Adaptive Filters Employing Partial Updates

Scott C. Douglas *
Department of Electrical Engineering
University of Utah
Salt Lake City, Utah 84112

Abstract– **In some adaptive filtering applications, the least-mean-square (LMS) algorithm may be too computationally- and memory-intensive to implement. In this paper, we analyze two adaptive algorithms that update only a portion of the coefficients of the adaptive filter per iteration. These algorithms use decimated versions of the error and regressor signals, respectively. Simulations verify the accuracy of the analyses, and the robustness of the algorithms is also explored.**

submitted to

IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II

ANALOG AND DIGITAL SIGNAL PROCESSING

Manuscript Received:

*Please address correspondence to: Scott C. Douglas, Department of Electrical Engineering, 2272 Merrill Engineering Building, University of Utah, Salt Lake City, UT 84112. (801) 581-4445. FAX: (801) 581-5281. Electronic mail address: douglas@ee.utah.edu.

[0]This research was supported in part by NSF Grant No. MIP-9501680.

1 Introduction

This paper explores algorithms for updating the coefficients of an adaptive filter in which only a portion of the parameters are adjusted at each sample time. In this way, the overall complexity of the adaptive system is less than that of the least-mean-square (LMS) adaptive filter. These algorithms are particularly suited for real-time applications that place great demands on computational and/or memory resources. For example, in acoustic echo cancellation, an adaptive finite-impulse-response (FIR) filter may require thousands of coefficients to accurately model the echo return path [1]. An LMS-based, L-coefficient FIR echo canceller employing at least $2L$ multiplication/addition, L signal read, L coefficient read, and L coefficient write operations per iteration could require overly-expensive processors and memory to meet the sampling-rate requirements of this application. Partial updating of the LMS adaptive filter coefficients has been used successfully in high-data-rate communications systems [2], and the general concept has undoubtedly been employed in numerous other undocumented applications to lower the implementation costs of the system.

In this paper, we analyze and compare two algorithms that employ decimated versions of the error and regressor signals, respectively. Each of these algorithms has particular advantages over the other depending on the nature of the chosen application and its implementation in hardware. For example, in feedforward active noise control, an adaptive algorithm employing decimated regressor vector signals that are computed by separate filters can be much less costly to implement compared to the standard filtered-X LMS adaptive controller. This benefit is particularly important for multichannel systems, as the computational complexity of the coefficient updates is often several times that of the controller output computation alone [3].

It should be noted that there exist several other methods for reducing the computational complexity of the LMS adaptive update. These methods include adaptive algorithms that employ quantization in the updates, such as the sign-error, sign-data, sign-sign, and power-of-two quantized algorithms [4, 5, 6, 7]. These algorithms require dedicated VLSI hardware to take advantage of their computational structure. The block LMS adaptive algorithm also can ease the computational requirements associated with the coefficient updates [8]; however, this algorithm also places significant demands upon processor memory and program storage [9]. Note that the algorithms described in this paper are similar to a recently-proposed algorithm based upon the Gauss-Seidel iterative method for solving a set of linear equations [10]. However, the motivation for the algorithm in [10] is different from ours, as the algorithm in [10] is more complex than the LMS algorithm.

The organization of the paper is as follows. The algorithms are presented in Section 2, and analyses of the algorithms' behaviors for statistically-stationary random inputs is presented in

Section 3. The robustness of the algorithms for periodic inputs is discussed in Section 4. Section 5 presents simulations of the algorithms, verifying the predictions of the analyses. Section 6 presents our conclusions.

2 Algorithm Descriptions

For the following descriptions of the algorithms, we assume a standard FIR adaptive filter configuration, in which the regressor signal is simply the input signal.

The first algorithm we consider is a slight variation on the partial update LMS algorithm described in [2, 11]. Termed the *periodic LMS algorithm,* the coefficient updates are given by

$$w_{i,k+1} = \begin{cases} w_{i,k} + \mu e_l x_{l-i+1} & \text{if } (k+i)\text{mod}N = 0 \text{ and } l = N\lfloor k/N \rfloor \\ w_{i,k} & \text{otherwise} \end{cases} \quad (1)$$

$$e_k = d_k - \mathbf{W}_k^T \mathbf{X}_k, \quad (2)$$

where $\mathbf{W}_k = [w_{1,k}\ w_{2,k}\ \cdots\ w_{L,k}]^T$ is the coefficient vector of the adaptive filter at time k, $\mathbf{X}_k = [x_k\ x_{k-1}\ \cdots\ x_{k-L+1}]^T$ is the input signal vector, d_k is the desired response signal, e_k is the error signal, and $\lfloor \cdot \rfloor$ denotes the truncation operation. For $N = 1$ and $N = L$, this algorithm reduces to the LMS and partial update LMS adaptive algorithms, respectively. For $N > 1$, the number of multiplies and coefficient memory accesses required for this algorithm are fewer than those required for the LMS algorithm. In addition, the coefficient updates for this algorithm are regular, as only L/N coefficients[1] are changed at each iteration.

By considering N iterations of the updates in (1), it can be shown that this algorithm is mathematically-equivalent to the following N-fold coefficient vector update:

$$\mathbf{W}_{k+N} = \mathbf{W}_k + \mu e_k \mathbf{X}_k. \quad (3)$$

Equation (3) describes a modified version of the LMS adaptive algorithm that uses every Nth instantaneous gradient to update the filter coefficients [12].

The second proposed algorithm, termed the *sequential LMS algorithm*, is given by

$$w_{i,k+1} = \begin{cases} w_{i,k} + \mu e_k x_{k-i+1} & \text{if } (k-i+1)\text{mod}N = 0 \\ w_{i,k} & \text{otherwise} \end{cases}. \quad (4)$$

For $N = 1$, this algorithm reduces to the LMS algorithm. This algorithm uses every Nth element of the regressor vector signal, saving computation if this signal must be computed. Like the periodic LMS algorithm, this algorithm allows a regular processing strategy. However, it is

[1] For ease in reporting analytical results, we assume throughout the paper that L/N is an integer; this restriction is not necessary for implementation purposes, however.

Algorithm	*Number per Iteration (Ave.)* *Multiplies*	*Adds*	*Data Memory*
LMS	$2L+1+[L_h]$	$2L+[L_h-1]$	$2L+[L_h]$
Periodic LMS	$\frac{1}{N}(2L+1)+\frac{1}{N}+[L_h]$	$\frac{2L}{N}+[L_h-1]$	$2L+1+[L_h]$
Sequential LMS	$\left(1+\frac{1}{N}\right)L+1+\left[\frac{L_h}{N}\right]$	$\left(1+\frac{1}{N}\right)L+\left[\frac{L_h-1}{N}\right]$	$2L+\left[\frac{L_h}{N}\right]$

Table 1: Complexity of the LMS, periodic LMS, and sequential LMS adaptive algorithms.

not mathematically-equivalent to (3), and its performance and stability behavior are in general different from that of the periodic LMS algorithm.

Table 1 shows the complexity of the three FIR adaptive filtering algorithms in terms of the average number of multiplies, adds, and memory locations required for each per iteration. Shown in brackets are the additional operations and storage required for a filtered-X implementation as would be employed in a single-channel feedforward adaptive control task, where L_h is the length of the plant modelling filter. The computational advantages of each of the algorithms depend on the type of application in which it is used. For example, if only the coefficient vector $\mathbf{W}_k$ is needed in a standard FIR filtering task, the periodic LMS algorithm is to be preferred as it uses the least amount of resources by not computing the error signal at every iteration. In a feedforward adaptive control task, the sequential LMS algorithm is to be preferred as it reduces the number of regressor vector elements to be computed.

