Complexity Reduction of the NLMS Algorithm via Selective Coefficient Update

T. Aboulnasr and K. Mayyas

***Abstract*—This correspondence proposes an algorithm for partial update of the coefficients of the normalized least mean square (NLMS) finite impulse response (FIR) adaptive filter. It is shown that while the proposed algorithm reduces the complexity of the adaptive filter, it maintains the closest performance to the full update NLMS filter for a given number of updates. Analysis of the MSE convergence and steady-state performance for independent and identically distributed (i.i.d.) signals is provided for the extreme case of one update/iteration.**

I. Introduction

Acoustic echo cancellation is a common application of adaptive filtering. The adaptive acoustic echo canceler requires several hundred taps in order to achieve satisfactory echo suppression. Even with the use of a NLMS-based echo canceler, the huge processing power required to implement such an echo canceler is beyond the capabilities of many digital signal processing (DSP) chips [1].

Several algorithms were proposed to reduce the computational cost of the NLMS algorithm. Such algorithms include the periodic NLMS algorithm and the partial update algorithms [2]–[4], where only a predetermined subset of the coefficients is updated every iteration, as well as the recent single update Max-NLMS algorithm [5], [6]. Inevitably, the penalty incurred when using these algorithms is a decrease in convergence speed, which is proportional to the reduction in complexity of the *coefficient updates*. This can sometimes be a major drawback in their implementation in the case of long impulse responses.

The algorithm proposed here is a member of the family of adaptive algorithms that updates a portion of their coefficients at each iteration. Those selected coefficients are the ones with larger magnitude gradient components on the error surface. The algorithms in [2]–[4], however, choose those coefficients, each iteration (or block of iterations), in a prespecified fixed way. The proposed algorithm adds a maximum of $2\log_2(N)+2$ (where N is the adaptive filter length) comparison operations over the computational overhead of the algorithms in [2]–[4].

II. Proposed Gradient-Based Partial Update Algorithm (M-Max Algorithm)

For LMS-type algorithms, it is generally noted that when updating all coefficients of the adaptive filter, the contribution to the error by some coefficients is small, whereas other coefficients have larger error contributions. In other words, the error function is not equally sensitive to variations in all coefficients. Accordingly, even if "less important" coefficients are not updated at a given iteration, the algorithm performance will be marginally affected. The sensitivity of the performance error to the individual *coefficient* at each iteration depends on two factors: the shape of the mean-square error (MSE) surface and the location of that coefficient at that instance relative to the bottom of the MSE surface. This sensitivity is reflected in the steepness of the gradient vector components. Consequently, a simple and direct criterion for the selection of coefficients to be updated is based on the magnitude of the corresponding gradient estimate in the direction of every individual coefficient at a given iteration. The instantaneous gradient estimate in the direction of the ith coefficient is $-2e(n)x(n-i+1)$, where $1 \le i \le N$. Clearly, all gradient components involve the quantity $-2e(n)$. Thus, we propose to select coefficients associated with the M largest values of $|x(n-i+1)|$ $1 \le i \le N$ for updating. The proposed algorithm is stated as follows. At each iteration, M out of the N coefficients are updated. Those M coefficients are the ones associated with the M largest $|x(n-i+1)|$, $i=1,\ldots,N$, at that iteration. The proposed M-Max NLMS algorithm update equation can be written as in (1), shown at the bottom of the next page. To investigate the performance of the proposed algorithm, we compare it with the full-update NLMS algorithm for the same μ. Define, at instant n, the error $e(n)=d(n)-\mathbf{X^T}(n)\mathbf{W}(n)$ and the *a posteriori* error $e_p(n)=d(n)-\mathbf{X^T}(n)\mathbf{W}(n+1)$ [7]. It is easy to see that $e_p(n)$ can be expressed in terms of $e(n)$ as

