FAST EXACT FILTERED-X LMS AND LMS ALGORITHMS FOR MULTICHANNEL ACTIVE NOISE CONTROL

S.C. Douglas

Department of Electrical Engineering
University of Utah
Salt Lake City, UT 84112 USA

ABSTRACT

In some situations where active noise control could be used, the well-known multichannel version of the filtered-X LMS adaptive filter is too computationally-complex to implement. In this paper, we develop a fast, exact implementation of this multichannel system whose complexity is approximately $O(2L)$ per filter channel, where L is the FIR filter length. In addition, we provide a computationally-efficient method for effectively removing the delays of the secondary paths within the coefficient updates, thus yielding a fast implementation of the LMS adaptive algorithm for multichannel active noise control. Examples illustrate both the equivalence of the algorithms to their original counterparts and the computational gains provided by the new algorithms.

1. INTRODUCTION

In active noise control, undesired sound in an acoustic region is attenuated by superimposing an equal-but-opposite acoustical field in the region. The "anti-noise" is created by measuring in real time the source of the unwanted noise with N_x input sensors, processing this information digitally to produce N_y output signals, and sending these signals to the desired quiet region using N_y actuators. An additional N_e error sensors are placed in the quieted region to provide feedback for the control system to adjust its characteristics. For the controller, finite-impulse-response (FIR) filters adapted using the filtered-X least-mean-square (LMS) algorithm are often used to calculate the system's output signals [1, 2]. Table 1 lists the equations for this algorithm, where $\mathbf{W}^{(i,j)}(n) = [w_0^{(i,j)}(n) \cdots w_{L-1}^{(i,j)}(n)]^T$ contains the filter coefficients for the (i,j)th channel of the controller, $\widehat{\mathbf{e}}(n) = [\epsilon^{(1)}(n) \cdots \epsilon^{(N_e)}(n)]^T$ contains samples from the N_e error sensors at time n, the $L \times N_e$-dimensional matrix $F^{(i,j)}(n)$ contains filtered input signal values, $\overline{F}^{(i,j)}(n)$ contains the first $L-1$ rows of $F^{(i,j)}(n)$, $y^{(j)}(n)$ is the jth output of the controller at time n, and $\mu(n)$ is the step size parameter. Here, $\mathbf{X}^{(i)}(n)$, $\underline{\mathbf{X}}^{(i)}(n)$, and $\underline{H}^{(j)}$ are defined as

$$\mathbf{X}^{(i)}(n) = [x^{(i)}(n) \cdots x^{(i)}(n-L+1)]^T \quad (1)$$

$$\underline{\mathbf{X}}^{(i)}(n) = [x^{(i)}(n) \cdots x^{(i)}(n-M+1)]^T \quad (2)$$

$$\underline{H}^{(j)} = [\underline{\mathbf{H}}^{(j,1)} \cdots \underline{\mathbf{H}}^{(j,N_e)}], \quad (3)$$

[0]Data was provided by SRI International, Menlo Park, CA.
This material is based upon work supported in part by the U.S. Army Research Office under Contract No. DAAH04-96-1-0085 and in part by the National Science Foundation under Grant No. MIP-9501680.

Table 1: The multichannel filtered-X LMS algorithm.

Equation	# Mults.
for $j = 1$ to N_y do	
$y^{(j)}(n) = \sum_{i=1}^{N_x} \mathbf{X}^{(i)T}(n)\mathbf{W}^{(i,j)}(n)$	N_xN_yL
for $i = 1$ to N_x do	
$\mathbf{f}^{(i,j)}(n) = \underline{H}^{(j)T}\underline{\mathbf{X}}^{(i)}(n-1)$	$N_xN_yN_eM$
$F^{(i,j)}(n) = \begin{bmatrix} \mathbf{f}^{(i,j)}(n) \\ \overline{F}^{(i,j)}(n-1) \end{bmatrix}$	
$\mathbf{W}^{(i,j)}(n+1) = \mathbf{W}^{(i,j)}(n) - F^{(i,j)}(n)(\mu(n)\widehat{\mathbf{e}}(n))$	$N_xN_yN_eL$ $+N_e$
end	
end	

respectively, where $\underline{\mathbf{H}}^{(j,k)} = [h_1^{(j,k)} \cdots h_M^{(j,k)}]^T$ contains the M impulse response values for the jth-output-to-kth-error secondary path transfer function.

