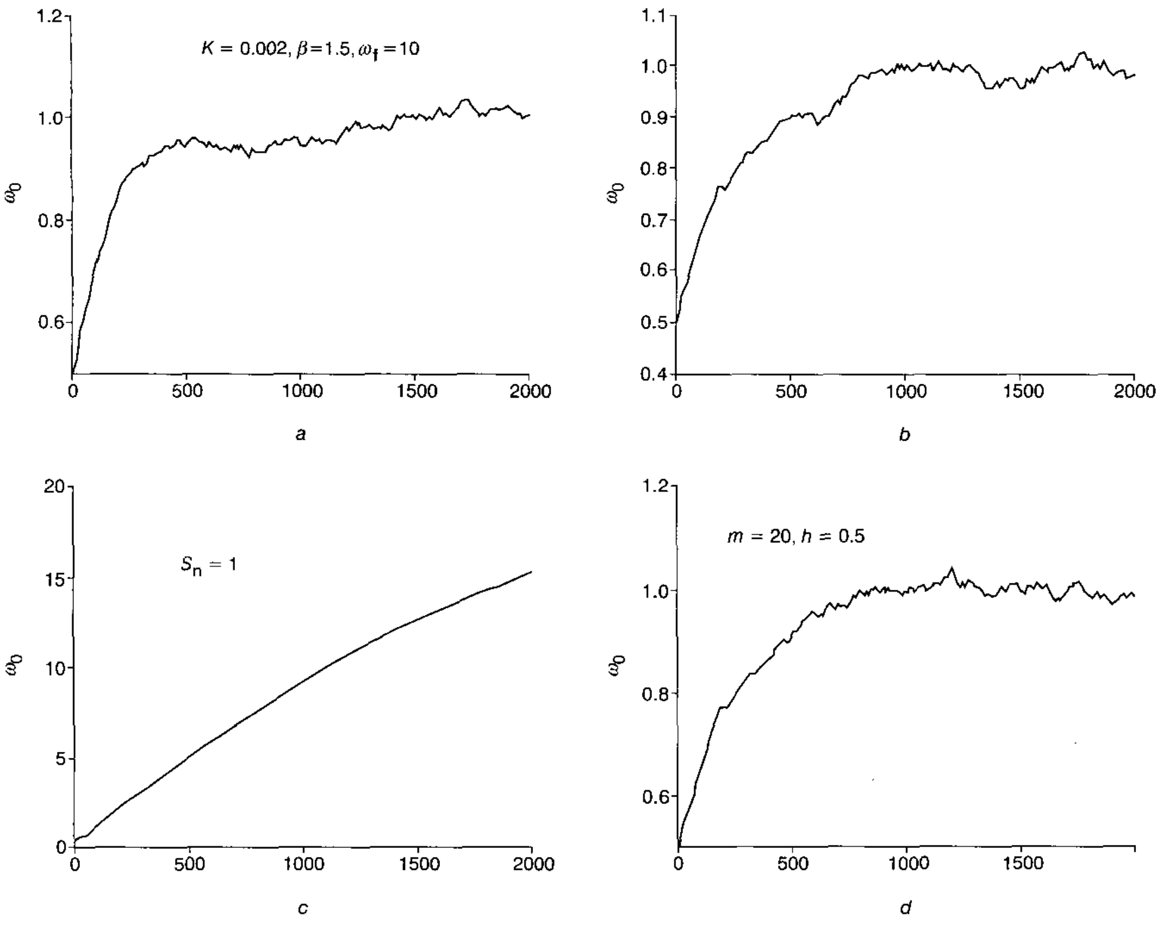


Fig. 18 *$S_n = 0.1$ W/Hz: band-pass filtering with $\beta = 1.5$*
a PLL_1
b SDM
c ZC
d HT

