14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

SELF-OPTIMIZING ADAPTIVE NOTCH FILTERS – COMPARISON OF THREE OPTIMIZATION STRATEGIES

Maciej Niedźwiecki, Piotr Kaczmarek

Faculty of Electronics, Telecommunications and Computer Science
Department of Automatic Control, Gdańsk University of Technology
Narutowicza 11/12, Gdańsk , Poland
Tel: + 48 58 3472519, Fax: +48 58 3416132
e-mail: maciekn@eti.pg.gda.pl, piokacz@proterians.net.pl

ABSTRACT

The paper provides comparison of three different approaches to on-line tuning of adaptive notch filters (ANF) – the algorithms used for extraction/elimination of complex sinusoidal signals (cisoids) buried in noise. Tuning is needed to adjust adaptation gains, which control tracking performance of ANF algorithms, to the unknown and/or time time-varying rate of signal nonstationarity. The first of the compared approaches incorporates sequential optimization of adaptation gains. The second solution is based on the concept of parallel estimation. It is shown that the best results are achieved when both approaches are combined in a judicious way. Such joint sequential/parallel optimization preserves advantages of both treatments: adaptiveness (sequential approach) and robustness to abrupt changes (parallel approach).

1. INTRODUCTION

Consider the problem of extraction or elimination of complex multifrequency signals buried in noise

$$y(t)=\sum_{i=1}^{k} s_i(t)+v(t), \quad s_i(t)=a_i(t)e^{j\sum_{\tau=1}^{t}\omega_i(\tau)} \tag{1}$$

where $t=1,2,\ldots$ denotes the normalized discrete time, $y(t)$ denotes the measured (noisy) signal and $v(t)$ is a complex white noise of variance σ_v^2. We will assume that $\mathrm{E}[v_R^2(t)] = \mathrm{E}[v_I^2(t)] = \sigma_v^2/2$, $\mathrm{E}[v_R(t)v_I(\tau)] = 0$, $\forall\, t,\tau$, where $v_R(t) = \mathrm{Re}[v(t)]$, $v_I(t) = \mathrm{Im}[v(t)]$. Since both the complex amplitudes $a_i(t)$ and the angular frequencies $\omega_i(t)$ in (1) are assumed to vary slowly with time, the extracted/cancelled signal $s(t)=\sum_{i=1}^{k} s_i(t)$ changes in a periodic-like, but not exactly periodic manner.

From several available adaptive notch filtering algorithms presented in [1], [2] and [3] we will pick a relatively new solution described and analyzed in [3]. This is by no means a critical choice as *all* algorithms referred to above have approximately the same local tracking properties. In particular, they all achieve the posterior Cramér-Rao frequency tracking bound for random walk frequency changes (under Gaussian assumptions) - see [2], [6] for more details.

In the single frequency case ($k=1$) the normalized, steady state version of the ANF algorithm presented in [3] can be written down in the form

$$\begin{aligned}
\varepsilon(t) &= y(t)-e^{j\widehat{\omega}(t)}\widehat{s}(t-1)\\
\widehat{s}(t) &= e^{j\widehat{\omega}(t)}\widehat{s}(t-1)+\mu\varepsilon(t)\\
g(t) &= \mathrm{Im}\left[\frac{\varepsilon^*(t)e^{j\widehat{\omega}(t)}}{\widehat{s}^*(t-1)}\right]\\
\widehat{\omega}(t+1) &= \widehat{\omega}(t)-\gamma g(t)
\end{aligned} \tag{2}$$

Tracking properties of this algorithm are determined by two user-dependent tuning coefficients: the adaptation gain μ, $0<\mu\ll 1$, which controls the rate of amplitude adaptation, and another adaptation gain γ, $0<\gamma\ll 1$, which decides upon the rate of frequency adaptation.

The multiple frequency algorithm can be obtained by combining k single-frequency ANFs, given by (2), into an appropriately designed parallel structure - see Section 3.