While the coefficient updates in Table 1 require only multiplies and adds to implement, the number of multiplies needed at each iteration is

$$C_{FXLMS} = N_xN_y((N_e+1)L + N_eM) + N_e, \quad (4)$$

a quantity that grows precipitously as the numbers of input sensors, output actuators, and error sensors are increased. A single-input, single-output, single-error system only requires $2L + M + 1$ multiplies per iteration to implement. Clearly, it is desirable to develop implementations of the multichannel system with complexities that are of $O(N_xN_y(2L))$. Recently, the multichannel adjoint LMS algorithm has been introduced whose complexity is [3]

$$C_{ALMS} = N_xN_y\left(2L + \left(\frac{N_e}{N_x}\right)M\right) + N_e. \quad (5)$$

Although the simulated performance of this new algorithm appears to be similar to that of the multichannel filtered-X LMS algorithm, little is known about its theoretical performance characteristics or its stability properties. No simple exact implementation of the multichannel filtered-X LMS adaptive algorithm has ever been presented.

In addition, the multichannel filtered-X LMS adaptive controller suffers from poor performance because the error signals $\epsilon^{(k)}(n)$ contain delayed versions of the controller coefficients $\mathbf{W}^{(i,j)}(n)$. These delays lead to a reduced stability range for the step size parameter $\mu(n)$ and slower convergence speeds [5]. One can approximately calculate the true LMS coefficient updates for the controller filters by recalculating the N_e error signals using the newest coefficients $\mathbf{W}^{(i,j)}(n)$, as described in [6] in the single-channel case. However, the overall complexity of the multichannel version of this algorithm is

$$C_{LMS} = N_x N_y\Big((2N_e+1)L + \Big(N_e + \frac{N_e}{N_x}\Big)M\Big) + N_e, \quad (6)$$

which is about twice that of the original filtered-X LMS controller. No simple way for computing the LMS updates for a multichannel controller has ever been presented.

In this paper, we provide fast, exact implementations of both the filtered-X LMS and LMS adaptive algorithms for multichannel active noise control systems. Our implementations produce the same output signals as their original implementations while requiring many fewer multiplies in situations where the controller filter length L is somewhat larger than the secondary path filter length M and is much larger than the number of error sensors N_e. In such cases, the complexities of our algorithms are $O(N_xN_y(2L))$. Example simulations show the equivalence of the new algorithms to their more-complex counterparts.

2. FAST MULTICHANNEL FILTERED-X LMS ALGORITHM

To derive the fast version of the multichannel filtered-X LMS algorithm, we can write the coefficient updates of the (i,j)th controller filter coefficients in Table 1 as

$$\mathbf{W}^{(i,j)}(n+1) = \mathbf{W}^{(i,j)}(n) - X^{(i)}(n-1)\underline{\mathcal{E}}^{(j)}(n). \quad (7)$$

where

$$X^{(i)}(n) = [\mathbf{X}^{(i)}(n) \cdots \mathbf{X}^{(i)}(n-M+1)] \quad (8)$$
$$= [\underline{\mathbf{X}}^{(i)}(n) \cdots \underline{\mathbf{X}}^{(i)}(n-L+1)]^T. \quad (9)$$

and the M-dimensional vector $\underline{\mathcal{E}}^{(j)}(n)$ for $1 \le j \le N_y$ is defined as

$$\underline{\mathcal{E}}^{(j)}(n) = [\varepsilon_0^{(j)} \cdots \varepsilon_{M-1}^{(j)}]^T \quad (10)$$
$$= \underline{H}^{(j)}(\mu(n)\widehat{e}(n)). \quad (11)$$