$$e_p(n)=e(n)-\sum_{j=1}^{N}x(n-j+1)\Delta w_j \tag{2}$$

where $\Delta w_j = w_j(n+1)-w_j(n)$. For the full update NLMS algorithm, Δw_j is given by $\Delta w_j=\frac{\mu}{\mathbf{X^T}(n)\mathbf{X}(n)}e(n)x(n-j+1), j=1,2,\ldots,N$, and then, $e_p^2(n)=e^2(n)(1-\mu)^2$. Note that for the full update NLMS algorithm $e_p(n)=0$ when $\mu=1$. For the proposed algorithm, we update only a set of M out of the N coefficients; $w_{i_1},w_{i_2},\ldots,w_{i_M}$. Δw_j is given by (3), shown at the bottom of the next page. Then, we have

$$e_p^2(n)=e^2(n)\left(1-\frac{\mu}{\mathbf{X^T}(n)\mathbf{X}(n)}\sum_{j=1}^{M}x^2(n-i_j+1)\right)^2. \tag{4}$$

It is clear that for $\underline{M=N}$, (4) reduces to that of the full update NLMS. In other words, as $\sum_{j=1}^{M}x^2(n-i_j+1)$ approaches $\mathbf{X^T}(n)\mathbf{X}(n)$, the convergence speed of the partial update algorithm approaches that of the full update NLMS. In the sequential NLMS (SNLMS) algorithm [2], [4], these M coefficients are chosen such that $(n-i_j+1) \bmod \frac{N}{M}=0$. In the sequential block NLMS [3] (SBNLMS), they are chosen such that $i_j=(n\times M)+j$. In both cases, the relative value of $\sum_{j=1}^{M}x^2(n-i_j+1)$ and $\mathbf{X^T}(n)\mathbf{X}(n)$ is not considered when selecting the set of M coefficients. However, in the proposed algorithm, the M coefficients to be updated are chosen to correspond to the M largest $x^2(n-j+1)$, $j=1,2,\ldots,N$, thus resulting in the largest $\sum_{j=1}^{M}x^2(n-i_j+1)$ at a given iteration n. Clearly, this results in the smallest possible $e_p^2(n)$ for the proposed algorithm and a given M, i.e., *the closest possible performance to the full-update NLMS.*

In terms of multiplications/additions, the proposed algorithm has the same complexity overhead of the SNLMS or SBNLMS algorithm. It should be noted that at each iteration, running the sorting procedure in descending order of $|x(n-i+1)|$, $i=1,\ldots,N$ is required. The M coefficients that belong to the first M elements of the sorted vector are updated. In [8], a fast algorithm for sorting of a sliding window

Manuscript received August 6, 1997; revised September 29, 1998. The associate editor coordinating the review of this paper and approving it for publication was Prof. Chi Chung Ko.

T. Aboulnasr is with the School of Information Technology and Engineering, University of Ottawa, Ottawa, Ont., Canada K1N 6N5.

K. Mayyas is with the Department of Electrical Engineering, Jordan University of Science and Technology, Irbid, Jordan.

Publisher Item Identifier S 1053-587X(99)03264-X.

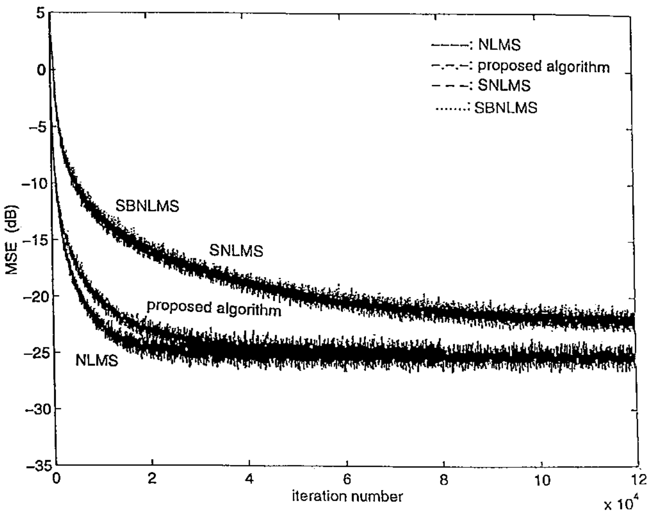

Fig. 1. Comparison of MSE between the full update NLMS, SBNLMS, SNLMS, and proposed Max-NLMS algorithm ($M = 25$) for the correlated input case SNR $\simeq$ 51 dB.