3 Analysis

For our analysis, we assume that the desired response is generated from a finite-impulse-response (FIR) model such that

$$d_k = \mathbf{W}_{opt}^T \mathbf{X}_k + n_k, \tag{5}$$

where $\mathbf{W}_{opt}$ are a set of optimal coefficients to be matched and $\{n_k\}$ is a zero-mean i.i.d. sequence that is independent of the input sequence $\{x_k\}$. In addition, we assume that the $L \times (L/N)$-dimensional matrices $X_{k+m(L/N)}$ and $X_{k+n(L/N)}$ are independent for $m \neq n$, with $\{X_k\}$ defined as

$$X_k = [\mathbf{X}_k \; \mathbf{X}_{k+1} \; \cdots \; \mathbf{X}_{k+(L/N)-1}]. \tag{6}$$

Technically, this assumption is not true, as these matrices share input signal elements for $|m - n| < L + (L/N)$. Even so, it yields accurate descriptions of adaptation behaviors for small step sizes.

For our analyses, we define a coefficient error vector as

$$\mathbf{V}_k = \mathbf{W}_k - \mathbf{W}_{opt}. \tag{7}$$

Then, using our assumptions, we can determine evolution equations for the mean coefficient error vector $E[\mathbf{V}_k]$ and the coefficient error correlation matrix $E[\mathbf{V}_k\mathbf{V}_k^T]$ for each algorithm.

3.1 Periodic LMS Algorithm

For comparative purposes, we summarize the results of the analysis for the periodic LMS algorithm given in [12]. For zero-mean signals, the evolution equation for the mean of the coefficient error vector is given by

$$E[\mathbf{V}_{k+N}] = (I_L - \mu R)E[\mathbf{V}_k], \tag{8}$$

where I_L is the L-dimensional identity matrix and $R = E[\mathbf{X}_k\mathbf{X}_k^T]$ is the input signal autocorrelation matrix. For zero-mean Gaussian signals, the evolution equation for the coefficient error correlation matrix is given by

$$\begin{aligned} E[\mathbf{V}_{k+N}\mathbf{V}_{k+N}^T] &= E[\mathbf{V}_k\mathbf{V}_k^T] - \mu\left(RE[\mathbf{V}_k\mathbf{V}_k^T] + E[\mathbf{V}_k\mathbf{V}_k^T]R\right) + \mu^2\sigma_n^2 R \\ &\quad + \mu^2(2RE[\mathbf{V}_k\mathbf{V}_k^T]R + R\mathrm{tr}[RE[\mathbf{V}_k\mathbf{V}_k^T]]). \end{aligned} \tag{9}$$

From this equation, we can determine a simple expression for the steady-state value of the excess mean-square-error (MSE) by neglecting the last term in (9) because it is much smaller than the other terms in the equation for small values of μ. The resulting expression is

$$\lim_{k\to\infty} E[(\mathbf{V}_k^T\mathbf{X}_k)^2] = \frac{\mu\sigma_n^2\mathrm{tr}R}{2}. \tag{10}$$

Moreover, it can be shown that a sufficient condition to guarantee the stability of (9) is

$$0 < \mu < \frac{2}{3\mathrm{tr}R}. \tag{11}$$

For i.i.d. input signals, it is sufficient to describe the evolution equation for the trace of the coefficient error correlation matrix, as given by

$$\mathrm{tr}E[\mathbf{V}_{k+N}\mathbf{V}_{k+N}^T] = (1 - 2\mu\sigma_x^2 + \mu^2((N-1)\sigma_x^4 + \eta))\mathrm{tr}E[\mathbf{V}_k\mathbf{V}_k^T] + \mu^2\sigma_n^2\sigma_x^2 L, \tag{12}$$

where $E[x_k^2] = \sigma_x^2$ and $E[x_k^4] = \eta$. The steady-state excess MSE for i.i.d. input signals is

$$\lim_{k\to\infty} E[(\mathbf{V}_k^T\mathbf{X}_k)^2] = \frac{\mu\sigma_n^2\sigma_x^4 L}{2\sigma_x^2 - \mu((L-1)\sigma_x^2 + \eta)}. \tag{13}$$

3.2 Sequential LMS Algorithm

3.2.1 Analysis Using the Independence Assumptions

We now analyze the sequential LMS algorithm's performance using the independence assumptions previously described. We can express the algorithm in (4) using the definition of $\mathbf{V}_k$ in (7) as

$$v_{i,k+1} = \begin{cases} v_{i,k} - \mu x_{k-i+1}\mathbf{X}_k^T\mathbf{V}_k + \mu n_k x_{k-i+1} & \text{if } (k-i+1)\mathrm{mod}N = 0 \\ v_{i,k} & \text{otherwise} \end{cases} . \quad (14)$$

Considering N iterations of this algorithm, the coefficient error vector update is

$$\mathbf{V}_{k+N} = A_k\mathbf{V}_k + \mathbf{B}_k, \quad (15)$$

where the elements of the $N \times N$ matrix A_k and vector $\mathbf{B}_k$ depend only on the elements of the input and noise signals. The exact form of A_k and $\mathbf{B}_k$ can be generated by successive application of the update relation in (14) over N iterations.

At this point, update equations for $E[\mathbf{V}_k]$ and $E[\mathbf{V}_k\mathbf{V}_k^T]$ can be developed by appropriate use of the relationship in (15). Unfortunately, the forms of A_k and $\mathbf{B}_k$ in (15) cannot be compactly described, and thus general forms of the expectations of these terms cannot be expressed. Even so, the technique for determining these evolution equations given the input and noise statistics is straightforward. We have used the computer-automated analysis technique described in [13] to derive the update equations for the mean coefficient error vector, given by

$$E[\mathbf{V}_{k+N}] = E[A_k]E[\mathbf{V}_k] + E[\mathbf{B}_k], \quad (16)$$

as well as the coefficient error correlation matrix, given by

$$E[\mathbf{V}_{k+N}\mathbf{V}_{k+N}^T] = E[A_k E[\mathbf{V}_k\mathbf{V}_k^T]A_k^T] + E[\mathbf{B}_k\mathbf{B}_k^T], \quad (17)$$

for input signals that is generated from the model

$$x_k = \mathbf{A}^T\mathbf{U}_k, \quad (18)$$

where $\mathbf{A} = [a_0 \ a_1 \ \cdots \ a_{M-1}]^T$ defines the correlation statistics of the input signal and $\mathbf{U}_k = [u_k \ u_{k-1} \ \cdots \ u_{k-M+1}]^T$, where u_k is a zero-mean i.i.d. signal. A description of the automated analysis technique appears in [13]. In this case, we enforce the assumptions previously described to simplify the forms of the equations produced by the analysis.