Selection of adaptation gains μ and γ in (2) is an important practical problem. By increasing these gains one increases the tracking speed of an ANF algorithm, but decreases its noise rejection capability. Decreasing adaptation gains has opposite consequences. The optimal choice of μ and γ, i.e. the choice that minimizes the mean-squared tracking errors, is therefore always a compromise between the filter's speed and it's accuracy. The outcome depends on the rate of nonstationarity of the identified signal. Quite obviously, if the rate of signal nonstationarity is also time-dependent, for example if periods of fast amplitude/frequency changes are followed by the periods of their slow variation and *vice versa*, it may be very difficult, if at all possible, to find fixed values of μ and γ that would guarantee satisfactory performance of an adaptive notch filter under such heterogenous conditions.

A typical way of increasing tracking capabilities of adaptive filters is by means of automatic gain tuning – see [4] for an interesting overview of different approaches to this problem. Generally speaking such on-line gain optimization can be performed using either sequential or parallel estimation techniques. The first case uses a single tracking algorithm equipped with adjustable adaptation gains. The second case takes several algorithms with different gain settings, runs them in parallel and compares them according to their predictive abilities. We will show that the best results can be obtained if both approaches mentioned above are appropriately combined.

2. AUTOMATIC TUNING OF AN ANF ALGORITHM

2.1 Pre-optimization

Denote by $\Delta s(t) = \widehat{s}(t) - s(t)$ the signal estimation error and suppose that the estimated signal is governed by $s(t) = e^{j\omega(t)}s(t-1)$, which means that variation of the instantaneous frequency $\omega(t)$ is the only source of signal nonstationarity. Furthermore, assume that $\omega(t)$ evolves according to the random walk (RW) model

$$\omega(t) = \omega(t-1) + w(t)$$

where $\{w(t)\}$ is a white noise sequence of variance σ_w^2, independent of $\{v(t)\}$.
Using the approximating linear filtering (ALF) technique, proposed by Tichavský and Händel [1] the following variance expression was derived in [3] for the case described above

$$\mathrm{E}[|\Delta s(t)|^2] \cong \left[\frac{\gamma}{4\mu} + \frac{\mu}{2}\right]\sigma_v^2 + \frac{b^2}{2\mu\gamma}\,\sigma_w^2 \qquad (3)$$

where $b^2 = |s(t)|^2$.
Denote by μ_o and γ_o the values of μ and γ that minimize (3). It is easy to check that

$$\mu_o = \sqrt[4]{2\xi}, \quad \gamma_o = \sqrt{2\xi} \qquad (4)$$

where the scalar coefficient $\xi = b^2\sigma_w^2/\sigma_v^2$ – the product of the signal-to-noise ratio b^2/σ_v^2 and the variance of frequency changes σ_w^2 – can be regarded a measure of signal nonstationarity.
According to (4), the optimal value of γ is equal to the square of the optimal value of μ: $\gamma_o = \mu_o^2$. This suggests that setting $\gamma = \mu^2$ may be a good way of reducing the number of design degrees of freedom of an ANF algorithm from two (μ,γ) to one (μ). The resulting pre-optimized version of (2) can be written down in the form

$$\begin{aligned}
\varepsilon(t) &= y(t) - e^{j\widehat{\omega}(t)}\widehat{s}(t-1)\\
\widehat{s}(t) &= e^{j\widehat{\omega}(t)}\widehat{s}(t-1) + \mu\varepsilon(t)\\
g(t) &= \mathrm{Im}\left[\frac{\varepsilon^*(t)e^{j\widehat{\omega}(t)}}{\widehat{s}^*(t-1)}\right]\\
\widehat{\omega}(t+1) &= \widehat{\omega}(t) - \mu^2 g(t)
\end{aligned} \qquad (5)$$

2.2 Sequential optimization

The adaptation gain μ can be adjusted recursively, by minimizing the following local measure of fit, made up of exponentially weighted prediction errors

$$V(t,\mu) = \frac{1}{2}\sum_{\tau=1}^{t}\lambda^{t-\tau}|\varepsilon(\tau,\mu)|^2$$

