Robust FxLMS Algorithms with Improved Convergence Performance

Markus Rupp and Ali H. Sayed, *Member, IEEE*

***Abstract*—This paper proposes two modifications of the filtered-x least mean squares (FxLMS) algorithm with improved convergence behavior 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 algorithms are all of filtered-error form. A choice of the stepsize parameter that guarantees faster convergence and conditions for robustness are also derived. Several simulation results are included to illustrate the discussions.**

***Index Terms*— Active noise control, feedback analyses, LMF algorithm, l_2-stability, robustness.**

I. Introduction

A widely used algorithm in active noise control is the *filtered-x least-mean-squares* (FxLMS) algorithm [1]–[4]. It can be motivated by referring to the simple noise control system depicted in Fig. 1. 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.

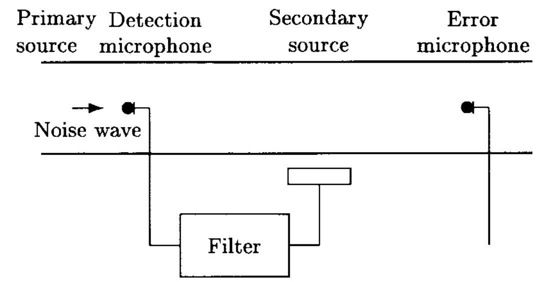


Fig. 1. Sketch of a simple active noise control system in a duct.

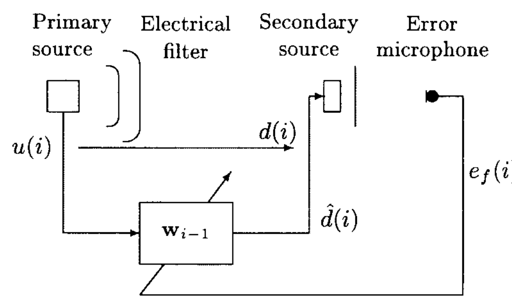


Fig. 2. Active noise control system using feedforward control.

Fig. 2 is a redrawing of the duct of Fig. 1, with emphasis on the particular structure of the adaptive antinoise generator. The figure 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 antinoise sequence $\hat{d}(i)$ is generated by a finite impulse response (FIR) filter of length M at the secondary source with the intent of canceling $d(i)$. 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 $(d(i), \hat{d}(i))$ have to further travel a path before reaching the right-most (error) microphone. This path is usually unknown, and the objective is to update the filter weights (denoted by $\boldsymbol{w}$) 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 stepsizes for stabilization purposes. This has the obvious disadvantage of slow adaptation and convergence.

The FxLMS algorithm (to be described further ahead) is a recursive procedure that has been suggested for the update of the adaptive weight estimate $\boldsymbol{w}_{i-1}$ (see, e.g., [1], [4], or [15]). It requires $2M$ operations per time step and has been shown to exhibit poor convergence behavior. 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 $3M$ operations (additions and multiplications) per time step [5], [6]. This figure is still prohibitive in several applications since the value of M can be significantly large.

Motivated by these facts, we invoke the feedback approach of [7] and [8], and use it to study both the robustness and convergence performance of these algorithms, and of new variants, within a purely deterministic framework. The analysis is carried out with two objectives in mind.

The first objective is to provide conditions on the stepsize parameter in order to guarantee robust performance (along the

Manuscript received March 31, 1995; revised March 13, 1997. This work was supported by the NSF under Award MIP-9796147. The work of M. Rupp was supported by a scholarship from the German Academic Exchange Service and the Scientific Division of NATO. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. James H. Snyder.

M. Rupp is with the Wireless Technology Research Department, Lucent Technologies, Holmdel, NJ 07733-0400 USA.

A. H. Sayed is with the Department of Electrical Engineering, University of California, Los Angeles, CA 90024 USA (e-mail: sayed@ee.ucla.edu).

Publisher Item Identifier S 1063-6676(98)00587-2.

TABLE I
COMPARISON OF DIFFERENT VARIANTS

Algorithm	Complexity ($M \gg M_F$)	Memory Capacity	Convergence Behavior
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

lines of H_∞ theory) of the FxLMS algorithm in the presence of disturbances and modeling uncertainties (or errors). Intuitively, a robust filter is one for which the estimation errors are consistent with the disturbances in the sense that "small" disturbances would lead to "small" estimation errors. This is not generally true for any adaptive filter: the estimation errors may still be relatively large even in the presence of "small" disturbances. A robust design would guarantee that the ratio of estimation error energy to disturbance energy will be bounded by a positive constant, say the constant one

$$\frac{\text{estimation error energy}}{\text{disturbance energy}} \leq 1. \tag{1}$$

A relation of form (1) is desirable from a practical point of view, since it guarantees that the resulting estimation error energy will be at most equal to the disturbance energy, no matter what the nature and the statistics of the disturbances are. In this sense, the algorithm will not unnecessarily magnify the disturbance energy and, consequently, small estimation errors will result when small disturbances occur. Hence, robust designs are useful in situations where prior statistical information is missing, since it would guarantee a desired level of robustness independent of the statistical nature of the noise and signals.

Our second objective is to suggest choices for the stepsize parameter in order to guarantee, in addition to robustness, an improved convergence speed. While the modified version of the FxLMS mentioned above—and discussed in [5] and [6]—exhibits improved convergence performance over the conventional FxLMS algorithm (to be described in the next section), it nevertheless achieves this improved performance at an increased computational cost of $3M$ computations per iteration. The modifications proposed in this work will lead to improved convergence but still at the same computational cost of $2M$ computations per iteration.

In particular, as a result of our analysis, we shall propose two modifications to the FxLMS algorithm. The results are summarized in Table I 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.