3.2.2 Approximate Analysis for Small Step Sizes

We now present a second analysis of the sequential LMS algorithm for small step sizes. For this analysis, we note that (4) can be written as

$$w_{i,k+1} = \begin{cases} w_{i,k} + \mu e_{m,l}x_{k-i+1} + O(\mu^2) & \text{if } (k-i+1)\mathrm{mod}N = 0,\ l = N\lfloor k/N \rfloor,\ m = k\mathrm{mod}N. \\ w_{i,k} & \text{otherwise} \end{cases} \quad (19)$$

$$e_{j,k} = d_{k+j} - \mathbf{W}_k^T \mathbf{X}_{k+j}, \tag{20}$$

where $O(\mu^2)$ represents terms that are of order μ^2 and higher. For small step sizes, these terms can be ignored. Collecting N updates of this equation yields the update given by

$$\mathbf{W}_{k+N} = \mathbf{W}_k + \mu \widetilde{\mathbf{X}}_k \otimes \mathbf{E}_k, \tag{21}$$

where $\mathbf{E}_k = [e_{0,k}\ e_{1,k} \cdots e_{(L/N)-1,k}]^T$ is an (L/N)-dimensional vector of errors, $\widetilde{\mathbf{X}}_k = [x_k\ x_{k-N}\ \cdots\ x_{k-L+N}]^T$ is an N-dimensional decimated version of the regressor vector, and $\otimes$ denotes the Kronnecker product.

The mean- and mean-square analyses of the update in (21) are similar to those for other block updating schemes [14]. Using (7), we can write an update for the coefficient error vector as

$$\mathbf{V}_{k+N} = (I_L - \mu \widetilde{\mathbf{X}}_k \otimes X_k^T)\mathbf{V}_k + \mu \widetilde{\mathbf{X}}_k \otimes \mathbf{N}_k, \tag{22}$$

where $\mathbf{N}_k = [n_k\ \cdots\ n_{k+(L/N)-1}]^T$ and X_k is as defined in (6).

For the evolution of the mean coefficient error vector, we can take expectations on both sides of (22) using our assumptions, which gives

$$E[\mathbf{V}_{k+N}] = (I_L - \mu E[\widetilde{\mathbf{X}}_k \otimes X_k^T])E[\mathbf{V}_k]. \tag{23}$$

It can be shown for stationary input signals that

$$E[\widetilde{\mathbf{X}}_k \otimes X_k^T] = R, \tag{24}$$

where R is the input signal autocorrelation matrix. Thus, from (23), we have

$$E[\mathbf{V}_{k+N}] = (I_L - \mu R)E[\mathbf{V}_k]. \tag{25}$$

This equation is the same as (8) for the periodic LMS algorithm in the mean.

We now examine the mean-square behavior of the sequential LMS algorithm for small step sizes. For this analysis, we assume that the input signal is zero mean and either Gaussian-distributed or i.i.d.-distributed with a known probability density. This additional assumption is necessary in order to evaluate the fourth-order moments within the analysis. The resulting equations are summarized here; the corresponding derivations are provided in the Appendix.

For zero-mean Gaussian input signals, the update equation for $E[\mathbf{V}_k\mathbf{V}_k^T]$ is given by

$$\begin{aligned} E[\mathbf{V}_{k+N}\mathbf{V}_{k+N}^T] &= E[\mathbf{V}_k\mathbf{V}_k^T] - \mu\left(RE[\mathbf{V}_k\mathbf{V}_k^T] + E[\mathbf{V}_k\mathbf{V}_k^T]R\right) + \mu^2\sigma_n^2 \widetilde{R} \otimes I_{(L/N)} \\ &\quad + \mu^2(2RE[\mathbf{V}_k\mathbf{V}_k^T]R + \widetilde{R} \otimes F(E[\mathbf{V}_k\mathbf{V}_k^T])), \end{aligned} \tag{26}$$

where $\widetilde{R}$ is an N-dimensional matrix whose i,jth value is defined by

$$\left[\widetilde{R}\right]_{i,j} = r_{(i-j)N}, \tag{27}$$

with $r_m = E[x_k x_{k-m}]$, and $F(\cdot)$ is a $(L/N)\times(L/N)$ matrix-valued function whose i,jth element is

$$\left[F(E[\mathbf{V}_k\mathbf{V}_k^T])\right]_{i,j} = \mathrm{tr}[R_{i-j}E[\mathbf{V}_k\mathbf{V}_k^T]], \tag{28}$$

with $R_m = E[\mathbf{X}_k\mathbf{X}_{k+m}^T]$. Because of the structure of $F(\cdot)$, the mean-square analysis depends upon autocorrelation lags that are not contained in the L-dimensional matrix R, and thus the above mean-square analysis cannot be simplified without further assumptions. However, if μ is small such that the last term in (26) can be neglected, then it can be shown that the steady-state excess MSE is approximately given by

$$\lim_{k\to\infty} E[(\mathbf{V}_k^T\mathbf{X}_k)^2] = \mathrm{tr}[RE[\mathbf{V}_k\mathbf{V}_k^T]] \tag{29}$$

$$= \frac{\mu\sigma_n^2 N\mathrm{tr}\widetilde{R}}{2} \tag{30}$$

$$= \frac{\mu\sigma_n^2\mathrm{tr}R}{2} \tag{31}$$

for stationary Gaussian input signals. Thus, the excess MSE in steady-state is approximately the same as that for the LMS adaptive filter with corresponding step size.

Because of the complexity of the form of (17), it is difficult to determine stability bounds on the step size μ from this equation. We instead use (26) to determine bounds on μ to guarantee convergence of the approximate mean-square analysis. The derivation is provided in the Appendix, and the resulting bounds are

$$0 < \mu < \frac{2}{3\mathrm{tr}[R]}. \tag{32}$$

These bounds are the same as those for the LMS algorithm for Gaussian input signals. As with all approximate analyses of the LMS algorithm, these bounds should be used as guidelines towards a good step size choice, as the actual stability bounds will differ somewhat from these results.

For i.i.d. input signals, it is shown in the Appendix that the update for $\mathrm{tr}E[\mathbf{V}_k\mathbf{V}_k^T]$ is given by

$$\mathrm{tr}E[\mathbf{V}_{k+N}\mathbf{V}_{k+N}^T] = (1-2\mu\sigma_x^2+\mu^2((N-1)\sigma_x^4+\eta))\mathrm{tr}E[\mathbf{V}_k\mathbf{V}_k^T]+\mu^2\sigma_n^2\sigma_x^2 L, \tag{33}$$

which the same as that for the periodic LMS algorithm.

To summarize these results, the overall behavior of the sequential LMS algorithm is approximately the same as that of the periodic LMS algorithm for stationary inputs. It should be noted, however, that both algorithms' convergence rates are approximately $1/N$th that of the LMS algorithm, from (8). That the LMS algorithm outperforms these simplified algorithms is not surprising, as the LMS algorithm coefficient updates require N times more arithmetical and memory operations than that required by the simplified algorithms.

