Modified FxLMS Algorithms with Improved Convergence Performance *

M. Rupp
Wireless Technology, Research Dept.
AT&T
791 Holmdel-Keyport Rd.
Holmdel NJ 07733-0400

A. H. Sayed
Dept. Electrical and Computer Eng.
University of California
Santa Barbara, CA 93106

Abstract

This paper proposes two modifications of the FxLMS algorithm with improved convergence behaviour albeit at the same computational cost of $2M$ operations per time step as the original FxLMS update. The paper further introduces a generalized FxLMS recursion and establishes that the various recursions are in fact of filtered-error form. An optimal choice of the step-size parameter in order to guarantee faster convergence, and conditions for robustness, are also derived. Several simulation results are included to illustrate the discussions.

1 Introduction

A widely used algorithm in active noise control is the so-called Filtered-x Least-Mean-Squares (FxLMS) algorithm [2, 4, 8]. Fig. 1 depicts a simple noise control system and serves as a motivation for the FxLMS scheme. The noise from an engine, usually in an enclosure such as a duct, is measured by a (detection) microphone and a filtered version of it is generated by a loudspeaker (secondary source) with the intent of diminishing the noise level at a certain location, say at the location of the right-most (error) microphone.

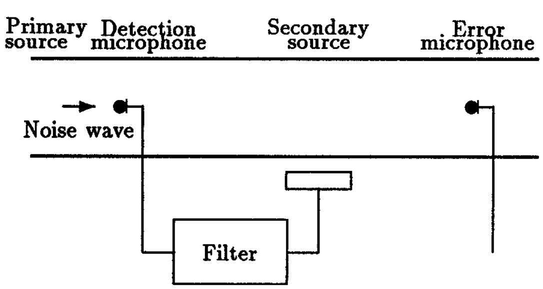

Figure 1: *Sketch of a simple active noise control system in a duct.*

Figure 2 is a redrawing of the duct example of Fig. 1. It shows the measured input noise signal $u(i)$ and a filtered version of it, denoted by $d(i)$, which corresponds to the signal $u(i)$ traveling further down the enclosure until it reaches the secondary source. An anti-noise sequence $\hat{d}(i)$ is generated by an FIR filter of length M at the secondary source with the intent of cancelling $d(i)$. The difference between both signals $d(i)$ and $\hat{d}(i)$ cannot be measured directly but only a filtered version of it, which is denoted by $e_f(i) = F[d(i) - \hat{d}(i)]$. The filter F is often assumed of FIR type and its presence is due to the fact that both signals have to travel a path before reaching the right-most (error) microphone. Since this path is unknown, the secondary controller can become unstable and the objective is to update the filter weights (denoted by $\mathbf{w}$) in order to minimize the filtered error $e_f(i)$ in a certain sense. The filter has to emulate the path that transforms $u(i)$ into $d(i)$. Depending on the situation, the whole device can be relatively large (with many tap weights) and people have often resorted to very small step-sizes for stabilization purposes. This has the obvious disadvantage of slow adaptation and convergence.

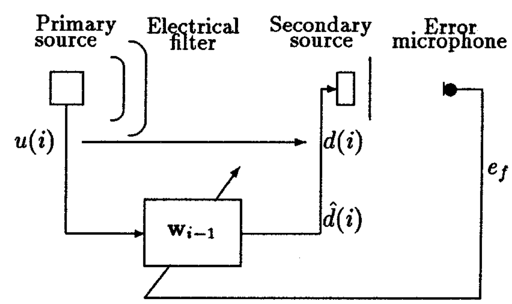

Figure 2: *Active noise control system using feedforward control.*

The FxLMS algorithm requires $2M$ operations per time step and has been shown to exhibit *poor convergence* behaviour. A modification of it (referred to as Modified FxLMS) has been proposed in the literature to ameliorate the convergence problem at the cost of increased computations, which are of the order of $3M$

*This work was supported in part by the National Science Foundation under Award No. MIP-9409319.

operations per time step [1, 5]. This figure is still prohibitive in several applications where the value of M is significantly large.

