A Fast Algorithm for Real-time Adaptive Notch Filtering

Haeng-Gihl Kim, *Member, KIMICS*

Abstract—A new algorithm is presented for adaptive notch filtering of narrow band or sine signals for embedded among broad band noise. The notch filter is implemented as a constrained infinite impulse response filter with a minimal number of parameters, Based on the recursive prediction error (RPE) method, the algorithm has the advantages of the fast convergence, accurate results and initial estimate of filter coefficient and its covariance is revealed. A convergence criterion is also developed. By using the information of the noise-to-signal power, the algorithm can self-adjust its initial filter coefficient estimate and its covariance to ensure convergence.

Index Terms—Adaptive Notch Filter, Recursive Prediction Error (RPE), convergence and, covariance.

I. INTRODUCTION

ADAPTIVE notch filter have a wide variety of applications in the field of signal processing for detecting and eliminating the time-varying narrow-band signals or other broad band noises. In the report, I present a fast recursive parameter estimation algorithm for infinite impulse response (IIR) adaptive notch filter (ANF), based on the recursive prediction error (RPE) method [1].

The original notch filter has a minimal number of parameters equal to n, the number of the input sine wave signals. Because of the spectral filter structure and the corresponding this can not meet the requirement of real-time processing. In most situations, the narrow-band signals are unknown, the true values of filter coefficients are unknown, and it is really difficult to determine what a proper value of the initial covariance matrix is in order to have the algorithm converge.

By designing a new notch bandwidth control parameter, the error performance surface will be improved; by resetting the initial filter coefficient estimate and its covariance, the convergent rate will be increased. A convergence criterion is developed as well. The new algorithm will self-adjust its initial condition, and has the advantages of faster convergence, more accurate results and computational efficiency than previous ANF algorithms.

Manuscript received November 22, 2001

Haeng-Gihl Kim, is with Yosu University, San 96-1, Dunduck-dong, Yosu, 550-749, Korea. He is now with the Dept. of Electronic Communication Eng., (Tel : +82-61-659-3231, Fax : +82-61-653-7706, e-mail : kcbs1004@yosu.ac.kr)

This study is supported by Yosu National University through Research Fund.

II. ERROR-PERFORMANCE SURFACE

In this section, we discuss the error performance surface of the special ANF [1]. For detailed advantages of the filter structure and derivation of the algorithm, refer to [1].

The system function of the ANF is

$$H(z^{-1}) = \frac{1 + a_1 z^{-1} + \cdots + a_n z^{-n} + \cdots + a_1 z^{-2n+1} + z^{-2n}}{1 + \rho a_1 z^{-1} + \cdots + \rho^n a_n z^{-n} + \cdots + \rho^{2n-1} a_1 z^{-2n+1} + \rho^{2n} z^{-2n}} \quad (1)$$

where, the filter coefficients $a_i (i = 1, \cdots, n)$ are the parameters to be estimated; n is the number of input sine waves; ρ is a parameter which controls the bandwidth of each notch.

Let $y(t)$ be the input of the filter at time t. Assume that $y(t)$ consists of the sum of n sine wave signals plus white noise.

The output of the filter is

$$\varepsilon(t) = \frac{A(q^{-1})}{A(\rho q^{-1})} y(t) \quad (2)$$

where q^{-1} is the unit delay operator, i.e. $q^{-1}y(t) = y(t-1)$,etc.

The RPE algorithm will adjust a_i to minimize the cost function

$$V = \frac{1}{2} \sum_{k=1}^{t} \beta(t,k) \varepsilon(k)^2 \quad (3)$$

where $\beta(t,k)$ is a parameter introduced in time-varying system [1].

This implies that if there are some other local minima exist, there is no guarantee that the algorithm will converge to the global minimum. Thus, it is very important to study the error-performance surface of the cost function. For simplicity, we first examine the error-performance surface for the input signal containing one sine wave signal plus white noise. Thus, the error-performance plot is a two-dimensional plot. Similar analysis can be applied to multi-dimensional performance surface.

III. A FAST ANF ALGORITHM

In this section, I will pursue a fast ANF algorithm based on the original ANF by the following new design: we will improve the error-performance surface of the filter by designing a new notch bandwidth control parameter; we will increase the convergent rate by resetting the initial filter coefficient estimate and its covariance; we will design a new criterion to provide information on the algorithm convergence and when to reset initial condition.

A. Design of Notch Bandwidth Control Parameter

The error-performance surface for original ANF is very difficult for RPE type of algorithm to search its global minimum because the global minimum lies in a very flat region. ρ is a notch bandwidth control parameter[2][3].