4 Robustness Issues

In this section, we explore the stability properties of the periodic and sequential LMS algorithms. It is well known that the LMS adaptive filter is exponentially asymptotically stable for inputs that satisfy persistence of excitation conditions given by the following [15]: for all k, there exists $K < \infty$, $\delta_1 > 0$, and $\delta_2 > 0$ such that

$$\delta_1 I_L \quad < \quad \sum_{i=k}^{k+K} \mathbf{X}_i \mathbf{X}_i^T \quad < \quad \delta_2 I_L. \tag{34}$$

This result can be naturally extended to the periodic and sequential LMS algorithms. It is easily seen that the first algorithm is exponentially asymptotically stable for signals that satisfy similar conditions for all k and for all j, $1 \leq j < L/N$:

$$\delta_1 I_L \quad < \quad \sum_{i=kL/N+j}^{(k+K)L/N+j} \mathbf{X}_i \mathbf{X}_i^T \quad < \quad \delta_2 I_L, \tag{35}$$

where K, L, N, δ_1, and δ_2 are as defined previously. As μ tends to zero, it can be shown that the persistence of excitation conditions to be satisfied for the sequential LMS algorithm are

$$\delta_1 I_L \quad < \quad \sum_{i=kL/N+j}^{(k+K)L/N+j} \tilde{\mathbf{X}}_i \otimes X_i^T \quad < \quad \delta_2 I_L. \tag{36}$$

We now show that there exists signals that satisfy the persistence of excitation condition for the LMS algorithm that destabilize the sequential LMS algorithm. Consider the case $N = L$, such that the sequential LMS algorithm updates one coefficient per sample time. Define the input signal x_k as

$$x_k \quad = \quad \begin{cases} 1 & \text{if } (k - n + 1) \operatorname{mod} L = 0 \\ a & \text{otherwise.} \end{cases} \tag{37}$$

for some integer value of n. This signal satisfies the persistence of excitation condition in (34) for the LMS algorithm for all real values of a except $a = 1$ and $a = -1/(L-1)$. However, for

$$a \quad < \quad -\frac{1}{L-1}, \tag{38}$$

the sequential LMS algorithm is unstable with this input for vanishingly small μ if $N = L$. An exact expression for the transition matrix A_k in (15) for this input signal is

$$\begin{aligned} A_k \quad &= \quad (I_L + \mu U^T)^{-1}(I_L - \mu(I_L + U)) &(39)\\ &= \quad I_L - \mu X_k^T + \mu^2 \left\{ U^T + U^T U + \sum_{i=2}^{\infty} (-\mu)^{i-2} (U^T)^i (I_L - \mu(I_L + U)) \right\}, &(40) \end{aligned}$$

where U is an upper-triangular matrix with zeros on the diagonal and values of a for all entries above the diagonal and $X_k^T = (I_L + U + U^T)$. Clearly, as μ tends to zero, the matrix A_k tends to the matrix $(I_L - \mu X_k^T)$. An eigenvalue analysis of X_k^T shows that it has $L-1$ eigenvalues equal to $(1-a)$ and one eigenvalue equal to $(1+(L-1)a)$. Thus, if $a < -1/(L-1)$, the matrix X_k^T has both positive and negative eigenvalues, and the resulting system is unstable with this input signal.

We have simulated the behavior of the sequential LMS adaptive algorithm and have verified that divergence does occur with this signal. This instability is sensitive to the phase of the input signal; thus, it may be possible to achieve either stable or unstable behavior of the sequential algorithm by a shift of the time axis. While we have not identified other signals that destabilize the algorithm, we suspect that there are other input signals, particularly those that contain sinusoidal components or that have cyclostationary statistics, that lead to algorithm instability. Problems of this nature have been observed before in other adaptive systems, most notably those that use blind equalization methods [16]. To stabilize the sequential algorithms, a multiplicative leak factor can be introduced into the updates [11], as follows:

$$w_{i,k+1} = \begin{cases} \beta w_{i,k} + \mu e_k x_{k-i+1} & \text{if } (k-i+1)\text{mod}N = 0 \\ w_{i,k} & \text{otherwise} \end{cases}, \tag{41}$$

where $0 \ll \beta < 1$. Considering the previous example, the corresponding transition matrix for the leaky algorithm will have eigenvalues bounded by one in magnitude for $a < -1/(L-1)$ if $0 < \beta < 1 + \mu(1+(L-1)a)$. Care must be taken with this approach, however, as the slower adaptation speeds of the sequential algorithm can limit the performance of such a modified system.

5 Simulations

5.1 Analysis Verification

We have evaluated the mean-square behaviors of the periodic and sequential LMS algorithms using our statistical analyses. For the sequential LMS algorithm, we have directly evaluated the expectations appearing in (17) using the automated techniques described in [13]. We first study a three-tap system identification problem with $N = L = 3$. The inputs to the two adaptive filters are zero-mean Gaussian with $E[x_k^2] = 1$, $E[x_k x_{k+1}] = 0.5$, and $E[x_k x_{k+i}] = 0$ for $i \geq 2$. For our simulations, the initial values of the filter coefficients have been chosen to be $w_{j,0} = w_{j,opt} + 2$ for $j \in \{1, 2, 3\}$. The results of one thousand trials have been averaged together to produce simulated convergence curves for comparison with the theoretical results.

Figure 1 shows the convergence of the total coefficient error powers $\text{tr}E[\mathbf{V}_k \mathbf{V}_k^T]$ for the periodic and sequential LMS adaptive filters as predicted by Equations (9), (17), and (26) and as found from simulations. As can be seen, the convergence behaviors of the two algorithms are similar initially; however, the steady-state coefficient error power for the sequential LMS adaptive filter is two-thirds greater than that for the periodic LMS adaptive filter for this input signal. This result can be predicted from (26) by neglecting the last term on the right-hand-side of this equation and solving for the steady-state value of the coefficient error covariance matrix as

$$\lim_{k\to\infty} \frac{\mathrm{tr}E[\mathbf{V}_k\mathbf{V}_k^T]_{seq}}{\mathrm{tr}E[\mathbf{V}_k\mathbf{V}_k^T]_{per}} = \frac{\sigma_x^2 \mathrm{tr} R^{-1}}{L}. \qquad (42)$$

Using the fact that $\sigma_x^2 = \sum_{i=1}^{L} \lambda_i / L$, where $\{\lambda_i\}$ are the eigenvalues of R, it is straightforward to show that $\sigma_x^2 \mathrm{tr} R^{-1}/L \geq 1$ for any positive definite covariance matrix. Thus, the steady-state coefficient error power produced by the sequential LMS algorithm is unlikely to be smaller than that of the periodic LMS algorithm for a given step size. Note that both analyses for the sequential LMS algorithm are quite accurate in predicting simulated behavior; thus, the simpler approximate analysis of Section 3.2.2 is to be preferred.

Equations (10) and (31) indicate that the excess MSEs of the algorithms will be similar, and Figure 2 shows the excess MSEs of the two algorithms for this example. As can be seen, the convergence of the two algorithms is quite similar throughout all stages of adaptation. The minor differences in convergence of the algorithms can be predicted by the differences in the fourth-moment terms in equations (9) and (26).

We now explore the behaviors of the two algorithms for a $L = 50$ coefficient system identification example. In this case, the correlated Gaussian input signals used for the experiments are zero mean with $r_m = 0.7^{|m|}$, and the initial values of the filter coefficients were chosen randomly from a uniform distribution over [-0.1,0.1]. One hundred simulations have been averaged for comparison with the theoretical predictions. Figures 3 and 4 show the total coefficient error powers and excess MSEs for both algorithms, respectively, for $N = 2$, $N = 5$, and $N = 10$. Also shown are the theoretical predictions of the systems' performances, showing that the theory is accurate. As can be seen, the speed of adaptation decreases proportionally for both algorithms as the number of coefficients updated per iteration is reduced. In this case, all systems produce the same steady-state excess MSE; however, the sequential LMS algorithms' steady-state coefficient error powers are approximately three times those of the periodic LMS algorithms. Thus, the periodic LMS algorithm is to be preferred if the most accurate estimates of the actual filter coefficient values are desired.

