ISSN 0005-1098

automatica

A Journal of IFAC The International Federation of Automatic Control

www.elsevier.com/locate/automatica

Available online at www.sciencedirect.com
SCIENCE DIRECT®

Automatica 41 (2005) 1503–1516

automatica

www.elsevier.com/locate/automatica

Estimation and tracking of complex-valued quasi-periodically varying systems☆

Maciej Niedźwiecki*, Piotr Kaczmarek

Faculty of Electronics, Telecommunications and Computer Science, Department of Automatic Control, Gdańsk University of Technology, Narutowicza 11/12, Gdańsk, Poland

Received 12 October 2004; received in revised form 31 January 2005; accepted 7 April 2005
Available online 14 June 2005

Abstract

The problem of identification/tracking of quasi-periodically varying complex systems is considered. This problem is a generalization, to the system case, of a classical signal processing task of either elimination or extraction of nonstationary sinusoidal signals buried in noise. The proposed solution is based on the exponentially weighted basis function (EWBF) approach. First, the basic EWBF algorithm is derived. Then its frequency-decoupled, parallel-form and cascade-form variants, with highly modular structure and reduced computational requirements, are described. Finally, the frequency-adaptive versions of all schemes are obtained using the recursive prediction error method.

Keywords: System identification; Time-varying processes; Frequency estimation; Basis function approach

1. Introduction

Consider the problem of identification/tracking of coefficients of a complex time-varying system governed by

$$y(t) = \sum_{l=1}^{n} \theta_l(t)u(t-l+1) + v(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}(t) + v(t), \tag{1}$$

where $t=1, 2, \ldots$ denotes the normalized discrete time, $y(t)$ denotes the system output, $\boldsymbol{\varphi}(t)=[u(t), \ldots, u(t-n+1)]^{\mathrm{T}}$ is the regression vector made up of the past input samples, $v(t)$ is an additive (white) noise, uncorrelated with $u(t)$, and $\boldsymbol{\theta}(t)=[\theta_1(t), \ldots, \theta_n(t)]^{\mathrm{T}}$ denotes the vector of time-varying impulse response coefficients, modeled as weighted sums of complex exponentials

$$\theta_l(t) = \sum_{i=1}^{k} a_{li}(t)\mathrm{e}^{\mathrm{j}\sum_{s=1}^{t}\omega_i(s)}, \quad l = 1, \ldots, n. \tag{2}$$

The main purpose of this paper is development of algorithms capable of tracking $\boldsymbol{\theta}(t)$. Since the amplitudes and frequencies in (2) are time varying, system parameters change over time in a periodic-like but not exactly periodic manner. We will assume that for every frequency component i, $i=1, \ldots, k$, the expansion coefficients $a_{li}(t), l=1, \ldots, n$ and the normalized angular frequencies $\omega_i(t) \in (-\pi, \pi]$ are slowly time varying. The system, governed by (1)–(2), which obeys the above-mentioned limitation, will be further referred to as *quasi-periodically* time varying.

One of the challenging potential applications, which under certain conditions admits formulation presented above, is adaptive equalization of rapidly fading communication channels. In modern wireless systems, distortion introduced into the transmitted signals is caused mostly by the multi-path effect—the signal reaches the receiver along different paths, i.e. with different time delays. When the multi-path effects are dominated by few strong reflectors (scatterers)

☆A part of this paper was presented at the 13th IFAC Symposium on System identification, Rotterdam, The Netherlands, August 2003. This paper was recommended for publication in revised form by Associate Editor Brett Ninness under the direction of Editor Torsten Soederstroem. This work was supported by KBN under Grant 4T 11A 01225.

* Corresponding author. Tel.: + 48 58 3472519; +48 58 3415821.
E-mail addresses: maciekn@eti.pg.gda.pl (M. Niedźwiecki), piokacz@proterians.net.pl (P. Kaczmarek).

0005-1098/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2005.04.011

and when the transmitter and/or receiver moves with a constant speed, the impulse response of the channel (along with the transmitter and receiver filters) can be modeled in the form

$$\theta_l(t) = \sum_{i=1}^{k} a_{li} e^{j\omega_i t}, \quad l = 1, \ldots, n. \tag{3}$$

In this particular case $y(t)$ denotes the sampled baseband signal, received by the mobile radio system, $u(t), t=1, 2, \ldots$ denotes the sequence of transmitted (complex) symbols, k stands for the number of different signal paths and $\omega_i, i = 1, \ldots, k$ are the corresponding Doppler shifts. When the speed of the vehicle changes over time, Doppler shifts are also time varying, which in a straightforward way leads to (2).

For mobile radio channels the sinusoidal model described above has a long history, which goes back to Aiken (1967). Quite recently, a number of papers explored the possibility of using it for equalization purposes—see e.g. Tsatsanis and Giannakis (1996), Giannakis and Tepedelenlioğlu (1998) and Bakkoury, Roviras, Ghogho, and Castanie (2000).

For $n = 1$ and $u(t) = 1 \ \forall t$ model (1)–(2) becomes a description of a noisy nonstationary multi-frequency signal

$$y(t) = \theta(t) + v(t) = \sum_{i=1}^{k} a_i(t) e^{j\sum_{s=1}^{t} \omega_i(s)} + v(t). \tag{4}$$

The problem of either elimination or extraction of nonstationary sinusoidal signals buried in noise has attracted a great deal of attention of researchers in the field of signal processing. Most of the solutions proposed so far are based on adaptive notch filtering. The conventional notch filter is designed to cancel a sinusoidal interference with known constant frequency ω_0. When the instantaneous frequency $\omega(t)$ of the signal is subject to small changes around ω_0, efficient cancellation is still possible, provided that the bandwidth of the notch filter is widened to accommodate for possible changes in $\omega(t)$. When the frequency changes are not local, e.g. there is a frequency drift, the frequency-adaptive versions of the notch filter must be designed. Several different algorithms for cancelling/retrieval of complex sinusoids, called cisoids, were proposed, e.g. by Li and Milstein (1983), Pei and Tseng (1994), Tichavský and Händel (1995) and Padmanabham and Tichavský (1996). An interesting comparative study of some of these solutions can be found in the paper by Tichavský and Nehorai (1997).

When restricted to the special signal case discussed above, the results developed in our paper offer new solutions to the problem of frequency tracking and adaptive notch filtering of complex signals. It should be stressed, however, that signal processing algorithms do not constitute the main contribution of the paper and are regarded here merely as a byproduct of the system-oriented analysis. Their relationship to the existing solutions was examined by Niedźwiecki and Kaczmarek (2005).

To our best knowledge, the paper presents the first rigorous treatment of the problem of identification of quasi-periodically varying systems. We are aware of only two earlier contributions to this problem, presented in the papers of Tsatsanis and Giannakis (1996) and Bakkoury et al. (2000). However, since the identification algorithms proposed there were based on the assumption that the estimated frequencies are unknown but constant, they are not capable of tracking time-varying frequencies unless some heuristic modifications are introduced.

The algorithms presented below are based on the exponentially weighted least squares (EWLS) approach to the problem of system parameter and frequency tracking. In the signal case ($n = 1$) an alternative, state space formulation of the tracking problem, which leads in a straightforward way to solutions based on extended Kalman filter (EKF), was proposed by Parker and Anderson (1990) and Hilands and Thomopoulos (1997)—for real-valued signals, and by Nishiyama (2000)—for complex-valued signals. Even though the same method can be, in principle, applied in the system case, we are not aware of any EKF-based contribution to the problem of identification of quasi-periodically varying systems.

The problem of identification of real systems (systems with input/output signals and parameters described by real numbers) was considered in the paper by Niedźwiecki and Kaczmarek (2004b). It should be stressed, that the results presented below cannot be obtained as a simple modification or extension of results derived for real systems.

The paper is organized as follows. In Section 2 the identification problem is solved for a system with known constant frequencies of parameter variation. Two frequency-decoupled versions of the resulting 'global' estimation algorithm (parallel-form and cascade-form), with reduced computational complexity, are next proposed. The decoupled algorithms are made up of identical (from the functional viewpoint) blocks, which take care of different frequency modes of the identified system. In Section 3 the recursive prediction error approach is used to derive frequency-adaptive versions of all algorithms described in Section 2. Section 4 contains a critical overview of two other algorithms for identification of quasi-periodically varying systems, proposed in the literature. Finally, Section 5 presents results of two simulation experiments.

2. Identification in the known frequencies case

Since the system governed by (1) and (2) is rapidly time varying (only the *changes of parameter changes* are assumed to be slow), the problem of estimation of $\boldsymbol{\theta}(t)$ cannot be solved satisfactorily using the weighted least squares (WLS) or least mean squares (LMS) approach. The WLS/LMS algorithms are designed for slowly varying systems and are not capable of tracking fast parameter changes (Niedźwiecki, 2000). To obtain good estimation results, one has to rely on more specialized adaptive filters, such as the basis function (BF) algorithms. The basis function approach derives its name from the underlying model of parameter changes: it is assumed that the parameter trajectory can be approximated by a linear combination of known functions of time, called basis functions. When the frequencies $\omega_1, \dots, \omega_k$ are known and constant, the obvious choice of basis functions for (2) is: $e^{j\omega_1 t}, \dots, e^{j\omega_k t}$. For a predefined basis the method of basis functions, which was originated by Subba Rao (1970), is a straightforward extension of the least squares approach. When the frequencies are not known and/or time varying, solution of the identification problem, discussed in Section 3, is not trivial any more.

Suppose, for the time being, that both the amplitudes and angular frequencies in (2) are constant, i.e. that the changes in system parameters are governed by (3). Let

$$\boldsymbol{\alpha}_i = [a_{1i}, \dots, a_{ni}]^{\mathrm{T}}, \quad \boldsymbol{\psi}_i(t) = \boldsymbol{\varphi}(t) e^{j\omega_i t}, \quad i = 1, \dots, k. \tag{5}$$

Using the short-hand notation introduced above, (1) can be rewritten in the form

$$y(t) = \sum_{i=1}^{k} \boldsymbol{\psi}_i^{\mathrm{T}}(t)\boldsymbol{\alpha}_i + v(t) = \boldsymbol{\psi}^{\mathrm{T}}(t)\boldsymbol{\alpha} + v(t), \tag{6}$$

where

$$\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1^{\mathrm{T}}, \dots, \boldsymbol{\alpha}_k^{\mathrm{T}}]^{\mathrm{T}},$$
$$\boldsymbol{\psi}(t) = [\boldsymbol{\psi}_1^{\mathrm{T}}(t), \dots, \boldsymbol{\psi}_k^{\mathrm{T}}(t)]^{\mathrm{T}} = \mathbf{f}(t) \otimes \boldsymbol{\varphi}(t),$$
$$\mathbf{f}(t) = [e^{j\omega_1 t}, \dots, e^{j\omega_k t}]^{\mathrm{T}}$$

and $\otimes$ denotes the Kronecker product. Note that $\boldsymbol{\alpha}_i$ is the vector of coefficients associated with a particular frequency ω_i and *not* with a particular impulse response parameter $\boldsymbol{\theta}_i(t)$. Similarly, $\boldsymbol{\psi}_i(t)$ is the generalized regression vector associated with the ith frequency component. This nonstandard parameterization was adopted deliberately. Later on it will allow us to easily derive the frequency-decoupled versions of the estimation algorithm.