The closer ρ is to 1, the narrower the notch bandwidth is. In practical situation, if no information is available on the input sine wave frequencies and the algorithm will not 'sense' the presence of the sine waves. This may prevent the algorithm from converging to the desired transfer functions. In order to increase the sensitivity of the ANF algorithm, wider notch should be used at the start of the data processing. After convergence, it is recommended that a large modulus of poles (narrow notch) be used in order to obtain a smaller distortion in the wideband component of the input signal by the filter [1].

The original notch filter algorithm is very sensitive to the initial conditions. If the initial filter coefficient estimate and its covariance are not chosen properly, even if the error performance surface is improved, the algorithm will not converge. In order to have the original algorithm converge, the initial value of $a_1[0]$ and $P[0]$ must be chosen properly to match the relationship with each other. Specifically, if the prior estimate of filter coefficient is far from its true value, its covariance should be large in order to ensure convergence. On the other hand, if the prior estimate of filter coefficient is near the true value, its covariance should be small in order to ensure convergence. We also notice that if $P[0]$ is too large to match $a_1[0]$, a_1 will pass into transient region on the error-performance surface within a few recursions (k = 2 for $\rho = 0.001$). This is because $P[0]$ is a large initial covariance [2]. The slow or non-convergence performance is because of the small $P[t]$ after a few recursions.

B. Design of Convergence Criterion

Fig. 1 is the frequency response for the ANF with $\rho = 0.8$, In Fig. 1, ω_0 is the zero on the frequency response, ω_1 is the input sine wave frequency. $\Delta\omega$ is the frequency error. Referring to Eq. (1), we have

$$\cos\omega_1 = -\frac{a_1}{2} \qquad \text{(4-a)}$$

$$\sin\omega_1 = \frac{\sqrt{4-a_1^2}}{2} \qquad \text{(4-b)}$$

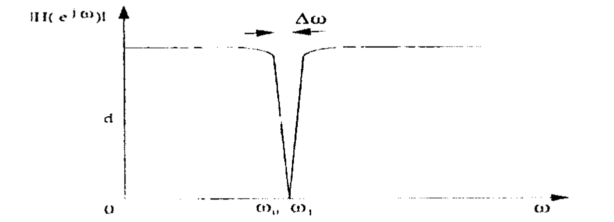

Fig. 1 Relationship between frequency magnitude response and frequency error

Denote $d = |H(e^{j\omega_0})|$. If the SNR is Large, the filter output is mainly the attenuated sine wave with amplitude of fC_1, where C_1 is the input sine wave amplitude.

Let $y(t)$ be the filter input and $\varepsilon(t)$ be the filter output, the signal-to-noise power can be expressed by

$$\frac{E\varepsilon(t)^2}{Ey(t)^2} = d^2 \qquad (5)$$

The above equation relates the filter signal-to-noise power to the frequency response at $\omega = \omega_0$.

From Eq. (1) and (5), we have

$$d^2 = \left||H(e^{j\omega_0})|\right|^2 = \left|H(e^{j(\omega_1-\Delta\omega)})\right|^2 = \frac{E\varepsilon(t)^2}{Ey(t)^2} \qquad (6)$$

Eq. (6) is expresses the signal-to-noise power in terms of the frequency error $\Delta\omega$.

Eq. (6) is derived with the assumption that the SNR is large. The higher the SNR, the smaller $\Delta\omega$ can be applied to Eq. (6). Denote the smallest $\Delta\omega$ that can be applied to Eq. (6) by $\Delta\omega_{min}$. For different SNR, $\Delta\omega_{min}$ is different. When $\omega < \Delta\omega_{min}$, the different between $\frac{E\varepsilon(t)^2}{Ey(t)^2}$ and $|H(e^{j(\omega_1-\Delta\omega)})|^2$ will become significant and Eq. (6) is no longer valid since white noise dominates the filter output instead of the attenuated sine wave signal. The variance of the white noise is constant at $\Delta\omega = 0$, so is the signal-to-noise power $\frac{E\varepsilon(t)^2}{Ey(t)^2}$.

Denote $\frac{E\varepsilon(t)^2}{Ey(t)^2}$ at $\Delta\omega = 0$ by eoylimit. The signal-to-noise ratio will approximately equal to $10\log\frac{1}{eoy\lim it}$.