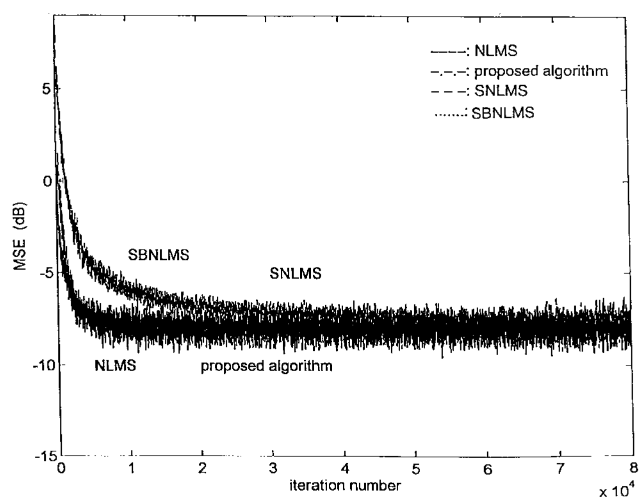

Fig. 2. Comparison of MSE between the full update NLMS, SBNLMS, SNLMS and proposed Max-NLMS algorithm ($M = 25$) for the correlated input case SNR $\simeq$ 21 dB.

of arbitrary N elements is proposed. The algorithm SORTLINE requires, *at most*, $2\log_2(N) + 2$ comparison operations per sample time. Consequently, the proposed partial update algorithm needs an extra $2\log_2(N) + 2$ comparisons in addition to multiplications and additions required by the SNLMS or SBNLMS algorithms. For large N, the savings of $(N - M)$ multiplications and $(N - M)$ additions will exceed the additional complexity of $2\log_2(N)$ comparison operations.

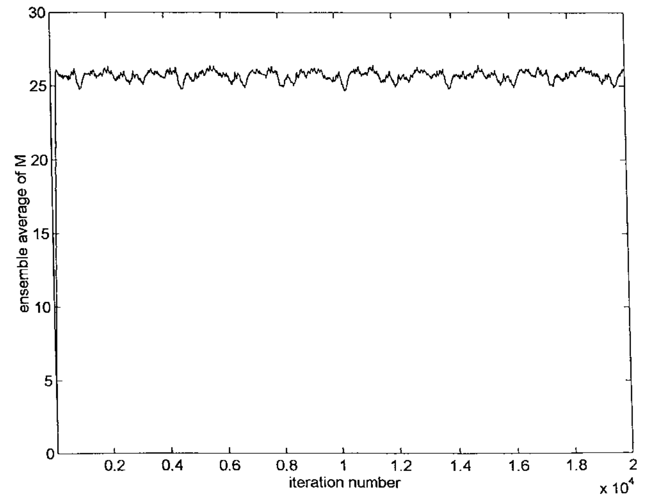

Fig. 3. Ensemble average of M selected per iteration by the modified proposedalgorithm with time-varying M for correlated inputcase with $PM_0 = 0.5$ and SNR $\simeq$ 51 dB.

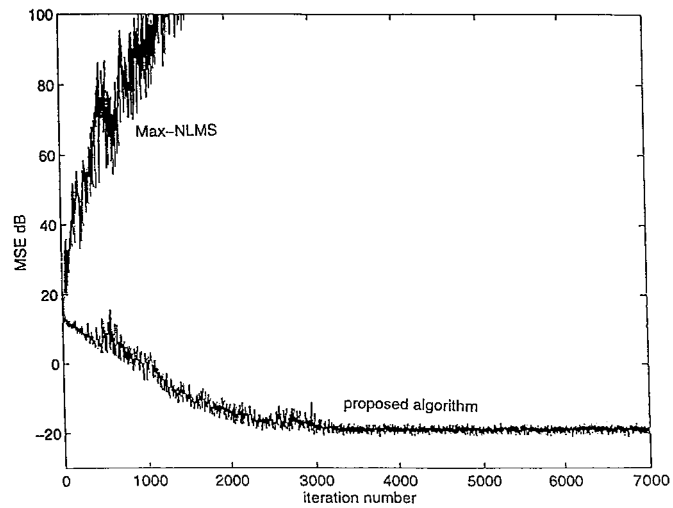

Fig. 4. Comparison of MSE between the Max-NLMS and proposed algorithm ($M = 1$) with $N = 10$ and a skewed binary distribution input.