2.1. Basic algorithm

Suppose now that the vector $\boldsymbol{\alpha}$ is slowly varying with time. It is known that, in the case considered, one can track $\boldsymbol{\alpha}(t)$ using the method of EWLS. The EWLS estimate of $\boldsymbol{\alpha}(t)$ can be obtained from

$$\widehat{\boldsymbol{\alpha}}(t) = \arg\min_{\boldsymbol{\alpha}} \sum_{s=1}^{t} \lambda^{t-s} |y(s) - \boldsymbol{\psi}^{\mathrm{T}}(s)\boldsymbol{\alpha}|^2, \tag{7}$$

where λ $(0 < \lambda < 1, 1 - \lambda \ll 1)$ denotes the so-called forgetting constant—the design parameter which controls the memory of the estimator, and hence allows one to trade-off between its tracking speed and tracking accuracy. Based on (7) one can estimate system parameters using

$$\widehat{\boldsymbol{\theta}}(t) = \mathbf{D}(t)\widehat{\boldsymbol{\alpha}}(t), \tag{8}$$

where $\mathbf{D}(t) = \mathbf{f}^{\mathrm{T}}(t) \otimes \mathbf{I}_n$ and $\mathbf{I}_n$ denotes the $n \times n$ identity matrix.

The recursive algorithm for evaluation of $\widehat{\boldsymbol{\theta}}(t)$ is given by

$$\begin{aligned}
\mathbf{f}(t) &= \mathbf{A}\mathbf{f}(t-1),\\
\boldsymbol{\psi}(t) &= \mathbf{f}(t) \otimes \boldsymbol{\varphi}(t),\\
\varepsilon(t) &= y(t) - \boldsymbol{\psi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\alpha}}(t-1),\\
\widehat{\boldsymbol{\alpha}}(t) &= \widehat{\boldsymbol{\alpha}}(t-1) + \mathbf{l}^*(t)\varepsilon(t),\\
\mathbf{l}(t) &= \frac{\mathbf{Q}(t-1)\boldsymbol{\psi}(t)}{\lambda + \boldsymbol{\psi}^{\mathrm{H}}(t)\mathbf{Q}(t-1)\boldsymbol{\psi}(t)},\\
\mathbf{Q}(t) &= \frac{1}{\lambda}[\mathbf{Q}(t-1) - \mathbf{l}(t)\boldsymbol{\psi}^{\mathrm{H}}(t)\mathbf{Q}(t-1)],\\
\widehat{\boldsymbol{\theta}}(t) &= \mathbf{D}(t)\widehat{\boldsymbol{\alpha}}(t),
\end{aligned} \tag{9}$$

where $\mathbf{A} = \mathrm{diag}\{e^{j\omega_1}, \dots, e^{j\omega_k}\}$.

Since the algorithm (9) combines the basis function parameterization with EWLS estimation, it will be further referred to as the exponentially weighted basis function (EWBF) algorithm (Niedźwiecki, 2000).

Another, equivalent form of the EWBF estimator, which will be very useful for our purposes, can be obtained by rewriting (9) in a different system of coordinates. Using the linear time-varying transformation

$$\begin{aligned}
&\widehat{\boldsymbol{\beta}}(t) = \mathbf{A}_n^{t+1}\widehat{\boldsymbol{\alpha}}(t), \quad \mathbf{P}(t) = \mathbf{A}_n^{-(t+1)}\mathbf{Q}(t)\mathbf{A}_n^{t+1},\\
&\mathbf{k}(t) = \mathbf{A}_n^{-t}\mathbf{l}(t),
\end{aligned} \tag{10}$$

where $\mathbf{A}_n = \mathbf{A} \otimes \mathbf{I}_n$, one can easily convert (9) into

$$\begin{aligned}
\varepsilon(t) &= y(t) - \boldsymbol{\varphi}_k^{\mathrm{T}}(t)\widehat{\boldsymbol{\beta}}(t-1),\\
\widehat{\boldsymbol{\beta}}(t) &= \mathbf{A}_n[\widehat{\boldsymbol{\beta}}(t-1) + \mathbf{k}^*(t)\varepsilon(t)],\\
\mathbf{k}(t) &= \frac{\mathbf{P}(t-1)\boldsymbol{\varphi}_k(t)}{\lambda + \boldsymbol{\varphi}_k^{\mathrm{H}}(t)\mathbf{P}(t-1)\boldsymbol{\varphi}_k(t)},\\
\mathbf{P}(t) &= \frac{1}{\lambda}\mathbf{A}_n^*[\mathbf{P}(t-1) - \mathbf{k}(t)\boldsymbol{\varphi}_k^{\mathrm{H}}(t)\mathbf{P}(t-1)]\mathbf{A}_n,\\
\widehat{\boldsymbol{\theta}}(t) &= \mathbf{D}_0\widehat{\boldsymbol{\beta}}(t),
\end{aligned} \tag{11}$$

where $\boldsymbol{\varphi}_k(t) = \mathbf{A}_n^{-t}\boldsymbol{\psi}(t) = \mathbf{f}(0) \otimes \boldsymbol{\varphi}(t) = [\underbrace{\boldsymbol{\varphi}^{\mathrm{T}}(t), \dots, \boldsymbol{\varphi}^{\mathrm{T}}(t)}_{k}]^{\mathrm{T}}$, and $\mathbf{D}_0 = \mathbf{D}(t)\mathbf{A}^{-(t+1)} = (\mathbf{f}^{\mathrm{T}}(t) \otimes \mathbf{I}_n)(\mathbf{A}^{-(t+1)} \otimes \mathbf{I}_n) = \mathbf{f}^{\mathrm{H}}(1) \otimes \mathbf{I}_n$. The last transformation follows from the identity $(\mathbf{X} \otimes \mathbf{Y})(\mathbf{P} \otimes \mathbf{Q}) = \mathbf{XP} \otimes \mathbf{YQ}$ which holds for Kronecker products.

Note that the regression vector $\boldsymbol{\varphi}_k(t)$, which appears in (11), does not depend on the frequencies $\omega_1, \dots, \omega_k$, while the components of the generalized regression vector $\boldsymbol{\psi}(t)$, appearing in the original algorithm (9), are frequency dependent. For this reason the transformed algorithm is a more convenient starting point for derivation of the frequency-adaptive version of the EWBF algorithm (see Section 3).

It is interesting to note that the reparameterized algorithm (11) corresponds to the following *backward-time* description of parameter changes

$$\theta_l(t-s+1)=\sum_{i=1}^{k} b_{li}(t)\mathrm{e}^{-\mathrm{j}\omega_i s}, \quad s\in T_t, \tag{12}$$

where $T_t=[1,\ldots,t]$ and $b_{li}(t)=\mathrm{e}^{\mathrm{j}\omega_i(t+1)}a_{li}$, $l=1,\ldots,n$, $i=1,\ldots,k$. Since, for every $t\geqslant k$, the subspace spanned by the basis set $\{\mathrm{e}^{\mathrm{j}\omega_1 s},\ldots,\mathrm{e}^{\mathrm{j}\omega_k s}, s\in T_t\}$ is identical with the subspace spanned by the time-reversed conjugate basis set $\{\mathrm{e}^{-\mathrm{j}\omega_1(t-s+1)},\ldots,\mathrm{e}^{-\mathrm{j}\omega_k(t-s+1)}, s\in T_t\}$, the backward-time description (12) is equivalent to the forward-time description (3). Note that in the second case the basis set is always 'fixed' at $s=t$, i.e. at the end of the analysis interval. Following (Niedźwiecki, 1990, 2000), the EWBF algorithm (9), based on the forward-time model, will be referred to as the running basis (RB) algorithm and the EWBF algorithm (11), based on the backward-time model, will be referred to as the fixed basis (FB) algorithm. It should be stressed that both algorithms are strictly input–output equivalent, i.e. they yield identical parameter estimates $\widehat{\boldsymbol{\theta}}(t)$ for identical data sets $\{u(s), y(s), s=1,\ldots,t\}$ and appropriately chosen initial conditions ($\widehat{\boldsymbol{\beta}}(0)=\mathbf{A}_n\widehat{\boldsymbol{\alpha}}(0)$, $\mathbf{P}(0)=\mathbf{A}_n^*\mathbf{Q}(0)\mathbf{A}_n$).

The tracking characteristics of the fixed basis EWBF estimator, such as its associated frequency response and the equivalent estimation memory, were derived and analyzed in the paper by Niedźwiecki and Kłaput (2003). Since the system parameterization adopted in the paper by Niedźwiecki and Kłaput (2003) is different from the currently used parameterization, the FB algorithm presented there is equivalent, but not identical with (11). Therefore, some caution is needed when using the results of Niedźwiecki and Kłaput (2003) in our current context.

2.2. *Parallel decomposition*

Denote by

$$y_i(t)=\boldsymbol{\psi}_i^{\mathrm{T}}(t)\boldsymbol{\alpha}_i+v(t)$$

the output of the ith subsystem of (6), i.e. subsystem associated with the frequency ω_i. Even though the signal $y_i(t)$ is not available, one can easily estimate it using the formula

$$\widehat{y}_i(t)=y(t)-\sum_{\substack{l=1\\ l\neq i}}^{k}\widehat{y}_l(t|t-1),$$

where

$$\widehat{y}_i(t|t-1)=\boldsymbol{\psi}_i^{\mathrm{T}}(t)\widehat{\boldsymbol{\alpha}}_i(t-1)=\boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\beta}}_i(t-1)$$

is the predicted value of $y_i(t)$ yielded by the estimation algorithm designed to track parameters of the ith subsystem.