We vary $\Delta\omega$ from 0 to 0.5. Each $\Delta\omega$ corresponds to a different a_1. Thus, each $\Delta\omega$ is related to a certain value of the noise-to-signal power. Denote the noise-to-signal power by eoy. The noise-to-signal power is calculated by

$$eoy = \frac{E\varepsilon(t)^2}{Ey(t)^2} = \frac{\sum_{K=1}^{n}\varepsilon[k]^2}{\sum_{k=1}^{N}y[k]^2} \qquad (7)$$

where N is an integer and should be a large value in order to minimize the window effect. If N is large enough, the sampling frequency should not affect the value of eoy. According to the experimental results, N = 300 will be reasonable value to use.

Each eoy corresponds to a different value of $\Delta\omega$; we use the same $\Delta\omega$ in Eq. (6) to calculate the corresponding $|H(e^{j(\omega_1-\Delta\omega)})|^2$. Since $|H(e^{j(\omega_1-\Delta\omega)})|^2$ is only effected by $\Delta\omega$ for a given ρ, we choose $a_1 = 1$ to calculate $|H(e^{j(\omega_1-\Delta\omega)})|^2$ with $\rho = 0.8$.

The $\Delta\omega$ in table I is obtained such that the difference between eoy and $|H(e^{j(\omega_1-\Delta\omega)})|^2$ is less than 1.0e-3.

Table 1 Convergence Criterion Reference Table

SNR	$\Delta\omega_{min}$	eoylimit	SNR	SNR-err
90	6.0e-5	1.1037e-9	89.571	4.761e-3
70	7.0e-4	1.1037e-7	69.571	6.123e-3
50	6.0e-3	1.1041e-5	49.570	8.599e-3
30	6.0e-2	1.1065e-3	29.560	1.465e-2
10	3.5e-1	1.0478e-1	9.797	2.026e-2
5	5.0e-1	2.8697e-1	5.422	8.434e-2

Table I shows $\Delta\omega_{min}$ and eoylimit for different SNRs. Signal-to-noise ratio is calculated by $SNR_0 = 10\log\frac{C_1}{2}$. SNR-err is calculated by $SNR-err = \frac{|SNR_0 - SNR|}{SNR_0}$.

Table I can be used as a convergence criterion reference table for obtaining convergence criterion. For a given SNR, if the maximum allowable frequency error $\Delta\omega_{max}$ of the filter is larger than $\Delta\omega_{min}$ in Table I, use $\Delta\omega = \Delta\omega_{max}$ Eq. (6) to calculate $|H(e^{j(\omega_1-\Delta\omega)})|^2$, use it as the convergence criterion eoylimit since Eq. (6) is still valid for $\Delta\omega = \Delta\omega_{max}$. If $\Delta\omega < \Delta\omega_{max}$, Eq. (5.5) is not valid for $\Delta\omega = \Delta\omega_{max}$, use the eoylimit in Table I as the convergence criterion.

Because the input sine wave and the SNR is unknown in actual situations, Table I can be used as a convergence criterion reference provide the SNR.

Fast Fourier Transform (FFT) provides a method to estimate the sine wave signal amplitude. Suppose a signal is $y(t) = C_1 \sin\omega_0 t$, where C_1 is the amplitude of the sine wave. $y[n]$ is the sampled sequence of $y(t)$. By applying a rectangular window $w[n]$ of length N to $y[n]$ and taking the FFT of the windowed sequence, we have $M = \frac{C_1}{2}N$, where M denotes the maximum value on the frequency magnitude response. Thus, the sine wave amplitude is $\frac{2M}{N}$.

If the signal contains both sine wave and white noise, $\frac{2M}{N}$ can still be used as a rough guess of the input sine wave magnitude. So the design of the new ANF will start by taking the FFT of the input signal to obtain an estimate of the SNR, then choose the closest SNR_0 In Table I to obtain the convergence criterion eoylimit.

Let eoyvar reflects the variation of eoy and how fast eoy(k) can move toward the convergent criterion eoylimit, in other word, it reflects how fast can converge to the true value.

In my design, eoyvar is defined by

$$eoy\,\mathrm{var} = \frac{\sum_{i=k-N}^{k-1} eoydiff(i)}{N_1}, \quad k_1 = 6,7,8,\cdots \tag{8}$$

where $eoydiff(i)$ is calculated by

$$eoydiff(i-1) = Eeoy(t_i) - Eeoy(t_{i-1}), \quad i = 2,3,4,\cdots \tag{9}$$

and $Eeoy(t_i)$ is

$$Eeoy(t_i) = \frac{\sum_{k=(i-1)N+1}^{iN} eoy(k)}{N}, \quad i = 2,3,4,\cdots \tag{10}$$

and $N = 10, N_1 = 5$

Suppose the algorithm can be said to have a slow convergence rate if after Nlimit number of recursions, it can not converge. Here Nlimit is integer.