A. Mean Square Analysis of the M-Max NLMS

In this section, we will study the convergence properties of the proposed algorithm. To make the analysis tractable, we consider the case for $M = 1$. Our objective is to show that the algorithm is guaranteed to converge for the extreme case of $M = 1$ for i.i.d. stationary zero-mean input (provided μ is chosen in the stability region) and that it will converge to the same steady-state error as the full update NLMS (we already know it converges for $M = N$).

$$w_i(n+1) = \begin{cases} w_i(n) + \dfrac{\mu}{\mathbf{X}^{\mathrm{T}}(n)\mathbf{X}(n)} e(n)x(n-i+1), & \text{if } i \text{ corresponds to one of the first} \\ & M \text{ maxima of } |x(n-i+1)|,\ i = 1, \ldots, N \\ w_i(n), & \text{otherwise} \end{cases} \tag{1}$$

$$\Delta w_j = \begin{cases} 0, & \text{if } j \neq i_k, \quad k = 1, 2, \ldots, M \\ \dfrac{\mu}{\mathbf{X}^{\mathrm{T}}(n)\mathbf{X}(n)} e(n)x(n-j+1), & \text{if } j = i_k, \quad k = 1, 2, \ldots, M \end{cases} \tag{3}$$

Assuming that $x(n)$ is a stationary zero-mean i.i.d. sequence and defining the coefficients error vector $\mathbf{V}(n) = \mathbf{W}(n) - \mathbf{W}^*$, for $M = 1$ and $N \geq 2$, (1) becomes (5), shown at the bottom of the page, where we used the relation $d(n) = \mathbf{X}^{\mathbf{T}}(n)\mathbf{W}^* + e^*(n)$, where $\mathbf{W}^*$ is the optimal weight vector, and $e^*(n)$ is a zero mean independent disturbance signal. Following the same approach in [5] and [6], and assuming that for high order adaptive filters $\mathbf{X}^{\mathbf{T}}(n)\mathbf{X}(n) \approx N\sigma_x^2$, it can be easily seen that the autocorrelation matrix governing the evolution of the mean error weight vector in (5) is $\mathbf{R} = \frac{\sigma_x^2}{N}\mathbf{I}$, where $\mathbf{I}$ is the $N \times N$ identity matrix. Note that $\mathbf{R}$ is symmetric and positive definite. This proves the mean convergence of the proposed algorithm $(M = 1)$ with a proper choice of the step size, irrespective of the type of the probability density function of the i.i.d. input signal. To find a limit on the step size of the proposed algorithm, we consider the mean square error analysis of the proposed algorithm. Along with the previous assumption on the input signal, we use the common independence theory. Let max be the index of the coefficient to be updated at time instant n, i.e., $w_{max}(n)$ is the coefficient to be updated. Then, from (5), it can be shown that the difference equation of the mean of the maxth coefficient for zero mean i.i.d. input signal is

$$E\{V_{max}^2\}(n+1) = \left(1 - 2\acute{\mu}\sigma_x^2 + \acute{\mu}^2\eta\right)E\{V_{max}^2\}(n) + \acute{\mu}^2\sigma_x^4 \sum_{j=1, j\neq max}^{N} E\{V_j^2(n)\} + \acute{\mu}^2\sigma_x^2\epsilon_{min} \quad (6)$$

where $\eta = E\{x^4(n)\}$, $\epsilon_{min} = E\{e^{*2}(n)\}$, and $\acute{\mu} = \frac{\mu}{N\sigma_x^2}$. For a zero mean independent Gaussian input signal, $\eta = 3\sigma_x^4$. In [5] and [6], it is shown that given the assumptions used here, the sequence of indices of updated coefficients is a Markov process with a uniform probability of selecting any coefficient for updating. Accordingly, we have $E\{V_{max}^2(n)\} = E\{V_j^2(n)\} = C(n)$, $\forall j = 1, 2, \ldots, N$. The probability of updating any coefficient at each sample time is $\frac{1}{N}$ [5], [6]; therefore

$$C(n+1) = \frac{1}{N}\left((N-1)E\{V_j^2(n+1)\} + E\{V_{max}^2(n+1)\}\right) \quad j \neq max. \quad (7)$$

