MODIFIED FILTERED-REFERENCE/FILTERED-ERROR LMS ALGORITHM: ALGORITHM AND STOCHASTIC MODEL

Juan R. V. López, Orlando J. Tobias, and Rui Seara

LINSE – Circuits and Signal Processing Laboratory
Department of Electrical Engineering
Federal University of Santa Catarina
88040-900 – Florianópolis – SC – Brazil
E-mail: {juan, orlando, seara}@linse.ufsc.br

ABSTRACT

This paper proposes a modified filtered-reference/filtered-error least-mean-square (MFxFeLMS) algorithm. In addition, a stochastic model for the first and second moments of the MFxFeLMS is derived. The proposed model is obtained without invoking the classic Independence Theory (IT), allowing for slow adaptation and Gaussian input signals. Numerical simulations confirm the very good agreement between the results obtained by the Monte Carlo (MC) method and from the proposed model for both white and colored Gaussian inputs.

1. INTRODUCTION

In active noise and vibration control a direct implementation of the adaptive structure is not always possible. A common problem is the fact that, in some situations, the error signal used to update the coefficients of the adaptive filter is not readily available or is inaccessible. In these cases, it is only possible to obtain a filtered version of either the adaptive filter output or of the error signal, as occurs in active noise/vibration control and other applications. In these applications, the family of filtered least-mean-square (FLMS) algorithms should then be used. However, such algorithms present poor behavior regarding convergence speed and stability issues. To circumvent such problems, in [1], a modification of the adaptive algorithm has been proposed, leading to a modified algorithm having a behavior equivalent to the standard LMS. That is, the characteristics of convergence and stability of the modified algorithm coincide with the standard LMS. Such an algorithm is termed modified FLMS (MFLMS) algorithm [1], [2].

In the open literature, there are four types of adaptive algorithms belonging to the family of filtered algorithms, namely: filtered-x (or filtered-reference) LMS (FxLMS) [3], [4]; filtered-error LMS (FeLMS) [5], [6]; modified FxLMS (MFxLMS) [2]; and modified FeLMS (MFeLMS) [7]. The first two algorithms are widely used in active noise control applications. The other ones are modified versions, aiming to increase the convergence speed of the first two. In addition, a generalized algorithm grouping the FxLMS and FeLMS has been introduced in [8], [9], termed filtered-reference/filtered-error LMS (FxFeLMS) algorithm.

Several statistical analyses of the different versions of the LMS algorithm have been accomplished under the light of the Independence Theory (IT) [10], [11]. However, there are certain situations in which such an analysis assumption can no longer be applied, such as modeling the FxLMS and MFeLMS algorithms. For instance, stochastic models for the first and second moments of the FxLMS and MFeLMS algorithms without invoking the IT are derived in [12]-[13] and [7], respectively. To the best of our knowledge, in the open literature there is neither a modified version FxFeLMS (MFxFeLMS) algorithm nor its stochastic model. Such an algorithm and model are the main contributions of this paper. Since the use of the independence assumption has been proved not adequate for this algorithm class (as discussed in [7],[12], and [13]), such an assumption is not invoked here. In this way, the proposed models for the mean weight behavior and learning curve are more accurate than those using IT. Through Monte Carlo (MC) numerical simulations, the performance of the FxFeLMS [8] and the proposed algorithm are compared. In addition, very good agreement between the MC simulation and model for the proposed algorithm is obtained. For all cases, both white and colored input data are used.

2. MODIFIED FxFeLMS ALGORITHM

2.1 FxFeLMS Algorithm

Figure 1 shows the block diagram of the FxFeLMS algorithm in which the following notation is used: $\mathbf{w}_\text{o} = [w_{\text{o},0}\ w_{\text{o},1} \cdots w_{\text{o},N-1}]^\text{T}$ represents the plant impulse response, $\mathbf{w}(n) = [w_0(n)\ w_1(n)\ \cdots w_{N-1}(n)]^\text{T}$ denotes the adaptive weight vector, $\mathbf{s}_1 = [s_{1,0}\ s_{1,1} \cdots s_{1,M_1-1}]^\text{T}$, $\mathbf{s}_2 = [s_{2,0}\ s_{2,1} \cdots s_{2,M_2-1}]^\text{T}$, and $\hat{\mathbf{s}} = [\hat{s}_0\ \hat{s}_1 \cdots \hat{s}_{\hat{M}-1}]^\text{T}$ are filter impulse responses which are placed, respectively, in the adaptive filter output, in the error signal path, and in the input signal path. Note that $\hat{\mathbf{s}}$ is an estimate of the convolution between $\mathbf{s}_1$ and $\mathbf{s}_2$. Signals $d(n)$ and $z(n)$ are the desired signal