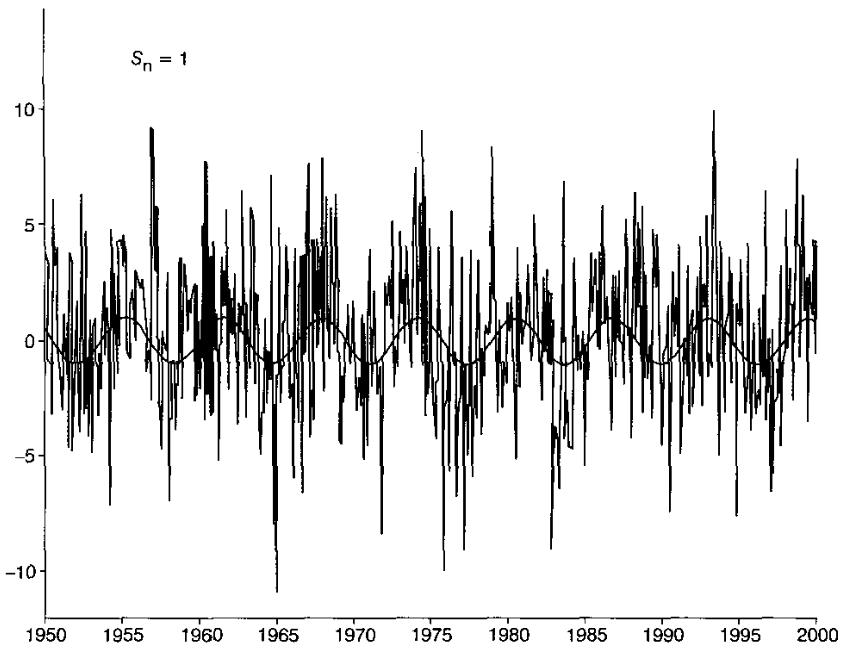


Fig. 19 *Lock-in the HTPLL with noisy data*

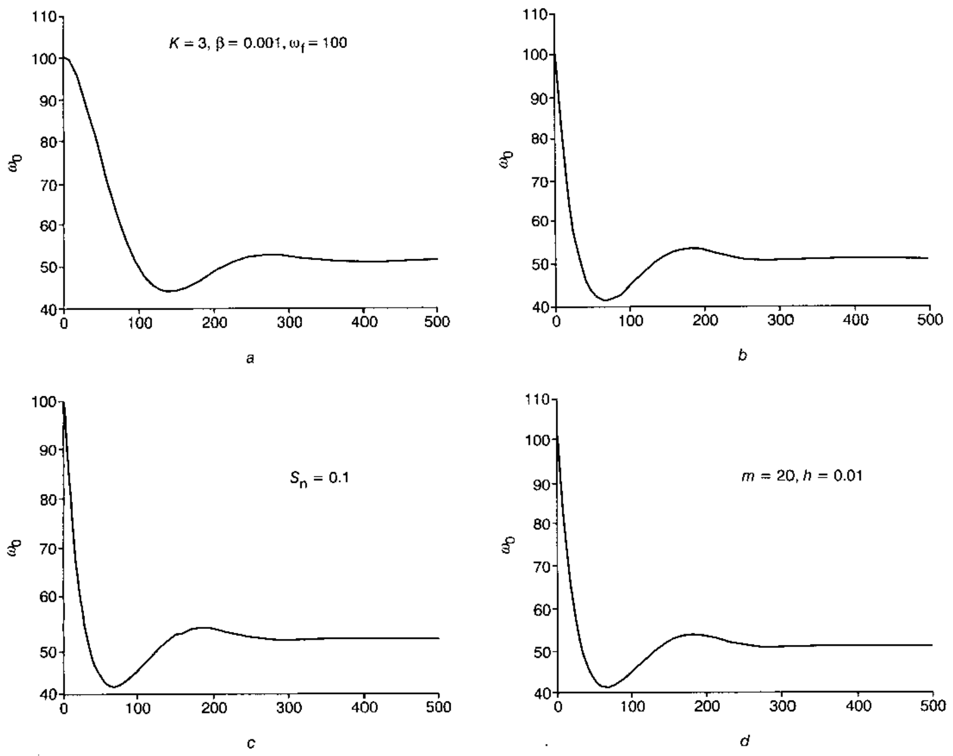


Fig. 20 *Testing G_r with a VPBF having $\beta = 0.001$: no noise*
a PLL$_1$
b SDM
c ZC
d HT