Motivated by these facts, we pursue here the feedback approach suggested in [6] and propose two modifications of the FxLMS algorithm with improved convergence behaviour albeit at the same cost of $2M$ operations per time step. The results are summarized in Table 1 where the last two lines refer to the two variants proposed in this work, and M_F denotes the length of the error filter F.

Algorithm	Complexity ($M \gg M_F$)	Memory Capacity	Convergence Behaviour
FxLMS	$2M + M_F$	$3M$	Poor
MFxLMS	$3M + 2M_F$	$3M + M_F$	Good
MFxLMS-1	$2M + 2M_F$	$3M + M_F$	Good
MFxLMS-2	$2M + 3M_F$	$3M + 2M_F$	Reasonable

Table 1: *Comparison of different variants.*

__Notation.__ We use small boldface letters to denote vectors (e.g., $\mathbf{u}$), "$*$" to denote Hermitian conjugation, and $\|\mathbf{x}\|_2$ to denote the Euclidean norm of a vector. We also use subscripts for time-indexing of vector quantities (e.g., $\mathbf{u}_i$) and parenthesis for time-indexing of scalar quantities (e.g., $v(i)$). All vectors are column vectors except for the row vectors $\mathbf{u}_i$.

2 The FxLMS Algorithm

The set-up for the FxLMS algorithm is depicted in Fig. 3. Let $\mathbf{w}$ be an unknown weight vector and assume $\{d(i)\}$ are noisy measurements that are related to $\mathbf{w}$ via $d(i) = \mathbf{u}_i\mathbf{w} + v(i)$. Here, the $\{\mathbf{u}_i\}$ are known input row vectors and the $\{v(i)\}$ are noise terms that may also account for modeling errors.

The FIR filter F is assumed known, of length M_F and coefficients $\{f_j\}_{j=0}^{M_F-1}$. The signal $\tilde{e}_a(i)$ denotes the difference $\tilde{e}_a(i) = d(i) - \mathbf{u}_i\mathbf{w}_{i-1}$, where $\mathbf{w}_{i-1}$ is an estimate for $\mathbf{w}$ that is generated as follows. Starting with an initial guess $\mathbf{w}_{-1}$, the FxLMS algorithm provides recursive estimates of $\mathbf{w}$ via the update relation (see[2]-[3]):

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu(i)\, F[\mathbf{u}_i^*]\, F[d(i) - \mathbf{u}_i\mathbf{w}_{i-1}], \qquad (1)$$

where the $\{\mu(i)\}$ are time-variant step-sizes.

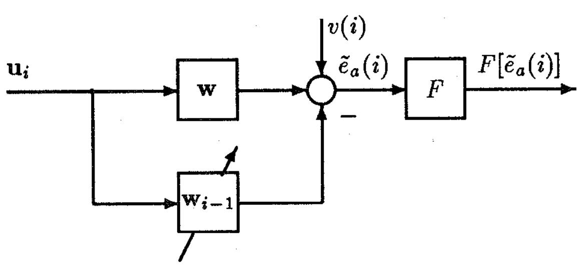

Figure 3: *Set-up for FxLMS.*

The following error quantities are useful for our later analysis: $\tilde{\mathbf{w}}_i$ denotes the difference between the true weight $\mathbf{w}$ and its estimate $\mathbf{w}_i$, $\tilde{\mathbf{w}}_i = \mathbf{w} - \mathbf{w}_i$ and $e_a(i)$ denotes the *a priori* estimation error, $e_a(i) = \mathbf{u}_i\tilde{\mathbf{w}}_{i-1}$.

3 A Generalized FxLMS Algorithm

For the sake of generality, and for reasons to become clear later, we study the more general update form:

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu(i) F[\mathbf{u}_i^*] G(i, q^{-1}) F[d(i) - \mathbf{u}_i\mathbf{w}_{i-1}], \qquad (2)$$

where a time-variant filter $G(i, q^{-1})$ has been included in the update relation (compare with (1)). We shall show in the sequel how to choose $G(i, q^{-1})$ in order to improve the convergence performance of (1).

