Analytical Development of the MMAXNLMS Algorithm

M. I. Haddad, K. A. Mayyas, and M. A. Khasawneh, Member of IEEE
Department of Electrical Engineering
P. O. Box 3030, Jordan University of Science & Technology
Irbid 221 10 Jordan, Fax: +962 (2) 295 018

Abstract

In this paper a recently presented adaptive algorithm with reduced complexity is analysed for the white Gaussian input case. The new analysis is extented for the proposed case where updating includes more than one component of the weight vector. The new algorithm, which updates the weights correponding to the element sizes of the data vector with the largest magnitude, is compared with the case where the updated weights are chosen randomly according to a uniform density function. Analysis is performed for both cases and the results are verified via computer simulations.

1 Introduction

Reduced complexity adaptive algorithms are becoming more and more important. They are particularly needed in high speed communication systems and in systems which require adaptive filters with large orders (thousands of taps). Several reduced-complexity algorithms have been proposed [3], but these algorithms reduce the speed of convergence in proportion to the reduction in the computational complexity. Recently, a new algorithm called the MMAXNLMS algorithm [1] was introduced which reduces the computational complexity but maintains the convergence speed close to the original one when all the elements in the weight vector are updated. The analysis for this algorithm is extended here for the case where updating includes more than one weight vector. Also, the performance of this algorithm is compared with the case where the elements in the weight vector chosen for updating at each time are determined according to a uniform random process. This algorithm shall henceforth be called MRSNLMS [1]. In this algorithm the indices for the weights to be updated at each iteration are chosen according to a uniform random process. Analytical results are derived for this new algorithm and they are verified by means of simulations.

2 Algorithm Description and Analysis

Recently a new reduced-complexity version of the normalized LMS algorithm was presented in [1]. The reduction in the computational cost of this algorithm was achieved by updating only M weights out of the total number of the weights, N, in the weight vector. The weights updated at each iteration are chosen according to the magnitude of the corresponding elements in the input data vector. The mathematical description of this algorithm is [1]

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

where

$$e(n)=d(n)-\mathbf{X}^T(n)\mathbf{W}(n) \tag{2}$$

where $d(n)$ is the desired signal and $\mathbf{X}(n)$ is the input data vector of lenght N, all at time iteration n. The minimum error, $e^*(n)$, is defined as

$$e^*(n)=d(n)-\mathbf{X}^T\mathbf{W}^*(n) \tag{3}$$

where $\mathbf{W}^*$ is the optimal weight vector.

The weight deviation vector, $\mathbf{V}(n)=\mathbf{W}(n)-\mathbf{W}^*$, is obtained using Eq.(1) as [1]

$$V_i(n+1)=\begin{cases} v_i(n)+\frac{\mu}{X^T(n)X(n)}e(n)x(n-i+1), & \\ . & \text{if } i \text{ corresponds to} \\ . & \text{one of the first} \\ . & M \text{ maxima of } |\mathbf{X}(n)| \\ 0 & \text{, otherwise}\end{cases} \tag{4}$$

When N is large, which is usually the case, we can use the approximation, $\mathbf{X}^T(n)\mathbf{X}(n)\simeq N\sigma_x^2$. Using a new variable $\alpha=\frac{\mu}{N\sigma_x^2}$ and utilizing the well-known independance assumptions [4], it is straightforward to show that

$$E[V_{(i)}^2(n+1)]=\left(1-2\alpha E[x_{(i)}^2]+\alpha^2 E[x_{(i)}^4]\right)E[V_i^2(n)]$$

$$+\alpha^2 E[x_{(i)}^2]\xi_{min}+\alpha^2(N-1)\sigma_x^2 E[V_i^2(n)]E[x_{(i)}^2] \tag{5}$$

where $\xi_{min}=E[e^{*2}(n)]$ and i corresponds to the indices of the updated weights in the weight vector. The index i takes the values $i=1,2,...,M$ in such away that $i=1$ corresponds to the weight vector component that is updated using the sample in $\mathbf{X}(n)$ with the maximum magnitude

[1]The letters RS denote Random Selection.

and $i = 2$ corresponds to the weight that is updated using the sample in $\mathbf{X}(n)$ with a magnitude that is next to maximum, and so on. For the other weights that are not updated we have

$$E[V_{(i)}^2(n+1)] = E[V_{(i)}^2(n)] \tag{6}$$

where $i = M+1, ..., N$ correspond to the indices of the weights that are not updated with no particular order. The trace of the matrix $E[\mathbf{V}(n+1)\mathbf{V}^T(n+1)]$ which represents the squared norm of the weight deviation vector is given as

$$C(n+1) = \sum_{i=1}^{M} E[V_{(i)}^2(n+1)] + \sum_{j=M+1}^{N} E[V_{(j)}^2(n+1)] \tag{7}$$