5.2 Multichannel Control Example

We now consider an example from the field of active noise and vibration control, in which an electro-acoustic system is used to cancel unwanted acoustical energy via destructive interference. For details on the system structure, notation, and the multichannel filtered-X LMS adaptive algorithm, the reader is referred to [3, 14]. For an N_x-input, N_y-output system with N_e feedback error sensors, the multichannel filtered-X LMS update is

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \mu U_k \overline{\mathbf{E}}_k, \tag{43}$$

where $\mathbf{W}_k$ is an $N_x N_y L_w$-dimensional vector describing the $N_x N_y$ controller transfer functions, L_w is the filter length of each controller, $\overline{\mathbf{E}}_k = [e_k^{(1)} \cdots e_k^{(N_e)}]^T$ is a vector of error measurements from the N_e error sensors, and U_k is an $(N_x N_y L_w) \times N_e$ matrix of filtered input signal values derived from the input signal channels and the output-actuator-to-error-sensor paths. The coefficient update in (43) can be difficult to implement in many real-world cases for several reasons. Firstly, the number of coefficients per controller channel often number in the hundreds for typical sampling rates, controller tasks, and input signals. Secondly, forming the filtered signal matrix U_k involves $N_y N_e L_h$ multiply/adds per time sample, and this computation can become significantly large for moderately-sized systems. Thirdly, the memory for storing U_k is N_e times that required to store the controller coefficients. Finally, the update calculation in (43) requires $N_x N_y N_e L_w$ multiply/adds to implement. These factors make the update in (43) unwieldy for many noise control tasks.

We can simplify the coefficient updates by using a partial update of the controller coefficients at each iteration. In this case, the following algorithm is suggested:

$$w_{i,k+1} = w_{i,k} - \mu [U_k]_{i,j} e_k^{(j)}, \quad 1 \leq i \leq N_x N_y, \tag{44}$$

where $[\cdot]_{i,j}$ denotes the i,jth element of the matrix and $j = (k - i + 1) \mathrm{mod} N_e$. In this case, every controller coefficient is updated at each iteration using only one of the N_e error measurements, and each error measurement is used to adjust $N_x N_y L_w / N_e$ filter coefficients. After N_e time samples, all controller coefficients have been adjusted using measurements taken from all N_e error sensors. This update only requires $N_x N_y L_w$ multiplies per iteration to adjust the controller coefficients. Moreover, since only subsampled versions of the filtered input signals are used in the update, this algorithm only needs $N_x N_y L_h$ multiplies and $N_x N_y L_w$ storage locations for the decimated filtered-signal matrix. The computational load of the algorithm is regular, and the programming methodology for implementing the algorithm is straightforward as well.

Figure 5a shows the total average mean-square error of the standard filtered-X controller in a four-input, four-output, four-error active noise control simulation example with a Gaussian disturbance as described in [14]. Figure 5b shows the corresponding convergence behavior for the proposed algorithm in (44). Note the difference in time scales for the two plots. The proposed algorithm converges at one-quarter the rate of the original controller; however, the update portion of the proposed algorithm requires approximately one-quarter the number of operations as compared to that of the original algorithm, and it requires only one-quarter the data storage of the original algorithm. Since the coefficient update represents approximately 80% of the total computational requirements of the system in this example, the overall controller's complexity is reduced by 60%. These savings are significant in that they enable a complex control system to be implemented on a significantly-simpler processor.

6 Conclusions

In this paper, we have analyzed and compared two adaptive algorithms that update only a portion of the coefficients of an adaptive system per iteration on average. Our analyses of these algorithms indicate that they achieve approximately the same level of misadjustment as the LMS algorithm for a given step size; however, their convergence speeds are reduced approximately in proportion to the number of coefficients updated per iteration divided by the filter length. These algorithms have different stability properties than that of the LMS adaptive algorithm, and we have shown that the sequential LMS algorithm can exhibit unstable behavior for certain periodic input signals. These algorithms are potentially useful for real-time applications of adaptive filters.

Appendix

To derive an update for $E[\mathbf{V}_k\mathbf{V}_k^T]$ for the algorithm in (19), we post-multiply the expression for $\mathbf{V}_{k+N}$ in (22) by its transpose and take expectations. The resulting equation is

$$\begin{aligned} E[\mathbf{V}_{k+N}\mathbf{V}_{k+N}^T] &= E[\mathbf{V}_k\mathbf{V}_k^T] - \mu\left(E[\widetilde{\mathbf{X}}_k \otimes X_k^T]E[\mathbf{V}_k\mathbf{V}_k^T] + E\left[\mathbf{V}_k\left(E[\widetilde{\mathbf{X}}_k \otimes X_k^T]\mathbf{V}_k\right)^T\right]\right) \\ &\quad + \mu^2 E\left[\widetilde{\mathbf{X}}_k \otimes \mathbf{N}_k\left(\widetilde{\mathbf{X}}_k \otimes \mathbf{N}_k\right)^T\right] + \mu^2 E\left[\widetilde{\mathbf{X}}_k \otimes X_k^T\mathbf{V}_k\left(\widetilde{\mathbf{X}}_k \otimes X_k^T\mathbf{V}_k\right)^T\right] \end{aligned} \quad (45)$$

where we have used our assumptions to separate expectations that depend upon input signal and coefficient vector terms. The input signal expectation terms of the form $E[\widetilde{\mathbf{X}} \otimes X_k^T]$ that appear in the second term of (45) evaluate to the input signal autocorrelation matrix, from (24).

To evaluate the third term of (45), we note that

$$\begin{aligned} E\left[\widetilde{\mathbf{X}}_k \otimes \mathbf{N}_k\left(\widetilde{\mathbf{X}}_k \otimes \mathbf{N}_k\right)^T\right] &= E\left[\left(\widetilde{\mathbf{X}}_k\widetilde{\mathbf{X}}_k^T\right) \otimes \left(\mathbf{N}_k\mathbf{N}_k^T\right)\right] \quad (46) \\ &= \widetilde{R} \otimes \sigma_n^2 I_{(L/N)}, \quad (47) \end{aligned}$$

where we have used the i.i.d. nature of the noise sequence and where $\widetilde{R}$ is as defined in (27). To evaluate the expectation within the fourth term of (45), we note that

$$E\left[\widetilde{\mathbf{X}}_k \otimes X_k^T\mathbf{V}_k\left(\widetilde{\mathbf{X}}_k \otimes X_k^T\mathbf{V}_k\right)^T\right] = E\left[\left(\widetilde{\mathbf{X}}_k\widetilde{\mathbf{X}}_k^T\right) \otimes \left(X_k^T\mathbf{V}_k\mathbf{V}_k^T X_k\right)\right]. \quad (48)$$

Thus, the $(m-1)N+i, (n-1)N+j$th element of the matrix in (48), for $1 \leq m \leq L/N$, $1 \leq n \leq L/N$, $1 \leq i \leq N$, and $1 \leq j \leq N$, is

$$\begin{aligned} & E\left[\left(\tilde{\mathbf{X}}_k\tilde{\mathbf{X}}_k^T\right) \otimes \left(X_k^T\mathbf{V}_k\mathbf{V}_k^TX_k\right)\right]_{(m-1)N+i,(n-1)N+j} \\ &= \sum_{l=1}^{L}\sum_{p=1}^{L} E[x_{k-(m-1)N}x_{k-(n-1)N}x_{k+i-l}x_{k+j-p}]E[v_{l,k}v_{p,k}]. \end{aligned} \tag{49}$$

For Gaussian input signals, we can evaluate the expectation in (49) as

$$\begin{aligned} E[x_{k-(m-1)N}x_{k-(n-1)N}x_{k+i-l}x_{k+j-p}] &= r_{(m-1)N-i+l}r_{(n-1)N-j+p} + r_{(n-1)N-i+l}r_{(m-1)N-j+p} \\ &\quad + r_{(m-n)N}r_{i-j+p-l}. \end{aligned} \tag{50}$$

Combining (49) and (50), it is easy to show that we can express the matrix in (48) as

$$E\left[\tilde{\mathbf{X}}_k\tilde{\mathbf{X}}_k^T \otimes X_k^T\mathbf{V}_k\mathbf{V}_k^TX_k\right] = 2RE[\mathbf{V}_k\mathbf{V}_k^T]R + \tilde{R} \otimes F(E[\mathbf{V}_k\mathbf{V}_k^T]), \tag{51}$$

where $F(\cdot)$ is as defined in (28) and $R_m = E[\mathbf{X}_k\mathbf{X}_{k+m}^T]$. Combining (24), (45), (47), and (51) gives the update in (26).