Note that for $\forall j \neq max$, $E\{V_j^2(n+1)\} = E\{V_j^2(n)\} = C(n)$. Then, substituting (6) in (7) results in

$$C(n+1) = \left(1 - 2\frac{\acute{\mu}}{N}\sigma_x^2 + \frac{\acute{\mu}^2}{N}\left[\eta + (N-1)\sigma_x^4\right]\right)C(n) + \frac{\acute{\mu}^2}{N}\sigma_x^2\epsilon_{min}. \quad (8)$$

To ensure the convergence of (8) and noting that $\acute{\mu} = \frac{\mu}{N\sigma_x^2}$, the step size μ of the proposed algorithm should be bounded by

$$0 < \mu < \frac{2N\sigma_x^4}{\eta + (N-1)\sigma_x^4}. \quad (9)$$

TABLE I
COMPARISON OF THEORETICAL AND EXPERIMENTAL PROPOSED ALGORITHM ($M = 1$) STEADY STATE EXCESS MSE

		Predicted $\epsilon_{ex}(\infty)$ (dB) Eq.(10)		Measured $\epsilon_{ex}(\infty)$ (dB)	
μ	ϵ_{min}	N=50	N=5	N=50	N=5
0.8	0.0001	-41.64	-40.41	-41.03	-41.25
0.8	0.01	-21.64	-20.41	-21.07	-21.24
0.8	1	-1.64	-0.41	-1.07	-1.15
0.4	0.0001	-45.97	-45.56	-45.58	-45.45
0.4	0.01	-25.97	-25.56	-25.61	-25.41
0.4	1	-5.97	-5.56	-5.61	-5.38
0.08	0.0001	-53.79	-53.72	-53.32	-53.25
0.08	0.01	-33.79	-33.72	-32.93	-33.21
0.08	1	-13.79	-13.72	-12.94	-13.40

The excess mean square error (MSE) $\epsilon_{\text{ex}}(n)$ of the adaptive filter when the input is an i.i.d. signal is given by $\epsilon_{\text{ex}}(n) = \sigma_x^2 \sum_{j=1}^{N} E\{V_j^2(n)\}$. Assuming that the step size of the proposed algorithm is chosen such that MSE convergence is guaranteed, and using (8), the steady-state excess MSE $\epsilon_{\text{ex}}(\infty)$ of the proposed algorithm has the form

$$\epsilon_{\text{ex}, M=1}(\infty) = \frac{\mu\epsilon_{min}}{2 - \frac{\mu}{N\sigma_x^4}\left(\eta + (N-1)\sigma_x^4\right)}. \quad (10)$$

In the case when all coefficients are updated at each iteration $(M = N)$ (i.e., full update NLMS algorithm), it can be shown that the steady-state excess MSE for high filter orders is given by $\epsilon_{\text{ex}, M=N}(\infty) = \frac{\mu\epsilon_{min}}{2 - \frac{\mu}{N\sigma_x^4}(\eta + (N-1)\sigma_x^4)}$ [9]. This shows that the case of $M = 1$ and the full update NLMS provide similar misadjustment when applied under same conditions and used with the same step-size value. Given that the algorithms are using FIR filters of the same order, and the input signal is of persistent excitation, it automatically implies they have both reached the same solution.

III. SIMULATIONS

In the first example, the proposed algorithm is compared with the full-update NLMS algorithm [1], the SNLMS algorithm [2], [4], and the SBNLMS algorithm [3] in an acoustic echo cancellation application. The echo path is that of an anechoic room of 200 taps measured at 8 kHz sampling rate. For the proposed algorithm, SNLMS and SBNLMS $M = 25$ coefficients are updated every iteration. Perfect modeling of the echo path is assumed, i.e., $N = 200$. A white noise of 0.0001 variance is added to the desired signal. The input signal is a highly correlated one generated by passing a zero mean white Gaussian signal with unity variance through the filter $H(z) = \frac{1}{1 - 1.58z^{-1} + 0.81z^{-2}}$ (SNR $\simeq 51$ dB). For the full-update NLMS algorithm $\mu = 1$, while the proposed algorithm, the SBNLMS and SNLMS algorithms have $\mu = 0.8$ chosen to achieve the same steady-state MSE of the NLMS algorithm. The superiority of the performance of the proposed algorithm compared with the SBNLMS