But for now we first show that (2) can be rewritten in a filtered-error form (see (6) and (7) further ahead). For this purpose, we use (2) to conclude that

$$\tilde{\mathbf{w}}_i = \tilde{\mathbf{w}}_{i-1} - \mu(i) F[\mathbf{u}_i^*]\, G(i, q^{-1})\, F[\tilde{e}_a(i)]\,, \qquad (3)$$

which allows us to express $\tilde{\mathbf{w}}_i$ in terms of $\tilde{\mathbf{w}}_{i-1-p}$, for some p,

$$\tilde{\mathbf{w}}_i = \tilde{\mathbf{w}}_{i-1-p} - \sum_{k=0}^{p} \mu(i-k) F[\mathbf{u}_{i-k}^*]\, G(i-k, q^{-1})\, F[\tilde{e}_a(i-k)]$$

or, in a form more suitable for our investigation,

$$\tilde{\mathbf{w}}_{i-1-p} = \tilde{\mathbf{w}}_{i-1} + \sum_{k=1}^{p} \mu(i-k) F[\mathbf{u}_{i-k}^*] G(i-k, q^{-1}) F[\tilde{e}_a(i-k)].$$

Now using the fact that $\tilde{e}_a(i) = v(i) + e_a(i)$, along with the linearity of F, we obtain

$$\begin{aligned} F[\tilde{e}_a(i)] &= F[v(i)] + F[\mathbf{u}_i]\tilde{\mathbf{w}}_{i-1} \\ &\quad + \sum_{k=1}^{M_F-1} c(i,k)\, G(i-k, q^{-1})\, F[\tilde{e}_a(i-k)]\,, \end{aligned}$$

where the coefficients $c(i,k)$ have been defined by

$$c(i,k) = \mu(i-k) F_k[\mathbf{u}_i] F[\mathbf{u}_{i-k}^*], \qquad (4)$$

for $k = 1, .., M_F - 1$, and where $F_k[\cdot]$ denotes the following filter: $F_k[\mathbf{u}_i] = \sum_{j=k}^{M_F-1} f_j\, \mathbf{u}_{i-j}$.

We therefore conclude that

$$F[\tilde{e}_a(i)] = \frac{1}{1 - C(i, q^{-1}) G(i, q^{-1})} \left[F[v(i)] + F[\mathbf{u}_i]\tilde{\mathbf{w}}_{i-1}\right], \quad (5)$$

which allows us to rewrite the weight-error update equation (3) in the equivalent form

$$\tilde{\mathbf{w}}_i = \tilde{\mathbf{w}}_{i-1} - \mu(i) F[\mathbf{u}_i^*] \frac{G(i, q^{-1})\, [F[v(i)] + F[\mathbf{u}_i]\tilde{\mathbf{w}}_{i-1}]}{1 - C(i, q^{-1}) G(i, q^{-1})}. \quad (6)$$

This equation is of the filtered-error type, as claimed earlier. In other words, if we introduce the new signals

$$v'(i) \leftarrow F[v(i)], \;\; \mathbf{u}'(i) \leftarrow F[\mathbf{u}_i], \;\; d'(i) \leftarrow \mathbf{u}_i'\mathbf{w} + v'(i),$$

then expression (6) corresponds to the weight-error update of the following algorithm:

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu(i) \mathbf{u}_i'^{*}\, M(i, q^{-1})[d'(i) - \mathbf{u}_i'\mathbf{w}_{i-1}], \quad (7)$$

where

$$M(i, q^{-1}) = G(i, q^{-1}) / (1 - C(i, q^{-1}) G(i, q^{-1})).$$

Algorithm (7) is a filtered-error algorithm.

4 An Optimal Choice for $G(i, q^{-1})$

It follows from (6) that the update equation for the generalized FxLMS recursion (2) is (7). If $M(i, q^{-1})$ were equal to one, then (7) would have exactly the same structure as an LMS update. In this case, the convergence performance of the modified algorithm would be similar in nature to that of an LMS algorithm (and superior to the original FxLMS update (1)).

The condition $M(i, q^{-1}) = 1$ can be met exactly, or approximately, in different ways as we now explain.