Estimation of $y_i(t)$, in the way described above, allows one to decompose the tracking algorithm, i.e. to replace one

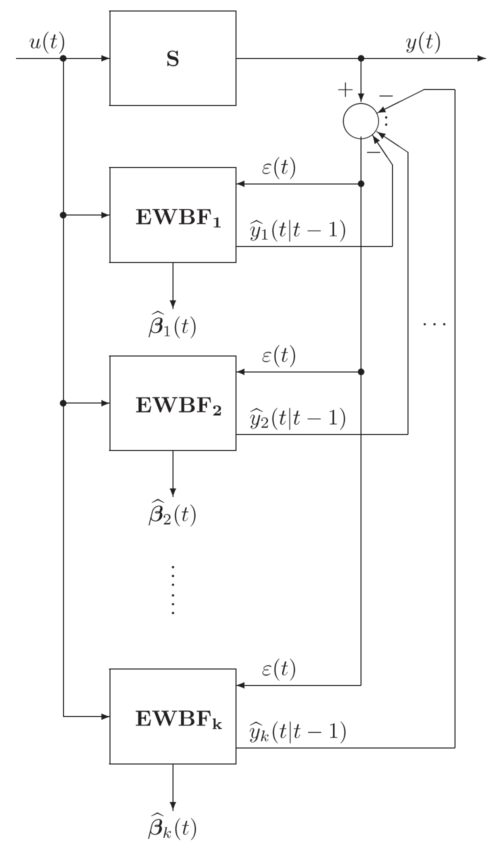

Fig. 1. Block diagram of the parallel-form implementation of the EWBF algorithm.

'global' algorithm (11) with k mutually coupled 'local' algorithms, each of which takes care of a particular subsystem. The ith component algorithm can be easily derived from (11) by setting $\mathbf{A}_n=\rho_i\mathbf{I}_n$, $\rho_i=\mathrm{e}^{\mathrm{j}\omega_i}$ and $\boldsymbol{\varphi}_k(t)=\boldsymbol{\varphi}(t)$. To add some extra design flexibility, we will equip each subalgorithm with an independently assigned forgetting factor λ_i. The resulting decoupled parallel-form EWBF algorithm (P-EWBF) can be written down as follows (see Fig. 1)

$$\varepsilon_i(t)=\widehat{y}_i(t)-\boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\beta}}_i(t-1),$$

$$\widehat{\boldsymbol{\beta}}_i(t)=\rho_i[\widehat{\boldsymbol{\beta}}_i(t-1)+\mathbf{k}_i^*(t)\varepsilon_i(t)],$$

$$\mathbf{k}_i(t)=\frac{\mathbf{P}_i(t-1)\boldsymbol{\varphi}(t)}{\lambda_i+\boldsymbol{\varphi}^{\mathrm{H}}(t)\mathbf{P}_i(t-1)\boldsymbol{\varphi}(t)},$$

$$\mathbf{P}_i(t)=\frac{1}{\lambda_i}[\mathbf{I}_n-\mathbf{k}_i(t)\boldsymbol{\varphi}^{\mathrm{H}}(t)]\mathbf{P}_i(t-1),$$

$$i=1,\ldots,k,$$

$$\widehat{\boldsymbol{\theta}}(t)=\sum_{i=1}^{k}\rho_i^*\widehat{\boldsymbol{\beta}}_i(t). \tag{13}$$

Observe that $\varepsilon_1(t)=\cdots=\varepsilon_k(t)=y(t)-\sum_{i=1}^{k}\boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\beta}}_i(t-1)=\varepsilon(t)$, i.e. all subalgorithms are in fact driven by the same global prediction error $\varepsilon(t)$.

To analyze the parameter tracking properties of algorithm (13) we will rewrite the system equation (1) in the form

$$y(t) = \sum_{i=1}^{k} \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}^{(i)}(t) + v(t),$$

where $\boldsymbol{\theta}^{(i)}(t) = \mathrm{e}^{\mathrm{j}\omega_i t}\boldsymbol{\alpha}_i$ is the vector made up of all terms that change with the frequency ω_i. Note that $\boldsymbol{\theta}(t) = \sum_{i=1}^{k} \boldsymbol{\theta}^{(i)}(t)$. The quantity $\boldsymbol{\theta}^{(i)}(t)$ should not be confused with $\theta_i(t)$—the ith component of $\boldsymbol{\theta}(t)$. The estimate of $\boldsymbol{\theta}^{(i)}(t)$ can be obtained from

$$\widehat{\boldsymbol{\theta}}^{(i)}(t) = \rho_i^t \widehat{\boldsymbol{\alpha}}_i(t) = \rho_i^* \widehat{\boldsymbol{\beta}}_i(t).$$

Assuming that the input sequence $u(t)$ is stationary and ergodic, it is possible to derive the following steady state formula (see Appendix)

$$\bar{\boldsymbol{\theta}}^{(i)}(t) = \mathrm{E}[\widehat{\boldsymbol{\theta}}^{(i)}(t)] \cong T_i(q^{-1})\boldsymbol{\theta}(t), \tag{14}$$

where

$$T_i(q^{-1}) = \frac{1-\lambda_i}{1-\lambda_i \rho_i q^{-1}}$$

and the expectation is taken over $\{\boldsymbol{\varphi}(s), v(s), s \leqslant t\}$. It is important to note that (14) holds for *any* parameter trajectory, i.e. the derivation is *not* based on the assumption that the true parameter trajectory obeys (3).

According to (14), the ith identification algorithm behaves, on the average, as a narrow band extraction filter $T_i(q^{-1})$ centered at the frequency ω_i. When the forgetting constant λ_i is close to one, the 3 dB bandwidth of this filter can be approximately expressed as $B_{3\,\mathrm{dB}} \cong 2(1-\lambda_i)$.

Observe that the mean parameter estimation error is given by

$$\boldsymbol{\theta}(t) - \bar{\boldsymbol{\theta}}^{(i)}(t) \cong (1 - T_i(q^{-1}))\boldsymbol{\theta}(t) = N_i(q^{-1})\boldsymbol{\theta}(t),$$

where the filter

$$N_i(q^{-1}) = \frac{\lambda_i(1-\rho_i q^{-1})}{1-\lambda_i \rho_i q^{-1}}$$

can be easily recognized as the notch filter centered at the frequency ω_i. This allows one to consider the EWBF algorithm a generalized notch filter.

2.3. Cascade decomposition

The decoupled algorithm presented in the previous subsection is a parallel structure made up of k identical (from the functional viewpoint) blocks. Each block is designed to track a particular frequency component of the parameter vector $\boldsymbol{\theta}(t)$. Connecting the same blocks so that they form a cascade (see Fig. 2), one obtains an interesting alternative to the parallel decomposition. To obtain the cascade variant of the EWBF algorithm (C-EWBF), the first recursion in (13) should be replaced with

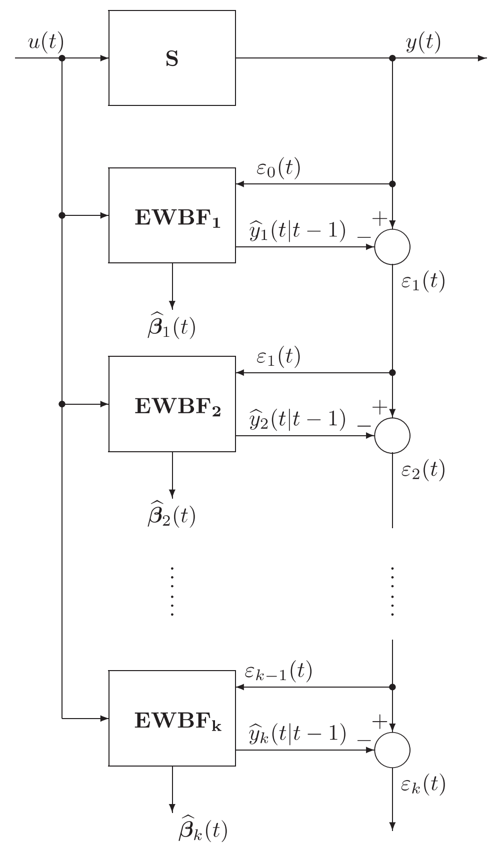

Fig. 2. Block diagram of the cascade-form implementation of the EWBF algorithm.

$$\varepsilon_i(t) = \varepsilon_{i-1}(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\beta}}_i(t-1), \tag{15}$$

where $\varepsilon_0(t) = y(t)$.

The cascade-form implementations are widespread in signal processing. Consider a multi-frequency notch filter realized as a cascade of single-frequency filters. When the bandwidths of the component filters are sufficiently narrow, so that they have negligible overlap, each section can eliminate one sinusoid. The first section cancels one sinusoid and transmits the rest of the signal, with small distortion, to the second stage, which cancels the second sinusoid (if present), etc. The cascade-form parameter tracking algorithm (15) exploits the same frequency decoupling property of narrowband filters. Suppose that the 'passbands' of the extraction filters $T_1(q^{-1}), \ldots, T_k(q^{-1})$ do not overlap, which can be always guaranteed by adopting forgetting factors $\lambda_1, \ldots, \lambda_k$ that are sufficiently close to one. Then

$$\begin{aligned}\varepsilon_1(t) &= y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\beta}}_1(t-1) \\ &= \sum_{i=2}^{k} \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}^{(i)}(t) + v(t) + \xi_1(t),\end{aligned}$$

where

$$\xi_1(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)[\boldsymbol{\theta}^{(1)}(t) - \widehat{\boldsymbol{\theta}}^{(1)}(t|t-1)].$$

If the bandwidth of the extraction filter is sufficiently narrow, the extraction error $\xi_1(t)$ is negligible giving

$$\varepsilon_1(t) \cong \sum_{i=2}^{k} \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}^{(i)}(t) + v(t).$$

Repeated use of this argument leads to

$$\varepsilon_j(t) \cong \sum_{i=j+1}^{k} \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}^{(i)}(t) + v(t), \quad j = 2, \ldots, k-1,$$

which means that the filters forming the cascade gradually reduce the 'spectral content' of the identified system.

In the parallel scheme, subsystems identified by different component algorithms do not overlap. In the cascade scheme the situation is different. Suppose that the frequency separation conditions, discussed earlier, are met. The subsystem identified ('seen') by the first algorithm ($\varepsilon_0(t) = y(t)$) is governed by

$$y(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}^{(1)}(t) + v(t) + e_1(t), \tag{16}$$

where

$$e_1(t) = \sum_{i=2}^{k} \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}^{(i)}(t).$$

According to (16), in addition to dealing with the measurement noise $v(t)$, the first filter, which aims at tracking $\boldsymbol{\theta}^{(1)}(t)$, must cope with an extra disturbance $e_1(t)$ due to the unmodeled process dynamics. And it is not until the last block in the cascade is reached that this unmodeled dynamics component is entirely eliminated. Based on this observation it is natural to expect that the estimates yielded by the first block will be prone to larger estimation errors than the estimates yielded by the second block, etc. Computer simulations, presented in the last section, confirm this conjecture.