Orlando J. Tobias is also with the Electrical Engineering and Telecom. Dept., Regional University of Blumenau (FURB), Blumenau, SC, Brazil. This work was supported in part by the National Council for Scientific and Technological Development (CNPq) and the Committee for Postgraduate Courses in Higher Education (CAPES).

and measurement noise, respectively. The latter is i.i.d, zero-mean and uncorrelated with any other signal in the system. In this analysis, the input vector is $\mathbf{x}(n)=[x(n)\; x(n-1)\cdots x(n-N+1)]^{\mathrm{T}}$, with $\{x(n)\}$ being a zero-mean Gaussian process with variance σ_x^2. The filtered input vector is given by $\mathbf{x}_{\mathrm{f}}(n)=[x_{\mathrm{f}}(n)\; x_{\mathrm{f}}(n-1)\cdots x_{\mathrm{f}}(n-N+1)]^{\mathrm{T}}$, with

$$\mathbf{x}_{\mathrm{f}}(n)=\sum_{i=0}^{\hat{M}-1}\hat{s}_i\mathbf{x}(n-i). \qquad (1)$$

In this work, we consider that vectors $\mathbf{w}_{\mathrm{o}}$ and $\mathbf{w}(n)$ have the same dimension, while the dimensions of vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ may be different.

From Figure 1, the error signal is given by

$$e(n)=\mathbf{w}_{\mathrm{o}}^{\mathrm{T}}\mathbf{x}(n)-\sum_{i=0}^{M_1-1}s_{1,i}\mathbf{w}^{\mathrm{T}}(n-i)\mathbf{x}(n-i)+z(n). \qquad (2)$$

The filtered error signal, used in the weight-updating rule of the adaptive algorithm, is given by

$$e_{\mathrm{f}}(n)=\sum_{i=0}^{M_2-1}s_{2,i}e(n-i). \qquad (3)$$

Now, substituting (2) into (3), we obtain

$$\begin{aligned} e_{\mathrm{f}}(n)=&\sum_{i=0}^{M_2-1}s_{2,i}\mathbf{w}_{\mathrm{o}}^{\mathrm{T}}\mathbf{x}(n-i)+z_{\mathrm{f}}(n)\\ &-\sum_{i=0}^{M_2-1}\sum_{j=0}^{M_1-1}s_{2,i}s_{1,j}\mathbf{w}^{\mathrm{T}}(n-i-j)\mathbf{x}(n-i-j)\end{aligned} \qquad (4)$$

with

$$z_{\mathrm{f}}(n)=\sum_{i=0}^{M_2-1}s_{2,i}z(n-i). \qquad (5)$$

Alternatively, (4) can be rewritten as

$$\begin{aligned} e_{\mathrm{f}}(n)=&\sum_{i=0}^{M_2-1}s_{2,i}\mathbf{w}_{\mathrm{o}}^{\mathrm{T}}\mathbf{x}(n-i)+z_{\mathrm{f}}(n)\\ &-\sum_{i=0}^{M_3-1}s_{3,i}\mathbf{w}^{\mathrm{T}}(n-i)\mathbf{x}(n-i)\end{aligned} \qquad (6)$$

with

$$\mathbf{s}_3=\mathbf{s}_1*\mathbf{s}_2 \qquad (7)$$

where $*$ denotes the convolution operator. Thus, the weight update equation of the FxFeLMS algorithm is finally given by [8]

$$\mathbf{w}(n+1)=\mathbf{w}(n)+\mu e_{\mathrm{f}}(n)\mathbf{x}_{\mathrm{f}}(n). \qquad (8)$$