4.1 The MFxLMS Algorithm

One way is the so-called Modified FxLMS algorithm, recently introduced in [1, 5]. It employs the following update:

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu(i)F[\mathbf{u}_i^*]\left(F[\tilde{e}_a(i)] + F[\mathbf{u}_i\mathbf{w}_{i-1}] - F[\mathbf{u}_i]\mathbf{w}_{i-1}\right). \tag{8}$$

The extra terms that are added in (8) to the original update recursion (1) have the effect of guaranteeing $M(i, q^{-1}) = 1$ in (7) since it can be verified that (using $F[\mathbf{u}_i\mathbf{w}] = F[\mathbf{u}_i]\mathbf{w}$)

$$F[\tilde{e}_a(i)] + F[\mathbf{u}_i\mathbf{w}_{i-1}] - F[\mathbf{u}_i]\mathbf{w}_{i-1} = F[v(i)] + F[\mathbf{u}_i]\tilde{\mathbf{w}}_{i-1}. \tag{9}$$

The additional terms correspond to filtering the input data $\mathbf{u}_i$ and the signal $\mathbf{u}_i\mathbf{w}_{i-1}$ by F; thus amounting to an increase in the computational complexity from $2M$ (as in the original FxLMS) to $3M$ operations per time step.

An alternative interpretation for the MFxLMS algorithm (8) is to note that it corresponds to employing a filter, say $G_o(i, q^{-1})$, such that $G_o/(1 - CG_o) = 1$, or, equivalently,

$$G_o(i, q^{-1}) = 1/(1 + C(i, q^{-1})). \tag{10}$$

This means that the MFxLMS recursion (8) can be equivalently rewritten in the form (2), viz.,

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu(i)F[\mathbf{u}_i^*]\frac{1}{1 + C(i, q^{-1})}F[d(i) - \mathbf{u}_i\mathbf{w}_{i-1}]. \tag{11}$$

4.2 The MFxLMS-1 Algorithm

We now propose two new modifications with lower computational requirements than the MFxLMS algorithm. They are based on approximating the optimal choice $G_o(i, q^{-1})$ with the intent of reducing the computational count to $2M$ operations per time step, as in the original FxLMS recursion (1).

The first modification, referred to as MFxLMS-1, replaces the time-variant coefficients $c(i, l)$ in (10) by constant approximations. This is especially useful when statistical information is available.

In particular, assume that the input sequence $u(i)$ is stationary with autocorrelation function $r_i = E[u(k)u^*(k-i)]$. If the process is ergodic and the order M of the input vector $\mathbf{u}_i$ (with shift structure) is sufficiently large, the terms $\mathbf{u}_{i-p}\mathbf{u}_i^*$ can be approximated by $\mathbf{u}_{i-p}\mathbf{u}_i^* \approx Mr_p$.

We further assume that the time-variant step size $\mu(i)$ in (11) is chosen as

$$\mu(i) = \alpha/\|F[\mathbf{u}_i]\|_2^2, \tag{12}$$

which is known as the projection step-size. The term $\|F[\mathbf{u}_i]\|_2^2$ in (12) can be approximated by

$$\|F[\mathbf{u}_i]\|_2^2 \approx M\sum_{i=0}^{M_F-1}\sum_{j=0}^{M_F-1} r_{i-j}\, f_i\, f_j, \tag{13}$$

and the filter coefficients $c(i, k)$ in (4) can also be approximated by $c(i, k) \approx \alpha\, \bar{c}(k)$, where we have defined the averaged coefficients

$$\bar{c}(k) = \frac{\sum_{i=k}^{M_F-1}\sum_{j=0}^{M_F-1} r_{i-j} f_i f_j}{\sum_{i=0}^{M_F-1}\sum_{j=0}^{M_F-1} r_{i-j}\, f_i\, f_j}. \tag{14}$$

These coefficients depend only on the error-filter F and on the autocorrelation coefficients $\{r_i\}$. They can therefore be computed in advance, assuming knowledge of the error filter F. Note also that in the special case of a white random process $u(i)$ with variance σ_u^2, the expression for $\bar{c}(k)$ can be further simplified since $\|F[\mathbf{u}_i]\|_2^2$ can be approximated by $M\sigma_u^2\sum_{i=0}^{M_F-1} f_i^2$ and, correspondingly,

$$\bar{c}(k) = \frac{\sum_{i=0}^{M_F-k-1} f_i f_{i+k}}{\sum_{i=0}^{M_F-1} f_i^2}. \tag{15}$$