The forgetting constant λ $(0 < \lambda < 1)$, which decides upon the effective averaging range, should be chosen so that $1-\lambda \ll \mu, \forall t$.
To evaluate the estimate $\widehat{\mu}(t) = \arg\min_\mu V(t,\mu)$ we will use the standard recursive prediction error (RPE) approach. According to Söderström and Stoica [7], the RPE algorithm can be expressed in the form

$$\widehat{\mu}(t) = \widehat{\mu}(t-1) - \left[V''(t,\widehat{\mu}(t-1))\right]^{-1}V'(t,\widehat{\mu}(t-1))$$

where

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

$$V''(t,\widehat{\mu}(t-1)) \cong \lambda V''(t-1,\widehat{\mu}(t-2)) + \left|\frac{\partial\varepsilon(t,\widehat{\mu}(t-1))}{\partial\mu}\right|^2$$

and all derivatives are taken with respect to μ. Denote

$$\zeta(t) = \frac{\partial\varepsilon(t,\widehat{\mu}(t-1))}{\partial\mu}, \quad \psi(t) = \frac{\partial\widehat{s}(t,\widehat{\mu}(t-1))}{\partial\mu}$$

$$\rho(t) = \frac{\partial g(t,\widehat{\mu}(t-1))}{\partial\mu}, \quad \chi(t) = \frac{\partial\widehat{\omega}(t,\widehat{\mu}(t-1))}{\partial\mu}$$

$$r(t) = V''(t,\widehat{\mu}(t-1))$$

Straightforward calculations lead to

$$\begin{aligned}
\varepsilon(t) &= y(t) - e^{j\widehat{\omega}(t)}\widehat{s}(t-1)\\
\zeta(t) &= -e^{j\widehat{\omega}(t)}[j\chi(t)\widehat{s}(t-1) + \psi(t-1)]\\
\psi(t) &= \varepsilon(t) - [1-\widehat{\mu}(t-1)]\zeta(t)\\
\rho(t) &= \mathrm{Im}\left\{\frac{e^{j\widehat{\omega}(t)}}{\widehat{s}^*(t-1)}\left[\zeta^*(t) + j\varepsilon^*(t)\chi(t) - \frac{\varepsilon^*(t)\psi(t-1)}{\widehat{s}^*(t-1)}\right]\right\}\\
r(t) &= \lambda r(t-1) + |\zeta(t)|^2\\
\widehat{\mu}(t) &= \left[\widehat{\mu}(t-1) - \frac{\mathrm{Re}[\varepsilon(t)\zeta^*(t)]}{r(t)}\right]_0^{\mu_{max}}\\
\widehat{s}(t) &= e^{j\widehat{\omega}(t)}\widehat{s}(t-1) + \widehat{\mu}(t)\varepsilon(t)\\
g(t) &= \mathrm{Im}\left[\frac{\varepsilon^*(t)e^{j\widehat{\omega}(t)}}{\widehat{s}^*(t-1)}\right]\\
\widehat{\omega}(t+1) &= \widehat{\omega}(t) - \widehat{\mu}^2(t)g(t)\\
\chi(t+1) &= \chi(t) - \widehat{\mu}(t)[2g(t) + \widehat{\mu}(t)\rho(t)]
\end{aligned} \qquad (6)$$

where

$$[x]_a^b = \begin{cases} a & \text{if} \quad x < a \\ x & \text{if} \quad a \le x \le b \\ b & \text{if} \quad x > b \end{cases}$$

Note that the algorithm was equipped with a "safety valve": when the calculated value of μ exceeds its upper limit, it is truncated to μ_{max}; similarly, μ is set to zero whenever the calculated value becomes negative.

2.3 Parallel optimization

The main idea behind the parallel optimization (or multiple-model) approach is to run simultaneously several adaptive filters, with fixed (different) adaptation gains, and choose at each time instant the most appropriate filter, namely the one that has shown the best performance in the "recent past". Following this line of thinking, consider L ANF algorithms of the form

$$\begin{aligned}
\varepsilon^{[l]}(t) &= y(t) - e^{j\widehat{\omega}^{[l]}(t)}\widehat{s}^{[l]}(t-1) \\
\widehat{s}^{[l]}(t) &= e^{j\widehat{\omega}^{[l]}(t)}\widehat{s}^{[l]}(t-1) + \mu^{[l]}\varepsilon^{[l]}(t) \\
g^{[l]}(t) &= \mathrm{Im}\left[\frac{(\varepsilon^{[l]}(t))^* e^{j\widehat{\omega}^{[l]}(t)}}{(\widehat{s}^{[l]}(t-1))^*}\right] \\
\widehat{\omega}^{[l]}(t+1) &= \widehat{\omega}^{[l]}(t) - (\mu^{[l]})^2 g^{[l]}(t) \qquad (7) \\
l &= 1,\ldots,L
\end{aligned}$$