3. Identification in the unknown frequencies case

Keeping the forgetting constant λ away from 1 guarantees that the EWBF algorithm will retain its tracking capabilities even if the corresponding frequencies $\omega_1, \ldots, \omega_k$ are subject to small changes around their nominal values—see Niedźwiecki and Kłaput (2003) for a detailed analysis of this property of EWBF filters. However, even though the EWBF filter is robust to small local changes in frequencies, it will fail to identify the system correctly in the presence of a frequency drift. For this reason in this section we will derive the frequency-adaptive EWBF algorithms, capable of tracking the time-varying frequencies $\omega_i(t), i = 1, \ldots, k$.

When system frequencies are unknown the identification problem becomes considerably more complicated. This is caused by the fact that in this case both the vector of expansion coefficients $\boldsymbol{\alpha}$ *and* the generalized regression vector $\boldsymbol{\psi}(t)$, which appear on the right-hand side of (6), are comprised of or depend on unknown and possibly time-varying quantities. As a result, model (6) becomes nonlinear in the estimated parameters.

Denote by $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_k]^{\mathrm{T}}$ the vector of unknown and/or time-varying frequencies and let $V(t, \boldsymbol{\omega})$ be the exponentially weighted measure of fit

$$V(t, \boldsymbol{\omega}) = \frac{1}{2} \sum_{s=1}^{t} \gamma^{t-s} |\varepsilon(s)|^2, \tag{17}$$

where γ, $0 < \gamma < 1$, is the forgetting constant, which will be used to control the speed of the frequency adaptation and $\varepsilon(s)$, as before, denotes the one-step-ahead prediction error yielded by the EWBF algorithm.

To evaluate the estimate

$$\widehat{\boldsymbol{\omega}}(t) = \arg\min_{\boldsymbol{\omega}} V(t, \boldsymbol{\omega})$$

we will use the recursive prediction error (RPE) approach. According to Söderström and Stoica (1988), the RPE algorithm can be expressed in the form

$$\widehat{\boldsymbol{\omega}}(t) = \widehat{\boldsymbol{\omega}}(t-1) - [V''(t, \widehat{\boldsymbol{\omega}}(t-1))]^{-1} V'(t, \widehat{\boldsymbol{\omega}}(t-1)),$$

where

$$V'(t, \widehat{\boldsymbol{\omega}}(t-1)) \cong \mathrm{Re}\left[\varepsilon(t, \widehat{\boldsymbol{\omega}}(t-1)) \frac{\partial \varepsilon^*(t, \widehat{\boldsymbol{\omega}}(t-1))}{\partial \boldsymbol{\omega}}\right],$$

$$V''(t, \widehat{\boldsymbol{\omega}}(t-1)) \cong \gamma V''(t-1, \widehat{\boldsymbol{\omega}}(t-2)) + \mathrm{Re}\left[\frac{\partial \varepsilon(t, \widehat{\boldsymbol{\omega}}(t-1))}{\partial \boldsymbol{\omega}} \frac{\partial \varepsilon^*(t, \widehat{\boldsymbol{\omega}}(t-1))}{\partial \boldsymbol{\omega}^{\mathrm{T}}}\right]$$

and all derivatives are taken with respect to $\boldsymbol{\omega}$. Observe that

$$\frac{\partial \varepsilon(t, \boldsymbol{\omega})}{\partial \boldsymbol{\omega}} = -\frac{\partial \widehat{\boldsymbol{\beta}}^{\mathrm{T}}(t-1, \boldsymbol{\omega})}{\partial \boldsymbol{\omega}} \boldsymbol{\varphi}_k(t),$$

$$\frac{\partial \widehat{\boldsymbol{\beta}}^{\mathrm{T}}(t, \boldsymbol{\omega})}{\partial \boldsymbol{\omega}} = \left[\frac{\partial \widehat{\boldsymbol{\beta}}(t, \boldsymbol{\omega})}{\partial \omega_1} \cdots \frac{\partial \widehat{\boldsymbol{\beta}}(t, \boldsymbol{\omega})}{\partial \omega_k}\right]^{\mathrm{T}},$$

$$\frac{\partial \mathbf{k}^{\mathrm{T}}(t, \boldsymbol{\omega})}{\partial \boldsymbol{\omega}} = \left[\frac{\partial \mathbf{k}(t, \boldsymbol{\omega})}{\partial \omega_1} \cdots \frac{\partial \mathbf{k}(t, \boldsymbol{\omega})}{\partial \omega_k}\right]^{\mathrm{T}}$$

and

$$\frac{\partial \widehat{\boldsymbol{\beta}}(t, \boldsymbol{\omega})}{\partial \omega_i} = \frac{\partial \mathbf{A}_n(t, \boldsymbol{\omega})}{\partial \omega_i} \mathbf{A}_n^*(t, \boldsymbol{\omega}) \widehat{\boldsymbol{\beta}}(t, \boldsymbol{\omega}) + \mathbf{A}_n(t, \boldsymbol{\omega}) \left[\frac{\partial \widehat{\boldsymbol{\beta}}(t-1, \boldsymbol{\omega})}{\partial \omega_i} + \frac{\partial \mathbf{k}^*(t, \boldsymbol{\omega})}{\partial \omega_i} \varepsilon(t, \boldsymbol{\omega}) + \mathbf{k}^*(t, \boldsymbol{\omega}) \frac{\partial \varepsilon(t, \boldsymbol{\omega})}{\partial \omega_i}\right],$$

$$\frac{\partial \mathbf{k}(t,\boldsymbol{\omega})}{\partial \omega_i} = \frac{\dfrac{\partial \mathbf{P}(t,\boldsymbol{\omega})}{\partial \omega_i}\boldsymbol{\varphi}_k(t)}{\lambda + \boldsymbol{\varphi}_k^{\mathrm{H}}(t)\mathbf{P}(t,\boldsymbol{\omega})\boldsymbol{\varphi}_k(t)} - \mathbf{k}(t,\boldsymbol{\omega})\frac{\boldsymbol{\varphi}_k^{\mathrm{H}}(t)\dfrac{\partial \mathbf{P}(t,\boldsymbol{\omega})}{\partial \omega_i}\boldsymbol{\varphi}_k(t)}{\lambda + \boldsymbol{\varphi}_k^{\mathrm{H}}(t)\mathbf{P}(t,\boldsymbol{\omega})\boldsymbol{\varphi}_k(t)},$$

$$\frac{\partial \mathbf{P}(t,\boldsymbol{\omega})}{\partial \omega_i} = \frac{j}{\lambda}[\boldsymbol{\Lambda}_i\mathbf{P}(t,\boldsymbol{\omega}) - \mathbf{P}(t,\boldsymbol{\omega})\boldsymbol{\Lambda}_i] + \frac{1}{\lambda}\mathbf{A}_n^*(t,\boldsymbol{\omega}) \times \left[\frac{\partial \mathbf{P}(t-1,\boldsymbol{\omega})}{\partial \omega_i} - \frac{\partial \mathbf{k}(t,\boldsymbol{\omega})}{\partial \omega_i}\boldsymbol{\varphi}_k^{\mathrm{H}}(t)\mathbf{P}(t-1,\boldsymbol{\omega}) - \mathbf{k}(t,\omega)\boldsymbol{\varphi}_k^{\mathrm{H}}(t)\frac{\partial \mathbf{P}(t-1,\boldsymbol{\omega})}{\partial \omega_i}\right]\mathbf{A}_n(t,\boldsymbol{\omega}),$$

where

$$\boldsymbol{\Lambda}_i = \mathrm{diag}\{\underbrace{0,\ldots,0}_{\mathrm{i-1}},1,\underbrace{0,\ldots,0}_{\mathrm{k-i}}\} \otimes \mathbf{I}_n.$$

Let

$$\varepsilon(t) = \varepsilon(t,\widehat{\boldsymbol{\omega}}(t)),$$

$$\boldsymbol{\eta}(t) = \partial\varepsilon(t,\widehat{\boldsymbol{\omega}}(t-1))/\partial\boldsymbol{\omega},$$

$$\mathbf{P}(t) = \mathbf{P}(t,\widehat{\boldsymbol{\omega}}(t)),$$

$$\mathbf{k}(t) = \mathbf{k}(t,\widehat{\boldsymbol{\omega}}(t)),$$

$$\mathbf{G}(t) = V''(t,\widehat{\boldsymbol{\omega}}(t-1)),$$

$$\widehat{\mathbf{A}}(t) = \mathrm{diag}\{\mathrm{e}^{\mathrm{j}\widehat{\omega}_1(t)},\ldots,\mathrm{e}^{\mathrm{j}\widehat{\omega}_k(t)}\},$$

$$\widehat{\mathbf{A}}_n(t) = \widehat{\mathbf{A}}(t) \otimes \mathbf{I}_n,$$

$$\widehat{\mathbf{D}}_0(t) = [\mathrm{e}^{-\mathrm{j}\widehat{\omega}_1(t)},\ldots,\mathrm{e}^{-\mathrm{j}\widehat{\omega}_k(t)}] \otimes \mathbf{I}_n,$$

$$\eta_i(t) = \partial\varepsilon(t,\widehat{\boldsymbol{\omega}}(t-1))/\partial\omega_i,$$

$$\boldsymbol{\xi}_i(t) = \partial\widehat{\boldsymbol{\beta}}(t,\widehat{\boldsymbol{\omega}}(t))/\partial\omega_i,$$

$$\widetilde{\mathbf{P}}_i(t) = \partial\mathbf{P}(t,\widehat{\boldsymbol{\omega}}(t))/\partial\omega_i,$$

$$\widetilde{\mathbf{k}}_i(t) = \partial\mathbf{k}(t,\widehat{\boldsymbol{\omega}}(t))/\partial\omega_i,$$

$$i = 1,\ldots,k.$$

Using the notation introduced above, the frequency-adaptive version of the fixed basis EWBF algorithm (11), further denoted as AEWBF, can be written down in the form

$$\varepsilon(t) = y(t) - \boldsymbol{\varphi}_k^{\mathrm{T}}(t)\widehat{\boldsymbol{\beta}}(t-1),$$

$$\eta_i(t) = -\boldsymbol{\varphi}_k^{\mathrm{T}}(t)\boldsymbol{\xi}_i(t-1), \quad i = 1,\ldots,k,$$

$$\mathbf{G}(t) = \gamma\mathbf{G}(t-1) + \mathrm{Re}[\boldsymbol{\eta}(t)\boldsymbol{\eta}^{\mathrm{H}}(t)],$$

$$\widehat{\boldsymbol{\omega}}(t) = \widehat{\boldsymbol{\omega}}(t-1) - \mathbf{G}^{-1}(t)\,\mathrm{Re}[\varepsilon(t)\boldsymbol{\eta}^*(t)],$$

$$\delta(t) = \lambda + \boldsymbol{\varphi}_k^{\mathrm{H}}(t)\mathbf{P}(t-1)\boldsymbol{\varphi}_k(t),$$