2.2 Modified FxFeLMS Algorithm

The MFxFeLMS algorithm is obtained by compensating the filtering operations performed by the filters $\mathbf{s}_1$ and $\mathbf{s}_2$. Such compensation is achieved by including the term $-\Lambda_{\mathrm{f}}(n)$ (named compensation term) into (6) [1]. In this way, the weight update and compensated error $\hat{e}_{\mathrm{f}}(n)$ expressions are, respectively, given by

$$\mathbf{w}(n+1)=\mathbf{w}(n)+\mu\hat{e}_{\mathrm{f}}(n)\mathbf{x}_{\mathrm{f}}(n) \qquad (9)$$

and

$$\begin{aligned}\hat{e}_{\mathrm{f}}(n)=&\sum_{i=0}^{M_2-1}s_{2,i}\mathbf{w}_{\mathrm{o}}^{\mathrm{T}}\mathbf{x}(n-i)+z_{\mathrm{f}}(n)\\ &-\sum_{i=0}^{M_3-1}s_{3,i}\mathbf{w}^{\mathrm{T}}(n-i)\mathbf{x}(n-i)\\ &-\Lambda_{\mathrm{f}}(n).\end{aligned} \qquad (10)$$

The compensation term $\Lambda_{\mathrm{f}}(n)$ is determined by enforcing (10) to be equal to the error signal of the standard LMS algorithm [1], whereby

$$\begin{aligned}e_{\mathrm{LMS}}(n)=&\sum_{i=0}^{M_2-1}s_{2,i}\mathbf{w}_{\mathrm{o}}^{\mathrm{T}}\mathbf{x}(n-i)+z_{\mathrm{f}}(n)\\ &-\sum_{i=0}^{M_3-1}s_{3,i}\mathbf{w}^{\mathrm{T}}(n)\mathbf{x}(n-i).\end{aligned} \qquad (11)$$

Note that in (11) the error signal now depends on the current value of the adaptive weight vector $\mathbf{w}(n)$ instead of $\mathbf{w}(n-i)$. From (10) and (11) the required compensation term is

$$\Lambda_{\mathrm{f}}(n)=\sum_{i=0}^{M_3-1}s_{3,i}[\mathbf{w}^{\mathrm{T}}(n)-\mathbf{w}^{\mathrm{T}}(n-i)]\mathbf{x}(n-i). \qquad (12)$$

In (12), the difference $[\mathbf{w}^{\mathrm{T}}(n)-\mathbf{w}^{\mathrm{T}}(n-i)]=0$ for $i=0$; otherwise, $[\mathbf{w}^{\mathrm{T}}(n)-\mathbf{w}^{\mathrm{T}}(n-i)]$ is obtained from (9) as follows. First, note that

$$i=1\Rightarrow\ \mathbf{w}(n)=\mathbf{w}(n-1)+\mu\hat{e}_{\mathrm{f}}(n-1)\mathbf{x}_{\mathrm{f}}(n-1) \qquad (13)$$

$$i=2\Rightarrow\ \mathbf{w}(n-1)=\mathbf{w}(n-2)+\mu\hat{e}_{\mathrm{f}}(n-2)\mathbf{x}_{\mathrm{f}}(n-2) \qquad (14)$$

$$i=L\Rightarrow\ \mathbf{w}(n-L+1)=\mathbf{w}(n-L)+\mu\hat{e}_{\mathrm{f}}(n-L)\mathbf{x}_{\mathrm{f}}(n-L) \qquad (15)$$

with $L=M_3-1$. Now, determining the algebraic differences between (9) and (13), (14), and (15), respectively, we get