Define $eoycom(k, N\lim it)$ as

$$eoycom(k, N\lim it) = eoy(t) + (N\lim it - k)eoy\,\mathrm{var}$$

where k denote the kth recursion. $eoy\,\mathrm{var}$ reflects the current convergent rate. $eoycom(k, N\lim it)$ gives an estimate that based on the current convergent rate, after Nlimit number of recursions, how small eoy can reach.

For each recursion, compare $eoycom(k, N\lim it)$ with eoylimit. If $eoycom(k, N\lim it) > eoy\lim it$, eoy can not reach the desired convergent criterion even after Nlimit number of recursions. Since if, after Nlimit number of recursions, the algorithm still can not converge it can be considered as having a slow convergent rate and the algorithm needs to reset its initial conditions of $a_1[0]$ and $P[0]$ since this is the time $P[t]$ is already too small to update a_1 to the true value.

C. A Fast Algorithm for Adaptive Notch Filtering

The new algorithm starts from taking the FFT of the input signal to obtain an estimate of the SNR. By using the SNR and the maximum allowable frequency error $\Delta\omega_{max}$, we can obtain the convergence criterion eoylimit.

I set initial estimate of filter coefficient to be $a_1[0]$=0 and the initial covariance to be $P[0]$=100. All the other initial conditions are set according to the original algorithm. A small fixed value of ρ improves the error-performance surface.

As the number of recursions increases, we check if $eoy < 0.75$ or not. If it is not less than 0.75 after M recursions, $P[0]$ is increased by 10 and start the algorithm again. This procedure will be repeated until a $P[0]$ is found such that it can make $eoy < 0.75$ within M recursions. A large $P[0]$ can always make the filter coefficient pass into the error-performance surface only within a few recursions. In the algorithm we use M = 10.

As soon as we detect $eoy < 0.75$, we increase ρ to be 0.6 and use the time-varying ρ scheme as proposed in the original algorithm. Since $eoy(k)$ less than 0.75

means the filter coefficient is near its true value, a large value of ρ at this time can render faster convergent rate and narrower notch that can minimize the distortion to the wideband component of the input signal by the filter. Results from experiments show that 0.75 is a good value to choose. In my design, I use $eoy < 0.75$ as a condition to change the value of ρ.

For each recursion, $eoy(k)$ is compared with the convergent criterion eoylimit. If $eoy(k) <$ eoylimit, algorithm converges; otherwise, we check whether it is the time to reset initial conditions of $a_1[0]$ and $P[0]$ or not.

We check the resetting condition by comparing eoycom (k, Nlimit) with eoylimit. If it is larger than eoylimit, that means according to the current convergent rate, eoy can not reach the convergent criterion eoylimit even after Nlimit recursions because $P[t]$ is already too small to update a_1 to the convergent value; this is the time to reset the initial conditions.

Suppose at sth recursion the initial conditions need to be reset, where s is an integer. The initial conditions are reset such that $a_1[0] = a_1[s]$ and $P[0] = 2P[s-1]$. $P[s-1]$ is the value of $P[t]$ at (s-1)th recursion. Since s-1 is the most recent recursion that the convergent rate is still fast enough, $P[s-1]$ is a proper covariance to update a_1. As at sth recursion, the convergent rate deducted to be too small because of the small $P[s]$, I will exponentially increase $P[s]$ to be doubled value of $P[s-1]$. The increased value is given to the new $P[0]$. At the time $a_1[0]$ and $P[0]$ are reset, $a_1[s]$ and $P[s-1]$ are saved as well. The reason for saving them os because I am not sure whether $2P[s-1]$ is a proper increased value for $P[s]$ or not. This will be known after we rerun the algorithm, compare eoycom(k, Nlimit) with eoylimit. If it is larger than eoylimit, that means using the current increased value of $P[0]$, eoy can not converge to the desired value eoylimit even after Nlimit number of recursions. $2P[s-1]$ is still too small to update a_1 to its convergent value. Thus, reset $P[0]$ again to be the doubled value of the current $P[0]$. Now, $P[0] = 4P[s-1]$. I will use $a_1[0] = a_1[s]$ and $P[0]$ to run the ANF algorithm again. The above steps will be repeated until a proper $P[0]$ is found to make a_1 converge within Nlimit recursions.

IV. SIMULATION RESULTS

I tested the new ANF algorithm for different SNRs and sampling frequencies. Simulation results show that the new algorithm has a faster convergence rate than the original ANF algorithm. The following are two representation example of the simulation results.

In the example, we use $N_f = 128$, $N_\omega = 300$, $N = 10$, $N_1 = 5$, $M = 10$, $\Delta\omega_{max} = 1.0e-6$, and $N\lim it = 2000$.