$$\mathbf{k}(t) = \delta^{-1}(t)\mathbf{P}(t-1)\boldsymbol{\varphi}_k(t),$$

$$\mathbf{P}(t) = \frac{1}{\lambda}\widehat{\mathbf{A}}_n^*(t)[\mathbf{P}(t-1) - \mathbf{k}(t)\boldsymbol{\varphi}_k^{\mathrm{H}}(t)\mathbf{P}(t-1)]\widehat{\mathbf{A}}_n(t),$$

$$\widehat{\boldsymbol{\beta}}(t) = \widehat{\mathbf{A}}_n(t)[\widehat{\boldsymbol{\beta}}(t-1) + \mathbf{k}^*(t)\varepsilon(t)],$$

$$\widetilde{\mathbf{k}}_i(t) = \delta^{-1}(t)[\widetilde{\mathbf{P}}_i(t-1)\boldsymbol{\varphi}_k(t) - \mathbf{k}(t)\boldsymbol{\varphi}_k^{\mathrm{H}}(t)\widetilde{\mathbf{P}}_i(t-1)\boldsymbol{\varphi}_k(t)],$$

$$\widetilde{\mathbf{P}}_i(t) = \frac{1}{\lambda}\widehat{\mathbf{A}}_n^*(t)[\widetilde{\mathbf{P}}_i(t-1) - \widetilde{\mathbf{k}}_i(t)\boldsymbol{\varphi}_k^{\mathrm{H}}(t)\mathbf{P}(t-1) - \mathbf{k}(t)\boldsymbol{\varphi}_k^{\mathrm{H}}(t)\widetilde{\mathbf{P}}_i(t-1)]\widehat{\mathbf{A}}_n(t) + \frac{j}{\lambda}[\mathbf{P}(t)\boldsymbol{\Lambda}_i - \boldsymbol{\Lambda}_i\mathbf{P}(t)],$$

$$\boldsymbol{\xi}_i(t) = j\boldsymbol{\Lambda}_i\widehat{\boldsymbol{\beta}}(t) + \widehat{\mathbf{A}}_n(t)[\boldsymbol{\xi}_i(t-1) + \widetilde{\mathbf{k}}_i^*(t)\varepsilon(t) + \mathbf{k}^*(t)\eta_i(t)],$$

$$i = 1,\ldots,k,$$

$$\widehat{\boldsymbol{\theta}}(t) = \widehat{\mathbf{D}}_0(t)\widehat{\boldsymbol{\beta}}(t). \tag{18}$$

Using the well-known matrix inversion lemma (Söderström & Stoica, 1988) one arrives at the following recursive scheme for computation of the matrix $\mathbf{G}^{-1}(t)$ in (18)

$$\mathbf{G}^{-1}(t) = \mathrm{Re}\{\mathbf{F}(t)\},$$

$$\mathbf{F}(t) = \frac{1}{\gamma}\left[\mathbf{F}(t-1) - \frac{\mathbf{F}(t-1)\boldsymbol{\eta}(t)\boldsymbol{\eta}^{\mathrm{H}}(t)\mathbf{F}(t-1)}{\gamma + \boldsymbol{\eta}^{\mathrm{H}}(t)\mathbf{F}(t-1)\boldsymbol{\eta}(t)}\right].$$

The procedure summarized above will be further referred to as the global search algorithm.

The frequency-adaptive counterparts of the decomposed EWBF algorithms (13) and (15) (denoted as P-AEWBF and C-AEWBF, respectively) can be written down in the following unified form

$$\varepsilon_i(t) = z_i(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\beta}}_i(t-1),$$

$$\eta_i(t) = -\boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\xi}_i(t-1),$$

$$g_i(t) = \gamma_i g_i(t-1) + |\eta_i(t)|^2,$$

$$\widehat{\omega}_i(t) = \widehat{\omega}_i(t-1) - g_i^{-1}(t)\,\mathrm{Re}[\varepsilon_i(t)\eta_i^*(t)],$$

$$\widehat{\rho}_i(t) = \mathrm{e}^{\mathrm{j}\widehat{\omega}_i(t)},$$

$$\mathbf{k}_i(t) = \frac{\mathbf{P}_i(t-1)\boldsymbol{\varphi}(t)}{\lambda_i + \boldsymbol{\varphi}^{\mathrm{H}}(t)\mathbf{P}_i(t-1)\boldsymbol{\varphi}(t)},$$

$$\mathbf{P}_i(t) = \frac{1}{\lambda_i}[\mathbf{I}_n - \mathbf{k}_i(t)\boldsymbol{\varphi}^{\mathrm{H}}(t)]\mathbf{P}_i(t-1),$$

$$\widehat{\boldsymbol{\beta}}_i(t) = \widehat{\rho}_i(t)[\widehat{\boldsymbol{\beta}}_i(t-1) + \mathbf{k}_i^*(t)\varepsilon_i(t)],$$

$$\boldsymbol{\xi}_i(t) = j\widehat{\boldsymbol{\beta}}_i(t) + \widehat{\rho}_i(t)[\boldsymbol{\xi}_i(t-1) + \mathbf{k}_i^*(t)\eta_i(t)],$$

$$i = 1,\ldots,k,$$

$$\widehat{\boldsymbol{\theta}}(t) = \sum_{i=1}^{k}\widehat{\rho}_i^*(t)\widehat{\boldsymbol{\beta}}_i(t). \tag{19}$$

In the parallel realization one should set $z_i(t)=\widehat{y}_i(t)$, which leads to $\varepsilon_1(t)=\cdots=\varepsilon_k(t)=y(t)-\boldsymbol{\varphi}^{\mathrm{T}}(t)\sum_{i=1}^{k}\widehat{\boldsymbol{\beta}}_i(t-1)=\varepsilon(t)$. In the cascade realization one should use $z_1(t)=y(t)$ and $z_i(t)=\varepsilon_{i-1}(t)$, $i>1$.

Since the quantities $\mathbf{k}_i(t)$ and $\mathbf{P}_i(t)$, which appear in (13) and (15), do not depend on $\boldsymbol{\omega}$, one arrives at $\widetilde{\mathbf{k}}_i(t)=\mathbf{0}$, $\widetilde{\mathbf{P}}_i(t)=\mathbf{O}$, $i=1,\ldots,k$. For this reason the corresponding terms are absent from the decomposed algorithm (19), which further simplifies its computational structure compared to (18).

3.1. Initialization

To start the global search algorithm one should set $\widehat{\boldsymbol{\beta}}(0)=\mathbf{0}$, $\mathbf{P}(0)=\delta\mathbf{I}_{kn}$ where δ denotes a large positive constant—this is a standard initialization procedure for all RLS-type recursive estimation algorithms (Söderström & Stoica, 1988). The recommended settings for the remaining startup variables are $\boldsymbol{\xi}_i(0)=\mathbf{0}$, $\widetilde{\mathbf{P}}_i(0)=\mathbf{0}$, $i=1,\ldots,k$ and $\mathbf{G}(0)=\varepsilon\mathbf{I}_k$, where ε is a small positive constant.

The analogous initial conditions for the decomposed algorithms are

$$\widehat{\boldsymbol{\beta}}_i(0)=\mathbf{0},\quad \mathbf{P}_i(0)=\delta\mathbf{I}_n,\quad g_i(0)=\varepsilon,\quad \boldsymbol{\xi}_i(0)=\mathbf{0},\quad i=1,\ldots,k. \tag{20}$$

Some caution is needed when choosing the initial frequency estimates. When no prior knowledge about the frequencies $\omega_1,\ldots,\omega_k$ is available, one can set

$$\widehat{\omega}_i(0)=0,\quad i=1,\ldots,k. \tag{21}$$

Such zero initial conditions yield good results when applied to the global search and cascade-form algorithms, but under certain circumstances they may cause problems when applied to the parallel-form algorithm.

Actually, suppose that all subalgorithms comprising the parallel scheme are initialized according to (20)–(21) and that they are equipped with the same forgetting constants, i.e. $\lambda_1=\cdots=\lambda_k=\lambda$ and $\gamma_1=\cdots=\gamma_k=\gamma$. Then, due to the symmetry of the parallel scheme, the identical initial conditions will force all subalgorithms to behave identically also for $t>0$. Since this implies that $\widehat{\omega}_1(t)=\cdots=\widehat{\omega}_k(t)\ \forall t$, it is clear that the frequency adaptation mechanism is in this case crippled. Note that the situation does not change if nonzero initial conditions are adopted as long as the same starting values are used in all subalgorithms.

The initial convergence deadlock, described above, is a typical unstable equilibrium point. Therefore, even the slightest differences in the initial conditions will make parallel-form algorithm work correctly. The easiest way of avoiding initial convergence problems is to use randomly generated initial conditions $\widehat{\omega}_i(t)\sim N(0,\sigma_0^2)$, $i=1,\ldots,k$ where σ_0^2 is a small number (e.g. $\sigma_0^2=10^{-4}$).

3.2. Selection of design variables

The tracking properties of the global search AEWBF algorithm (18) depend on two design variables: the forgetting constant λ, which controls the bandwidth of parameter tracking, and the forgetting constant γ, which controls the bandwidth of frequency tracking. Generally speaking, both forgetting factors should be chosen so as to trade-off the tracking speed of a generalized adaptive notch filter (which decreases with growing λ and γ) and its noise rejection capability (which increases with growing λ and γ). We have found out experimentally that good results are usually obtained when the frequency tracking bandwidth is larger (but not significantly larger) than the parameter tracking bandwidth, e.g.

$$1-\gamma=2(1-\lambda). \tag{22}$$

The tuning problem is then reduced to selection of a single design variable λ.

For a simplified, gradient version of the AEWBF algorithm some analytical results are now available, showing how performance of a generalized adaptive notch filter depends on selection of its design variables—see Niedźwiecki and Kaczmarek (2005). Even though restricted to a simple case (single-frequency mode, random walk frequency variation), the analysis presented in the paper by Niedźwiecki and Kaczmarek (2005) provides interesting insights into the tracking capabilities and tracking limitations of generalized adaptive notch filters, including the associated speed/accuracy trade-offs, performance optimization issues and the achievable tracking bounds. Extension of these results to the RPE algorithm (18) seems to be a difficult task. Nevertheless, most of the qualitative observations made by Niedźwiecki and Kaczmarek (2005) remain valid in this more complicated case.

When subsystems corresponding to different frequency modes change with different speeds, the frequency-decoupled parallel-form and cascade-form algorithms (19), allowing one to assign forgetting factors λ_i and γ_i individually for each of k subalgorithms, have some potential advantage over the global search algorithm (18), which applies the same forgetting factors to all frequency modes. Of course, to make good use of this extra design flexibility, one should either know or estimate the degree of nonstationarity of the identified modes. Therefore, unless the forgetting factors are assigned in an adaptive fashion (such fully adaptive schemes were already proposed in the signal identification case, see e.g. Dragošević & Stanković, 1995a,b), we recommend using identical settings in all subalgorithms ($\lambda_1=\cdots=\lambda_k$, $\gamma_1=\cdots=\gamma_k$).