The update term in (7) resembles a similar term that appears in the fast affine projection (FAP) algorithm [7, 8]. Thus, we can use a similar method as is used in [7, 8] to efficiently compute the coefficient updates. Define the auxiliary coefficient vector $\widehat{\mathbf{W}}^{(i,j)}(n)$ such that

$$\widehat{\mathbf{W}}^{(i,j)}(n) = \mathbf{W}^{(i,j)}(n) + \widetilde{X}^{(i)}(n-1)\overline{\mathbf{E}}^{(j)}(n-1), (12)$$

where $\widetilde{X}^{(i)}(n)$ is a matrix containing the last $M-1$ columns of $X^{(i)}(n)$ and $\overline{\mathbf{E}}^{(j)}(n)$ contains the first $M-1$ elements of the vector $\underline{\mathbf{E}}^{(j)}(n)$ defined as

$$\underline{\mathbf{E}}^{(j)}(n) = \begin{bmatrix} \varepsilon_0^{(j)}(n) \\ \varepsilon_1^{(j)}(n) + \varepsilon_0^{(j)}(n-1) \\ \vdots \\ \varepsilon_{M-1}^{(j)}(n) + \cdots + \varepsilon_0^{(j)}(n-M+1) \end{bmatrix}. (13)$$

Note that $\underline{\mathbf{E}}^{(j)}(n)$ can be easily updated as

$$\underline{\mathbf{E}}^{(j)}(n) = \begin{bmatrix} 0 \\ \overline{\mathbf{E}}^{(j)}(n-1) \end{bmatrix} + \underline{\mathcal{E}}^{(j)}(n). \quad (14)$$

Using the definition of $\widehat{\mathbf{W}}^{(i,j)}(n)$ in (12), we can develop a simple update for this vector given by

$$\widehat{\mathbf{W}}^{(i,j)}(n+1) = \widehat{\mathbf{W}}^{(i,j)}(n) - \mathbf{X}^{(i)}(n-M)E_{M-1}^{(j)}(n), \quad (15)$$

Table 2: The fast multichannel filtered-X LMS algorithm.

Equation	# Mults.
$\widetilde{\mathbf{R}}(n) = \widetilde{\mathbf{R}}(n-1) + \sum_{i=1}^{N_x}\Big(\widetilde{\mathbf{X}}^{(i)}(n-1)x^{(i)}(n) - \widetilde{\mathbf{X}}^{(i)}(n-L-1)x^{(i)}(n-L)\Big)$	$2N_x(M-1)$
for $j = 1$ to N_y do	
$y^{(j)}(n) = \Big(\sum_{i=1}^{N_x}\mathbf{X}^{(i)T}(n)\widehat{\mathbf{W}}^{(i,j)}(n)\Big)$	N_xN_yL
$-\widetilde{\mathbf{R}}^T(n)\overline{\mathbf{E}}^{(j)}(n-1)$	$+N_y(M-1)$
$\underline{\mathcal{E}}^{(j)}(n) = \underline{H}^{(j)}(\mu(n)\widehat{e}(n))$	$N_yN_eM + N_e$
$\underline{\mathbf{E}}^{(j)}(n) = \begin{bmatrix} 0 \\ \overline{\mathbf{E}}^{(j)}(n-1) \end{bmatrix} + \underline{\mathcal{E}}^{(j)}(n)$	
for $i = 1$ to N_x do	
$\widehat{\mathbf{W}}^{(i,j)}(n+1) = \widehat{\mathbf{W}}^{(i,j)}(n) - \mathbf{X}^{(i)}(n-M)E_{M-1}^{(j)}(n)$	N_xN_yL
end	
end	

where $E_{M-1}(n)$ is the last element of $\underline{\mathbf{E}}^{(j)}(n)$. Note that equation (15) requires only L multiplies to implement.

Because the true coefficient vector $\mathbf{W}^{(i,j)}(n)$ is not available, we employ correction terms to compute the N_y controller outputs using the vectors $\{\widehat{\mathbf{W}}^{(i,j)}(n)\}$. By premultiplying both sides of (12) by $\mathbf{X}^{(i)T}(n)$, the quantity $y^{(i,j)}(n) = \mathbf{X}^{(i)T}(n)\mathbf{W}^{(i,j)}(n)$ can be written as

$$y^{(i,j)}(n) = \mathbf{X}^{(i)T}(n)\widehat{\mathbf{W}}^{(i,j)}(n) - \mathbf{X}^{(i)T}(n)\widetilde{X}^{(i)}(n-1)\overline{\mathbf{E}}^{(j)}(n-1). \quad (16)$$

Define the $M-1$-dimensional vector $\widetilde{\mathbf{R}}(n)$ as

$$\widetilde{\mathbf{R}}(n) = \sum_{i=1}^{N_x}\widetilde{X}^{(i)T}(n-1)\mathbf{X}^{(i)}(n). \quad (17)$$