Finally, we should stress that the analysis carried out in this paper is significantly different, both in scope and objectives, from earlier works in the literature on filtered error algorithms (especially [9]). Reference [9], and many of the references therein, are primarily interested in conditions under which adaptive algorithms are guaranteed to be exponentially asymptotically stable in the noise-free case. The derivations in these references usually invoke results from averaging (and ODE) analysis [10]–[12] and their conclusions only hold under the assumptions of *very* small stepsizes and persistently exciting regressors. However, it is always desirable to be able to quantify how "large" or how "small" the step size, and other relevant quantities, can be, and such quantification is usually difficult to pursue in these frameworks (as explicitly stated on p. 397 of [9]).

In this paper, we are not explicitly interested in the exponential convergence of the FxLMS adaptive scheme and its variants, but rather in how reasonably they perform in the presence of both disturbances and modeling errors. For this purpose, we pursue a feedback analysis that allows us to quantify how large or how small the stepsize should be in order to guarantee a certain level of performance in the face of ever present disturbances. The analysis also allows us to suggest choices for the step size, as well as algorithm modifications, in order to improve the convergence and robustness performance (see also the studies in [13] and the last section of [14]).

A. Notation

We use small boldface letters to denote vectors and capital boldface letters to denote matrices. Also, the symbol "*" denotes Hermitian conjugation (complex conjugation for scalars). The symbol $\boldsymbol{I}$ denotes the identity matrix of appropriate dimensions, and the boldface letter $\mathbf{0}$ denotes either a zero vector or a zero matrix. The notation $\|\boldsymbol{x}\|$ denotes the Euclidean norm of a vector. All vectors are column vectors except for the input data vector denoted by $\boldsymbol{u}_i$, which is taken to be a row vector. We further employ the shift operator notation $q^{-1}u(k) = u(k-1)$. Hence, applying an operator $W(q^{-1}) = \Sigma_{k=1}^{M} w_k q^{-k}$ to a sequence $d(i)$ means $W(q^{-1})d(i) = \Sigma_{k=1}^{M} w_k d(i-k)$.

B. The FxLMS Algorithm

The set-up for the FxLMS algorithm is depicted in Fig. 3. Let $\boldsymbol{w}$ be an unknown weight vector and assume $\{d(i)\}$ are noisy measurements that are related to $\boldsymbol{w}$ via

$$d(i) = \boldsymbol{u}_i \boldsymbol{w} + v(i). \tag{2}$$

Here, the $\{\boldsymbol{u}_i\}$ 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

$$\tilde{e}_a(i) = d(i) - \boldsymbol{u}_i \boldsymbol{w}_{i-1} \tag{3}$$

where $\boldsymbol{w}_{i-1}$ is an estimate for $\boldsymbol{w}$ that is generated as follows. Starting with an initial guess $\boldsymbol{w}_{-1}$, the FxLMS algorithm provides recursive estimates for $\boldsymbol{w}$ via the update relation (see [1], [15]):

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu(i) F[\boldsymbol{u}_i^*] F[d(i) - \boldsymbol{u}_i \boldsymbol{w}_{i-1}] \tag{4}$$

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

The following error quantities are useful for our later analysis: $\tilde{\boldsymbol{w}}_i$ denotes the difference between the true weight

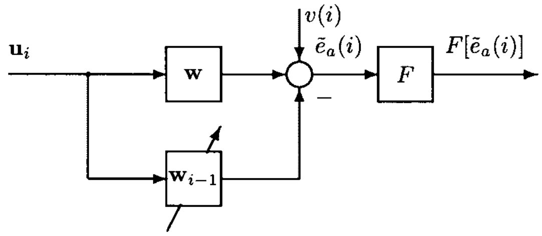


Fig. 3. Set-up for FxLMS.

$\boldsymbol{w}$ and its estimate $\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i = \boldsymbol{w} - \boldsymbol{w}_i$ and $e_a(i)$ denotes the *a priori* estimation error, $e_a(i) = \boldsymbol{u}_i\tilde{\boldsymbol{w}}_{i-1}$.

II. A Generalized FxLMS Algorithm

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

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu(i)F[\boldsymbol{u}_i^*]G(i,q^{-1})F[d(i) - \boldsymbol{u}_i\boldsymbol{w}_{i-1}] \tag{5}$$

where a time-variant filter $G(i,q^{-1})$ has been included in the update relation [compare with (4)]. We shall show in the sequel how to choose $G(i,q^{-1})$ in order to improve the convergence performance of (4). But first we show that (5) can be rewritten in a filtered-error form [see (11) and (12) further ahead].

For this purpose, we note from (5) that

$$\tilde{\boldsymbol{w}}_i = \tilde{\boldsymbol{w}}_{i-1} - \mu(i)F[\boldsymbol{u}_i^*]G(i,q^{-1})F[\tilde{e}_a(i)] \tag{6}$$

which allows us to express $\tilde{\boldsymbol{w}}_i$ in terms of $\tilde{\boldsymbol{w}}_{i-1-p}$, for any $p \geq 0$, as follows:

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

or, in a form more suitable for our investigation

$$\tilde{\boldsymbol{w}}_{i-1-p} = \tilde{\boldsymbol{w}}_{i-1} + \sum_{k=1}^{p} \mu(i-k)F[\boldsymbol{u}_{i-k}^*]G(i-k,q^{-1}) \cdot F[\tilde{e}_a(i-k)]. \tag{7}$$

Now using the fact that

$$\tilde{e}_a(i) = \boldsymbol{u}_i\boldsymbol{w} + v(i) - \boldsymbol{u}_i\boldsymbol{w}_{i-1} = v(i) + e_a(i)$$

along with the assumed linearity of F, we can show (a proof is provided in Appendix A) that

$$F[\tilde{e}_a(i)] = F[v(i)] + F[\boldsymbol{u}_i]\tilde{\boldsymbol{w}}_{i-1} + \sum_{k=1}^{M_F-1} \cdot c(i,k)G(i-k,q^{-1})F[\tilde{e}_a(i-k)] \tag{8}$$

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

$$c(i,k) = \mu(i-k)F_k[\boldsymbol{u}_i]F[\boldsymbol{u}_{i-k}^*] \tag{9}$$

for $k = 1, \cdots, M_F - 1$, and where the notation $F_k[\cdot]$ denotes the following filter

$$F_k[\boldsymbol{u}_i] = \sum_{j=k}^{M_F-1} f_j\boldsymbol{u}_{i-j}.$$

(Note that the lower index starts at k).