Substituting Eqs.(5),(6) in Eq.(7) we get

$$C(n+1) = \Bigg[M + \alpha^2 \sum_{i=1}^{M} E[x_i^4] + [\alpha^2(N-1)\sigma_x^2 - 2\alpha] \sum_{i=1}^{M} E[x_i^2](N-M) \Bigg] C(n) + \alpha^2 \xi_{min} \sum_{i=1}^{M} E[x_i^2] \tag{8}$$

Further simplification gives

$$C(n+1) = \left(1 + \frac{\alpha^2}{N} \sum_{i=1}^{M} E[x_i^4] + \frac{\alpha^2(N-1)\sigma_x^2 - 2\alpha}{N} \times \sum_{i=1}^{M} E[x_i^2] \right) C(n) + \alpha^2 \xi_{min} \sum_{i=1}^{M} E[x_i^2] \tag{9}$$

If we represent this equation in terms of $\overline{x}$ rather than x, where $\overline{x}$ has the same properties of x but with the variance of $\overline{x}$ normalized to 1, then the last equation will take the following equivalent form :

$$C(n+1) = \left(1 + \frac{\alpha^2 \sigma_x^4}{N} \sum_{i=1}^{M} E[\overline{x}_i^4] + \frac{(\alpha^2(N-1)\sigma_x^2 - 2\alpha)}{N} \times \sigma_x^2 \sum_{i=1}^{M} E[\overline{x}_i^2] \right) C(n) + \sigma_x^2 \alpha^2 \xi_{min} \sum_{i=1}^{M} E[\overline{x}_i^2] \tag{10}$$

Defining a new variable $S^{(p)}(k)$ where

$$S_k^{(p)} = \sum_{m=1}^{k} E[\overline{x}_{(m)}^p] \tag{11}$$

where $\overline{x}$ is a zero-mean unit variance Gaussian random variable and the subscript (m) denotes order statistics such that $\overline{x}_{(1)}$ is the maximum. Accordingly, and substituting for α, Eq.(10) becomes

$$C(n+1) = \left(1 + \frac{\mu^2}{N^3} S_M^{(4)} - \left[\frac{2\mu - \mu^2}{N^2} + \frac{\mu^2}{N^3}\right] S_M^{(2)} \right) C(n) + \frac{\mu^2 \xi_{min}}{N^2 \sigma_x^2} S_M^{(2)} \tag{12}$$

The quantities, $S_M^{(2)}$ and $S_M^{(4)}$ can only be evaluated numerically because they can not be evaluated analytically; refer to Appendix A. Following a straightforward procedure we can find a range for μ over which the stability of the algorithm is guaranteed. The stability region for μ is

$$0 < \mu < \frac{2N}{N + (1 + S_M^{(4)}/S_M^{(2)})} \tag{13}$$

and the value of μ that gives maximum convergence speed is

$$\mu^* = \frac{N}{N + 1 + S_M^4 / S_M^2} \tag{14}$$

The excess MSE for this algorithm is given as

$$\begin{aligned} \text{MSE}_{excess} &= \sigma_x^2 C(\infty) \\ &= \frac{N\mu\xi_{min} S_M^{(2)}}{((2-\mu)N - \mu)S_M^{(2)} - \mu S_M^{(4)}} \end{aligned} \tag{15}$$

where $C(\infty)$ is found from Eq.(10) to be equal to

$$C(\infty) = \frac{\mu N \xi_{min} S_M^{(2)} / \sigma_x^2}{((2-\mu)N - \mu)S_M^{(2)} - \mu S_M^{(4)}} \tag{16}$$

The results for the new algorithm, MRSNLMS, are derived from the above results by setting $S_M^{(2)} = M$ and $S_M^{(4)} = 3M$. The range for μ is given as

$$0 < \mu < \frac{2N}{N+4} \tag{17}$$

The excess MSE is also given as

$$\begin{aligned} \text{MSE}_{excess} &= \sigma_x^2 C(\infty) \\ &= \frac{N\mu\xi_{min}}{(2-\mu)N - 4\mu} \end{aligned} \tag{18}$$

The value of μ which maximizes the convergence speed is found to be

$$\mu^* = \frac{N}{N+4} \tag{19}$$

The computational complexity of the MMAXNLMS algorithm is a little bit higher than other reduced-complexity algorithms, if the same number of weights is updated. The increase in the number of computations is a result of the sorting that is needed to determine the M elements with the largest magnitude in the data vector. The additional computational count is determined by the sorting algorithm. A fast on-line sorting algorithm is found in [5]. This algorithm requires a maximum of $2\log_2(N) + 2$ comparisons per time iterations. This relative increase in computational complexity comes in such a way that the MMAXNLMS algorithm can acheive faster rates of convergence at lower computational loads compared with those of algorithms introduced in [3].