Once $\bar{C}(q^{-1}) = \sum_{k=1}^{M_F-1}\bar{c}(k)q^{-k}$ is evaluated, the $\mathbf{w}_i$ is updated via

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu(i)F[\mathbf{u}_i^*]\frac{1}{1 + \bar{C}(q^{-1})}F[d(i) - \mathbf{u}_i\mathbf{w}_{i-1}],$$

where $\mu(i)$ is given by (12) and (13).

The above solution requires of the order of $2M$ computations per time step. It however requires exact (or approximate) knowledge of the autocorrelation function of the input process. If this is not available, estimates for r_i can be calculated (e.g., by sample covariances) and the optimal coefficients can be computed at every time instant via (14). However, the final computational load of the algorithm may exceed $3M$ depending on how the coefficients are estimated. For this reason, we suggest here a second modification that might be more appropriate in such cases.

4.3 The MFxLMS-2 Algorithm

We have shown in Sec. 4.1 that the MFxLMS algorithm can be written in two equivalent forms. The first one is (recall (8) and (9)):

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu(i)F[\mathbf{u}_i^*]\,\bar{e}(i), \tag{16}$$

$$\bar{e}(i) = F[v(i)] + F[\mathbf{u}_i]\tilde{\mathbf{w}}_{i-1}, \tag{17}$$

and the second one is (11).

The difference between both representations is that the second one operates directly on the available signal $F[\tilde{e}_a(i)]$ by filtering it through $1/(1 + C)$, while the first one modifies the update equation as in (8) and uses the additional filtering operations $F[\mathbf{u}_i\mathbf{w}_{i-1}]$ and $F[\mathbf{u}_i]\mathbf{w}_{i-1}$. The net result, however, is the same since we already know that both representations are equivalent and, in particular, that

$$\bar{e}(i) = \frac{1}{1 + C(i, q^{-1})}F[\tilde{e}_a(i)] = G_o(i, q^{-1})F[\tilde{e}_a(i)]. \tag{18}$$

If we knew $\bar{e}(i)$ then it could be used in the update form (16), without the explicit need for the additional filtering operations of (8).

This suggests the following modification. We have in (18) a relation between $\bar{e}(i)$ and $1/(1+C)$. The only known quantity in (18) is $F[\tilde{e}_a(i)]$, and we can rewrite the expression in the form $\bar{e}(i) = F[\tilde{e}_a(i)] - C(i,q^{-1})\bar{e}(i)$. Since $\bar{e}(i)$ is unknown, we need an estimate for it, say $\hat{\bar{e}}(i)$. The above equation can then be replaced by $\hat{\bar{e}}(i) = F[\tilde{e}_a(i)] - C(i,q^{-1})\hat{\bar{e}}(i)$. An approximate solution would be to use a gradient-type algorithm to estimate both $C(i,q^{-1})$ and $\hat{\bar{e}}(i)$:

$$\hat{\bar{e}}(i) = F[\tilde{e}_a(i)] - \sum_{k=1}^{M_F-1} \hat{c}(i-1,k)\hat{\bar{e}}(i-k), \quad (19)$$

$$\hat{c}(i,k) = \hat{c}(i-1,k) + \frac{\hat{\bar{e}}(i)\hat{\bar{e}}(i-k)}{1+\sum_{k=1}^{M_F-1}|\hat{\bar{e}}(i-k)|^2}. \quad (20)$$

Once the $\hat{\bar{e}}(i)$ is evaluated, it is used in $\mathbf{w}_i = \mathbf{w}_{i-1} + \mu(i)F[\mathbf{u}_i^*]\,\hat{\bar{e}}(i)$. The reader may refer back to Table 1, where the different variants of the FxLMS algorithm are compared in terms of computational complexity and storage requirements, as well as in terms of their convergence behaviour as suggested by the simulation results at the end of this paper.