We therefore conclude that

$$F[\tilde{e}_a(i)] = \frac{1}{1 - C(i,q^{-1})G(i,q^{-1})}[F[v(i)] + F[\boldsymbol{u}_i]\tilde{\boldsymbol{w}}_{i-1}] \tag{10}$$

which allows us to rewrite the weight-error update equation (6) in the equivalent form

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

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

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

then expression (11) corresponds to the weight-error update of the following algorithm:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu(i)\boldsymbol{u}_i'^*Q(i,q^{-1})[d'(i) - \boldsymbol{u}_i'\boldsymbol{w}_{i-1}] \tag{12}$$

where

$$Q(i,q^{-1}) = \frac{G(i,q^{-1})}{1 - C(i,q^{-1})G(i,q^{-1})}.$$

Recursion (12) is a filtered-error algorithm. All we have done so far is to show that the generalized form that we introduced in (5) can be rewritten in the alternative form (12). This alternative form involves only filtering of the error signal $d'(i) - \boldsymbol{u}_i'\boldsymbol{w}_{i-1}$ by Q, but not of the regressor $\boldsymbol{u}_i'^*$. Note that this equivalence has been established without any approximations.

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

Once this equivalent rewriting of recursion (5) has been established, we now note that if $Q(i,q^{-1})$ were equal to one, then (12) would have exactly the same structure as a standard LMS update. In this case, the convergence performance of (12) would be similar in nature to that of an LMS algorithm [and, hence, superior to the original FxLMS update (4)].

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

B. The MFxLMS Algorithm

Different choices for $Q(i,q^{-1})$ would correspond to different choices for $G(i,q^{-1})$ in (5) and, hence, to different modifications of the original FxLMS update (4).

We now verify that a recent modification of the FxLMS update (4), which we henceforth refer to it as the modified FxLMS algorithm (MFxLMS) [5], [6], can be interpreted as providing one such particular choice for $Q(i,q^{-1})$.

More specifically, the MFxLMS algorithm employs the following update:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu(i)F[\boldsymbol{u}_i^*](F[\tilde{e}_a(i)] + F[\boldsymbol{u}_i\boldsymbol{w}_{i-1}] - F[\boldsymbol{u}_i]\boldsymbol{w}_{i-1}). \tag{13}$$

The extra terms that are added in (13) to the original update recursion (4) have the precise effect of guaranteeing $Q(i,q^{-1}) = 1$ in (12). This can be verified as follows. First note that since $\boldsymbol{w}$ is a fixed vector [it is the true vector assumed in the model (2)] we have

$$F[\boldsymbol{u}_i\boldsymbol{w}] = F[\boldsymbol{u}_i]\boldsymbol{w}. \tag{14}$$

Moreover, by linearity of F and since $\tilde{e}_a(i) = e_a(i) + v(i)$, we obtain

$$\begin{aligned} F[\tilde{e}_a(i)] &= F[\boldsymbol{u}_i\boldsymbol{w}] - F[\boldsymbol{u}_i\boldsymbol{w}_{i-1}] + F[v(i)] \\ &= F[\boldsymbol{u}_i]\boldsymbol{w} - F[\boldsymbol{u}_i\boldsymbol{w}_{i-1}] + F[v(i)] \end{aligned} \tag{15}$$

where we used (14) in the last equality. Using (15), we can now express the sum that appears in the update relation (13) as follows:

$$F[\tilde{e}_a(i)] + F[\boldsymbol{u}_i\boldsymbol{w}_{i-1}] - F[\boldsymbol{u}_i]\boldsymbol{w}_{i-1} = F[v(i)] + F[\boldsymbol{u}_i]\tilde{\boldsymbol{w}}_{i-1}. \tag{16}$$

The additional terms in (13) correspond to filtering the input data $\boldsymbol{u}_i$ and the signal $\boldsymbol{u}_i\boldsymbol{w}_{i-1}$ by F, thus amounting to an increase in computational complexity from $2M$ (as in the original FxLMS) to $3M$ operations per time step.

An alternative interpretation for the MFxLMS algorithm (13) is to note that it corresponds to employing a filter, say $G_o(i, q^{-1})$, in (5) such that $G_o/(1-CG_o) = 1$ or, equivalently

$$G_o(i, q^{-1}) = \frac{1}{1 + C(i, q^{-1})}. \tag{17}$$

This means that the MFxLMS recursion (13) can be equivalently rewritten in the form (5), viz.,

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu(i)F[\boldsymbol{u}_i^*]\frac{1}{1 + C(i, q^{-1})}F[d(i) - \boldsymbol{u}_i\boldsymbol{w}_{i-1}] \tag{18}$$

where the coefficients of $C(i, q^{-1})$ are computed as in (9).

C. 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.

The first modification, referred to as MFxLMS-1, replaces the time-variant coefficients $c(i, l)$ (9) in (17) 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 $\boldsymbol{u}_i$ (with shift structure) is sufficiently large, the inner product terms $\boldsymbol{u}_{i-p}\boldsymbol{u}_i^*$ can be approximated by $\boldsymbol{u}_{i-p}\boldsymbol{u}_i^* \approx Mr_p$.