each set to a different value of μ.
A simple way of combining the estimates yielded by the competing algorithms is to set

$$\widehat{s}(t) = \widehat{s}^{[\widehat{l}(t)]}(t) \qquad (8)$$

where

$$\widehat{l}(t) = \arg\min_{1\leq l\leq L}\sum_{m=1}^{M}|\varepsilon^{[l]}(t-m)|^2$$

and M is the length of the local analysis interval.
The switching rule (8) makes its choice on the basis of comparing predictive abilities of different filters, but it does not take into account the distribution of the accumulated prediction error over the set of competitive algorithms. To further improve results one can replace (8) with the following weighted estimation formula

$$\overline{s}(t) = \sum_{l=1}^{L} c^{[l]}(t)\widehat{s}^{[l]}(t) \qquad (9)$$

where the weighting coefficients $c^{[l]}(t), l = 1,\ldots,L$ (called credibility coefficients in [5]) obey

$$c^{[l]}(t) = \frac{\eta^{[l]}(t)}{\sum_{l=1}^{L}\eta^{[l]}(t)}$$

$$\eta^{[l]}(t) = \left[\sum_{m=1}^{M}|\varepsilon^{[l]}(t-m)|^2\right]^{-M/2}, \quad l = 1,\ldots,L$$

When designing a multiple-model adaptive scheme, one is interested in selecting the adaptation gains of competing algorithms so as to increase robustness of the parallel structure, i.e. decrease its sensitivity to unknown and/or time-varying degree of nostationarity of the identified signal. Using similar arguments as those presented in [5, Ch. 8], one can show that in order to maximize robustness of the parallel scheme the gains $\mu^{[1]},\ldots,\mu^{[L]}$ should form a geometric progression, i.e. $\mu^{[i+1]} = \delta\mu^{[i]}$, $i = 1,\ldots,L-1$ where $\delta > 1$ is a constant multiplier.

2.4 Combined approach

Parallel optimization based on (8) or (9) is not restricted to banks of fixed-gain filters. This allows one to combine in a judicious way parallel optimization with sequential optimization. As an example, consider a parallel scheme made up of three GANF algorithms: the "center", self-optimizing filter (6), which works out the estimate $\widehat{\mu}(t)$, and two "side" filters – a "slow" filter, with the gain $\widehat{\mu}(t)/\delta$, and a "fast" one, with the gain $\widehat{\mu}(t)\delta$, where $\delta > 1$ is the appropriately chosen multiplier. Note that the gains of the side filters are simply the rescaled versions of the gain of the center filter, i.e. side filters do not estimate μ on their own. Experimental results show clearly that such joint sequential/parallel optimization preserves advantages of both treatments: adaptiveness (sequential approach) and robustness (parallel approach).

3. ALGORITHM FOR MULTIPLE FREQUENCIES

3.1 Estimation of signal components

Let

$$y_i(t) = s_i(t) + v(t), \quad i = 1,\ldots,k$$

If the signal components defined above were known, the multiple-frequency sequential optimization algorithm could have been designed in a decoupled form as a collection of single-frequency filters of the form (6), driven by $y_1(t),\ldots,y_k(t)$, respectively. When such knowledge is not available, which is a typical situation in practice, one can replace the outputs $y_1(t),\ldots,y_k(t)$ with the following estimates

$$\widehat{y}_i(t) = y(t) - \sum_{\substack{n=1\\ n\neq i}}^{k}\widehat{s}_n(t|t-1) \qquad (10)$$