$$V_i(n+1) = \begin{cases} \left(1 - \frac{\mu}{\mathbf{X}^{\mathbf{T}}(n)\mathbf{X}(n)}x^2(n-i+1)\right)V_i(n) \\ - \frac{\mu}{\mathbf{X}^{\mathbf{T}}(n)\mathbf{X}(n)} \sum_{j=1, j\neq i}^{N} x(n-i+1)x(n-j+1)V_j(n) \\ + \frac{\mu}{\mathbf{X}^{\mathbf{T}}(n)\mathbf{X}(n)}x(n-i+1)e^*(n) \\ \quad \text{if } i \text{ corresponds to the maximum of } |x(n-i+1)|, \quad i = 1, \ldots, N \\ V_i(n), \quad \text{otherwise} \end{cases} \quad (5)$$

and SNLMS is obvious from Fig. 1. Note that the performance of the proposed algorithm is quite comparable with the NLMS algorithm. The slow convergence speed of the SBNLMS and SNLMS algorithms relative to the NLMS algorithm in Fig. 1 is not surprising since the convergence of the SBNLMS and SNLMS algorithms is expected to be eight times slower than the NLMS algorithm. On the other hand, the strategy of adapting the *set* of coefficients to be updated has minimized the loss in performance compared with the full-update case. This example is repeated for lower SNR using a disturbance noise of 0.1 variance (SNR $\simeq$ 21 dB). Fig. 2 illustrates that the proposed algorithm maintains its superior performance under low SNR conditions.

The choice of M is generally limited by the allowable complexity. However, the algorithm can be expanded to maintain a predetermined performance, irrespective of the input signal condition. We define the instantaneous performance measure $PM(n) = \sum_{j=1}^{M} \frac{x^2(n-i_j+1)}{\mathbf{X}^{\mathbf{T}}(n)\mathbf{X}(n)}$ and require that M be selected at each instant n such that $PM(n) \approx PM_0$, where $0 < PM_0 \leq 1$, and $x(n-i_j+1), j = 1, \ldots, M$ are the largest M values. Accordingly, $(1-\mu PM(n))^2 \approx (1-\mu PM_0)^2$, and the reduction in the *a posteriori* error $e_p(n)$ after adaptation relative to that of $e(n)$ previous is kept approximately constant at each iteration. Note that $PM_{\mathrm{NLMS}}(n) = 1$, and therefore, PM_0 can be indicative of the relative performance of the modified algorithm compared with the NLMS. The modified algorithm is used in the above example (SNR $\simeq 51$ dB) with $\mu = 1$ and $PM_0 = 0.5$, and the ensemble average of M is plotted in Fig. 3. Fig. 3 indicates that for this particular distribution, $E\{PM(n)\}$ is a constant that depends only on the value of M.

In the second example, we compare the performance of the proposed M-Max NLMS algorithm for $M = 1$ with the Max-NLMS algorithm described in [5] and [6]. Both algorithms will pick identical coefficients to update (coefficient with maximum $x(n-i+1)$). However, the update term is slightly different (compare [5, Eq. (5)] for the Max NLMS and (1) for the proposed algorithm). It is shown in [5] and [6] that the Max NLMS algorithm diverges for some input signals with certain nonsymmetric distribution for any step size. Our first example tests the proposed M-Max NLMS to verify its convergence in this case. The input signal used in [5] and [6] is a continuous approximation to a skewed binary distribution with a p.d.f. given by

$$p(x)_X = \begin{cases} \frac{1}{\delta}(0.5-\alpha), & \text{if } 1-\beta < x < 1-\beta+\delta \\ \frac{1}{\delta}(0.5+\alpha), & \text{if } -1-\beta < x < -1-\beta+\delta \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where $0 < \delta \ll \beta$. The unknown system is an FIR system with 10 coefficients, and the FIR adaptive filter has $N = 10$. A zero mean white Gaussian noise of 0.01 variance is added to the desired signal. The proposed algorithm $(M = 1)$ and the Max-NLMS are used with $\mu = 0.4$ and $\mu_{\mathrm{Max\text{-}NLMS}} = 0.24$, respectively. The input signal parameters are $\alpha = 0.02$, $\beta = 0.02$, and $\delta = 0.001$ [5], [6]. As expected from the analysis in Section II, Fig. 4 shows that the proposed algorithm with $M = 1$ and i.i.d. input maintains stability and achieves convergence even with asymmetric distribution, whereas the Max-NLMS algorithm diverges for input signals with this distribution. When $\alpha = 0$ and $\beta = 0$, both the Max-NLMS and the proposed algorithm converge without stability problems.