If we further assume that the time-variant step size $\mu(i)$ in (18) is chosen as

$$\mu(i) = \frac{\alpha}{||F[\boldsymbol{u}_i]||^2} \tag{19}$$

which is known as the *projection stepsize*, then the term $||F[\boldsymbol{u}_i]||^2$ in (19) can be approximated by

$$||F[\boldsymbol{u}_i]||^2 \approx M\sum_{i=0}^{M_F-1}\sum_{j=0}^{M_F-1} r_{i-j}f_i f_j \tag{20}$$

and the filter coefficients $c(i, k)$ in (9) can also be approximated by $c(i, k) \approx \alpha\bar{c}(k)$, where we have defined the averaged coefficients

$$\bar{c}(k) = \frac{\displaystyle\sum_{i=k}^{M_F-1}\sum_{j=0}^{M_F-1} r_{i-j}f_i f_j}{\displaystyle\sum_{i=0}^{M_F-1}\sum_{j=0}^{M_F-1} r_{i-j}f_i f_j}. \tag{21}$$

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 F. Note also that since $u(i)$ resembles noise, it can be assumed in many cases that $u(i)$ is a white random sequence with variance σ_u^2. For this case, the expression for $\bar{c}(k)$ can be further simplified, since $||F[\boldsymbol{u}_i]||_2^2$ can be approximated by $M\sigma_u^2\sum_{i=0}^{M_F-1} f_i^2$ and, correspondingly

$$\bar{c}(k) = \frac{\displaystyle\sum_{i=0}^{M_F-k-1} f_i f_{i+k}}{\displaystyle\sum_{i=0}^{M_F-1} f_i^2}. \tag{22}$$

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

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu(i)F[\boldsymbol{u}_i^*]\frac{1}{1 + \alpha\bar{C}(q^{-1})}F[d(i) - \boldsymbol{u}_i\boldsymbol{w}_{i-1}] \tag{23}$$

where $\mu(i)$ is given by (19) and (20). We summarize this first variant in the following statement.

Algorithm 1 (MFxLMS-1): Consider a stationary input sequence $\{u(i)\}$ with autocorrelation coefficients $\{r_i\}$. Let $\{f_j\}_{j=0}^{M_F-1}$ denote the coefficients of the error filter F. The MFxLMS-1 algorithm proceeds as follows.

- Compute the coefficients $\bar{c}(k)$, for $k = 1, 2, \cdots, M_F$, using (21), and define the time-invariant filter $\bar{C}(q^{-1}) = \sum_{k=1}^{M_F-1}\bar{c}(k)q^{-l}$..
- Use $\bar{C}(q^{-1})$ in the update equation (23).

In the case of a white random noise process $\{u(i)\}$ with variance σ_u^2, the expression for $\bar{c}(k)$ collapses to (22).

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 (21). 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.

D. The MFxLMS-2 Algorithm

We have shown above that the MFxLMS algorithm can be written in two equivalent forms. The first one [recall (13) and (16)] is

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu(i)F[\boldsymbol{u}_i^*]\bar{e}(i) \tag{24}$$

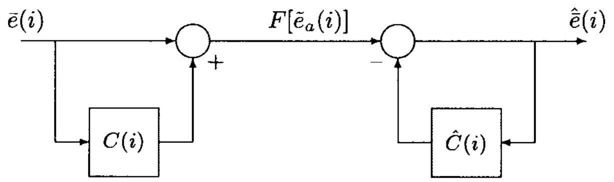


Fig. 4. Construction of $F[\tilde{e}_a(i)]$ and its approximate inversion.

$$\bar{e}(i) = F[v(i)] + F[\boldsymbol{u}_i]\tilde{\boldsymbol{w}}_{i-1} \tag{25}$$

and the second one is (18).

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 (13) and uses the additional filtering operations $F[\boldsymbol{u}_i\boldsymbol{w}_{i-1}]$ and $F[\boldsymbol{u}_i]\boldsymbol{w}_{i-1}$. The net result, however, is the same, since we already know that both representations are equivalent and, in particular, that

$$\bar{e}(i) = \frac{1}{1+C(i,q^{-1})}\, F[\tilde{e}_a(i)] = G_o(i,q^{-1})F[\tilde{e}_a(i)]. \tag{26}$$

If we knew $\bar{e}(i)$, then it could be used in the update form (24), without the explicit need for the additional filtering operations of (13).

This suggests the following modification. We have in (26) a relation between $\bar{e}(i)$ and $1/(1+C)$. The only known quantity in (26) is $F[\tilde{e}_a(i)]$, and we can rewrite the expression in the form $\bar{e}(i) = F[\tilde{e}_a(i)] - C(i,q^{-1})\bar{e}(i)$. The left part of Fig. 4 depicts this relation. Since $\bar{e}(i)$ is unknown, we need an estimate for it, say $\hat{\bar{e}}(i)$. The above relation can then be replaced by $\hat{\bar{e}}(i) = F[\tilde{e}_a(i)] - C(i,q^{-1})\hat{\bar{e}}(i)$. In order to come up with an estimate $\hat{\bar{e}}(i)$ we can use the inverse structure of $1+C(i,q^{-1})$ (which is $1/[1+C(i,q^{-1})]$) as depicted in the right part of Fig. 4. If $F[\tilde{e}_a(i)]$ is regarded as a filtered (correlated) version of $\bar{e}(i)$, then $\hat{\bar{e}}(i)$ can be interpreted as the estimated unfiltered (uncorrelated) version of it. The coefficients $\hat{c}(i,k)$ can be estimated by using a linear prediction scheme.