3.3. Selection of the number of frequency modes

So far we have assumed that the number of harmonic components, denoted by k, is known a priori. The problem of estimation of k was recently analyzed by Niedźwiecki and Kaczmarek (2004a). The proposed solution is based on localization of dominant peaks of the so-called system periodogram, evaluated for a short initial fragment of input/output data. System periodogram is a generalization, to the system case, of a classic concept of signal periodogram. As shown in the paper by Niedźwiecki and Kaczmarek (2004a), the number of statistically significant periodogram peaks can be reliably determined using the appropriately redefined version of the Akaike's final prediction error (FPE) criterion. Moreover, the results of such nonparametric analysis can be used to determine initial conditions needed to smoothly start (start without initialization transients), or restart, the model-based tracking algorithms.

Another approach to the problem of selection of k, based on simultaneous detection of pairs of related peaks in two cyclic spectra of system output, was described by Tsatsanis and Giannakis (1996).

From the practical viewpoint it is important to know what happens when the adopted number of frequency modes is greater or smaller than the number of true frequencies. The rigorous analysis of the effects of overparameterization and underparametrization is beyond the scope of this paper. We note, however, that we have no simulation evidence of any serious problems caused by overparameterization. Quite obviously, the cascade-form implementation is insensitive to overparameterization. Since all signals are propagated in one direction, from the first block to the last one, addition of superfluous blocks at the end of the cascade does not affect operation of the preceding blocks. The corresponding superfluous frequency estimates seem to randomly fluctuate, with large variance, around the zero frequency, without adversely affecting the true frequency estimates. In the parallel structure, the superfluous estimates tend to lock onto the true frequencies. The same behavior was observed for the global search algorithm.

If the number of frequencies is underestimated (which is not very likely to happen if the FPE test is used, since the Akaike's criteria show tendency to overfit the selected models), all algorithms tend to focus on dominant, i.e. largest-power frequency modes.

3.4. *Computational complexity*

Table 1 presents comparison of computational complexity (the number of complex multiply/add operations per time update) of the basic (global search) frequency-adaptive algorithm (AEWBF), given by (18), and for its decomposed, parallel-form (P-AEWBF) and cascade-form (C-AEWBF) variants, given by (19). To show computational overhead due to frequency adaptation, the analogous data for the fixed-frequency RB EWBF algorithm (9), FB EWBF algorithm (11) and the decomposed P-EWBF and C-EWBF algorithms (given by (13) and (15), respectively), were also included in Table 1. All evaluations take into consideration special features of some of the involved matrices, such as conjugate symmetry ($\mathbf{Q}(t)$, $\mathbf{P}(t)$, $\mathbf{F}(t)$, $\mathbf{P}_i(t)$, $\widetilde{\mathbf{P}}_i(t)$, $i=1,\ldots,k$) and diagonal or block diagonal structure ($\mathbf{A}$, $\mathbf{A}_n$, $\widehat{\mathbf{A}}_n(t)$, $\mathbf{D}(t)$, $\mathbf{D}_0$, $\widehat{\mathbf{D}}_0(t)$). Note very high complexity of the frequency-adaptive version of the basic algorithm and low complexity (linear in the number of frequencies) of its decomposed variants.

Table 1
Comparison of computational complexity of different EWBF algorithms

Algorithm	Total complexity
RB EWBF	$2n^2k^2+6nk+k$
FB EWBF	$2.5n^2k^2+6.5nk-n^2k+k$
P-EWBF, C-EWBF	$2n^2k+7nk$
AEWBF	$2.5n^2k^2+7.5nk-n^2k+2k^2$ $+[4n^2k^2+6nk-n^2k+k]k$
P-AEWBF, C-AEWBF	$2n^2k+10nk+4k$

n denotes the number of system coefficients and k is the number of frequencies.

4. Comparison with known results

To our best knowledge the only results published so far on frequency tracking for the purpose of identification of quasi-periodically varying systems, are those presented in the papers of Tsatsanis and Giannakis (1996) and Bakkoury et al. (2000). Both papers suggest gradient frequency corrections that can be used in combination with the running basis algorithm (9)–(8). Unfortunately, in both cases the derivations were based on the assumption that the estimated frequencies are unknown but constant, leading to frequency tracking algorithms that simply do not work unless some heuristic modifications are introduced.

Adopting the instantaneous measure of fit $J(t,\boldsymbol{\omega})=|\varepsilon(t)|^2$ and assuming that

$$\boldsymbol{\psi}_i(t)=\boldsymbol{\varphi}(t)\mathrm{e}^{\mathrm{j}\omega_i t}. \tag{23}$$

Bakkoury et al. arrive at the following gradient search procedure:

$$\widehat{\omega}_i(t+1)=\widehat{\omega}_i(t)-\mu J_i'(\widehat{\boldsymbol{\omega}}(t)), \tag{24}$$

where

$$J_i'(\widehat{\boldsymbol{\omega}}(t))=\left.\frac{\partial J(t,\boldsymbol{\omega})}{\partial\omega_i}\right|_{\boldsymbol{\omega}=\widehat{\boldsymbol{\omega}}(t)} =2\,\mathrm{Re}\{jt\varepsilon(t)\mathrm{e}^{-\mathrm{j}\widehat{\omega}_i(t)t}\boldsymbol{\varphi}^{\mathrm{H}}(t)\widehat{\boldsymbol{\alpha}}_i^*(t-1)\} \tag{25}$$

and $\mu>0$ denotes a small adaptation gain (stepsize). The problem is caused by the fact that relationship (23) implies that the estimated frequencies ω_i are time-invariant. Therefore it leads to the algorithm capable of estimating frequencies that are unknown but *constant*. Note that the gradient, evaluated according to (25), increases with time, which is understandable assuming that the seeked frequencies are time-invariant, but is completely unreasonable if they are time varying.

The same assumption was made in the earlier paper of Tsatsanis and Giannakis (1996), which describes the gradient search frequency tracker based on the averaged measure of fit $\bar{J}(t,\boldsymbol{\omega}) = 1/N\sum_{n=t-N+1}^{t} J(n,\boldsymbol{\omega})$. The authors, probably motivated by the negative simulation evidence, suggested elimination of the factor t in (25) ('so that the gradient does not increase indefinitely with time') and periodic resetting of the phase $\widehat{\omega}_i(t)t$ using $\bmod 2\pi$ ('to avoid overflow and numerical problems'). Without these two heuristic modifications the algorithm proposed by Tsatsanis and Giannakis (1996) does not work satisfactorily.

The computational complexity of both algorithms mentioned above is equal to $2n^2k^2 + 7nk + k$.

5. Computer simulations

A large number of simulation experiments were performed to check the initial convergence and frequency tracking/matching capabilities of algorithms discussed in Sections 3 and 4. The results presented below were obtained for a hypothetical time-varying communication channel with two impulse response coefficients $\theta_1(t)$ and $\theta_2(t)$ ($n=2$), each of which was modeled as a linear combination of two complex exponentials ($k=2$). The weighting coefficients in (2) had constant values $\boldsymbol{\alpha}=[a_{11}, a_{12}, a_{21}, a_{22}]^{\mathrm{T}}=[2, 0.5j, -j, 1.5]^{\mathrm{T}}$. The input signal was the white 4-QAM sequence ($u(t)=\pm 1 \pm j$, $\sigma_u^2=2$) and the noise was complex Gaussian with variance $\sigma_v^2=0.2$.

Four frequency-adaptive algorithms were compared: the global search algorithm (AEWBF), given by (18), the parallel-form (P-AEWBF) and cascade-form (C-AEWBF) variants of the global search algorithm, given by (19), and the algorithm proposed by Tsatsanis and Giannakis (T–G). Since the algorithm proposed by Bakkoury et al. (2000) is not capable of tracking time-varying frequencies, it was excluded from the comparison.

Since all compared algorithms use EWLS in their parameter estimation loops, they were equipped with the same forgetting constants $\lambda = \lambda_1 = \lambda_2 = 0.99$. In agreement with (22), the forgetting constants which control the frequency tracking loops in the AEWBF, P-AEWBF and C-AEWBF algorithms, were set to $\gamma = \gamma_1 = \gamma_2 = 0.98$. To make the comparison fair, the analogous tuning variables of the T–G algorithm (μ and N) were chosen so as to guarantee that the compared algorithms behave similarly under time-invariant conditions. All design parameters and the corresponding steady state tracking variances are shown in Table 2. Note that for the selected tuning variables the mean square prediction errors of the AEWBF and P-AEWBF algorithms are slightly smaller, and the mean square frequency estimation errors are slightly larger, than the corresponding quantities measured for the T–G algorithm (we were unable to find settings yielding exactly the same prediction and frequency tracking performance of the compared algorithms). Note also that the C-AEWBF algorithm yields larger steady state errors than the P-AEWBF algorithm, which is a consequence of specific error propagation properties of the cascade scheme, discussed in Section 2.3.

Table 2
Design parameters of the compared algorithms

Algorithm	Parameter tracking	Frequency tracking	σ_ε^2	$\bar{\sigma}_\omega^2$
AEWBF	$\lambda = 0.99$	$\gamma = 0.98$	0.206	3.23×10^{-8}
P-AEWBF	$\lambda = 0.99$	$\gamma = 0.98$	0.207	3.46×10^{-8}
C-AEWBF	$\lambda = 0.99$	$\gamma = 0.98$	0.255	2.13×10^{-6}
T–G	$\lambda = 0.99$	$\mu = 2.2 \times 10^{-5}$ $N = 20$	0.218	2.32×10^{-8}

The last two columns show the steady state values of the prediction error variance σ_ε^2 and the average frequency estimation error variance $\bar{\sigma}_\omega^2 = (\sigma_{\omega_1}^2 + \sigma_{\omega_2}^2)/2$ observed under time-invariant conditions.

One of the most important aspects of the problem of frequency estimation/tracking is correct frequency matching. The difficulty is caused by the fact that the minimized cost function has one global minimum, corresponding to the correct frequency assignment, and a large number of local minima, corresponding to incorrect or partially correct assignments. By partially correct assignments we mean all situations where two or more estimates $\widehat{\omega}_i(t)$ lock onto the same frequency, while some other frequencies are not matched at all. Basically, the incorrect frequency matching can occur in three situations: (i) in the initial convergence phase, if the starting values of frequency estimates are far from the true frequency values; (ii) in the 'steady state' tracking phase, when some of the frequencies become too closely spaced; (iii) after a sudden frequency change—which, from the qualitative viewpoint, is similar to (i) above.