where $\widehat{s}_n(t|t-1)$ denotes the predicted value of $s_n(t)$, based on information available at instant $t-1$.
When designing multiple-frequency parallel optimization schemes, the estimation technique described above results in huge computational savings. Actually, suppose that the signal has k frequency components, and that for each component i one considers L different levels of the adaptation gain: $\mu_i^{[1]},\ldots,\mu_i^{[L]}$. To realize a multiple-frequency parallel optimization scheme, analogous to (7), one should run in parallel and compare L^k different algorithms, each consisting of k subalgorithms. The competing filters correspond to different combinations of adaptation gains, namely: $\{\mu_1^{[1]},\mu_2^{[1]},\ldots,\mu_k^{[1]}\}$, $\{\mu_1^{[2]},\mu_2^{[1]},\ldots,\mu_k^{[1]}\}$, ..., $\{\mu_1^{[L]},\mu_2^{[1]},\ldots,\mu_k^{[1]}\}$, ..., $\{\mu_1^{[L]},\mu_2^{[L]},\ldots,\mu_k^{[L]}\}$. Even for relatively small values of k and L the number of structural variants becomes impractically large, e.g. for three frequencies and three gain levels the total number of algorithms (each made up of three subalgorithms) that should be run in parallel is 27.
Incorporating the estimates $\widehat{y}_1(t),\ldots,\widehat{y}_k(t)$ one can estimate prediction errors for each of the k frequency components $\varepsilon_i^{[l]}(t) = \widehat{y}_i(t) - e^{j\widehat{\omega}_i^{[l]}(t)}\widehat{s}_i^{[l]}(t-1)$, $l = 1,\ldots,L$, $i = 1,\ldots,k$. Hence, when the technique described above is used, there is only a need to run *one* algorithm consisting of kL subalgorithms.

3.2 Proposed algorithm

Denote by $\widehat{s}_i(t),\widehat{\omega}_i(t+1)$ the estimates yielded by the ith subalgorithm of the multiple-frequency sequential optimization algorithm. Denote by $\widehat{s}_i^-(t),\widehat{\omega}_i^-(t+1)$ and $\widehat{s}_i^+(t),\widehat{\omega}_i^+(t+1)$ the estimates yielded by the corresponding side filters - the slow one and the fast one, respectively. Finally, denote by $\overline{s}_i(t),\overline{\omega}_i(t+1)$ the weighted estimates of $s_i(t),\omega_i(t+1)$, obtained by means of averaging $\widehat{s}_i^-(t),\widehat{\omega}_i^-(t+1)$, $\widehat{s}_i(t),\widehat{\omega}_i(t+$

1) and $\widehat{s}_i^+(t), \widehat{\omega}_i^+(t+1)$ in the way described in Section 2.3. Note, that the predicted value of $s_i(t)$, based on information available at instant $t-1$, can be obtained from

$$\widehat{s}_i(t|t-1) = e^{j\bar{\omega}_i(t)}\bar{s}_i(t-1) \tag{11}$$

Combining (11) with (10) and using the technique described in the preceding subsection one arrives at the following multiple-frequency combined algorithm

center filter:

$$\begin{aligned}
\widehat{y}_i(t) &= y(t) - \sum_{\substack{n=1\\ n\neq i}}^{k} e^{j\bar{\omega}_n(t)}\bar{s}_n(t-1)\\
\varepsilon_i(t) &= \widehat{y}_i(t) - e^{j\widehat{\omega}_i(t)}\widehat{s}_i(t-1)\\
\zeta_i(t) &= -e^{j\widehat{\omega}_i(t)}[j\chi_i(t)\widehat{s}_i(t-1) + \psi_i(t-1)]\\
\psi_i(t) &= \varepsilon_i(t) - [1-\widehat{\mu}_i(t-1)]\zeta_i(t)\\
\rho_i(t) &= \mathrm{Im}\left\{ \frac{e^{j\widehat{\omega}_i(t)}}{\widehat{s}_i^*(t-1)} \left[\, \zeta_i^*(t) + j\varepsilon_i^*(t)\chi_i(t) \right.\right.\\
&\qquad \left.\left. - \frac{\varepsilon_i^*(t)\psi_i(t-1)}{\widehat{s}_i^*(t-1)} \,\right] \right\}\\
r_i(t) &= \lambda_i r_i(t-1) + |\zeta_i(t)|^2\\
\widehat{\mu}_i(t) &= \left[\widehat{\mu}_i(t-1) - \frac{\mathrm{Re}[\varepsilon_i(t)\zeta_i^*(t)]}{r_i(t)} \right]_0^{\mu_{max}}\\
\widehat{s}_i(t) &= e^{j\widehat{\omega}_i(t)}\widehat{s}_i(t-1) + \widehat{\mu}_i(t)\varepsilon_i(t)\\
g_i(t) &= \mathrm{Im}\left[\frac{\varepsilon_i^*(t)e^{j\widehat{\omega}_i(t)}}{\widehat{s}_i^*(t-1)} \right]\\
\widehat{\omega}_i(t+1) &= \widehat{\omega}_i(t) - \widehat{\mu}_i^2(t)g_i(t)\\
\chi_i(t+1) &= \chi_i(t) - \widehat{\mu}_i(t)[2g_i(t) + \widehat{\mu}_i(t)\rho_i(t)]\\
\eta_i(t) &= \left[\sum_{m=0}^{M-1} |\varepsilon_i(t-m)|^2 \right]^{-M/2}
\end{aligned} \tag{12}$$