An approximate solution would be to use a gradient-type algorithm to estimate both $\hat{\bar{e}}(i)$ and the coefficients of $C(i,q^{-1})$ as follows (for $k = 1,\cdots,M_F-1$):

$$\hat{\bar{e}}(i) = F[\tilde{e}_a(i)] - \sum_{k=1}^{M_F-1} \hat{c}(i-1,k)\hat{\bar{e}}(i-k) \tag{27}$$

$$\hat{c}(i,k) = \hat{c}(i-1,k) + \frac{\hat{\bar{e}}(i)\hat{\bar{e}}(i-k)}{1+\sum_{k=1}^{M_F-1}|\hat{\bar{e}}(i-k)|^2}. \tag{28}$$

Once the $\hat{\bar{e}}(i)$ is evaluated, it is used in

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu(i)F[\boldsymbol{u}_i^*]\hat{\bar{e}}(i). \tag{29}$$

Algorithm 2 (MFxLMS-2): Given $\{\boldsymbol{u}_i\}$ and $\{F[\tilde{e}_a(i)]\}$, evaluate the estimates $\hat{\bar{e}}(i), \hat{c}(i,k)\}$ as in (27) and (28) and update the weight estimate using (29).

III. Robustness and l_2-Stability

We now analyze the robustness performance of the generalized FxLMS recursion (11) [or, equivalently, (12)]. As mentioned before, the recursion (12) is in filtered-error form and, therefore, the small gain (feedback) analysis of [7] and [8] is applicable. In particular, the result stated in the next theorem is a special case of Theorem 2 in Section V-C of [7].

Define $\bar{\mu}(i) = 1/\|F[\boldsymbol{u}_i]\|^2$ and introduce the diagonal matrices

$$\boldsymbol{M}_N = \text{diag}\{\mu(i)\}_{i=0}^N, \quad \overline{\boldsymbol{M}}_N = \text{diag}\{\bar{\mu}(i)\}_{i=0}^N.$$

Let also $\{\boldsymbol{G}_N, \boldsymbol{C}_N\}$ be lower triangular matrices that describe the action of the filters $\{G, C\}$ on sequences at their input. These are generally band matrices. For example, the matrix $\boldsymbol{C}_N$ for $M_F = 3$ takes the form (it is strictly lower triangular)

$$\boldsymbol{C}_N = \begin{bmatrix} 0 & & & & \\ c(0,1) & 0 & & & \\ c(0,2) & c(1,1) & 0 & & \\ 0 & c(1,2) & c(2,1) & 0 & \\ & \ddots & \ddots & \ddots & \ddots \end{bmatrix}.$$

We further write $\|\boldsymbol{A}\|_{2,ind}$ to denote the 2-induced norm of a matrix $\boldsymbol{A}$ (i.e., its largest singular value). Now define

$$\Delta(N) = \|\boldsymbol{I} - \boldsymbol{M}_N\overline{\boldsymbol{M}}_N^{-(1/2)}\boldsymbol{G}_N(\boldsymbol{I}-\boldsymbol{C}_N\boldsymbol{G}_N)^{-1} \cdot \overline{\boldsymbol{M}}_N^{-(1/2)}\|_{2,ind},$$
$$\gamma(N) = \|\overline{\boldsymbol{M}}_N^{-(1/2)}\boldsymbol{M}_N\boldsymbol{G}_N(\boldsymbol{I}-\boldsymbol{C}_N\boldsymbol{G}_N)^{-1} \cdot \overline{\boldsymbol{M}}_N^{-(1/2)}\|_{2,ind}.$$

Theorem 1: Consider the recursion (12) over $0 \le i < N$ and define $e_a'(i) = F[\boldsymbol{u}_i]\tilde{\boldsymbol{w}}_{i-1}$. If $\Delta(N) < 1$ then the map from $\{\sqrt{\mu(\cdot)}F[v(\cdot)], \tilde{\boldsymbol{w}}_{-1}\}$ to $\{\sqrt{\mu(\cdot)}e_a'(\cdot)\}$ is l_2-stable in the following sense. It satisfies

$$\sqrt{\sum_{i=0}^N \mu(i)|e_a'(i)|^2} \le \frac{\gamma^{1/2}(N)}{1-\Delta(N)}\left[\|\tilde{\boldsymbol{w}}_{-1}\|_2 + \gamma^{1/2}(N) \cdot \sqrt{\sum_{i=0}^N \mu(i)|F[v(i)]|^2}\right]. \tag{30}$$

We can regard the sum in the right-hand side of (30) as the "weighted" energy of the disturbances in the problem, viz., $\{\tilde{\boldsymbol{w}}_{-1}, F[v(\cdot)]\}$. Likewise, we can regard the sum in the left-hand side of (30) as the "weighted" energy of the resulting estimation (filtered) errors $\{e_a'(\cdot)\}$. Hence, according to (30), the amplification (energy-wise) from $\{\tilde{\boldsymbol{w}}_{-1}, F[v(\cdot)]\}$ to $\{e_a'(\cdot)\}$ is bounded from above by $\gamma^{1/2}(N)/(1-\Delta(N))$, which is reasonable especially when $\Delta(N)$ is sufficiently far from one. This means that, regardless of the nature of the "disturbances," their energy will not be unnecessarily amplified (but only if $\Delta(N)$ is guaranteed to be less than one).

A. Stepsize Conditions for the FxLMS Algorithm

In the special case of the original FxLMS algorithm (4), which corresponds to $G = 1$, and for a constant step-size parameter $\mu(i) = \mu$, the condition $\Delta(N) < 1$ can be guaranteed by requiring

$$\left\| \boldsymbol{I} - \frac{\mu}{\overline{\mu}} \left(\boldsymbol{I} - \boldsymbol{C}_N\right)^{-1} \right\|_{2,ind} < 1 \tag{31}$$

where we have defined $\overline{\mu}$ to be any positive scalar satisfying (assuming $\|\boldsymbol{u}'_i\|$ is uniformly bounded from above)

$$\overline{\mu} \leq \min_{0 \leq i \leq N} \{\overline{\mu}(i)\}.$$

Expression (31) therefore provides a condition that guarantees an l_2-stable performance for the *original* FxLMS algorithm. The condition is in terms of the filter $C(i, q^{-1})$ that we showed how to construct in (21). Note also that condition (31) explicitly incorporates the ratio of the stepsize parameter μ to the input energy. This may be compared with results in the literature (e.g., [9], [16]) regarding the exponential stability of filtered-error algorithms. These results usually require the error filter to be strictly positive-real and the analysis holds as long as the stepsize parameter is small enough. Expression (31), on the other hand, does not impose a small condition assumption on the stepsize, nor does it require a positive real condition on the filter F. It instead blends the conditions on F, the input signal energy, and μ into a single contractivity condition. A related condition arises in [17] in the study of the global uniform asymptotic stability of an output error adaptive filter in the noiseless case; it incorporates both the stepsize and the input signal energy.