5 Robustness/l_2-Stability

Another point to remark here is that once the generalized FxLMS algorithm (2) is written in filtered-error form (7), the feedback analysis of [6] (see also [7]) can be applied to it. This allows us to provide conditions on the step-size parameters $\mu(i)$ in order to guarantee i) robustness and ii) improved convergence.

Intuitively, a robust algorithm is one for which the estimation errors (say $e_a(i)$) are consistent with the disturbances (or noise) in the sense that "small" disturbances would lead to "small" estimation errors, no matter what the disturbances (and their statistics) are! This is not generally true for any adaptive filter: the estimation errors can still be large even in the presence of small disturbances.

Following the analysis of [6], an l_2-stable (or robust) mapping from the (filtered) disturbance signals $\{\tilde{\mathbf{w}}_{-1}, \sqrt{\mu(\cdot)}\, v'(\cdot)\}$ to the (filtered) estimation errors $\{\sqrt{\mu(\cdot)}\, e_a'(\cdot)\}$ can be guaranteed by the generalized FxLMS recursion (2) if a certain contractivity requirement is met. This condition can be stated as follows. Define $\bar{\mu}(i) = 1/\|F[\mathbf{u}_i]\|_2^2$, as well as the diagonal matrices $\mathbf{M}_N = \text{diag}(\mu(0), \mu(1), ..., \mu(N))$ and $\overline{\mathbf{M}}_N = \text{diag}(\bar{\mu}(0), \bar{\mu}(1), ..., \bar{\mu}(N))$. Let also $\mathbf{C}_N$ and $\mathbf{G}_N$ be lower triangular band matrices that describe the action of the linear time-variant filters $C(i,q^{-1})$ and $G(i,q^{-1})$. If we denote the 2-induced norm (i.e., maximum singular value) of a matrix by $\|\cdot\|_{2,ind}$, then the robustness condition is to require

$$\left\| \mathbf{I} - \mathbf{M}_N \overline{\mathbf{M}}_N^{-\frac{1}{2}} \mathbf{G}_N(\mathbf{I}-\mathbf{C}_N\mathbf{G}_N)^{-1}\overline{\mathbf{M}}_N^{-\frac{1}{2}} \right\|_{2,ind} < 1,$$

In the special case of the original FxLMS algorithm (1), which corresponds to $G=1$, and for a constant step-size $\mu(i) = \mu$, the above condition collapses to requiring the following:

$$\left\| \mathbf{I} - \frac{\mu}{\bar{\mu}}(\mathbf{I}-\mathbf{C}_N)^{-1} \right\|_{2,ind} < 1 \quad (21)$$

where $\bar{\mu}$ is such that $\bar{\mu} \le \min_{0\le i\le N}\{1/\|F[\mathbf{u}_i]\|_2^2\}$.

Likewise, under the same conditions of Sec. 4.2, with $\mu(i) = \alpha/\|F[\mathbf{u}_i]\|_2^2$ and replacing $c(i,k)$ by the averaged coefficients $\bar{c}(k)$, we can show that an approximate stability (robustness) condition is

$$\max_{\Omega}\left| 1 - \frac{\alpha}{1-\alpha\bar{C}(e^{j\Omega})}\right| < 1. \quad (22)$$

This condition can also be shown (cf. [6]) to suggest a choice for α for faster convergence for FxLMS, viz.,

$$\alpha_{opt} = \min_{\alpha}\max_{\Omega}\left| 1 - \frac{\alpha}{1-\alpha\bar{C}(e^{j\Omega})}\right|. \quad (23)$$

6 Simulation Results

In all experiments we have chosen a Gaussian white random sequence with variance one as the input signal $u(i)$, and the additive noise was set at $-60dB$ below the input power. We provide plots of learning curves for the relative system mismatch, defined as $S_{rel}(i) = \|\tilde{\mathbf{w}}_i\|_2^2/\|\tilde{\mathbf{w}}_{-1}\|_2^2$. The curves are averaged over 50 Monte Carlo runs in order to approximate $\text{E}[S_{rel}(i)]$. The results in the figures are also indicated in dB. In all experiments we employed the projection normalization (12).

6.1 The Delayed LMS Algorithm

In our first example, a transversal filter of order $M=10$ is to be identified in the case of a pure delay filter $F(q^{-1}) = q^{-4}$. The FxLMS algorithm in this case corresponds to the so-called Delayed LMS:

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \frac{\alpha}{\|\mathbf{u}_{i-4}\|_2^2}\mathbf{u}_{i-4}^*\,[d(i-4) - \mathbf{u}_{i-4}\mathbf{w}_{i-5}].$$

The curve for the standard LMS algorithm with projection step-size is also given as a comparison, viz.,

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \frac{1}{\|\mathbf{u}_i\|_2^2}\mathbf{u}_i^*\,[d(i) - \mathbf{u}_i\mathbf{w}_{i-1}].$$

As Figure 4 shows, the delay causes a degradation in the convergence behaviour of DLMS algorithm.

In a second experiment, the modified version of the DLMS algorithm, using the optimal $G_o(i,q^{-1})$ as suggested by (10), has been used

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \frac{\alpha}{\|\mathbf{u}_{i-4}\|_2^2}\mathbf{u}_{i-4}^*\,\frac{1}{1+C(i,q^{-1})}[d(i-4) - \mathbf{u}_{i-4}\mathbf{w}_{i-5}],$$

where,

$$C(i,q^{-1}) = \alpha\left[\frac{\mathbf{u}_{i-4}\mathbf{u}_{i-5}^*}{\|\mathbf{u}_{i-5}\|_2^2}q^{-1} + \frac{\mathbf{u}_{i-4}\mathbf{u}_{i-6}^*}{\|\mathbf{u}_{i-6}\|_2^2}q^{-2} + \frac{\mathbf{u}_{i-4}\mathbf{u}_{i-7}^*}{\|\mathbf{u}_{i-7}\|_2^2}q^{-3} + \frac{\mathbf{u}_{i-4}\mathbf{u}_{i-8}^*}{\|\mathbf{u}_{i-8}\|_2^2}q^{-4}\right].$$

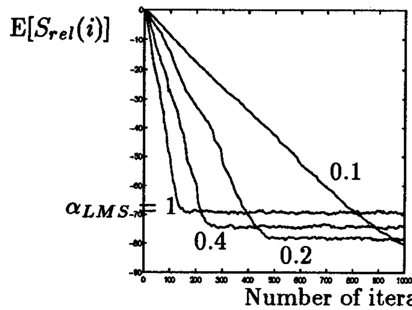

Figure 4: *Learning curves for DLMS algorithm* $\alpha = 0.1, 0.2, 0.4$ *in comparison to LMS* $\alpha = 1.0$.

This is of course equivalent to a MFxLMS form,

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \frac{\alpha}{\|\mathbf{u}_{i-4}\|_2^2}\mathbf{u}_{i-4}^* \left(d(i-4) - \mathbf{u}_{i-4}\mathbf{w}_{i-1}\right).$$

As Figure 5 demonstrates, the modification restores the convergence performance of the algorithm to a level comparable to the standard LMS case; the learning curves of the modified DLMS algorithm and the LMS algorithm almost coincide. A second curve for $\alpha = 1.5$ is given, a step-size for which the conventional DLMS algorithm was already unstable.

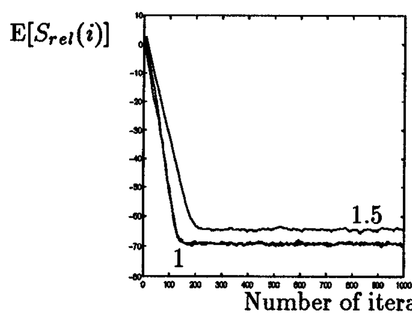

Figure 5: *Learning curves for Modified DLMS algorithm* $\alpha = 1$ *and* $\alpha = 1.5$

6.2 The FxLMS Algorithm

Another simulation was performed with the intent of identifying a 20th order filter ($M = 20$) with the error filter path being now given by $F(q^{-1}) = 1 + q^{-1} + q^{-2} + q^{-3}$, indicating a low pass behaviour as it is common in acoustic ducts. The coefficients of the corresponding averaged filter $\bar{C}(q^{-1})$ were given by $\bar{c}_1 = 0.75$, $\bar{c}_2 = 0.5$, $\bar{c}_3 = 0.25$, i.e., $\bar{C}(q^{-1}) = 0.75q^{-1} + 0.5q^{-2} + 0.25q^{-3}$.