Note that $\widetilde{\mathbf{R}}(n)$ can be updated as

$$\widetilde{\mathbf{R}}(n) = \widetilde{\mathbf{R}}(n-1) + \sum_{i=1}^{N_x}\Big(\widetilde{\mathbf{X}}^{(i)}(n-1)x(n) - \widetilde{\mathbf{X}}^{(i)}(n-L-1)x(n-L)\Big), \quad (18)$$

where $\widetilde{\mathbf{X}}^{(i)}(n)$ contains the last $M-1$ elements of $\underline{\mathbf{X}}^{(i)}(n)$. Summing both sides of (16) over $1 \le i \le N_x$, we obtain

$$y^{(j)}(n) = \Big(\sum_{i=1}^{N_x}\mathbf{X}^{(i)T}(n)\widehat{\mathbf{W}}^{(i,j)}(n)\Big) - \widetilde{\mathbf{R}}^T(n)\overline{\mathbf{E}}^{(j)}(n-1) \quad (19)$$

Equations (11), (14), (15), (18), and (19) define the fast multichannel filtered-X LMS algorithm. Table 2 lists the equations for the algorithm and the number of multiplies required for each step. The complexity of the algorithm at each iteration is

$$C_{FXLMS}^{(f)} = N_xN_y\Big(2L + \Big(\frac{N_e}{N_x} + \frac{1}{N_x} + \frac{2}{N_y}\Big)M\Big) - 2N_x - N_y + N_e. \quad (20)$$

If $N_x = N_y = N_e = N$, then as N gets large, the overall complexity of the system approaches that of N independent single-channel filtered-X LMS controllers.

Remark: The coefficient updates for $\widehat{\mathbf{W}}^{(i,j)}(n)$ in (15) are of the same form as those for $\mathbf{W}^{(i,j)}(n)$ of the algorithm in [3]. Thus, the multichannel adjoint LMS algorithm is a modified version of the fast multichannel filtered-X LMS algorithm in which the N_e correction terms $\widetilde{\mathbf{R}}^T(n)\overline{\mathbf{E}}^{(j)}(n-1)$ in (19) are ignored. Since the elements of $\overline{\mathbf{E}}^{(j)}(n-1)$ are scaled by the step size $\mu(n)$, this approximation is valid for small step sizes. However, the two algorithms are different for nonzero step sizes. Since the complexity of the multichannel adjoint LMS algorithm is nearly the same as that of our algorithm in many cases, the latter algorithm is to be preferred.

3. FAST MULTICHANNEL LMS ALGORITHM

We now derive a fast implementation of the LMS adaptive algorithm for the multichannel controller. The single-channel version of this algorithm is described in [9]. The multichannel LMS adaptive filter update is given by

$$\mathbf{W}^{(i,j)}(n+1) = \mathbf{W}^{(i,j)}(n) - F^{(i,j)}(n)(\mu(n)\mathbf{e}(n)), \qquad (21)$$

where the modified error vector $\mathbf{e}(n)$ is given by

$$\mathbf{e}(n) = \widehat{\mathbf{e}}(n) - \left(\sum_{j=1}^{N_y} \underline{H}^{(j)T}\underline{\mathbf{Y}}^{(j)}(n-1) - \sum_{i=1}^{N_x} F^{(i,j)T}(n)\mathbf{W}^{(i,j)}(n)\right), \qquad (22)$$

where $\underline{\mathbf{Y}}^{(j)}(n) = [y^{(j)}(n) \cdots y^{(j)}(n-M+1)]^T$. If the secondary path transfer functions contained in $\{\underline{H}^{(j)}\}$ are accurate, then $\mathbf{e}(n)$ in (22) contains the instantaneous errors of the system and does not depend on past coefficient values $\mathbf{W}^{(i,j)}(k)$, $k < n$.