A similar conclusion to (31) can be obtained when we replace the time-variant matrix $\boldsymbol{C}_N$ by a time-invariant approximation as described in (23) and apply $\mu(i) = \alpha\overline{\mu}(i)$. In other words, once a filter structure of the form (23) has been devised, with a given $\overline{C}$, we can now pursue a robustness analysis of the resulting algorithm.

For this purpose, all we need to do is replace $\boldsymbol{C}_N$ by a lower triangular matrix that corresponds to $\alpha\overline{\boldsymbol{C}}_N$, which now becomes Toeplitz due to the time invariance of $\overline{C}$. The sufficient condition for l_2-stability then collapses to requiring

$$\|\boldsymbol{I} - \alpha(\boldsymbol{I} - \alpha\overline{\boldsymbol{C}}_N)^{-1}\|_{2,ind} < 1 \tag{32}$$

which is guaranteed if we require

$$\max_{\Omega} \left| 1 - \frac{\alpha}{1 - \alpha\overline{C}(e^{j\Omega})} \right| < 1. \tag{33}$$

Moreover, the energy arguments in [7] (see below for an intuitive explanation) suggest that, in general, improved convergence can be obtained by posing an optimization problem for the selection of α, (which determines the stepsize in the FxLMS algorithm) as follows:

$$\alpha_{opt} = \min_{\alpha} \max_{\Omega} \left| 1 - \frac{\alpha}{1 - \alpha\overline{C}(e^{j\Omega})} \right|. \tag{34}$$

The resulting α_{opt} is the value that makes the magnitude in (33) the lowest possible. That is, it forces the value of Δ to be the furthest from one. Intuitively, this is the case that results in the "smallest" estimation error energy over an interval of length N [in view of (30)] and would result in faster convergence. The simulation results in the next section demonstrate this remark. We shall also compare the above choice(s) for α with the one suggested in [1], viz.,

$$\alpha_{\max} = \frac{1}{1 + \dfrac{M_F}{M}}.$$

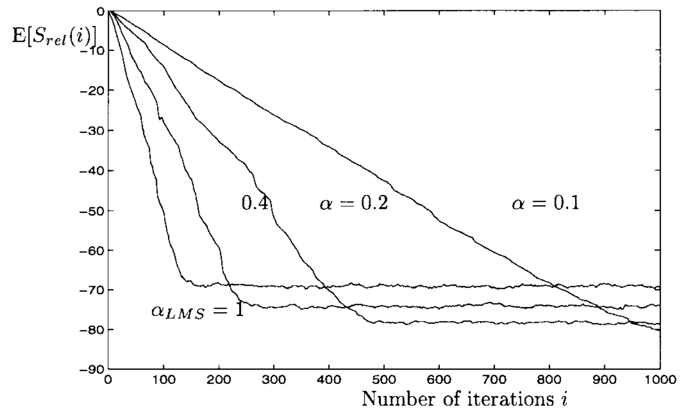


Fig. 5. Learning curves for DLMS algorithm $\alpha = 0.1, 0.2, 0.4$ in comparison to LMS $\alpha = 1.0$.

IV. 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 -60 dB below the input power. We provide plots of learning curves for the relative system mismatch, defined as

$$S_{\text{rel}}(i) = \frac{\|\tilde{\boldsymbol{w}}_i\|^2}{\|\tilde{\boldsymbol{w}}_{-1}\|_2^2}.$$

The curves are averaged over 50 Monte Carlo runs in order to approximate $\mathrm{E}[S_{\text{rel}}(i)]$. The results in the figures are also indicated in dB. In all experiments, we employed the projection normalization (19).

A. 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 $F(q^{-1}) = q^{-4}$ (which is not positive real). The FxLMS algorithm in this case corresponds to the *delayed LMS* (DLMS) [18]–[20], as follows:

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

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

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

As Fig. 5 shows, the delay causes a degradation in the convergence behavior of the DLMS algorithm.

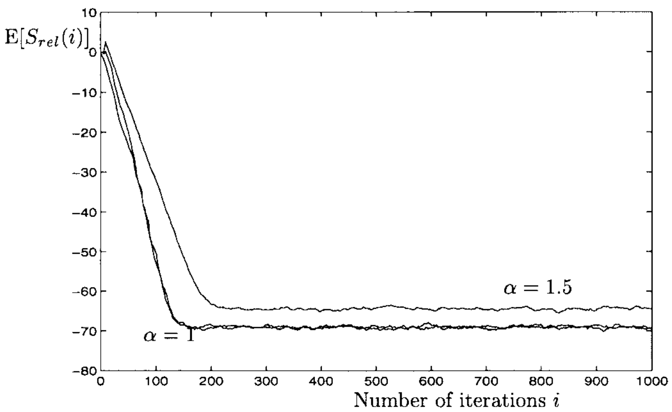


Fig. 6. Learning curves for modified DLMS algorithm $\alpha = 1$ and $\alpha = 1.5$.