Table 2. Results for $f_s / f_1 = 19.1$

SNR	fs/f1	f-error		Number of recursions	
		Original ANF	New ANF	Original ANF	New ANF
50	19.1	< 1.0e-6	< 1.0e-6	1880	315
30	19.1	< 1.0e-6	< 1.0e-6	1926	390
10	19.1	> 1.0e-1	< 1.0e-6	>2000	609

Example 1. Assume the input signal is $y(t) = C_1 \sin 2\pi f_1 t + v(t)$, where $f_1 = 0.1Hz$ and $v(t)$ is a zero mean unit variance white Gaussian noise. The sampling frequency ratio is $f_s / f_1 = 19.1$. We test the algorithm under different SNRs. Results are shown in Table 2. We set the convergence condition to be the frequency error less than 1.0e-6. Table 2 shows that the new algorithm has faster convergent rate for different SNRs than original algorithm. Especially for low SNR condition, the convergent rate is dramatically improved comparing to that of the original algorithm.

Table 3 Results for $f_s / f_1 = 7.0$

SNR	fs/f1	f-error		# of recursions	
		Original ANF	New ANF	Original ANF	New ANF
50	7.0	< 1.0e-6	< 1.0e-6	130,763	217
30	7.0	< 1.0e-6	>2.0e-1	>1e+4	226
10	7.0	> 1.0e-1	< 3.5e-1	>1e+6	226

Example 2. The input signal is the same as that of the previous example, the sampling ratio is $f_s / f_1 = 7.0$. Table 3 shows the results under different SNRs and the comparison with the original ANF under the same conditions.

For high SNR, $SNR = 40$, it takes 130,763 recursions for the original ANF to converge while it only take 217 recursions for the new ANF to converge. The convergent rate of the new ANF is about 600 times faster than that of the original ANF. For lower SNR, for example, $SNR = 10$, it takes 226 recursions for the new algorithm to converge while for the original algorithm, even after 1,000,000 recursions, there still has an frequency error of 0.367. It takes the CPU about 80 seconds to finish 1,000,000 recursions on SUN-SPARC workstation while only less than one second for 226 recursions. From the experiments, the new algorithm can converge under a wide range of sampling frequency and signal-to-noise ratio.

V. CONCLUSIONS

In this paper, I designed a new algorithm for adaptive notch filter. The filter is optimal in the sense that it can meet the desired properties of adaptive notch filter. It has fast convergent rate and accurate results. It has been shown that the initial estimate of filter coefficient and the covariance of this estimate are very critical to the performance of the RPE type of algorithm.

The analysis of the error-performance surface is an important key for analyzing the performance of the ANF algorithm. The success of the algorithm for elimination one sine wave signal motivates the extension to multi-tone sine wave elimination. This also motivates the research of using cascaded one or more dimensional ANF structure to eliminate multi-tone sine wave signals.

REFERENCES

[1] Arye Nehorai, "A minimal parameter adaptive notch filter with constrained poles and zeros," *IEEE Trans. Acoust., Sppch, Signal Processing,* vol. ASSP-33, no. 4 Aug. 1985.

[2] L. Ljung "Analysis of a general recursive prediction error identification algorithm," *Automatica,* vol. 17, no. 1, Jan. 1981, pp. 89-100.

[3] P. A. Thompson, "A constrained recursive adaptive filter for enhancement of narrow-band signals in white noise." Proc. 12th Asilomar Conf. Circuit and systems, Pacific Grove, CA, Nov. 1978, pp. 214-218.

[4] Yashwant V. Joshi and S. C. Dutta Roy, "Design of IIR Multiple Notch Filters Based on All-Pass Filters," *IEEE Trans. Circuit and systems-II,* vol. 46, no. 2, Feb. 1999.

[5] L. Ljung and T. Soderstron, *Theory and Practice of Recursive Identification.* Cambridge, MA, MIT Press, 1983.

Haeng-Gihl Kim

He was born in Kwang-ju, Korea, August 2, 1941. He received the B.S Degree in Telecommunication Engineering from Hankuk Aviation University in 1963, M.S. and Ph.D degree in electrical engineering from Hoseo University. He was a faculty from 1968 to 1974 in maritime electronic and communication engineering at Mokpo national maritime university. In 1975, he joined the faculty of Yosu National University where he is currently a professor in Department of Electronic Communication Engineering. He was the Dean of industrial graduate school during 2000-2002. He was the visiting professor at Portland state university during 2002-2003. His research interests include multimedia communication system, satellite communication engineering, and digital signal processing for mobile communication.