Define the vector $\underline{\mathbf{U}}^{(j)}(n)$ as

$$\underline{\mathbf{U}}^{(j)}(n) = \left(\sum_{i=1}^{N_x} X^{(i)T}(n)\mathbf{W}^{(i,j)}(n+1)\right) - \underline{\mathbf{Y}}^{(j)}(n). \qquad (23)$$

Then, $\mathbf{e}(n)$ can be computed as

$$\mathbf{e}(n) = \widehat{\mathbf{e}}(n) + \sum_{j=1}^{N_y} \underline{H}^{(j)T}\underline{\mathbf{U}}^{(j)}(n-1). \qquad (24)$$

We can show that $\underline{\mathbf{U}}^{(j)}(n)$ can be recursively computed as

$$\underline{\mathbf{U}}^{(j)}(n) = \begin{bmatrix} 0 \\ \overline{\mathbf{U}}^{(j)}(n-1) \end{bmatrix} - \sum_{i=1}^{N_x} X^{(i)T}(n)\Delta^{(i,j)}(n), \qquad (25)$$

where $\overline{\mathbf{U}}^{(j)}(n)$ contains the first $M-1$ elements of $\underline{\mathbf{U}}^{(j)}(n)$ and $\Delta^{(i,j)}(n) = \mathbf{W}^{(i,j)}(n+1) - \mathbf{W}^{(i,j)}(n)$. Substituting for $\Delta^{(i,j)}(n)$ from (21) into (25), we find that

$$\underline{\mathbf{U}}^{(j)}(n) = \begin{bmatrix} 0 \\ \overline{\mathbf{U}}^{(j)}(n-1) \end{bmatrix} - \underline{R}_{xf}^{(j)}(n)(\mu(n)\mathbf{e}(n)), \qquad (26)$$

where we have defined the $M \times N_e$ matrix $\underline{R}_{xf}^{(j)}(n)$ as

$$\underline{R}_{xf}^{(j)} = \sum_{i=1}^{N_x} X^{(i)T}(n)F^{(i,j)}(n). \qquad (27)$$

Table 3: The fast multichannel LMS algorithm.

Equation	*# Mults.*
$\widetilde{\mathbf{R}}(n) = \widetilde{\mathbf{R}}(n-1) + \sum_{i=1}^{N_x} \left(\widetilde{\mathbf{X}}^{(i)}(n-1)x^{(i)}(n) - \widetilde{\mathbf{X}}^{(i)}(n-L-1)x^{(i)}(n-L)\right)$	$2N_x(M-1)$
$\mathbf{e}(n) = \widehat{\mathbf{e}}(n) + \sum_{j=1}^{N_y} \underline{H}^{(j)T}\underline{\mathbf{U}}^{(j)}(n-1)$	N_yN_eM
$\mathbf{e}^{(\mu)}(n) = \mu(n)\mathbf{e}(n)$	N_e
for $j = 1$ to N_y do	
for $i = 1$ to N_x do	
$\mathbf{f}^{(i,j)}(n) = \underline{H}^{(j)T}\underline{\mathbf{X}}^{(i)}(n-1)$	$N_xN_yN_eM$
$\mathbf{f}^{(i,j)}(n-L) = \underline{H}^{(j)T}\underline{\mathbf{X}}^{(i)}(n-L-1)$	$N_xN_yN_eM$
end	
$\underline{R}_{xf}^{(j)}(n) = \underline{R}_{xf}^{(j)}(n-1) + \sum_{i=1}^{N_x} \left(\underline{\mathbf{X}}^{(i)}(n)\mathbf{f}^{(i,j)T}(n) - \underline{\mathbf{X}}^{(i)}(n-L)\mathbf{f}^{(i,j)T}(n-L)\right)$	$2N_xN_yN_eM$
$\underline{\mathbf{U}}^{(j)}(n) = \begin{bmatrix} 0 \\ \overline{\mathbf{U}}^{(j)}(n-1) \end{bmatrix} - \underline{R}_{xf}^{(j)}(n)\mathbf{e}^{(\mu)}(n)$	N_yN_eM
$y^{(j)}(n) = \left(\sum_{i=1}^{N_x} \mathbf{X}^{(i)T}(n)\widehat{\mathbf{W}}^{(i,j)}(n)\right) - \widetilde{\mathbf{R}}^T(n)\overline{\mathbf{E}}^{(j)}(n-1)$	N_xN_yL $+N_y(M-1)$
$\underline{\mathcal{E}}^{(j)}(n) = \underline{H}^{(j)}\mathbf{e}^{(\mu)}(n)$	N_yN_eM
$\underline{\mathbf{E}}^{(j)}(n) = \begin{bmatrix} 0 \\ \overline{\mathbf{E}}^{(j)}(n-1) \end{bmatrix} + \underline{\mathcal{E}}^{(j)}(n)$	
for $i = 1$ to N_x do	
$\widehat{\mathbf{W}}^{(i,j)}(n+1) = \widehat{\mathbf{W}}^{(i,j)}(n) - \mathbf{X}^{(i)}(n-M)E_{M-1}^{(j)}(n)$	N_xN_yL
end	
end	

Note that $\underline{R}_{xf}^{(j)}(n)$ can be updated recursively in a fashion similar to (18).