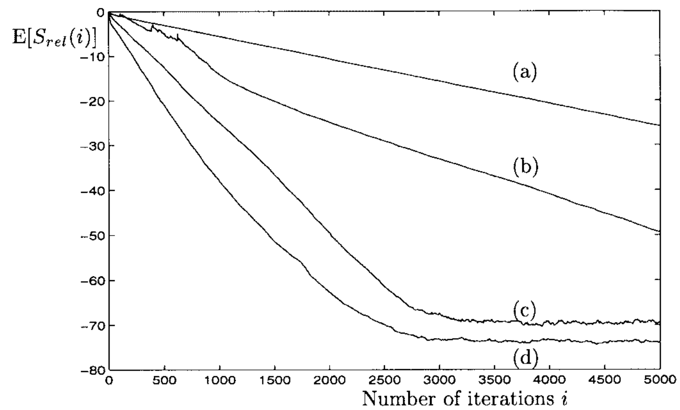


Fig. 7. Learning curves for FxLMS algorithm with $\alpha = 0.5$ (a) and modifications: MFxLMS-2; (b) $\alpha = 1.15$ and MFxLMS-1; (c) $\alpha = 1.2$ in comparison to MFxLMS; (d) $\alpha = 1.2$.

In a second experiment, the modified version of the DLMS algorithm, using the optimal $G_o(i, q^{-1})$ as suggested by (17), has been used, viz.,

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

where

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

This is, of course, equivalent to a MFxLMS form

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

As Fig. 6 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 stepsize for which the conventional DLMS algorithm was already unstable.

B. The FxLMS Algorithm and Modifications

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 lowpass behavior as it is common in acoustic ducts. Note that this filter is also not positive real. The coefficients of the corresponding averaged filter $\overline{C}(q^{-1})$ were given by $\bar{c}_1 = 0.75, \bar{c}_2 = 0.5, \bar{c}_3 = 0.25$, i.e., $\overline{C}(q^{-1}) = 0.75q^{-1} + 0.5q^{-2} + 0.25q^{-3}$.

If we use the above averaged coefficients as approximations, we obtain an approximate stability range for the FxLMS algorithm at $0 < \alpha < 0.5$ [recall (33)]; the optimal convergence speed is attained at $\alpha_o = 0.45$ [recall (34)]. 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 stepsize from [1] for this case is 0.8333, which is already in the unstable region.

Fig. 7 shows learning curves that correspond to the FxLMS algorithm with the above-mentioned optimal step-size [curve (a)] and the proposed version MFxLMS-1 that leads to a learning curve [indicated by "(c)"] which is close to the optimal one [i.e., the one that corresponds to the MFxLMS recursion and is indicated by "(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 *a priori.* 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 $\boldsymbol{u}_i$ and $v(i)$ prefiltered by F, is not explicitly shown in the figure since it essentially coincides with the MFxLMS algorithm [curve (d)].

V. Concluding Remarks

We presented a time-domain analysis of a generalized FxLMS recursion (5) and have shown that it can be reexpressed in the form of a filtered-error variant (11). In particular, we have shown that the MFxLMS recursion (13) corresponds to a special case of the generalized algorithm (5), viz., the choice $G_o(i, q^{-1})$ in (17). But other choices are also possible.

We then proceeded to provide two approximations for the optimal $G_o(i, q^{-1})$. This led us to two solutions: The MFxLMS-1 and the MFxLMS-2. Both have the same computational requirement as the original FxLMS algorithm, viz., of the order of $2M$ computations per step and, hence, less than the MFxLMS variant, which requires of the order of $3M$ computations. Both algorithms, MFxLMS-1 and MFxLMS-2, also have improved convergence behavior when compared with FxLMS but only MFxLMS-1 exhibits a good enough behavior that is comparable with MFxLMS.

APPENDIX A
PROOF OF (8)

The result (8) follows from the following sequence of easily verifiable identities:

$$\begin{aligned} F[\tilde{e}_a(i)] &= F[v(i)] + F[e_a(i)] = F[v(i)] + F[\boldsymbol{u}_i\tilde{\boldsymbol{w}}_{i-1}] \\ &= F[v(i)] + \sum_{l=0}^{M_F-1} f_l \boldsymbol{u}_{i-l}\tilde{\boldsymbol{w}}_{i-1-l} \\ &= F[v(i)] + \sum_{l=0}^{M_F-1} f_l \boldsymbol{u}_{i-l}\tilde{\boldsymbol{w}}_{i-1} + X_1 \\ &= F[v(i)] + \sum_{l=0}^{M_F-1} f_l \boldsymbol{u}_{i-l}\tilde{\boldsymbol{w}}_{i-1} + X_2 \\ &= F[v(i)] + F[\boldsymbol{u}_i]\tilde{\boldsymbol{w}}_{i-1} + X_3 \\ &= F[v(i)] + F[\boldsymbol{u}_i]\tilde{\boldsymbol{w}}_{i-1} \\ &\quad + \sum_{k=1}^{M_F-1} c(i,k)G(i-k,q^{-1})F[\tilde{e}_a(i-k)] \end{aligned}$$

where we used (7) in order to obtain

$$X_1 = \sum_{l=1}^{M_F-1} f_l \boldsymbol{u}_{i-l} \sum_{k=1}^{l} \mu(i-k)F[\boldsymbol{u}_{i-k}^*]G(i-k,q^{-1}) \cdot F[\tilde{e}_a(i-k)].$$

Simple rearrangements lead to the expressions

$$\begin{aligned} X_2 &= \sum_{k=1}^{M_F-1} \left[\sum_{l=k}^{M_F-1} f_l \boldsymbol{u}_{i-l}\right] \mu(i-k)F[\boldsymbol{u}_{i-k}^*] \\ &\quad \cdot G(i-k,q^{-1})F[\tilde{e}_a(i-k)], \\ X_3 &= \sum_{k=1}^{M_F-1} \mu(i-k)F_k[\boldsymbol{u}_i]F[\boldsymbol{u}_{i-k}^*]G(i-k,q^{-1}) \\ &\quad \cdot F[\tilde{e}_a(i-k)]. \end{aligned}$$

REFERENCES

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

[2] J. R. Glover, Jr., "Adaptive noise canceling applied to sinusoidal interferences," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 484–491, Dec. 1977.