If we use the above averaged coefficients as approximations, we obtain an approximate stability range for the MFxLMS-1 algorithm at $0 < \alpha < 0.5$ (recall (22)); the optimal convergence speed is attained at $\alpha_o = 0.45$ (recall (23)). In the simulations that were carried out, the results were very close to these values with a stability bound at 0.57 and fastest convergence at 0.5. In particular, the optimal step-size from [2] for this case is 0.8333, which is already in the unstable region.

As Figure 6 shows, the average filter solution that corresponds to the proposed version MFxLMS-1 leads to a learning curve (indicated by the letter (c)) which is close to the optimal one (i.e., the one that corresponds to the MFxLMS recursion and is indicated by the letter (d) in the figure).

The figure also indicates the result of the second modification MFxLMS-2 (curve (b)), which is appropriate when the statistics of the input sequence is not known apriori. While curve (b) is less appealing than the curves (c) and (d), it nevertheless improves on the convergence of the original FxLMS recursion, which is indicated by curve (a). The optimal convergence speed for the MFxLMS-2 algorithm was found for $\alpha_o = 1.15$ and stability bound at 1.3. A fifth learning curve for the LMS algorithm, with $\mathbf{u}_i$ and $v(i)$ prefiltered by F, is not explicitly shown in the figure since it essentially coincides with the MFxLMS algorithm (curve (d)).

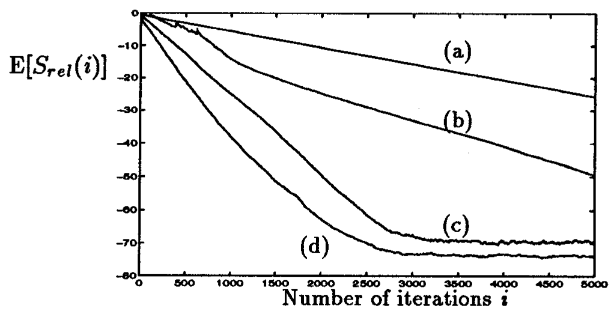

Figure 6: *Learning curves for FxLMS algorithm with* $\alpha = 0.5$ *(a) and modifications: MFxLMS-2 (b)* $\alpha = 1.15$ *and MFxLMS-1 (c)* $\alpha = 1.2$ *in comparison to MFxLMS (d)* $\alpha = 1.2$.

References

[1] E. Bjarnason, "Noise cancellation using a modified form of the filtered–XLMS algorithm," in *Proc. Eusipco Signal Processing V*, Brüssel, 1992.

[2] S.J. Elliott and P.A. Nelson, *Active noise control*, IEEE Signal Processing Magazine, Oct. 1993.

[3] L.J. Eriksson, "Development of the filtered-U algorithm for active noise control," *J. Acoust. Soc. Am.*, vol. 89, no. 1, pp. 257–265, Apr. 1991.

[4] J. R. Glover, Jr. , "Adaptive noise canceling applied to sinusoidal interferences," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 25, no. 6., pp. 484–491, Dec. 1977.

[5] I. Kim, H. Na, K. Kim and Y. Park, "Constraint filtered-x and filtered-u algorithms for the active control of noise in a duct," *J. Acoust. Soc. Am.*, vol. 95, no. 6, pp. 3397-3389, June 1994.

[6] A. H. Sayed and M. Rupp, "A time-domain feedback analysis of adaptive gradient algorithms via the small gain theorem," *Proc. SPIE Conference on Advanced Signal Processing*, vol. 2563, pp. 458–469, San Diego, CA, July 1995.

[7] A. H. Sayed and M. Rupp, "A feedback analysis of perceptron learning for neural networks", *Proc. this conference* (Asilomar 95).

[8] B. Widrow and S.D. Stearns, *Adaptive Signal Processing*, Englewood Cliffs, NJ: Prentice–Hall, Signal Processing Series, 1985.