Signal Processing 129 (2016) 38–47

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

A self-optimization mechanism for generalized adaptive notch smoother

Michał Meller

Department of Automatic Control, Gdańsk University of Technology, Faculty of Electronics, Telecommunications and Computer Science, Narutowicza 11/12, 80-233 Gdańsk, Poland

ARTICLE INFO

Article history:
Received 25 March 2016
Received in revised form
18 May 2016
Accepted 27 May 2016
Available online 2 June 2016

Keywords:
Generalized adaptive notch smoothing
System identification
Automatic tuning

ABSTRACT

Tracking of nonstationary narrowband signals is often accomplished using algorithms called adaptive notch filters (ANFs). Generalized adaptive notch smoothers (GANSs) extend the concepts of adaptive notch filtering in two directions. Firstly, they are designed to estimate coefficients of nonstationary quasi-periodic systems, rather than signals. Secondly, they employ noncausal processing, which greatly improves their accuracy and can be applied whenever additional delay can be tolerated. The paper develops a novel performance assessment mechanism for GANS. It allows one to evaluate tracking accuracy of the smoother without prior knowledge of the true values of the system's frequency or coefficients. The extension can be employed to build a parallel bank of filters, which automatically chooses the one which is best matched to unknown and possibly time-varying tracking conditions.

1. Introduction

A broad spectrum of applications involves estimation of linear-in-parameters' systems whose coefficients are time varying, i.e. systems governed by

$$y(t) = \boldsymbol{\phi}^{\mathrm{T}}(t)\boldsymbol{\theta}(t) + v(t) \tag{1}$$

where $t = \dots, -1, 0, 1, \dots$ denotes discrete, dimensionless time, $y(t)$ is the system output, $v(t)$ is a wideband measurement noise, $\boldsymbol{\phi}(t) = [\phi_1(t)\ \phi_2(t)\ \dots\ \phi_n(t)]^{\mathrm{T}}$ is the regression vector and $\boldsymbol{\theta}(t) = [\theta_1(t)\ \theta_2(t)\ \dots\ \theta_n(t)]^{\mathrm{T}}$ denotes the vector of system coefficients.

Depending on the underlying source of nonstationarity, the system of interest may be loosely classified as being either a 'slowly' or a 'rapidly' time varying one [1]. Obviously, the border between the two cases is rather blurry, to say the least. Quite often the answer to the question whether or not one is facing the rapid variation case depends on the required estimation accuracy. The case of slow variations can usually be solved using standard estimation tools, such as the recursive least squares (RLS) or the least mean squares (LMS) algorithms, among others [2]. The second class of problems arises when these tools can no longer deliver estimates of sufficient accuracy. Solutions to this class of problems typically rely on additional knowledge, in the form of a model of the system coefficients' behavior.

The case of particular interest to this paper is when the coefficient vector $\boldsymbol{\theta}(t)$ varies in an approximately complex exponential manner

$$\boldsymbol{\theta}(t) = \boldsymbol{\beta}(t)e^{j\sum_{\tau=1}^{t}\omega(\tau)}, \tag{2}$$

where $\boldsymbol{\beta}(t) = [\beta_1(t)\ \beta_2(t)\ \dots\ \beta_n(t)]^{\mathrm{T}}$, denotes slowly time-varying complex valued 'amplitude' vector and $\omega(t)$ is the slowly time-varying (real-valued) instantaneous frequency. Note that the system (1)–(2) can be more accurately classified as quasi-periodic – since both $\boldsymbol{\beta}(t)$ and $\omega(t)$ are time-varying quantities, the behavior of $\boldsymbol{\theta}(t)$ can be approximated using stationary complex sinusoids (cisoids) only in a short time frame.

The importance of quasi-periodic case stems from the fact that it is encountered in numerous RF applications where Doppler effect takes place [3–7]. In these applications $y(t)$ typically represents the complex-valued baseband received signal, $\boldsymbol{\theta}(t)$ is made up of past samples of the transmitted waveform, while $\boldsymbol{\beta}(t)$ and $\omega(t)$ are the scatterer's 'impulse response' and Doppler frequency, respectively. Note that, since spatial relationships between the transmitter, the reflectors, and the receiver change due to their relative movement, $\boldsymbol{\beta}(t)$ and $\omega(t)$ are indeed time varying quantities. On the other hand, the time frame in which these changes become significant spans tens or even hundreds of periods of their movement-induced complex sinusoidal Doppler terms, which means that behavior of system coefficients is well modeled by Eq. (2).