To determine sufficient bounds on the step size μ to guarantee convergence of the mean-square analysis equation in (26), we first note that for any $1 \leq i \leq j \leq L$,

$$\left|\left[F(E[\mathbf{V}_k\mathbf{V}_k^T])\right]_{i,j}\right| \leq \mathrm{tr}[RE[\mathbf{V}_k\mathbf{V}_k^T]], \tag{52}$$

a result that can be easily shown using the Cauchy-Schwartz inequality. Thus, by taking the trace of both sides of (26), we have

$$\mathrm{tr}[E[\mathbf{V}_{k+N}\mathbf{V}_{k+N}^T]] - \mathrm{tr}[E[\mathbf{V}_k\mathbf{V}_k^T]] \leq -\mu(2-3\mu\mathrm{tr}[R])\mathrm{tr}[RE[\mathbf{V}_k\mathbf{V}_k^T]] + \mu\sigma_n^2\sigma_x^2L. \tag{53}$$

Now, to guarantee convergence of the filter coefficients in mean square, it is necessary to make the right hand side of (53) negative. For $\sigma_n^2\sigma_x^2L/(2-3\mu\mathrm{tr}[R]) < \mathrm{tr}[RE[\mathbf{V}_k\mathbf{V}_k^T]] < \infty$, this will be true if μ satisfies the bounds as given in (32).

For i.i.d. input signals, the expectation in (49) evaluates to

$$E[x_{k-(m-1)N}x_{k-(n-1)N}x_{k+i-l}x_{k+j-p}] = \begin{cases} \eta & \text{if } (m-1)N = (n-1)N = l-i = p-j \\ \sigma_x^4 & \begin{array}{l}\text{if } l-i = (m-1)N \neq p-j = (n-1)N \\ \text{or } l-i = (n-1)N \neq p-j = (m-1)N\end{array} \\ 0 & \text{otherwise.} \end{cases} \tag{54}$$

Thus, by taking the traces of both sides of (45) and noting that $E[\tilde{\mathbf{X}}_k \otimes X_k^T] = \sigma_x^2 I_L$, we can find an update for $\mathrm{tr}E[\mathbf{V}_k\mathbf{V}_k^T]$. The resulting expression is given in (33).

References

[1] A. Gilloire, "Experiments with sub-band acoustic echo cancellers for teleconferencing," *Proc. IEEE International Conf. Acoust., Speech, Signal Processing,* Dallas, TX, vol. IV, pp. 2141-2144, April 1987.

[2] V. Wolff, R. Gooch, and J. Treichler, "Specification and development of an equalizer-demodulator for wideband digital microwave radio signals," *Proc. IEEE Military Communications Conference,* New York, NY, vol. 2, pp. 461-467, October 1988.

[3] P.A. Nelson and S.J. Elliott, *Active Control of Sound* (London: Academic Press Ltd., 1992).

[4] A. Gersho, "Adaptive filtering with binary reinforcement," *IEEE Trans. Inform. Theory,* vol. IT-30, no. 2, pp. 191-199, March 1984.

[5] J.L. Moschner, "Adaptive filtering with clipped input data," Stanford Univ. Technical Report No. 6796-1, Stanford Univ. Center for Systems Research, Stanford, CA, June 1970.

[6] R.W. Lucky, "Techniques for adaptive equalization of digital communication systems," *Bell Syst. Tech. J.,* vol. 45, pp. 1151-1162, February 1966.

[7] P. Xue and B. Liu, "Adaptive equalizer using finite-bit power-of-two quantizer," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no. 6, pp. 1603-1611, December 1986.

[8] G.A. Clark, S.K. Mitra, and S.R. Parker, "Block implementation of adaptive digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, no. 3, pp. 744-752, June 1981.

[9] N.K. Jablon, "On the complexity of frequency-domain adaptive filtering," *IEEE Trans. Signal Processing*, vol. SP-39, no. 10, pp. 2331-2334, October 1991.

[10] J.T. Cilke and D.M. Etter, "A new adaptive algorithm to reduce weight fluctuations caused by high variance data," *IEEE Trans. Signal Processing*, vol. SP-40, no. 9, pp. 2324-2327, September 1992.

[11] J.R Treichler, C.R. Johnson, Jr., and M. G. Larimore, *Theory and Design of Adaptive Filters* (New York: Wiley-Interscience, 1987).

[12] W.A. Gardner, "Learning characteristics of stochastic-gradient-descent algorithms: A general study, analysis, and critique," *Signal Processing*, vol. 6, no. 2, pp. 113-133, April 1984.

[13] S.C. Douglas and W. Pan, "Exact expectation analysis of the LMS adaptive filter," *IEEE Trans. Signal Processing,* vol. SP-43, no. 12, pp. 2863-2871, December 1995.

[14] S.C. Douglas, "Analysis of the multiple-error and block least-mean-square adaptive algorithms," *IEEE Trans. Circuits and Systems II: Analog Digital Signal Processing,* vol. 42, no. 2, pp. 92-101, February 1995.

[15] W.A. Sethares and C.R. Johnson, Jr., "A comparison of two quantized state adaptive algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-37, no. 1, pp. 138-143, January 1989.

[16] J.R. Treichler, V. Wolff, and C.R. Johnson, Jr., "Observed misconvergence in the constant modulus adaptive algorithm," *Proc. 25th Asilomar Conf. Signals, Systems, and Computers,* vol. 2, pp. 663-667, November 1991.

List of Figures

List of Footnotes

Manuscript received:

The author is with the Department of Electrical Engineering, University of Utah, Salt Lake City, UT.

This research was supported in part by NSF Grant No. MIP-9501680.

[1]For notational convenience and ease of reporting analytical results, we assume throughout the paper that L/N is an integer; this restriction is not necessary for implementation purposes, however.