In all initial convergence tests the frequencies ω_1 and ω_2 were kept constant. To avoid the initial convergence 'deadlock', which occurs when identical initial conditions are adopted for all blocks in the parallel estimation scheme, the starting values of the frequency estimates $\widehat{\omega}_1(0)$ and $\widehat{\omega}_2(0)$ were set to two different numbers close to zero. The typical results are depicted in Fig. 3. For all simulation runs, corresponding to different realizations of the input and noise sequences, the frequency estimates yielded by the decomposed algorithms P-AEWBF and C-AEWBF, locked onto the true values. The global search algorithm AEWBF properly matched the basis frequencies in a majority of cases (more than 80%). In contrast with this, the gradient search algorithm of T–G failed to find the proper frequency values in *all* initial convergence tests.

It should be stressed that none of the algorithms guarantees global convergence of the frequency estimates. In the presence of large estimation errors, e.g. due to frequency jumps or adverse initial frequency assignment, all schemes

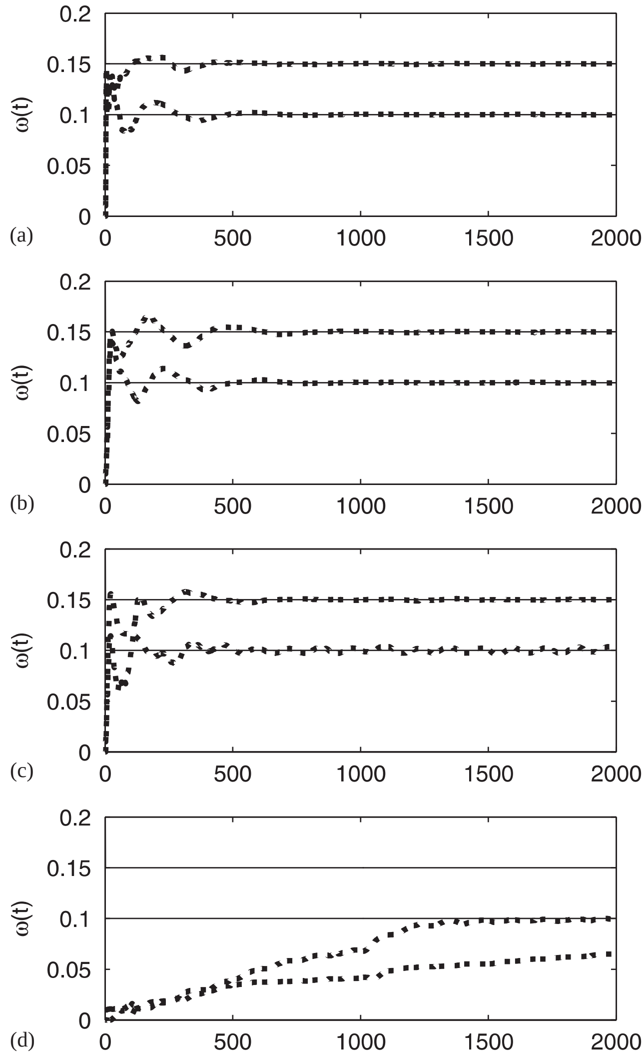

Fig. 3. True basis function frequencies (solid lines) and their estimates (dotted lines) obtained in the initial convergence period using the proposed AEWBF algorithm (a) and its decomposed parallel-form (b) and cascade-form (c) variants. Plot (d) shows the results yielded by the gradient search algorithm proposed by Tsatsanis and Giannakis (1996).

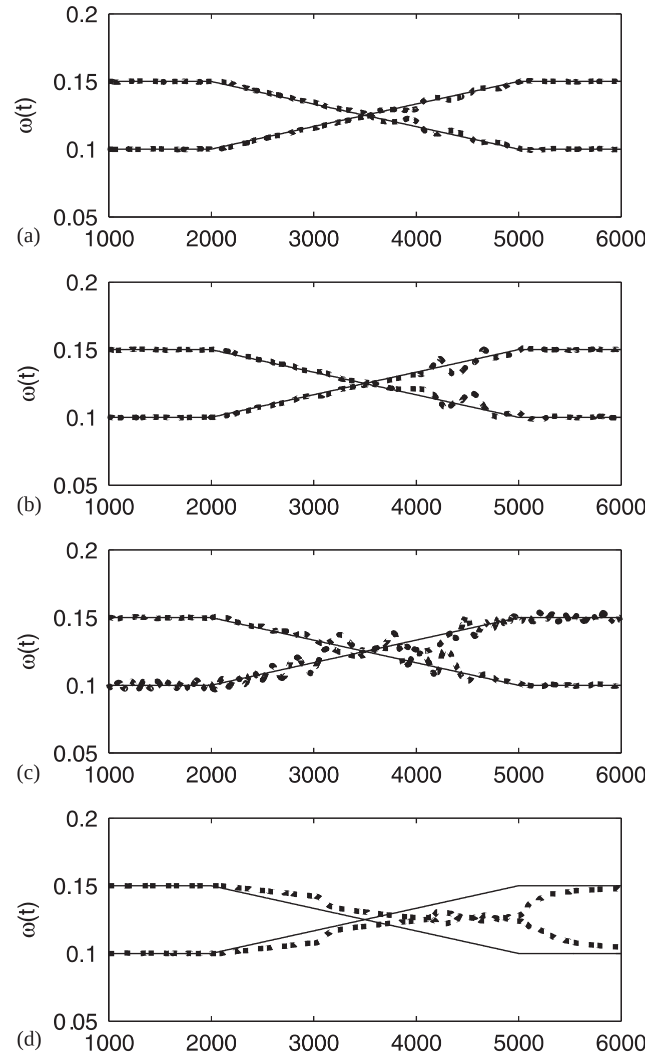

Fig. 4. Instantaneous basis function frequencies (solid lines) and their estimates (dotted lines) obtained using the proposed AEWBF algorithm (a) and its decomposed parallel-form (b) and cascade-form (c) variants. Plot (d) shows the results yielded by the gradient search algorithm proposed by Tsatsanis and Giannakis (1996).

have problems with correct frequency matching. Therefore, the results summarized above should be regarded as an indication of robustness of the compared algorithms to large errors, and not as a convergence statement. In the signal identification case an interesting robustness study (for a single cisoid) was provided by Händel, Tichavský, and Savaresi (1998). Finally, we note that the initial convergence problems can be almost completely avoided if the initialization technique described by Niedźwiecki and Kaczmarek (2004a) is used.

To check the steady state frequency tracking/matching capabilities of the compared algorithms, linear changes in frequencies were enforced after the initial convergence period was over (to guarantee correct frequency matching in the startup phase, the initial frequency estimates were set to the true values). The trajectories of $\omega_1(t)$ and $\omega_2(t)$ intersected in the middle of the analysis interval. One of our main concerns was behavior of the tracking algorithms after reaching the crossover point.

The results obtained for the proposed algorithms (AEWBF, P-AEWBF and C-AEWBF) were satisfactory, both in terms of frequency matching (see Fig. 4) and in terms of parameter tracking (see Fig. 5)—after passing the crossover point both frequency estimates $\widehat{\omega}_1(t)$ and $\widehat{\omega}_2(t)$ usually followed the correct values $\omega_1(t)$ and $\omega_2(t)$, respectively. The AEWBF algorithm consistently showed the best performance. This was hardly a surprise, since the global search algorithm (18) incorporates cross-coupling between the estimates of different frequency components, while its decomposed variants use the local search strategy. As expected (see remarks at the end of Section 2.3), the parallel-form algorithm P-AEWBF yielded better results than the

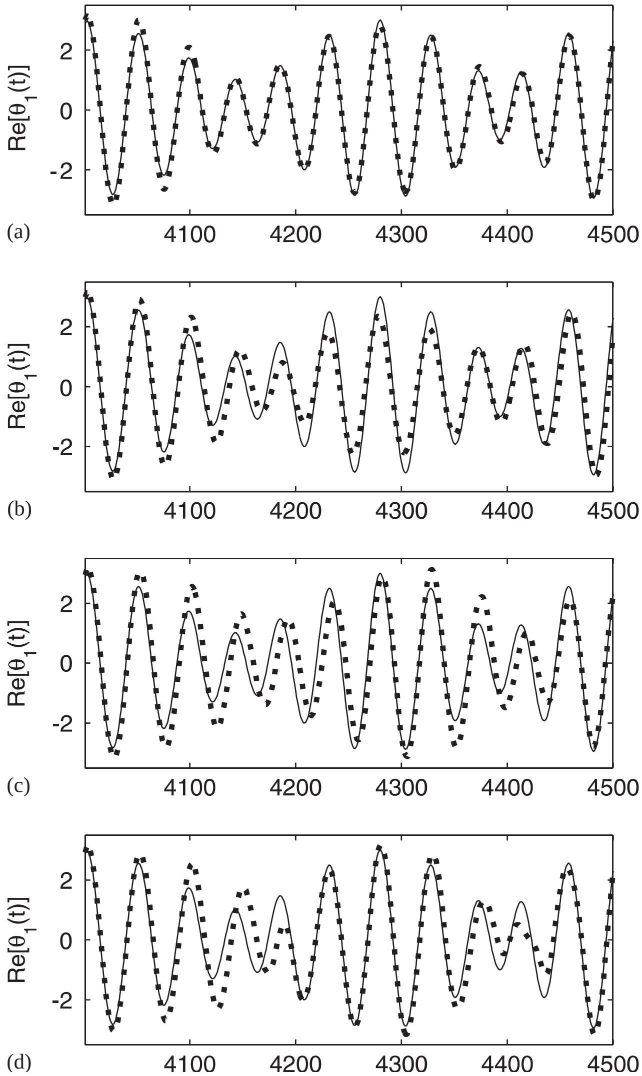

Fig. 5. Real part of the true system parameter $\theta_1(t)$ (solid lines) and its estimates (dotted lines) obtained using the proposed AEWBF algorithm (a) and its decomposed parallel-form (b) and cascade-form (c) variants. Plot (d) shows the results yielded by the gradient search algorithm proposed by Tsatsanis and Giannakis (1996).

Table 3
Estimates of the prediction error variance and frequency estimation variance obtained for 50 simulation runs

Algorithm	σ_ε^2	$\bar{\sigma}_\omega^2$
AEWBF	0.318	1.99×10^{-6}
P-AEWBF	0.760	5.55×10^{-6}
C-AEWBF	1.039	2.07×10^{-5}
T–G	1.164	1.18×10^{-4}

cascade-form algorithm C-AEWBF. The algorithm of T–G performed worse than any of the proposed algorithms.