3 Computer Simulations

In this section computer simultions are used to verify analytical results. A filter of 140-taps is used throughout the simultions. The input signal is a zero-mean unit variance white Gaussian random signal. A zero-mean white Gaussian noise which has a variance of 0.1 is added to the output signal to simulate measurement noise. The results are the average of 100 independent experiments.

Fig.(1) shows $C(n)$ for the MRSNLMS algorithm for $\mu = 0.986$ and $M = 15, 30, 60,$ and 140. Fig.(2) shows $C(n)$ for the MMAXNLMS algorithm for $\mu = 0.986$ and $M = 15, 30, 60,$ and 140. In these two figures the continuous line represents experimental curves from the simulations and the dashed line represents $C(n)$ from Eq.(12). The values for $S_M^{(2)}$ and $S_M^{(4)}$ used in the generation of the analytical curves for $C(n)$ are shown in tables 1 and 2, respectively.

$S_{15}^{(2)}$	$S_{30}^{(2)}$	$S_{60}^{(2)}$	$S_{140}^{(2)}$
63.05	93.09	123.8	140

Table.(1) : The values for $S_M^{(2)}$ for $M = 15, 30, 60,$ and 140.

.

$S_{15}^{(4)}$	$S_{30}^{(4)}$	$S_{60}^{(4)}$	$S_{140}^{(4)}$
311.71	374.55	408.84	420

. Table.(2) : The values for $S_M^{(4)}$ for $M = 15, 30, 60,$ and 140.

.

Note how close the simulations are to the analytical results. Figs.(3) and (4) show the steady state level of $C(n)$ for $M = 60$ for both algorithms. Notice that $C(\infty)$ is approximately the same for the two algorithms while the MMAXNLMS algorithm achieves much faster convergence than the MRSNLMS algorithm. In both cases, the analysis and simulations are in close agreement.

The learning curves for the MRSNLMS algorithm and the MMAXNLMS algorithm are shown in Fig.(5) and Fig.(6), respectively. Notice the difference between the two algorithms in the speed of convergence, while the steady-state error is almost the same. Note, also, that the speed of convergence is less affected for the MMAXNLMS algorithm than for the MRSNLMS algorithm when a reduction in M is made.

4 Conclusions

The analysis of the MMAXNLMS algorithm has been extended to the case when updating includes more than one weight vector. Simulation results were in excellent agreement with the analysis. Analysis was, also, made for the case where updating is based on a uniform random process irrespective of the magnitude of the corresponding elements in the weight vector. The analysis for this algorithm, MRSNLMS, is a special case of the first algorithm, MMAXNLMS. In both cases, the performance of the two algorithms was very close to that predicted by analysis. It was, also, shown that the number of weights updated at each iteration, M, has negligeable effect on the steady state error for both algorithms.

5 APPENDIX

Let $y = |x|$, the CDF of y is given as

$$F_y(y) = F_x(y) - F_x(-y), y > 0 \tag{20}$$

The PDF for order statistics is [2]

$$f_{y_{(k)}}(y) = N f_y(y) \frac{(N-1)!}{(N-k)!(k-1)!} (F_y(y))^{N-k} \times$$

$$(F_y(y))^{N-k} (1 - F_y(y))^{k+1} \tag{21}$$

where N is the number of the elements in the input data vector, $\mathbf{X}(n)$. The expected value of $x_{(k)}^p$ can be expressed as

$$E[x_{(k)}^p] = \int_{-\infty}^{\infty} x^p f_{x_{(k)}}(x) dx \tag{22}$$

If p is assumed to be an even number then

$$E[x_{(k)}^p] = E[y_{(k)}^p] \tag{23}$$

Therefore, $S_M^{(p)}$ can be represented by the following expression :

$$\begin{aligned} S_M^{(p)} &= \sum_{k=1}^{M} E[y_{(k)}^p] \\ &= \sum_{k=1}^{M} \{ N \int_{-\infty}^{\infty} y^p f_y(y) \frac{(N-1)!}{(N-k)!(k-1)!} \times \\ & \quad (F_y(y))^{N-k} (1 - F_y(y))^{k+1} \} \end{aligned} \tag{24}$$

The last equation is only valid when p is an even number.