Next, we compare results for the steady-state MSE $\epsilon_{\mathrm{ex}}(\infty)$ obtained from theoretical analysis via (10) with results of simulations for $N = 5$ and $N = 50$ and different values of μ and ϵ_{min}. The values of μ and ϵ_{min} are selected to cover several experimental conditions of low and high SNR, as well as small and large step sizes. The input is a zero mean white Gaussian signal with unity variance. The proposed algorithm is used with $M = 1$. The unknown system has the same length of N in each case. Table I shows that (10) predicts very closely the actual $\epsilon_{\mathrm{ex}}(\infty)$, as obtained from simulations.

References

[1] P. S. R. Diniz, *Adaptive Filtering, Algorithms and Practical Implementation*. Norwell, MA: Kluwer, 1997.

[2] D. Messerschmitt *et al.*, *Digital Voice Echo Canceler with TMS32020*, Digital Signal Processing Applications with the TMS3C20 Family, Texas Instrum., Inc. Dallas, TX, 1986.

[3] S. M. Kuo and J. Chen, “Multiple microphone acoustic echo cancellation system with the partial adaptive process,” *Digital Signal Process.*, vol. 3, pp. 54–63, 1993.

[4] S. C. Douglas, “Adaptive filters employing partial updates,” *IEEE Trans. Circuits Syst. II*, vol. 44, pp. 209–216, Mar. 1997.

[5] ——, “Analysis and implementation of the Max-NLMS adaptive filter,” in *Proc. 29th Asilomar Conf. Signals Systs. Comput.*, Pacific Grove, CA, Oct. 1995, vol. 1, pp. 659–663.

[6] ——, “Analysis of the Max-NLMS adaptive filter,” submitted for publication.

[7] S. Shaffer and C. S. Williams, “Comparison of the LMS, αLMS, and the data reusing LMS algorithm,” in *Proc. ICASSP*, 1984, pp. 260–264.

[8] I. Pitas, “Fast algorithms for running ordering and max/min calculation,” *IEEE Trans. Circuits Syst.*, vol. 36, pp. 795–804, June 1989.

[9] K. Mayyas and T. Aboulnasr, “The leaky LMS algorithm: MSE analysis for Gaussian data,” *IEEE Trans. Signal Processing*, vol. 45, pp. 927–934, Apr. 1997.

A General Efficient Method for Chaotic Signal Estimation

Ling Cong, Wu Xiaofu, and Sun Songgeng

***Abstract*— This correspondence presents a general, computationally efficient method for chaotic signal estimation based on the connection between the symbolic sequence and the initial condition of a chaotic system. The performance of the method in white Gaussian noise is evaluated. The new method is asymptotically unbiased and attains the Cramér–Rao lower bound at high signal-to-noise ratios.**

***Index Terms*—Chaos, estimation, Gaussian noise, maximum likelihood estimation, signal processing.**

I. Introduction

Chaotic signals are appealing candidates for use in many engineering contexts. In order to exploit chaotic signals in engineering applications, there is a need for robust and efficient algorithms for estimation of these signals in the presence of noise. A variety of algorithms have been proposed for estimating a chaotic signal $x(n)$ that is embedded in white Gaussian noise $w(n)$ [1]–[5]. The data

Manuscript received November 6, 1997; revised September 9, 1998. The associate editor coordinating the review of this paper and approving it for publication was Dr. Ali H. Sayed.

The authors are with the Nanjing Institute of Communications Engineering, Nanjing, China (e-mail: lingc@ice.edu.cn).

Publisher Item Identifier S 1053-587X(99)03254-7.