The method for calculating $\mathbf{e}(n)$ in (23)–(27) can be combined with the previously-derived algorithm to obtain a fast version of the LMS algorithm for multichannel active noise control. Table 3 lists the complete algorithm. The number of multiplies at each iteration is

$$C_{LMS}^{(f)} = N_xN_y\left(2L + \left(N_e\left(4 + \frac{3}{N_x}\right) + \frac{1}{N_x} + \frac{2}{N_y}\right)M\right) - 2N_x - N_y + N_e. \qquad (28)$$

As in the previous case, the complexity of this algorithm is also $O(N_xN_y(2L))$ if N_eM is somewhat less than L

Remark: The quantity $\underline{H}^{(j)T}\underline{\mathbf{U}}^{(j)}(n-1)$ is of $O(\mu(n))$. Thus, if $\mu(n)$ is an extremely small value, then this quantity can be neglected in (24). The resulting coefficient updates are then the same as those of the multichannel filtered-X LMS controller. The filtered-X LMS algorithm was originally derived assuming "slow adaptation" [10]. Our algorithm quantitatively defines the difference between the filtered-X LMS and LMS coefficient updates and provides an alternate justification for the former algorithm for small step sizes.

Table 4: Complexities of the various algorithms for different controller configurations, $L = 500$, $M = 125$.

N_x	N_y	N_e	C_{ALMS}	$C^{(f)}_{FXLMS}$	C_{FXLMS}	$C^{(f)}_{LMS}$	C_{LMS}
1	2	2	2502	2998	3502	5998	6002
2	2	2	4502	5246	7002	10246	11502
2	4	2	9002	9994	14002	19994	23002
2	4	4	10004	10996	24004	30996	42004
4	3	4	13504	14868	36004	41868	61504
4	4	4	18004	19492	48004	55492	82004
4	8	4	36004	37998	96004	109998	164004
4	8	8	40008	41992	176008	185992	312008
4	16	4	72004	74980	192004	218980	328004
8	8	8	72008	74984	352008	346084	616008
4	16	16	96016	98992	672016	674992	1216016
8	16	16	160016	163984	1344016	1251984	2400016
16	16	16	288016	293968	2688016	2405968	4768016

4. EXAMPLES AND SIMULATIONS

We now compare the complexities of the fast filtered-X LMS and LMS algorithms with those of their original counterparts and with that of the adjoint LMS algorithm in [3]. Table 4 shows the number of multiplies per time instant for each implementation for different choices of N_x, N_y, and N_e, where $L = 500$ and $M = 125$ for the controller and secondary path filter lengths, respectively. As can be seen, the complexities of the fast filtered-X LMS and LMS algorithms are smaller than their original implementations in every case, and for systems with large number of channels, the savings is significant. Moreover, the complexity differences between the fast filtered-X LMS and adjoint LMS algorithms are minor for most systems. Since the behavior of the fast filtered-X LMS algorithm is well-understood, this algorithm is to be preferred. In addition, in cases where the complexity of the fast LMS algorithm is similar to that of the original filtered-X LMS algorithm, current users of the latter algorithm could potentially achieve delayless LMS adaptation for the same filter lengths and sampling rates on their existing hardware platforms.