$$\begin{aligned}\begin{bmatrix}\mathbf{w}^{\mathrm{T}}(n+1)-\mathbf{w}^{\mathrm{T}}(n)\\ \mathbf{w}^{\mathrm{T}}(n+1)-\mathbf{w}^{\mathrm{T}}(n-1)\\ \mathbf{w}^{\mathrm{T}}(n+1)-\mathbf{w}^{\mathrm{T}}(n-2)\\ \vdots\\ \mathbf{w}^{\mathrm{T}}(n+1)-\mathbf{w}^{\mathrm{T}}(n-L+1)\end{bmatrix}=&\begin{bmatrix}\mathbf{w}^{\mathrm{T}}(n)-\mathbf{w}^{\mathrm{T}}(n-1)\\ \mathbf{w}^{\mathrm{T}}(n)-\mathbf{w}^{\mathrm{T}}(n-2)\\ \mathbf{w}^{\mathrm{T}}(n)-\mathbf{w}^{\mathrm{T}}(n-3)\\ \vdots\\ \mathbf{w}^{\mathrm{T}}(n)-\mathbf{w}^{\mathrm{T}}(n-L)\end{bmatrix}\\ &+\mu\hat{e}_{\mathrm{f}}(n)\begin{bmatrix}\mathbf{x}_{\mathrm{f}}^{\mathrm{T}}(n)\\ \mathbf{x}_{\mathrm{f}}^{\mathrm{T}}(n)\\ \mathbf{x}_{\mathrm{f}}^{\mathrm{T}}(n)\\ \vdots\\ \mathbf{x}_{\mathrm{f}}^{\mathrm{T}}(n)\end{bmatrix}-\mu\begin{bmatrix}\hat{e}_{\mathrm{f}}(n-1)\mathbf{x}_{\mathrm{f}}^{\mathrm{T}}(n-1)\\ \hat{e}_{\mathrm{f}}(n-2)\mathbf{x}_{\mathrm{f}}^{\mathrm{T}}(n-2)\\ \hat{e}_{\mathrm{f}}(n-3)\mathbf{x}_{\mathrm{f}}^{\mathrm{T}}(n-3)\\ \vdots\\ \hat{e}_{\mathrm{f}}(n-L)\mathbf{x}_{\mathrm{f}}^{\mathrm{T}}(n-L)\end{bmatrix}.\end{aligned} \qquad (16)$$

Now, applying (13)-(15) to determine, respectively, $\hat{e}_{\mathrm{f}}(n-1)\mathbf{x}_{\mathrm{f}}^{\mathrm{T}}(n-1)$ to $\hat{e}_{\mathrm{f}}(n-L)\mathbf{x}_{\mathrm{f}}^{\mathrm{T}}(n-L)$, and substituting the resulting expressions into (16), after some straightforward algebra, we get

$$\begin{bmatrix} \boldsymbol{\alpha}_1^{\mathrm{T}}(n+1) \\ \boldsymbol{\alpha}_2^{\mathrm{T}}(n+1) \\ \vdots \\ \boldsymbol{\alpha}_{M_3-1}^{\mathrm{T}}(n+1) \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{\alpha}_1^{\mathrm{T}}(n) \\ \vdots \\ \boldsymbol{\alpha}_{M_3-2}^{\mathrm{T}}(n) \end{bmatrix} + \mu\hat{e}_{\mathrm{f}}(n) \begin{bmatrix} \mathbf{x}_{\mathrm{f}}^{\mathrm{T}}(n) \\ \mathbf{x}_{\mathrm{f}}^{\mathrm{T}}(n) \\ \vdots \\ \mathbf{x}_{\mathrm{f}}^{\mathrm{T}}(n) \end{bmatrix} \qquad (17)$$

with

$$\boldsymbol{\alpha}_j^{\mathrm{T}}(n) = \mathbf{w}^{\mathrm{T}}(n) - \mathbf{w}^{\mathrm{T}}(n-j), \qquad j = 1, 2, ..., M_3 - 1. \quad (18)$$

Expression (17) can be rewritten as

$$\boldsymbol{\alpha}_j(n+1) = \begin{cases} \mu\hat{e}_{\mathrm{f}}(n)\mathbf{x}_{\mathrm{f}}(n), & j = 1 \\ \boldsymbol{\alpha}_{j-1}(n) + \mu\hat{e}_{\mathrm{f}}(n)\mathbf{x}_{\mathrm{f}}(n), & 2 \le j \le M_3 - 1. \end{cases} \qquad (19)$$