Table 3 shows the average values of the frequency estimation error variance $\bar{\sigma}_\omega^2 = (\sigma_{\omega_1}^2 + \sigma_{\omega_2}^2)/2$ and the prediction error variance σ_ε^2, observed for $t \in [2001, 5000]$ in 50 simulation runs. Note that the quantitative results, gathered in Table 3, fully confirm our qualitative ratings of the compared algorithms.

6. Conclusion

The problem of identification/tracking of quasi-periodically varying complex systems was considered.

In the first part of the paper, the exponentially weighted basis function approach was used to solve the estimation/tracking problem for almost periodically varying systems, i.e. systems characterized by known constant frequencies. Adoption of a special system parameterization allowed for a frequency decomposition of the basic algorithm. The resulting parallel-form and cascade-form algorithms have a highly modular structure and reduced computational complexity. They are made up of identical functional blocks which, based on their first-order tracking characteristics established in the paper, can be considered generalized notch filters. Each block (subalgorithm) takes care of a particular frequency component (subsystem) of the identified system. Such frequency decomposition is a new concept in identification of time-varying systems.

In the second part of the paper the frequency-adaptive versions of all algorithms were derived using the recursive prediction error approach. The best results, both in terms of parameter tracking and frequency matching, are obtained for the frequency-adaptive version of the basic algorithm. Out of the two decomposed versions of this basic solution, the parallel-form algorithm performs better than the cascade-form algorithm. All proposed algorithms show much better performance than the gradient search algorithms described in the literature.

At the conceptual level the paper establishes interesting connections between identification of quasi-periodically varying systems and filtering theory.

Appendix

It is straightforward to check, that the recursive formula (13) can be expressed in the following explicit form

$$\widehat{\boldsymbol{\beta}}_i(t) = (\mathbf{R}_i^*(t))^{-1}\mathbf{p}_i(t),$$

where

$$\mathbf{R}_i(t) = \sum_{s=0}^{t-1} \lambda_i^s \boldsymbol{\varphi}(t-s)\boldsymbol{\varphi}^{\mathrm{H}}(t-s),$$

$$\mathbf{p}_i(t) = \sum_{s=0}^{t-1} \lambda_i^s \rho_i^{s+1} y(t-s)\boldsymbol{\varphi}^*(t-s).$$

Hence, in the steady state it holds

$$\widehat{\boldsymbol{\beta}}_i(t) = \left(\sum_{s=0}^{\infty} \lambda_i^s \boldsymbol{\varphi}^*(t-s)\boldsymbol{\varphi}^{\mathrm{T}}(t-s) \right)^{-1} \times \left(\sum_{s=0}^{\infty} \lambda_i^s \rho_i^{s+1} y(t-s)\boldsymbol{\varphi}^*(t-s) \right).$$

One can show that if the process $u(t)$ is stationary and ergodic then

$$\lim_{\lambda_i \mapsto 1} (1-\lambda_i) \sum_{s=0}^{\infty} \lambda_i^s \boldsymbol{\varphi}^*(t-s)\boldsymbol{\varphi}^{\mathrm{T}}(t-s) = \boldsymbol{\Phi},$$

where $\boldsymbol{\Phi} = \mathrm{E}[\boldsymbol{\varphi}^*(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)]$. This leads to the following approximation

$$\widehat{\boldsymbol{\beta}}_i(t) \cong (1-\lambda_i)\boldsymbol{\Phi}^{-1} \sum_{s=0}^{\infty} \lambda_i^s \rho_i^{s+1} y(t-s)\boldsymbol{\varphi}^*(t-s).$$

Combining the above relationship with $y(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}(t) + v(t)$ one obtains

$$\widehat{\boldsymbol{\beta}}_i(t) \cong (1-\lambda_i)\boldsymbol{\Phi}^{-1} \sum_{s=0}^{\infty} \lambda_i^s \rho_i^{s+1} \boldsymbol{\varphi}^*(t-s)\boldsymbol{\varphi}^{\mathrm{T}}(t-s)\boldsymbol{\theta}(t-s) + (1-\lambda_i)\boldsymbol{\Phi}^{-1} \sum_{s=0}^{\infty} \lambda_i^s \rho_i^{s+1} \boldsymbol{\varphi}^*(t-s)v(t-s).$$

Since the noise $v(t)$ is uncorrelated with $\boldsymbol{\varphi}(t)$, one arrives at

$$\mathrm{E}[\widehat{\boldsymbol{\beta}}_i(t)] \cong (1-\lambda_i) \sum_{s=0}^{\infty} \lambda_i^s \rho_i^{s+1} \boldsymbol{\theta}(t-s) = \frac{(1-\lambda_i)\rho_i}{1-\lambda_i \rho_i q^{-1}}\, \boldsymbol{\theta}(t).$$

Finally, one obtains

$$\mathrm{E}[\widehat{\boldsymbol{\theta}}_i(t)] = \rho_i^* \mathrm{E}[\widehat{\boldsymbol{\beta}}_i(t)] \cong \frac{1-\lambda_i}{1-\lambda_i \rho_i q^{-1}}\, \boldsymbol{\theta}(t) = T_i(q^{-1})\boldsymbol{\theta}(t).$$

References

Aiken, R. T. (1967). Communication over the discrete-path fading channel. *IEEE Transactions on Information Theory*, *IT-13*, 346–347.

Bakkoury, J., Roviras, D., Ghogho, M., & Castanie, F. (2000). Adaptive MLSE receiver over rapidly fading channels. *Signal Processing*, *80*, 1347–1360.

Dragošević, M. V., & Stanković, S. S. (1995a). An adaptive notch filter with improved tracking properties. *IEEE Transactions on Signal Processing*, *43*, 2068–2077.

Dragošević, M. V., & Stanković, S. S. (1995b). Fully adaptive constrained notch filter for tracking multiple frequencies. *Electronics Letters*, *31*, 1215–1217.

Giannakis, G. B., & Tepedelenlioğlu, C. (1998). Basis expansion models and diversity techniques for blind identification and equalization of time-varying channels. *Proceedings of the IEEE*, *86*, 1969–1986.

Händel, P., Tichavský, P., & Savaresi, S. M. (1998). Large error recovery for a class of frequency tracking algorithms. *International Journal on Adaptive Control and Signal Processing*, *12*, 417–436.

Hilands, T. W., & Thomopoulos, S. C. A. (1997). Nonlinear filtering method for harmonic retrieval and model order selection in Gaussian and non-Gaussian noise. *IEEE Transactions on Signal Processing*, *45*, 982–995.

Li, L. M., & Milstein, L. B. (1983). Rejection of pulsed CW interference in PN spread-spectrum systems using complex adaptive filters. *IEEE Transactions on Communications*, *CM-31*, 10–20.

Niedźwiecki, M. (1990). Recursive functional series modeling approach to identification of time-varying plants—more bad news than good?. *IEEE Transactions on Automatic Control*, *AC-35*, 610–616.

Niedźwiecki, M. (2000). *Identification of time-varying processes*. New York: Wiley.

Niedźwiecki, M., & Kaczmarek, P. (2004a). New approach to identification of quasi-periodically varying systems. *10th IEEE international conference on methods and models in automation and robotics*, Międzyzdroje, Poland, pp. 1045–1050.

Niedźwiecki, M., & Kaczmarek, P. (2004b). Generalized adaptive notch filters for identification of real quasi-periodically varying systems. *12th European signal processing conference* (*EUSIPCO 2004*), Vienna, Austria, pp. 1915–1918.

Niedźwiecki, M., & Kaczmarek, P. (2005). Tracking analysis of a generalized adaptive notch filter. Technical report ETI-1/2005, Gdańsk University of Technology. *IEEE Transactions on Signal Processing*, to appear.

Niedźwiecki, M., & Kłaput, T. (2003). Fast algorithms for identification of periodically varying systems. *IEEE Transactions on Signal Processing*, *51*, 3270–3279.

Nishiyama, K. (2000). A nonlinear multiple complex sinusoidal estimator. *IEICE Transactions on Fundamentals*, *E83*, 497–506.

Padmanabham, M., & Tichavský, P. (1996). Hyperstable adaptive line enhancer for tracking of multiple noisy cisoids. *Proceedings of the eighth European signal processing conference* (*EUSIPCO-96*), Trieste, Italy, pp. 1271–1274.

Parker, P. J., & Anderson, B. D. O. (1990). Frequency tracking of nonsinusoidal periodic signals in noise. *Signal Processing*, *20*, 127–152.

Pei, S.-C., & Tseng, C.-C. (1994). Complex adaptive IIR notch filter algorithm and its applications. *IEEE Transactions on Circuits and Systems*, *41*, 158–163.

Söderström, T., & Stoica, P. (1988). *System identification*. Englewood Cliffs, NJ: Prentice-Hall.

Subba Rao, T. (1970). The fitting of non-stationary time-series models with time-dependent parameters. *Journal of the Royal Statistical Society, Series B*, *32*, 312–322.

Tichavský, P., & Händel, P. (1995). Two algorithms for adaptive retrieval of slowly time-varying multiple cisoids in noise. *IEEE Transactions on Signal Processing*, *43*, 1116–1127.

Tichavský, P., & Nehorai, A. (1997). Comparative study of four adaptive frequency trackers. *IEEE Transactions on Signal Processing*, *45*, 1473–1484.

Tsatsanis, M. K., & Giannakis, G. B. (1996). Modeling and equalization of rapidly fading channels. *International Journal on Adaptive Control and Signal Processing*, *10*, 159–176.

Maciej Niedźwiecki was born in Poznań, Poland in 1953. He received the M.Sc. and Ph.D. degrees from the Gdańsk University of Technology, Gdańsk, Poland, and the Dr. Hab. (D.Sc.) degree from the Technical University of Warsaw, Warsaw, Poland, in 1977, 1981 and 1991, respectively.

He spent three years as a Research Fellow with the Department of Systems Engineering, Australian National University, 1986–1989. During 1990–1993 he served as a Vice Chairman of Technical Committee on Theory of the International Federation of Automatic Control (IFAC). He is the author of the book *Identification of Time-varying Processes* (Wiley, 2000).

He works as a Professor and Head of the Department of Automatic Control, Faculty of Electronics, Telecommunications and Computer Science, Gdańsk University of Technology. His main areas of research interests include system identification, signal processing and adaptive systems.

Piotr Kaczmarek received the M.Sc. degree in automatic control from the Gdańsk University of Technology, Gdańsk, Poland, in 2000. He has also graduated from the European Master Degree Course in Control and Management of Lean Manufacturing in Network Systems conducted in co-operation between Gdańsk University of Technology (Poland), Catholic University of Louvain (Belgium), and University of Karlsruhe (Germany).

Since 1998 he has been involved in many projects for the biggest Polish air conditioning systems manufacturers. Now he is a Ph.D. student at the Gdańsk University of Technology. His interests include system identification and adaptive filtering as well as optimization of production techniques.