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

side filters:

$$\begin{aligned}
\widehat{\mu}_i^+(t) &= \widehat{\mu}_i(t)\delta\\
\widehat{\mu}_i^-(t) &= \widehat{\mu}_i(t)/\delta\\
\varepsilon_i^\pm(t) &= \widehat{y}_i(t) - e^{j\widehat{\omega}_i^\pm(t)}\widehat{s}_i^\pm(t-1)\\
\widehat{s}_i^\pm(t) &= e^{j\widehat{\omega}_i^\pm(t)}\widehat{s}_i^\pm(t-1) + \widehat{\mu}_i^\pm(t)\varepsilon_i^\pm(t)\\
g_i^\pm(t) &= \mathrm{Im}\left[\frac{(\varepsilon_i^\pm(t))^* e^{j\widehat{\omega}_i^\pm(t)}}{(\widehat{s}_i^\pm(t-1))^*} \right]\\
\widehat{\omega}_i^\pm(t+1) &= \widehat{\omega}_i^\pm(t) - (\widehat{\mu}_i^\pm(t))^2 g_i^\pm(t)\\
\eta_i^\pm(t) &= \left[\sum_{m=0}^{M-1} |\varepsilon_i^\pm(t-m)|^2 \right]^{-M/2}
\end{aligned} \tag{13}$$

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

output filter:

$$\begin{aligned}
c_i(t) &= \frac{\eta_i(t)}{\eta_i^-(t) + \eta_i(t) + \eta_i^+(t)}\\
c_i^\pm(t) &= \frac{\eta_i^\pm(t)}{\eta_i^-(t) + \eta_i(t) + \eta_i^+(t)}\\
\bar{\omega}_i(t+1) &= c_i^-(t)\widehat{\omega}_i^-(t+1) + c_i(t)\widehat{\omega}_i(t+1)\\
&\quad + c_i^+(t)\widehat{\omega}_i^+(t+1)\\
\bar{s}_i(t) &= c_i^-(t)\widehat{s}_i^-(t) + c_i(t)\widehat{s}_i(t) + c_i^+(t)\widehat{s}_i^+(t)
\end{aligned} \tag{14}$$

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

$$\bar{s}(t) = \sum_{i=1}^{k} \bar{s}_i(t) \tag{15}$$

4. COMPUTER SIMULATIONS

In the first simulation experiment the analyzed signal was governed by

$$y(t) = s(t) + v(t) = ae^{j\sum_{\tau=1}^{t}\omega(\tau)} + v(t)$$

where the amplitude a was set to 1 and the frequency $\omega(t)$ evolved according to the random walk model.

The evaluation of different gain scheduling rules was started at the instant $t = 1001$, after all compared algorithms have reached their steady state. Since the adopted noise variances σ_v^2 and σ_w^2 were time-varying: $\sigma_v^2(t) = \{1$ for $t \in [1,7000]; 0.2$ for $t \in [7001,10000]\}$, $\sigma_w^2(t) = \{10^{-5}$ for $t \in [1,4000]; 10^{-6}$ for $t \in [4001,10000]\}$, the entire analysis interval $T = [1001,10000]$, covering 9000 samples, was divided into three subintervals $T_1 = [1001,4000]$, $T_2 = [4001,7000]$ and $T_3 = [7001,10000]$, corresponding to three different tracking conditions. The steady state optimal values of μ, computed according to (4), were: $\mu_o(t) = 0.0067$ for $t \in T_1$, $\mu_o(t) = 0.0038$ for $t \in T_2$ and $\mu_o(t) = 0.0056$ for $t \in T_3$.