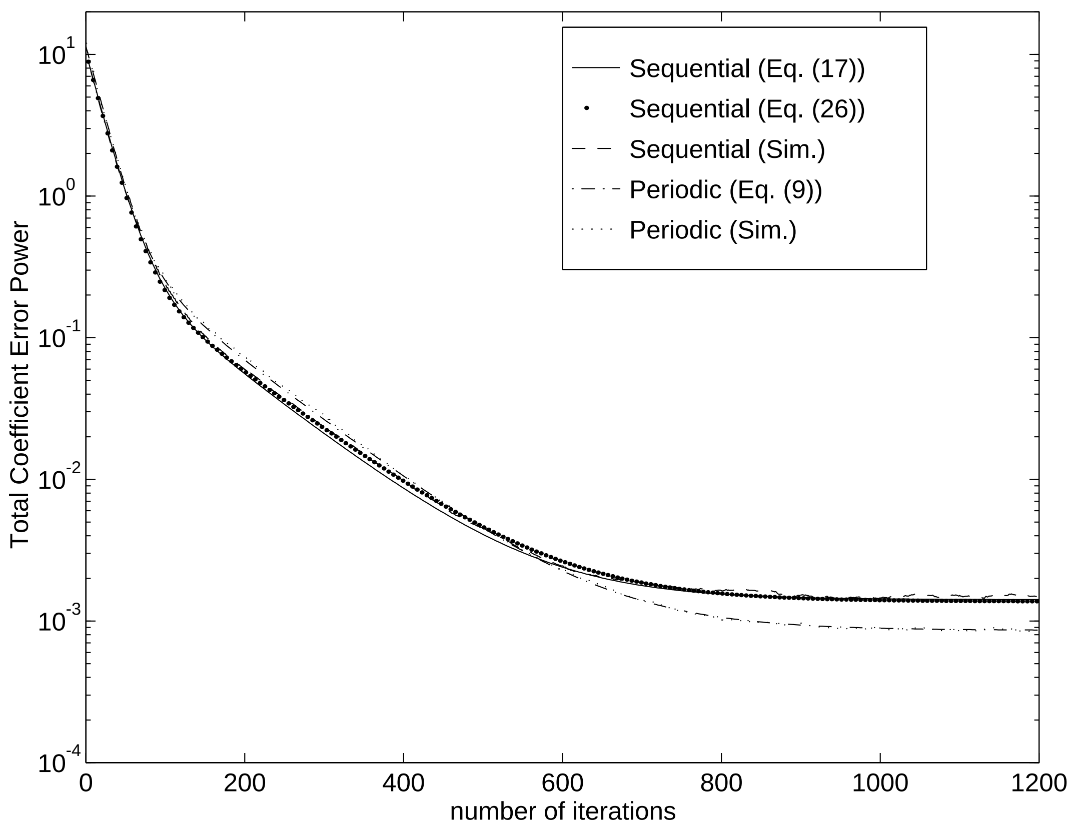

Figure 1: Convergence of total coefficient error power, theory and simulation: sequential and periodic LMS adaptive filters, correlated Gaussian input signals, $L = N = 3$, $\mu = 0.05$.

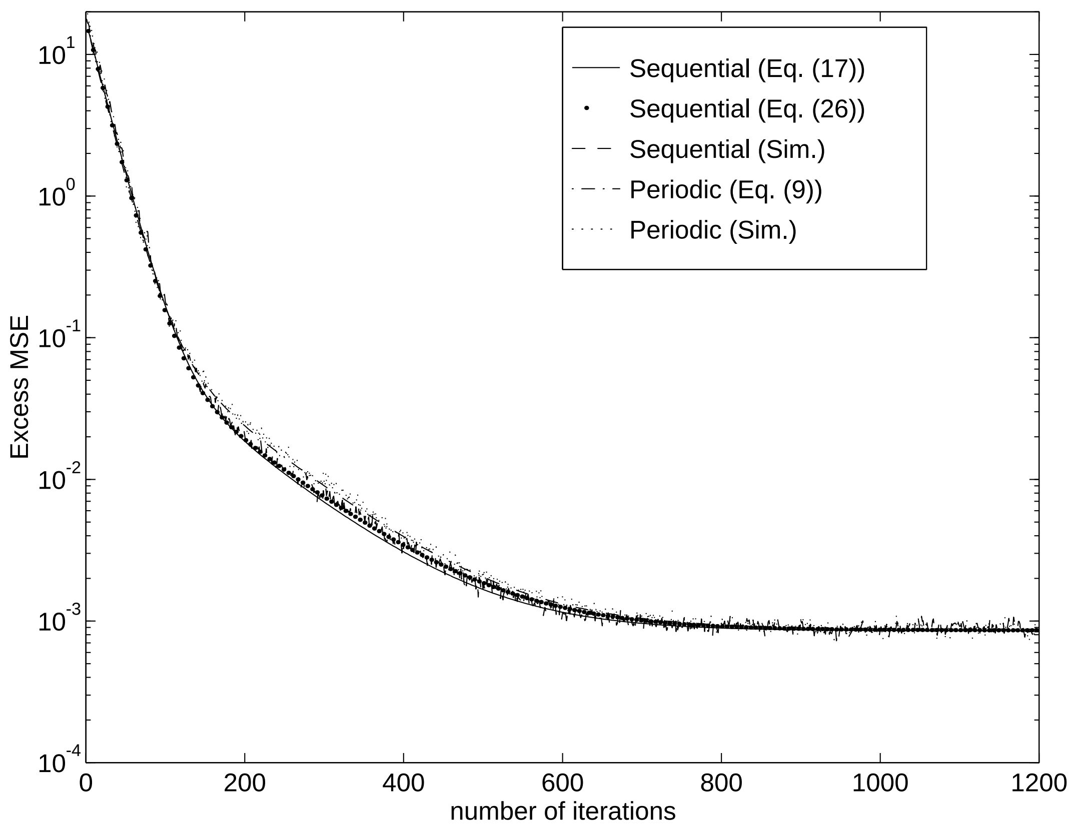

Figure 2: Convergence of excess MSE, theory and simulation: sequential and periodic LMS adaptive filters, correlated Gaussian input signals, $L = N = 3$, $\mu = 0.05$.

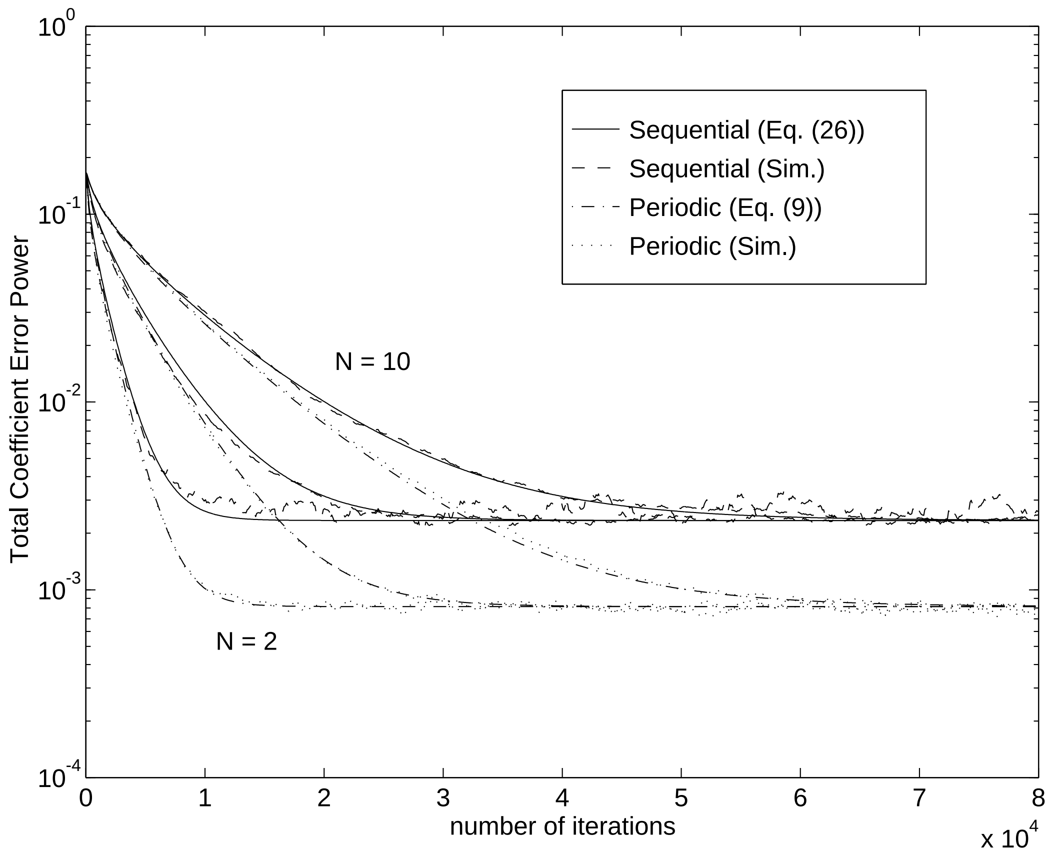

Figure 3: Convergence of total coefficient error power, theory and simulation: sequential and periodic LMS adaptive filters, correlated Gaussian input signals, $L = 50$, $\mu = 0.003$.