References

[1] T. Aboulnasr and K. Mayyas, "Complexity reduction of the NLMS algorithm via selective coefficient update," To Be Published.

[2] J. Pitman, *Probability,* Springer-Verlag, 1993.

[3] S. C. Douglas,"Adaptive filters employing partial updates," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, pp. 209-216, March 1997.

[4] B. Widrow, and S. Stearns, *Adaptive Signal Processing*, Englewood Cliffs, NJ: Printice-Hall, 1985.

[5] Ioannis Pitas,"Fast Algorithms for Running Ordering and Max/Min Calculation ," *IEEE Trans. Circuits and Systems*, vol. 36, no. 6, pp. 795-804, June 1989.

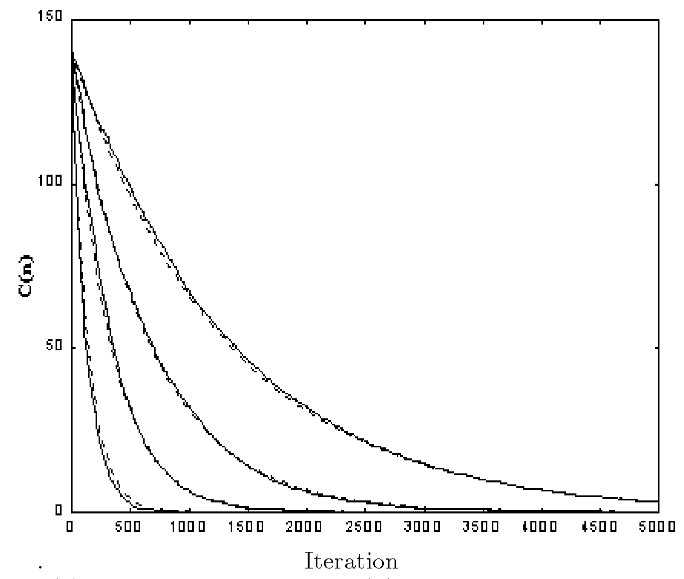

Fig.(1) : The transient part of $C(n)$ for MRSNLMS.

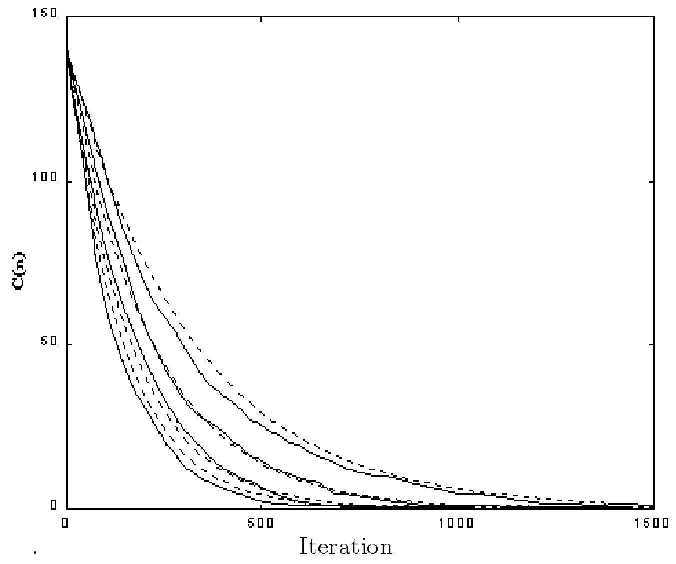

Fig.(2) : The transient part of $C(n)$ for MMAXNLMS.

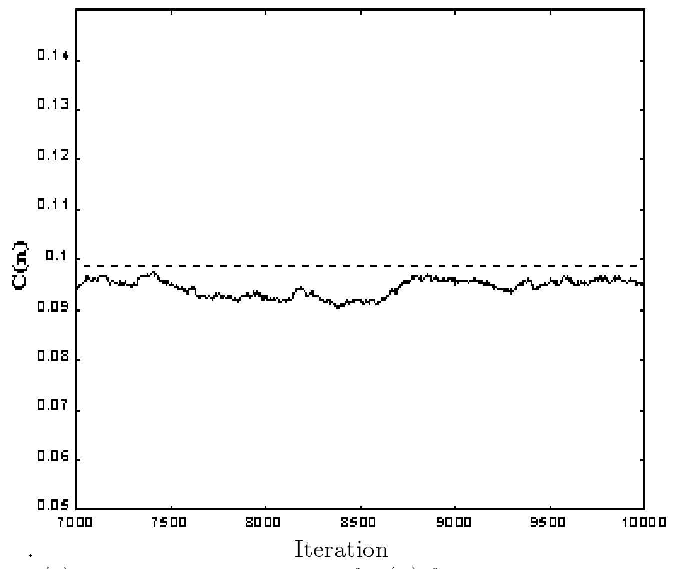

Fig.(3) : The steady state part of $C(n)$ for MRSNLMS.