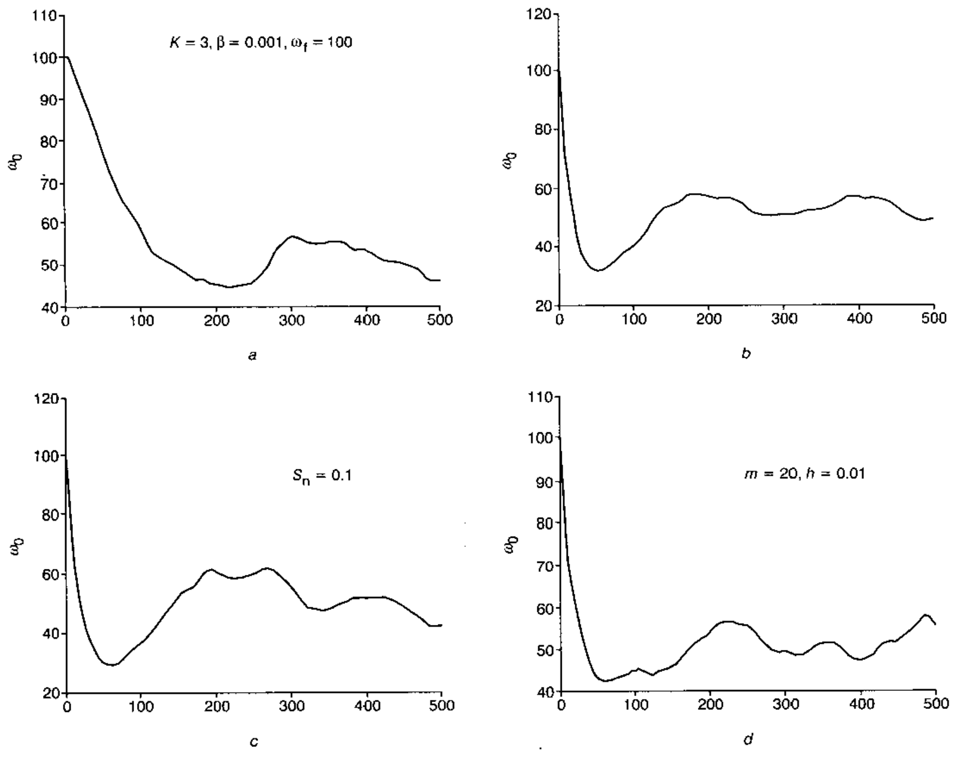


Fig. 21 *Testing G_r with SNR$\approx$1:100*
a PLL$_1$
b SDM
c ZC
d HT

There remain several interesting and open questions. The speed of convergence, which can be quite rapid, is determined by the PSD dynamics and by possible violations of the frozen-parameter assumption. Analysis of the phase-response of candidate $G(s)$ to ramping frequency inputs would be useful in this context, as it could provide guidelines for the design of faster algorithms. The effect of the VBPF on convergence, though seemingly small in the examples shown above, might become significant if very tight bandwidths are envisaged (small β). But the main questions lie in the prior choices of K, $\omega_0(0)$ when the plant G is indeed unknown. Some preliminary information is required, perhaps derived from a simple step-test. It is conjectured that, given the nice behaviour of the algorithms described here, a general-purpose method for plant testing, tuning and monitoring can be developed that is competitive with the well-established autotuning approach.

7 References

1 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

2 YU, C.-C.: 'Autotuning of PID controllers' (Springer-Verlag, London, UK, 1999)

3 ZIEGLER, J.G., and NICHOLS, N.B.: 'Optimum settings for automatic controllers', *Trans. ASME*, 1942, **12**, pp. 759–768

4 HARRIS, T.J.: 'Assessment of control loop performance', *Can. J. Chem. Eng.* 1989, **67**, pp. 856–861

5 BEST, R.E.: 'Phase-locked loops' (McGraw-Hill, New York, NY, 1997)

6 EGAN, W.F.: 'Phase-lock basics' (Wiley, New York, NY, 1998)

7 JOHNSON, M.A., and CROWE, J.: 'New methods of non-parametric methods for PID control'. Proceedings of workshop on adaptive control and signal processing, Glasgow, Scotland, 1998

8 CROWE, J., and JOHNSON, M.A.: 'Process identifier and its application to industrial control', *IEE Proc., Control Theory Appl.*, 2000, **147**, (2), pp. 196–204

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 BALESTRINO, A., LANDI, A., SANI, L., and DI MOIA, A.: 'Sinusoidal automatic tuning variation technique for PID controllers'. Proceedings of Automation and decision making, Milan, Italy, 2000, pp. 143–152

11 BALESTRINO, A., LANDI, A., and SANI, L.: 'ATV techniques: troubles and remedies'. Proceedings of International Symposium on Advanced control in chemical processes, Pisa, Italy, 2000, pp. 755–759

12 CLARKE, D.W., Phase-locked loops: a control perspective. Technical report, Department of Engineering Science, Oxford University, 2001

13 OPPENHEIM, A.V., and SCHAFER, R.W.: 'Digital signal processing' (Prentice-Hall, Englewood Cliffs, NJ, 1975)

14 GHAOUD, T., and CLARKE, D.W.: 'Modelling and tracking a vortex flow-meter signal', *Flow Meas. Instrum.*, 2002, **13**, (3), pp. 103–117