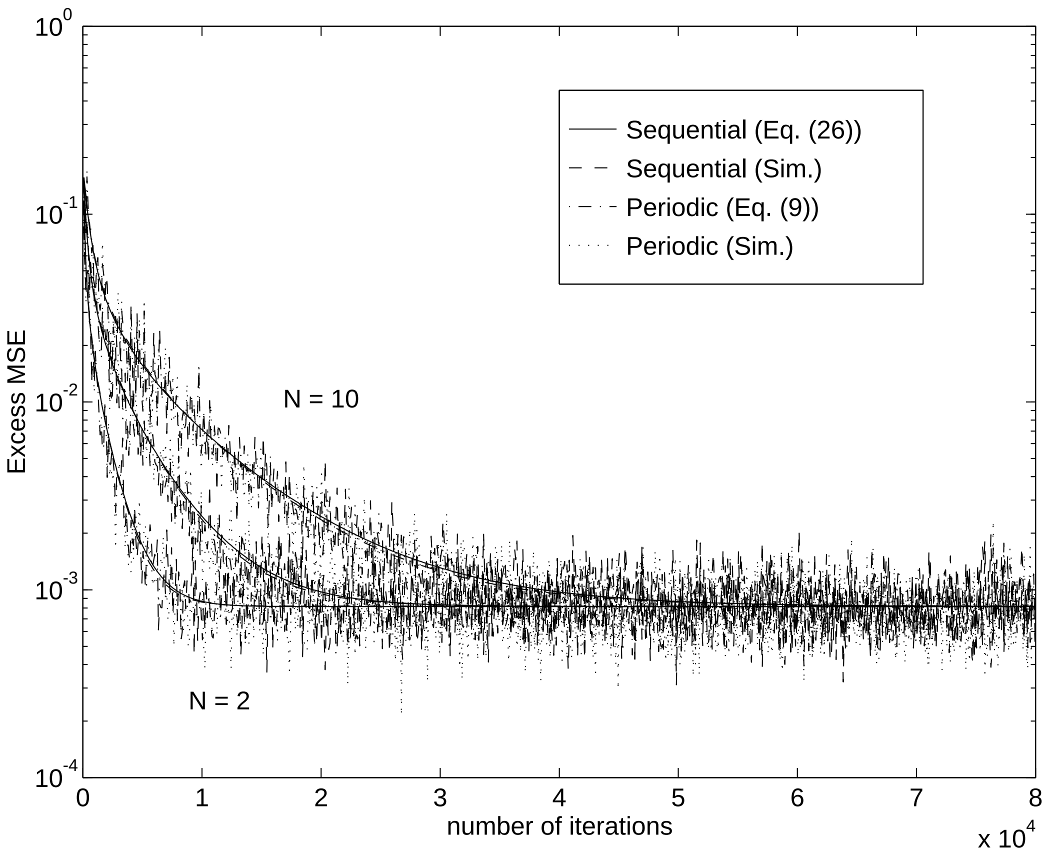

Figure 4: Convergence of excess MSE, theory and simulation: sequential and periodic LMS adaptive filters, correlated Gaussian input signals, $L = 50$, $\mu = 0.003$.

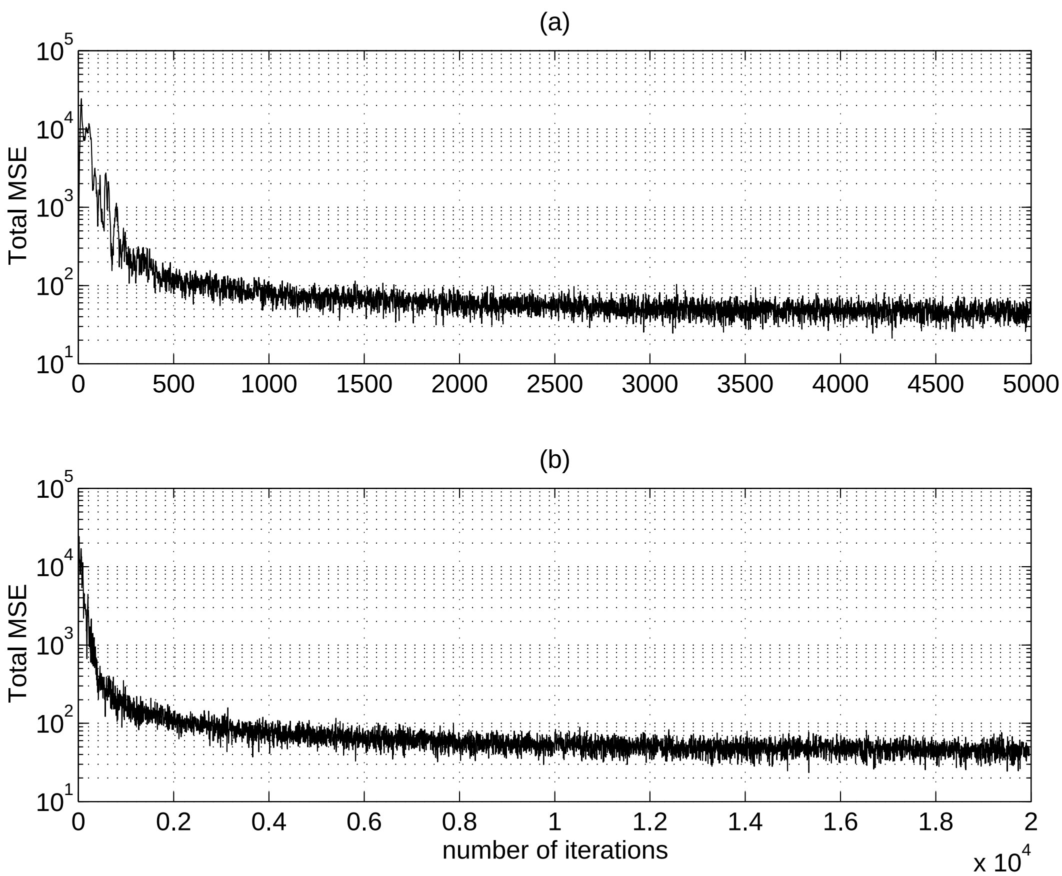

Figure 5: Convergence of total MSE, a) filtered-X LMS and b) sequential filtered-X LMS algorithms: $N_x = N_y = N_e = 4$, exponentially-decaying actuator filters, unit-variance white Gaussian input signal channels, $\mu = 0.0001$.