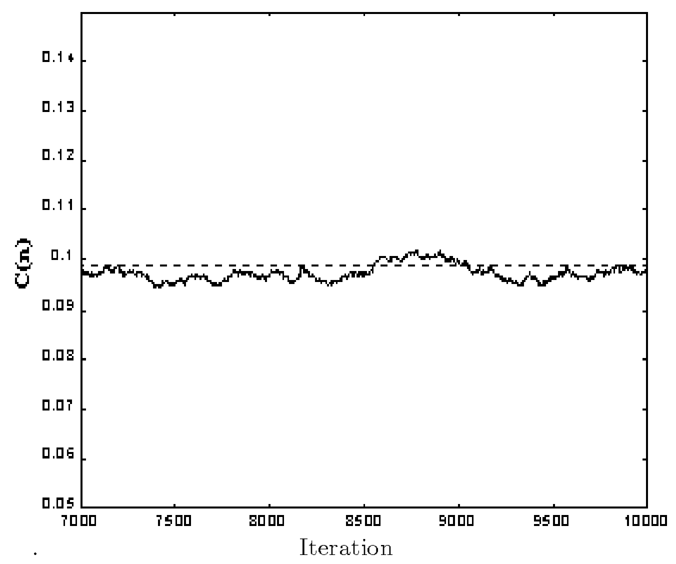

Fig.(4) : The steady state part of $C(n)$ for MMAXNLMS.

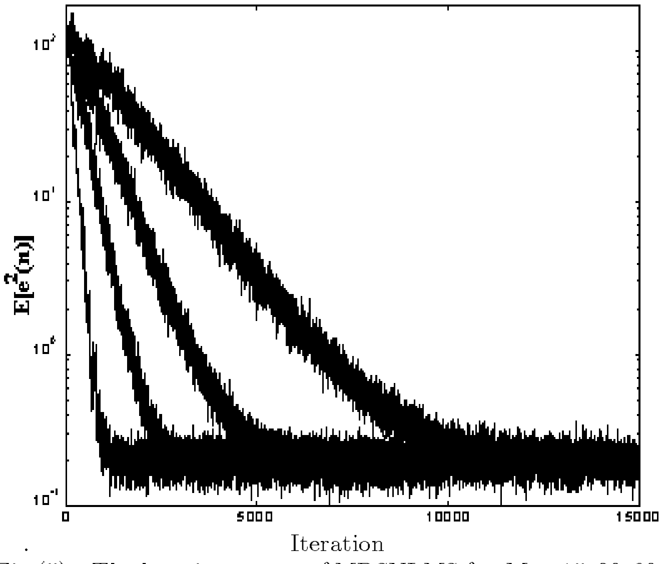

Fig.(5) : The learning curves of MRSNLMS for $M = 15, 30, 60,$ and 140.

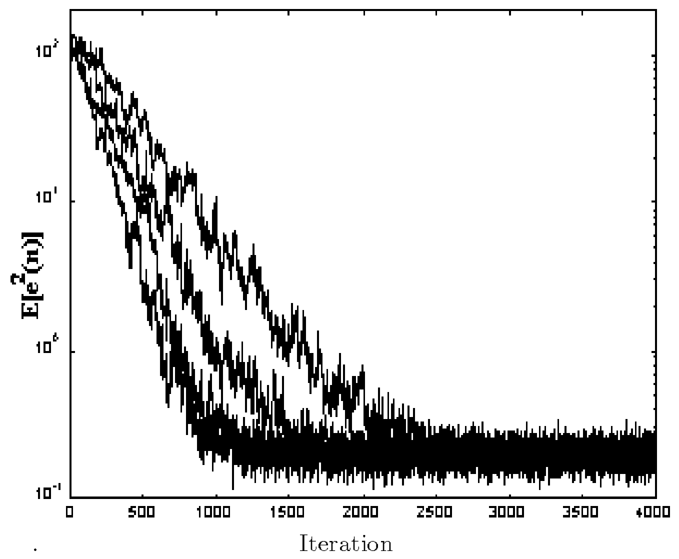

Fig.(6) : The learning curves of MMAXNLMS for $M = 15, 30, 60,$ and 140.