Five algorithms were compared: the optimally tuned ANF filter (5), two variants of the parallel optimization algorithm (7), (9) ("tuned": $L = 3$, $\mu^{[1]} = 0.03$, $\mu^{[2]} = 0.045$, $\mu^{[3]} = 0.067$, and "detuned": $L = 3$, $\mu^{[1]} = 0.02$, $\mu^{[2]} = 0.03$, $\mu^{[3]} = 0.045$), the sequential optimization algorithm (6) ($\lambda = 0.99$, $\mu_{max} = 0.2$), and the algorithm based on the combined approach ($\lambda = 0.99$, $\mu_{max} = 0.2$, $\delta = 1.5$). The length of the local analysis interval was set to $M = 30$. Observe that for the tuned parallel optimization algorithm it holds that $\mu_o(t) \in [\mu^{[1]}, \mu^{[3]}]$, $\forall t \in T$; the detuned algorithm does not have this property.

Figure 1 shows tracking results yielded by the sequential algorithm - note that the proposed scheme is doing a pretty good job in optimizing the adaptation gain μ.

According to Table 1, the tuned parallel optimization approach yields better results than the sequential approach. However, the above conclusion does not remain true if the filter gains are chosen less carefully - see results obtained for the detuned algorithm in intervals T_1 and T_3. Therefore, from the robustness point of view, the combined scheme seems to be the most advisable one: while only slightly inferior to the carefully designed parallel scheme, it is self-dependent in choosing the right range of μ's, i.e. it does not require any prior knowledge of the degree of signal nonstationarity.

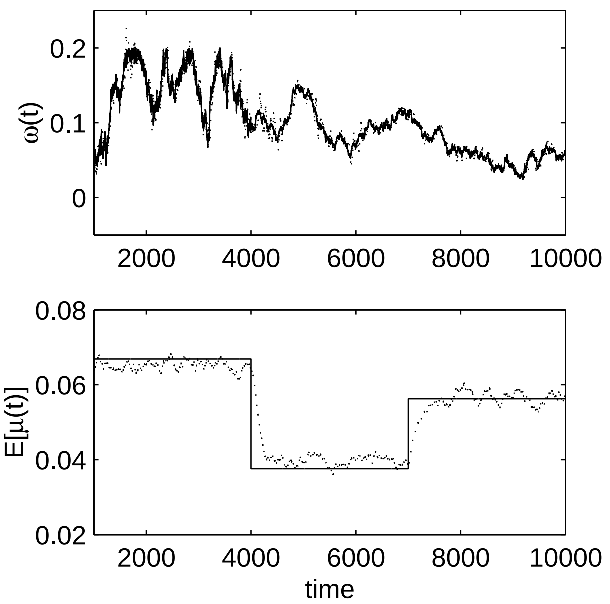

Figure 1: Tracking results for a single cisoid obtained using the sequential algorithm. The upper plot shows evolution of the true frequency (solid line) and its estimate (dotted line). The lower plot shows the optimal value $\mu_o(t)$ and the ensemble average of its estimates $\widehat{\mu}(t)$, corresponding to 100 different realizations of measurement noise.

Table 1: Comparison of the mean-squared values of signal estimation errors $\Delta s(t) = \widehat{s}(t) - s(t)$ for different algorithms (evaluated from the results of 100 simulation runs).