[3] S. Laugesen and S. Elliott, "Multichannel active control of random noise in a small reverberant room," *IEEE Trans. Speech Audio Processing*, vol. 1, pp. 241–249, Apr. 1993.

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

[5] E. Bjarnason, "Noise cancellation using a modified form of the filtered-XLMS algorithm," in *Proc. Eusipco Signal Processing V*, Brussels, Belgium, 1992.

[6] 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. Amer.*, vol. 95, pp. 3397–3389, June 1994.

[7] M. Rupp and A. H. Sayed, "A time-domain feedback analysis of filtered-error adaptive gradient algorithms," *IEEE Trans. Signal Processing*, vol. 44, pp. 1428–1439, June 1996.

[8] A. H. Sayed and M. Rupp, "An l_2-stable feedback structure for nonlinear adaptive filtering and identification," *Automatica*, vol. 33, pp. 13–30, 1997.

[9] W. A. Sethares, B. D. O. Anderson, and C. R. Johnson, "Adaptive algorithms with filtered regressor and filtered error," *Math. Contr., Signals, Syst.*, vol. 2, pp. 381–403, 1989.

[10] A. Benveniste, M. Métivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*. New York: Springer-Verlag, 1990.

[11] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*. Cambridge, MA: MIT Press, 1983.

[12] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms: Stability and Performance*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[13] B. Hassibi, A. H. Sayed, and T. Kailath, "H^∞ optimality of the LMS algorithm," *IEEE Trans. Signal Processing*, vol. 44, pp. 267–280, Feb. 1996.

[14] A. H. Sayed and T. Kailath, "A state-space approach to adaptive RLS filtering," *IEEE Signal Processing Mag.*, vol. 11, pp. 18–60, July 1994.

[15] L. J. Eriksson, "Development of the filtered-U algorithm for active noise control," *J. Acoust. Soc. Amer.*, vol. 89, pp. 257–265, Apr. 1991.

[16] A. C. Orgren, S. Dasgupta, C. E. Rohrs, and N. R. Malik, "Noise cancellation with improved residuals," *IEEE Trans. Signal Processing*, vol. 39, Dec. 1991.

[17] M. Tomizuka, "Parallel MRAS without compensation block," *IEEE Trans. Automat. Contr.*, vol. AC-27, pp. 505–506, 1982.

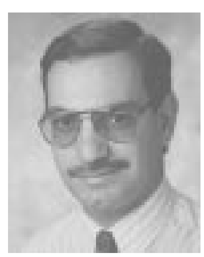
[18] P. Kabal, "The stability of adaptive minimum mean square error equalizers using delayed adjustment," *IEEE Trans. Commun.*, vol. COMM–31, pp. 430–432, Mar. 1983.

[19] G. Long, F. Ling, and J. A. Proakis, "The LMS algorithm with delayed coefficient adaptation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1397–1405, Sept. 1989; also *IEEE Trans. Signal Processing*, vol. 40, pp. 230–232, Jan. 1992.

[20] M. Rupp and R. Frenzel "The behavior of LMS and NLMS algorithms with delayed coefficient update in the presence of spherically invariant processes," *IEEE Trans. Signal Processing*, vol. 42, pp. 668–672, Mar. 1994.

Markus Rupp received the Dipl. in electrical engineering from FHS Saarbruecken, Germany, and Universitaet of Saarbruecken in 1984 and 1988, respectively, and the Doctoral degree in 1993 (summa cum laude) at the TH Darmstadt in the field of acoustical echo cancellation.

During his undergraduate training, he he was a lecturer in digital signal processing and high-frequency techniques at the FHS. He received the DAAD postdoctoral fellowship, and spent November 1993 to September 1995 in the Department of Electrical and Computer Engineering, University of California, Santa Barbara. Since October 1995, he has been with the Wireless Technology Research Department, Lucent Technologies, Holmdel, NJ, where he is currently working on new adaptive equalization schemes. His interests involve algorithms for speech applications, speech enhancement and recognition, echo compensation, adaptive filter theory, H_∞-filtering, neural nets, classification algorithms, and signal detection. He has over 30 publications in the field of adaptive filters.



Ali H. Sayed (SM'90–M'92) was born in São Paulo, Brazil. In 1981, he graduated in first place in the National Lebanese Baccalaureat, with the highest score in the history of the examination. In 1987, he received the degree of Engineer in Electrical Engineering from the University of São Paulo, Brazil, and was the first-place graduate of the School of Engineering. He received the M.S. degree (with distinction) in electrical engineering from the University of São Paulo, Brazil, in 1989, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1993. He has over 90 publications.

From September 1992 to August 1993, he was a Research Associate with the Information Systems Laboratory, Stanford University, Stanford, CA, after which he joined the Department of Electrical and Computer Engineering, University of California, Santa Barbara, as an Assistant Professor. Since July 1996, he has been an Associate Professor in the Department of Electrical Engineering, University of California, Los Angeles. His research interests are in the areas of adaptive and statistical signal processing, linear and nonlinear filtering and estimation, interplays between signal processing and control methodologies, interpolation theory, and structured computations in systems and mathematics.

Dr. Sayed is a member of SIAM and ILAS. He was awarded the Institute of Engineering Prize and the Conde Armando Alvares Penteado Prize, both in 1987, in Brazil. He is a recipient of a 1994 NSF Research Initiation Award and the 1996 IEEE Donald G. Fink Prize Award. He is an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING.