A Complex Adaptive IIR notch Filter Algorithm with Optimal Convergence Factor

Yaohui Liu[1], Timo I. Laakso[1], and Paulo S. R. Diniz[2]

Abstract **- In this paper, a variable convergence factor is introduced to Gauss-Newton type algorithm based complex adaptive IIR notch filters to estimate the frequencies of the sinusoids embedded in white noise. The variable convergence factor is optimized in the sense of maximal reduction of mean square error (MSE) at each iteration as the adaptation algorithm is executed. Simulation results confirm that the proposed convergence factor leads to faster convergence than a fixed convergence factor.**

1 Introduction

Adaptive notch filters (ANF) are widely used in many signal processing applications to extract, eliminate or trace narrow-band or sinusoidal signals embedded in broadband noise [1]. If such signal consists of in-phase and quadrature components, a complex coefficient ANF must be implemented. Most of such applications are in radar and communication systems. To yield sharp-cutoff bandpass characteristics, IIR filter formulation is more efficient than its FIR counterpart. An early contribution by Nehorai [2] imposed constraints on a notch transfer function, which leads to simple relations between poles and zeros, thus, it can be exploited advantageously in adaptive filter design.

Numerous algorithms for ANF have been proposed (e.g., [2-5]), most of them belonging to the recursive prediction error type. An important issue to consider when implementing these ANF algorithms is the choice of the convergence factor associated with the algorithm for coefficient updating and the pole radius factor associated with the notch bandwidth. These two factors affect the stability and the convergence speed of the algorithm. The choice of these factors is a tradeoff between tracking ability and noise sensitivity [1]. In this paper we propose a variable convergence factor that optimizes a well-defined instantaneous error criterion not requiring *a priori* assumptions about the signal and noise characteristics. The paper is organized as follows. Section 2 defines the system model for a Gauss-Newton algorithm based complex adaptive IIR notch filter. The optimal convergence factor is derived in section 3. In section 4, simulation examples are provided. Finally section 5 concludes the paper.

[1] Helsinki University of Technology, Signal Processing Laboratory, , P.O. Box 3000, FIN-02150, Finland, email: {yaohui.liu, timo.laakso}@hut.fi Tel: +358-9-451-5975, Fax: +358-9-460-224
[2] COPPE / Universidade Federal do Rio de Janeiro Caixa Postal 68504, RJ, Brazil, 21945-970, email: diniz@lps.ufrj.br

2 System Models

2.1 Constrained Notch Filters

Adaptive notch filters are designed for signal environments consisting of sinusoidal components of unknown frequencies immersed in white noise. In communications systems, the sinusoidal signals are often modeled with complex exponentials. Such signals take the form

$$u(k) = \sum_{i=1}^{M} P_i \exp\left(j\omega_i k + \varphi_i\right) + n(k) \tag{1}$$

where P_i is the amplitude of the i^{th} sinusoid and φ_i is its phase. $n(k)$ denotes the noise which is assumed independent of the sinusoid terms, and will usually be considered white. In order to estimate the sinusoidal frequencies ω_i and consequently eliminate the corresponding sinusoids, the following constrained form of the notch filter has been considered in many papers [5].

$$H_M(z) = \prod_{i=1}^{M} \frac{1 - e^{j\hat{\omega}_i} z^{-1}}{1 - re^{j\hat{\omega}_i} z^{-1}} = \frac{1 + \sum_{i=1}^{M} a_i^* z^{-i}}{1 + \sum_{i=1}^{M} r^i a_i^* z^{-i}} \tag{2}$$

where ω_i represents the i^{th} notch frequency and r is the pole radius. The bandwidth of the complex notch created by each pole-zero pair is $B_N = \pi(1-r)$ [3]. $a_i = a_{iR} + ja_{iI}$ is a complex coefficient and the notation * denotes the complex conjugate. Note that the above notch filter consists of cascades of M first-order filters which have their zeros on the unit circle, resulting in exactly zero gain at each notch frequency. Observe (2), the complex-valued coefficients can be parameterized in vector form

$$\boldsymbol{\theta} = \left[a_1 \ a_2 \ \cdots \ a_M\right]^T \tag{3}$$

and (2) can be expressed in the form [2]

$$H(z) = \frac{A(z)}{A(rz)} \tag{4}$$

The input-output description of a notch filter can also be expressed in difference equation form

$$e(k) = u(k) + \sum_{i=1}^{M} a_i^* \left[u(k-i) - r^i e(k-i)\right] \quad (5)$$
$$= u(k) - \boldsymbol{\theta}^H \boldsymbol{\Phi}(k)$$

where

$$\boldsymbol{\Phi}(k) = [\phi_1(k) \;\; \phi_2(k) \;\; \cdots \;\; \phi_M(k)]^T \quad (6)$$
$$\phi_i(k) = -u(k-i) + r^i e(k-i) \quad i = 1 \cdots M$$

and the superscript H denotes Hermitian operator (conjugate transpose). The reference signal for notch filter is considered as zero and $e(k)$ is the output error.

2.2 ANF Algorithms

We can use Gauss-Newton type algorithm [5] to adjust parameter vector $\boldsymbol{\theta}$ of the above notch filter form. First, we calculate the output error $e(k)$ based on an estimate of $\boldsymbol{\theta}$ and input-output data.

$$e(k) = u(k) - \boldsymbol{\theta}^H(k)\boldsymbol{\Phi}(k) \quad (7)$$

Because we have a causal system, the data are zero if their time indexes are smaller than 1. The initial $\boldsymbol{\theta}$ can be defined as $\boldsymbol{\theta}_0 = [0,\ 0,\ldots,1]^T$ Now, take the gradient vector of $e(k)$ with respect to $\boldsymbol{\theta}$, which can be defined as $\boldsymbol{\Psi}(k)$ and update the covariance matrix $\mathbf{R}$

$$\mathbf{R}(k+1) = \lambda(k)\mathbf{R}(k) + \alpha(k)\boldsymbol{\Psi}(k)\boldsymbol{\Psi}^H(k) \quad (8)$$

where $\lambda(k)$ is the forgetting factor and $\alpha(k) = 1 - \lambda(k)$ is regarded as the convergence factor,

$$0 \le \alpha(k) \le 1 \quad (9)$$

which determines the convergence speed and the stability of the algorithm. Since computing the inverse of a matrix is computationally expensive, the inversion is usually updated by using the matrix inversion lemma. Defining $\mathbf{P} = \mathbf{R}^{-1}$, we have

$$\mathbf{P}(k+1) = \frac{1}{\lambda(k)}\left[\mathbf{P}(k) - \left(\frac{\mathbf{P}(k)\boldsymbol{\Psi}(k)\boldsymbol{\Psi}^H(k)\mathbf{P}(k)}{\frac{\lambda(k)}{\alpha(k)} + \boldsymbol{\Psi}^H(k)\mathbf{P}(k)\boldsymbol{\Psi}(k)}\right)\right] \quad (10)$$

Then we can update $\boldsymbol{\theta}$ according to the Newton algorithm with the available estimate of the covariance matrix.

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + \alpha(k)\mathbf{P}(k+1)\boldsymbol{\Psi}(k)e^*(k) \quad (11)$$

After obtaining $\boldsymbol{\theta}(k+1)$, we can improve the estimation of $e(k)$ by its *a posteriori* estimate value

$$\hat{e}(k) = u(k) - \boldsymbol{\theta}^H(k+1)\boldsymbol{\Phi}(k) \quad (12)$$

The reason why $\hat{e}(k)$ is superior to $e(k)$ is because it includes the influence of the new incoming data to the filter coefficients when calculating the filter output. From this point of view, $\alpha(k)$ can be interpreted as the weight for the new incoming data.

Calculating the gradient $\boldsymbol{\Psi}(k)$ can be computationally expensive. However, we can also solve it recursively. Define the negative gradient vector as

$$\boldsymbol{\Psi}(k) = \frac{-1}{2}\nabla e(k) = \frac{-1}{2}(\nabla_R e(k) + j\nabla_I e(k)) \quad (13)$$

where

$$\nabla_R e(k) = \left[\frac{\partial e(k)}{\partial a_{1R}}\ \frac{\partial e(k)}{\partial a_{2R}} \cdots \frac{\partial e(k)}{\partial a_{MR}}\right]^T \quad (14)$$

$$\nabla_I e(k) = \left[\frac{\partial e(k)}{\partial a_{1I}}\ \frac{\partial e(k)}{\partial a_{2I}} \cdots \frac{\partial e(k)}{\partial a_{MI}}\right]^T \quad (15)$$

According to [9], assuming that the coefficients adapt slowly, i.e.,

$$\boldsymbol{\theta}(k) \approx \boldsymbol{\theta}(k-1) \approx \cdots \approx \boldsymbol{\theta}(k-M+1) \quad (16)$$

we have

$$\boldsymbol{\Psi}(k) = \frac{\boldsymbol{\Phi}(k)}{A(rz)} \quad (17)$$

So, we can compute $\boldsymbol{\Psi}$ recursively. According to (6), $\boldsymbol{\Psi}(k)$ can be expressed as

$$\boldsymbol{\Psi}(k) = [\psi_1(k) \;\; \psi_2(k) \;\; \cdots \;\; \psi_M(k)]^T \quad (18)$$
$$\psi_i(k) = -u_g(k-i) + r^i \hat{e}_g(k-i) \quad i = 1 \cdots M$$

where $u_g(k)$ and $\hat{e}_g(k)$ can be updated by the following difference equations

$$u_g(k) = u(k) - \sum_{i=1}^{M} r^i a_i(k) u_g(k-i) \quad (19)$$

$$\hat{e}_g(k) = \hat{e}(k) - \sum_{i=1}^{M} r^i a_i(k) \hat{e}_g(k-i) \quad (20)$$

3 Optimal Convergence Factor

In the Gauss-Newton algorithm, prediction error $e(k)$ is replaced by *a posteriori* prediction error $\hat{e}(k)$. In each iteration, we need to determine $\alpha(k)$ for the purpose of updating $\mathbf{P}$ and $\boldsymbol{\theta}$. Since the statistics of the input signal and the system to be modeled are not known in practice, a fixed convergence factor tends to yield unsatisfactory results. An optimal convergence factor $\alpha(k)$ puts a proper weight on the new incoming data at each updating step, which will lead to the maximal reduction of MSE, thus speeding up the convergence.

In the updating step (7) the MSE is a nonquadratic function of $\boldsymbol{\theta}$. Nevertheless, in the neighborhood of a given point on the MSE surface, the MSE can be approximated by a quadratic function

$$\varepsilon(k)=\frac{|e(k)|^2}{2}\approx\frac{|u(k)|^2}{2}+\boldsymbol{\theta}^H(k)\boldsymbol{\rho}(k)+\frac{1}{2}\boldsymbol{\theta}^H(k)\mathbf{R}(k)\boldsymbol{\theta}(k) \quad (21)$$

where $\boldsymbol{\rho}(k)$ is defined as $\nabla_{\boldsymbol{\theta}}\varepsilon(k)|_{\boldsymbol{\theta}=0}$ and $\mathbf{R}(k)$ is the Hessian matrix in (8). Differentiating (21) with respect to $\boldsymbol{\theta}$ yields an expression for $\boldsymbol{\rho}(k)$

$$\boldsymbol{\rho}(k)=\nabla\varepsilon(k)-\mathbf{R}(k)\boldsymbol{\theta}(k) \quad (22)$$

where

$$\nabla\varepsilon(k)=\nabla\left(\frac{|e(k)|^2}{2}\right)=-\boldsymbol{\Psi}(k)e^*(k) \quad (23)$$

In the process of updating $\boldsymbol{\theta}$, $\varepsilon(k)$ can be substituted by its *a posteriori* estimate

$$\hat{\varepsilon}(k)=\frac{|u(k)|^2}{2}+[\boldsymbol{\theta}(k)+\Delta\boldsymbol{\theta}(k)]^H\boldsymbol{\rho}(k)+\frac{1}{2}[\boldsymbol{\theta}(k)+\Delta\boldsymbol{\theta}(k)]^H\mathbf{R}(k)[\boldsymbol{\theta}(k)+\Delta\boldsymbol{\theta}(k)] \quad (24)$$

where

$$\Delta\boldsymbol{\theta}(k)=\boldsymbol{\theta}(k+1)-\boldsymbol{\theta}(k)=\alpha(k)\mathbf{P}(k)\boldsymbol{\Psi}(k)e^*(k) \quad (25)$$

By inserting (22) into (24), it can be easily shown that

$$\Delta\varepsilon(k)=\Delta\boldsymbol{\theta}(k)\nabla\varepsilon(k)+\frac{1}{2}\Delta\boldsymbol{\theta}^H(k)\mathbf{R}(k)\Delta\boldsymbol{\theta}(k) \quad (26)$$

where

$$\Delta\varepsilon(k)=\hat{\varepsilon}(k)-\varepsilon(k) \quad (27)$$

By inserting (25) (23) into (26), noticing that $\mathbf{P}(k)$ is a Hermitian matrix, i.e., $\mathbf{P}^H(k)=\mathbf{P}(k)$, we obtain

$$\Delta\varepsilon(k)=|e(k)|^2\tau_k\frac{\alpha^2(k)/2-\alpha(k)}{1+\alpha(k)(\tau_k-1)} \quad (28)$$

where

$$\tau_k=\boldsymbol{\Psi}^H(k)\mathbf{P}(k)\boldsymbol{\Psi}(k) \quad (29)$$

The optimal $\alpha(k)$ that yields maximal MSE reduction can be found by setting the derivative to zero,

$$\frac{\partial\Delta\varepsilon(k)}{\partial\alpha(k)}=0 \quad (30)$$

Which yields the optimal solution

$$\alpha_{opt}(k)=\frac{2}{1+\sqrt{2\tau_k-1}} \quad (31)$$

If $2\tau_k-1$ is less than zero, it should be replaced by zero. Note that the above result is similar to the one in [8], which is derived under the assumption of quadratic MSE surface. However, in our case the formation of $\boldsymbol{\theta}$ is complex valued and is constrained due to the structure of the filter, thus, the MSE surface is non-quadratic, an empirical reduction parameter κ must be included

$$\alpha_{opt}(k)=\frac{\kappa}{1+\sqrt{2\tau_k-1}} \quad (32)$$

Based on the simulation for the first order notch filter, as a rule of thumb, κ = 0.05 is a good choice. It should be noted that the initial convergence factor is not critical in the case of using optimal convergence factor. On the other hand, for a fixed step size, a proper choice of α depends on a priori knowledge of SNR. In practice, small enough α has to be chosen to guarantee the convergence in the case of low SNR conditions. Furthermore, note that τ_k is an intermediate result of (10), therefore, finding the optimal convergence factor does not increase the complexity of the algorithm.

4. Simulation Results

Due to the constraint form of the notch filter, it can be completely characterized by its first-order stages. For simplicity, we will consider only the single-notch case in the simulation. The input signal consists of a complex sinusoid with noise and the sinusoid power is 10 dB higher than the noise power. Suppose single sinusoid exists in the model (1), we compare the convergence of the algorithm with fixed convergence factor α and the algorithm with optimal α. Pole radius affects the convergence speed. To compare the influence of convergence factor, we fix the pole radius at $r=0.89$. The fixed convergence factor is set to be $\alpha=0.0005$ to guarantee the stability of the estimate. The reduction parameter κ equals to 0.05.

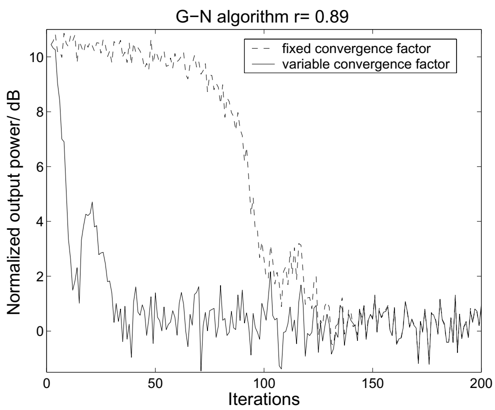

Figure 1: Convergence of the algorithm for a first order notch filter

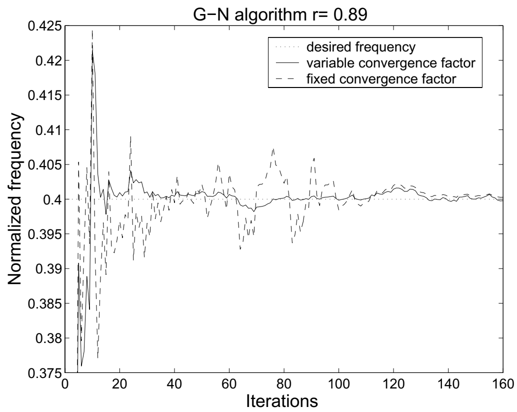

Figure 2: Frequency estimation results for the first order notch filter.

The output MSE is shown in Fig. 1, which is the average of 50 independent runs. It can be seen that using optimal α leads to the optimal solution in ca. 40 iterations, whereas the fixed step size requires ca. 130 iterations. Furthermore, the frequency estimate in Fig. 2 shows better performance for the variable convergence factor too.

The second example in Fig. 3 shows the MSE convergence for high noise case. The pole radius again equals to 0.89. The sinusoid power is now only 3 dB higher than the noise power. Now, the convergence of the notch filter is slower for both algorithms. The fixed step size converges in ca. 200 iterations, whereas the optimal step size has converged in ca. 60 iterations, which is an improvement of ca. 140 iterations.

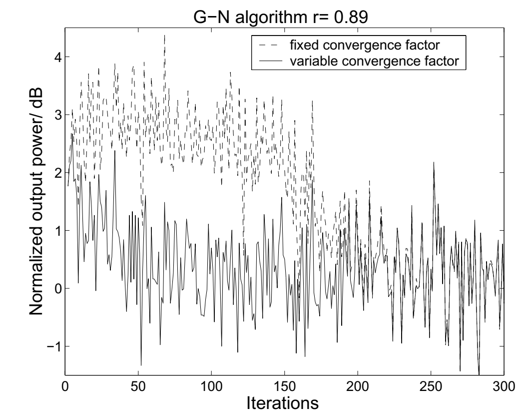

Figure 3: Convergence of the algorithm for a first order notch filter for high noise case.

5 Conclusions

A complex constrained IIR notch filter based on Gauss-Newton algorithm using optimal convergence factor has been proposed. The scheme to optimize the convergence factor α is to maximize the reduction of MSE in each iteration, which leads to faster convergence and makes it easier to choose α without *a priori* knowledge of the signal model. Furthermore, because the calculation of α only requires the intermediate result in the updating algorithms, it does not increase the complexity of the algorithm essentially.

References

[1] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, 1983

[2] A. Nehorai, "A minimal parameter adaptive notch filter with constrained poles and zeros," *IEEE Trans.on ASSP*, vol. ASSP-33, no. 4, pp. 983-996, Aug. 1985.

[3] P. Stoica and, A. Nehorai, “Performance analysis of an adaptive notch filter with constrained poles and zeros,” *IEEE Trans. on ASSP*, Vol. ASSP-36, No.6, pp. 911-919, June 1998.

[4] P. Händel and, A. Nehorai, “Tracking analysis of an adaptive notch filter with constrained poles and zeros,” *IEEE Trans. Signal Processing*, Vol.42, No.2, pp. 281-291, Feb. 1994.

[5] S. C. Pei and C. Tseng, "Complex adaptive IIR notch filter algorithm and its applications," *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 41, no. 2, pp. 158-163, Feb. 1994.

[6] P. S. R. Diniz, *Adaptive filtering: algorithm and practical implementation*, Kluwer, Boston, 1997.

[7] J.E. Cousseau and, P.S.R. Diniz, “On optimal convergence factor for IIR adaptive filters,” *in Proc. IEEE ISCAS '94*, Vol 2, pp.137–140, London, May, 1994.

[8] J.E. Cousseau, P.S.R. Diniz, and A. Antoniou, “Improved parallel realisation of IIR adaptive filters”, *Proc. IEE Proceedings part G: Circuits, Devices and Systems*, Vol: 140 5, pp.322 –328, Oct. 1993.

[9] J. J. Shynk, “A complex adaptive algorithm for IIR filtering,” *IEEE Trans. on ASSP*, Vol. ASSP-34, No.5, pp. 1342-1344, Oct. 1986.