μ	T_1	T_2	T_3
optimal	0.063	0.037	0.011
parallel optimization (tuned)	0.064	0.038	0.011
parallel optimization (detuned)	0.083	0.037	0.013
sequential optimization	0.068	0.043	0.012
combined approach	0.063	0.039	0.012

Figure 2 shows results of application of the combined sequential/parallel optimization algorithm (12)–(15) (with $M = 30$, $\delta = 1.5$ and $\lambda_1 = \lambda_2 = \lambda_3 = 0.99$) to estimation of a nonstationary multifrequency signal buried in white measurement noise

$$y(t) = s(t) + v(t)$$

$$s(t) = 1.5e^{j\sum_{\tau=1}^{t}\omega_1(\tau)} - je^{j\sum_{\tau=1}^{t}\omega_2(\tau)} + 2e^{j\sum_{\tau=1}^{t}\omega_3(\tau)}$$

Again, the results are satisfactory, both in terms of frequency tracking and signal tracking.

5. CONCLUSION

On-line optimization of tracking performance of adaptive notch filters (ANF) can be done either sequentially or by means of parallel estimation. We have shown that the best results are obtained if both approaches are appropriately combined. For a single cisoid the combined sequential/parallel algorithm is made up of three ANF filters: the center filter, which performs sequential gain optimization, and two side filters (“slow” and “fast”) with adaptation gains that are the rescaled versions of the gain of the center filter. The combined approach can be easily extended to multifrequency signals. The resulting algorithm has moderate computational requirements and superb tracking properties, confirmed by the simulation evidence.

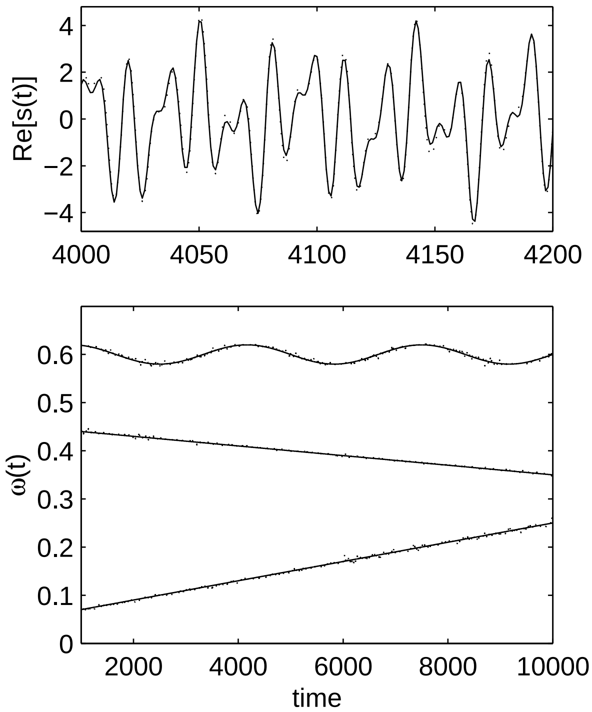

Figure 2: Typical tracking results for a multifrequency signal obtained using the combined algorithm. The upper plot shows the real part of the true signal (solid line) and its estimate (dotted line). The lower plot shows evolution of the true frequencies $\omega_i(t), i = 1,\dots,3$ and their estimates (SNR=12dB).

REFERENCES

[1] P. Tichavský and P. Händel, “Two algorithms for adaptive retrieval of slowly time-varying multiple cisoids in noise,” *IEEE Trans. on Signal Processing*, vol. 43, pp. 1116–1127, 2005

[2] P. Tichavský and A. Nehorai, “Comparative study of four adaptive frequency trackers,” *IEEE Trans. on Signal Processing*, vol. 45, pp. 1473–1484, 1997.

[3] M. Niedźwiecki and P. Kaczmarek, “Signal tracking properties of a class of adaptive notch filters,” *Proc. 13th European Signal Processing Conference, EUSIPCO 2005*, Antalya, Turkey, 2005, pp. 1–4.

[4] Y.-C.Lim, Y.-X. Zou and N. Zheng, “A piloted adaptive notch filter,” *IEEE Trans. on Signal Processing*, vol. 53, pp. 1310–1323, 2005.

[5] M. Niedźwiecki, *Identification of Time-varying Processes*, Wiley, 2000.

[6] M. Niedźwiecki and P. Kaczmarek, “Tracking analysis of a generalized adaptive notch filter,” *IEEE Trans. on Signal Processing*, vol. 54, pp. 304–314, 2006.

[7] T. Söderström and P. Stoica, *System Identification*, Prentice Hall, 1988.