The problem of estimating coefficients of the system (1)–(2) is sometimes called generalized adaptive notch filtering problem, and the algorithm designed for this task is referred to as

E-mail address: michal.meller@eti.pg.gda.pl

http://dx.doi.org/10.1016/j.sigpro.2016.05.028
0165-1684/

generalized adaptive notch filters (GANFs). To explain why GANFs can be regarded as extension of classical adaptive notch filters (ANFs) [8–12], consider the case when $\boldsymbol{\phi}(t) = \phi(t) \equiv 1$. Under such restriction the task of tracking coefficients of the system (1)–(2) is equivalent to tracking a nonstationary complex sinusoid

$$s(t) = \beta(t)e^{j\sum_{\tau=1}^{t}\omega(t)} \tag{3}$$

embedded in wideband noise

$$y(t) = s(t) + v(t), \tag{4}$$

which is the backbone of the ANF problem. Practical applications which fall under the scope of (3) include, among others, filtering power signal from electrocardiogram (ECG) recordings [13,14], tracking harmonic currents in power applications [15–17], fault detection [18] or active control of narrowband acoustic noise [19–21].

Both ANFs and GANFs are causal algorithms – to work out their estimates, they employ only data from the past and the current moment in time. However, in many applications one is allowed to make use of future data as well. Such a situation can occur, e.g. when the data was prerecorded in advance or when additional processing delay can be tolerated. In cases like these one can employ noncausal extensions of ANFs and GANFs, called notch smoothers (ANSs and GANSs, respectively). Such algorithms can offer substantially better performance than their causal counterparts, both in terms of frequency and system (signal) tracking accuracy – see [22,23] for more details.

These benefits are, unfortunately, not free – GANSs, being more complex than GANFs, require more skill from their user. The paper addresses these difficulties and proposes two useful improvements. First, an on-line performance assessment mechanism is developed, which allows one to evaluate frequency tracking accuracy of the GANS without prior knowledge of the true frequency trajectory. This design is far from trivial, because smoothers, being noncausal devices, can easily create an illusion of good performance. Second, on the basis of the proposed assessment mechanism, an automatic optimization mechanism is constructed. The final solution employs the parallel approach – it runs a bank of smoothers predesigned to match different tracking conditions and selects the one with (locally) best accuracy.

The paper is organized as follows. Section 2 revises the GANS algorithm of interest in this study. Section 3 develops and analyzes an automatic performance assessment mechanism. Section 4 discusses implementation-related issues. Section 5 gathers all the partial results derived in Sections 2–4 and proposes the self-optimizing variant of the GANS. Section 6 presents simulation results which validate the analytical part of the paper. Section 7 concludes.

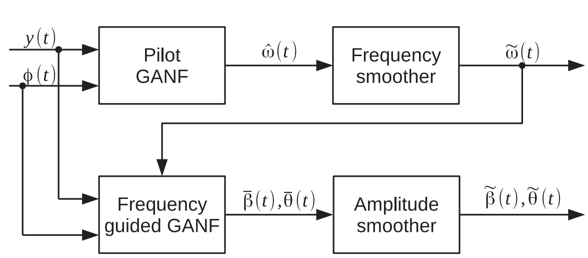

Fig. 1. GANS algorithm and signals appearing at outputs of consecutive stages of processing.

2. Generalized adaptive notch smoothing revised

2.1. Basic generalized adaptive notch smoother

Suppose that the sequence of regression vectors $\boldsymbol{\phi}(t)$ is a wide-sense stationary, persistently exciting random process with known correlation matrix $\boldsymbol{\Phi} = \mathrm{E}[\boldsymbol{\phi}^*(t)\boldsymbol{\phi}^{\mathrm{T}}(t)]$. The GANS proposed in [24] is a multi-step algorithm which consists of four parts (see Fig. 1 for an outline of the algorithm and signals appearing at output of different stages). First, the so-called pilot GANF is employed

$$\begin{aligned}
\hat{f}(t) &= e^{j[\hat{\omega}(t-1)+\hat{\alpha}(t-1)]}\hat{f}(t-1)\\
\epsilon(t) &= y(t) - \boldsymbol{\phi}^{\mathrm{T}}(t)\hat{f}(t)\hat{\boldsymbol{\beta}}(t-1)\\
\hat{\boldsymbol{\beta}}(t) &= \hat{\boldsymbol{\beta}}(t-1) + \mu\boldsymbol{\Phi}^{-1}\boldsymbol{\phi}^*(t)\hat{f}^*(t)\epsilon(t)\\
\delta(t) &= \mathrm{Im}\,\frac{\epsilon^*(t)\boldsymbol{\phi}^{\mathrm{T}}(t)\hat{f}(t)\hat{\boldsymbol{\beta}}(t-1)}{\hat{\boldsymbol{\beta}}^{\mathrm{H}}(t-1)\boldsymbol{\Phi}\hat{\boldsymbol{\beta}}(t-1)}\\
\hat{\alpha}(t) &= \hat{\alpha}(t-1) - \gamma_\alpha\delta(t)\\
\hat{\omega}(t) &= \hat{\omega}(t-1) + \hat{\alpha}(t-1) - \gamma_\omega\delta(t)\\
\hat{\boldsymbol{\theta}}(t) &= \hat{\boldsymbol{\beta}}(t)\hat{f}(t)
\end{aligned} \tag{5}$$

where $*$ denotes complex conjugation, T and H stand for transposition and conjugate transposition, respectively, $\hat{f}(t)$ is a phase term, $\epsilon(t)$ is a prediction error; $\hat{\omega}(t)$ and $\hat{\alpha}(t)$ denote the estimates of instantaneous frequency and frequency rate $[\alpha(t) = \omega(t+1) - \omega(t)]$ respectively. The parameters $\mu > 0$, $\gamma_\omega > 0$, $\gamma_\alpha > 0$, $\gamma_\alpha \ll \gamma_\omega \ll \mu$ are small adaptation gains which govern rates of amplitude, frequency, and frequency rate adaptation, respectively.

Even though the pilot GANF includes tracking of system coefficients, its actual purpose is to work out frequency estimates, $\hat{\omega}(t)$. These estimates are, in fact, preliminary and undergo further processing to deliver more accurate ones.

This improvement is achieved by making use of available 'future' information. Smoothing of frequency estimates is performed using the following cascade of simple linear filters, where the second one runs in reversed time

$$\begin{aligned}
\bar{\omega}(t) &= -c_1\bar{\omega}(t-1) + b_1\hat{\omega}(t-1)\\
\tilde{\omega}(t) &= -f_1\tilde{\omega}(t+1) - f_2\tilde{\omega}(t+2) - f_3\tilde{\omega}(t+3) + \gamma_\alpha\bar{\omega}(t+1).
\end{aligned} \tag{6}$$

The coefficients appearing in the above two recursive equations depend on adaptation gains of the pilot filter in the following way:

$$\begin{aligned}
b_1 &= \gamma_\alpha/\gamma_\omega\\
c_1 &= (\gamma_\alpha - \gamma_\omega)/\gamma_\omega\\
f_1 &= \mu + \gamma_\omega + \gamma_\alpha - 3\\
f_2 &= 3 - 2\mu - \gamma_\omega\\
f_3 &= \mu - 1
\end{aligned} \tag{7}$$

The final two steps are amplitude filtering, performed using the so-called frequency guided GANF,

$$\begin{aligned}
\bar{f}(t) &= e^{j\tilde{\omega}(t)}\bar{f}(t-1)\\
\bar{\epsilon}(t) &= y(t) - \boldsymbol{\phi}^{\mathrm{T}}(t)\bar{f}(t)\bar{\boldsymbol{\beta}}(t)\\
\bar{\boldsymbol{\beta}}(t) &= \bar{\boldsymbol{\beta}}(t-1) + \mu_a\boldsymbol{\Phi}^{-1}\boldsymbol{\phi}^*(t)\bar{f}^*(t)\bar{\epsilon}(t)\\
\bar{\boldsymbol{\theta}}(t) &= \bar{\boldsymbol{\beta}}(t)\bar{f}(t),
\end{aligned} \tag{8}$$

followed with amplitude smoothing

$$\begin{aligned}
\tilde{\boldsymbol{\beta}}(t) &= (1-\mu_a)\tilde{\boldsymbol{\beta}}(t+1) + \mu_a\bar{\boldsymbol{\beta}}(t)\\
\tilde{\boldsymbol{\theta}}(t) &= \tilde{\boldsymbol{\beta}}(t)\bar{f}(t).
\end{aligned} \tag{9}$$