We now explore the behaviors of the systems in an active noise control task. Figure 1 shows the total squared errors, given by filtered versions of $\sum_{k=1}^{N_e} \epsilon^{(k)2}(n)$, for three different four-input, three-output, four-error active noise control systems with $L = 500$ and $M = 200$-coefficient controller and secondary path filters, respectively, as applied to data taken from an air conditioner compressor located in an anechoic chamber [11]. Shown for comparison are three systems adapted using the adjoint LMS, fast filtered-X LMS, and fast LMS algorithms, respectively, where the step sizes for each algorithm have been chosen to give fast convergence without a large increase in the total steady-state mean-squared error. As can be seen, the LMS algorithm provides the fastest convergence, as a large step size can be chosen for this algorithm. The filtered-X LMS controller has better performance than the adjoint LMS controller, and the range of stable step sizes for the filtered-X LMS algorithm is larger than that for the adjoint LMS algorithm for this data. For step sizes less than $\mu \approx 0.05$, however, all three algorithms perform similarly on this data set, indicating that the adjoint LMS algorithm is to be preferred when computational and memory resources are at a premium. In each case, the fast versions of the algorithms produced exactly the same controller outputs as their more-complex counterparts, up to finite-precision errors in the computations.

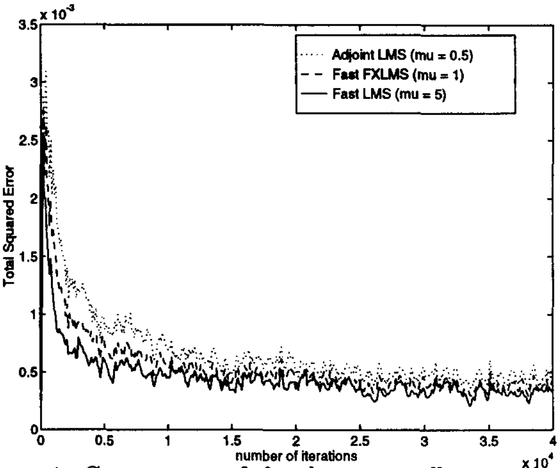

Figure 1: Convergence of the three controllers on air conditioner compressor data.

5. CONCLUSIONS

In this paper, we have presented implementations of the filtered-X LMS and LMS algorithms for multichannel active noise control that are more efficient than their standard implementations when the number of controller channels is large. The two methods can also be applied to active noise control algorithms employing normalized step sizes as well as sgn(·) and other nonlinearities in the coefficient updates. Simulations on data taken from a physical active noise control system show the equivalence of the new algorithms to their original counterparts.

REFERENCES

[1] S.M. Kuo and D.M. Morgan, *Active Noise Control Systems: Algorithms and DSP Implementations* (New York: Wiley-Interscience, 1996).

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

[3] E.A. Wan, "Adjoint LMS: An efficient alternative to the filtered-X LMS and multiple error LMS algorithms," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Atlanta, GA, vol. 3, pp. 1842-1845, May 1996.

[4] G. Long, F. Ling, and J.G. Proakis, "The LMS algorithm with delayed coefficient adaptation," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 37, no. 9, pp. 1397-1405, Sept. 1989; vol. 40, no. 1, pp. 230-232, Jan. 1992 (corrections).

[5] E. Bjarnason, "Analysis of the filtered-X LMS algorithm," *IEEE Trans. Speech Audio Processing,* vol. 3, no. 6, pp. 504-514, Nov. 1995.

[6] E. Bjarnason, "Active noise cancellation using a modified form of the filtered-X LMS algorithm," *Proc. EUSIPCO-92, Signal Processing VI,* Brussels, Belgium, vol. 2, pp. 1053-1056, Aug. 1992.

[7] S. Gay and S. Tavathia, "The fast affine projection algorithm," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Detroit, MI, vol. 5, pp. 3023-3026, May 1995.

[8] M. Tanaka, Y. Kaneda, S. Makino, and J. Kojima, "Fast projection algorithm and its step size control," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Detroit, MI, vol. 2, pp. 945-948, May 1995.

[9] S.C. Douglas, "Efficient implementation of the modified filtered-X LMS algorithm," submitted to *IEEE Signal Processing Lett.*

[10] B. Widrow, D. Shur, and S. Shaffer, "On adaptive inverse control," *Proc. 15th Asilomar Conf. Circuits, Systems, and Computers,* Pacific Grove, CA, pp. 185-195, Nov. 1981.

[11] D.K. Peterson, W.A. Weeks, and W.C. Nowlin, "Active control of complex noise problems using a broadband multichannel controller," *Proc. NOISE-CON,* Ft. Lauderdale, FL, p. 315, May 1994.