Finally, substituting (18) into (12) and the resulting expression into (10), the compensated error signal is determined as follows:

$$\begin{aligned} \hat{e}_{\mathrm{f}}(n) = & \sum_{i=0}^{M_2-1} s_{2,i}\mathbf{w}_{\mathrm{o}}^{\mathrm{T}}\mathbf{x}(n-i) - \sum_{i=0}^{M_3-1} s_{3,i}\mathbf{w}^{\mathrm{T}}(n-i)\mathbf{x}(n-i) \\ & - \sum_{i=1}^{M_3-1} s_{3,i}\boldsymbol{\alpha}_i^{\mathrm{T}}(n)\mathbf{x}(n-i) + z_{\mathrm{f}}(n). \end{aligned} \qquad (20)$$

3. ANALYSIS

To determine model expressions for the MFxFeLMS algorithm, we make the following assumption A1: the correlations between input vectors at different lags are much more important than the correlations between the input and weight vectors. The same assumption is used for the correlations between input and $\boldsymbol{\alpha}_j$ vectors.

3.1 Mean Weight Behavior

In this section, a model expression for the first moment of the adaptive weight vector is derived. Taking the expected value of both sides of (9), one has

$$E[\mathbf{w}(n+1)] = E[\mathbf{w}(n)] + \mu E[\hat{e}_{\mathrm{f}}(n)\mathbf{x}_{\mathrm{f}}(n)]. \qquad (21)$$

The term $\hat{e}_{\mathrm{f}}(n)\mathbf{x}_{\mathrm{f}}(n)$ is obtained by post-multiplying both sides of (20) by $\mathbf{x}_{\mathrm{f}}(n)$. Then, substituting (1) into all r.h.s. terms of the resulting expression, taking the expected value, and according A1, we obtain

$$\begin{aligned} E[\hat{e}_{\mathrm{f}}(n)\mathbf{x}_{\mathrm{f}}(n)] = & \sum_{i=0}^{\hat{M}-1}\sum_{j=0}^{M_2-1} \hat{s}_i s_{2,j}\mathbf{R}_{j-i}\mathbf{w}_{\mathrm{o}} \\ & - \sum_{i=0}^{\hat{M}-1}\sum_{j=0}^{M_3-1} \hat{s}_i s_{3,j}\mathbf{R}_{j-i}E[\mathbf{w}(n-j)] \\ & - \sum_{i=0}^{\hat{M}-1}\sum_{j=1}^{M_3-1} \hat{s}_i s_{3,j}\mathbf{R}_{j-i}E[\boldsymbol{\alpha}_j(n)] \end{aligned} \qquad (22)$$

where the autocorrelation matrix $\mathbf{R}_{j-i}$ in (22) is obtained according to the general form $\mathbf{R}_{\beta-\alpha} = E[\mathbf{x}(n-\alpha)\mathbf{x}^{\mathrm{T}}(n-\beta)]$.

In (22), the term $E[\boldsymbol{\alpha}_j(n)]$ is determined by taking the expected value of (19). Thus,

$$E[\boldsymbol{\alpha}_j(n+1)] = \begin{cases} \mu E[\hat{e}_{\mathrm{f}}(n)\mathbf{x}_{\mathrm{f}}(n)], & j = 1 \\ E[\boldsymbol{\alpha}_{j-1}(n)] + \mu E[\hat{e}_{\mathrm{f}}(n)\mathbf{x}_{\mathrm{f}}(n)], & 2 \le j \le M_3 - 1. \end{cases} \qquad (23)$$

3.2 Steady-State Value of $\mathbf{w}(n)$

By substituting (22) into (21) and taking the limit as $n \to \infty$ of the resulting expression, we have

$$\begin{aligned} \lim_{n\to\infty} E[\mathbf{w}(n+1)] = & \lim_{n\to\infty} E[\mathbf{w}(n)] + \mu\sum_{i=0}^{\hat{M}-1}\sum_{j=0}^{M_2-1} \hat{s}_i s_{2,j}\mathbf{R}_{j-i}\mathbf{w}_{\mathrm{o}} \\ & - \mu\sum_{i=0}^{\hat{M}-1}\sum_{j=0}^{M_3-1} \hat{s}_i s_{3,j}\mathbf{R}_{j-i} \lim_{n\to\infty} E[\mathbf{w}(n-j)] \\ & - \mu\sum_{i=0}^{\hat{M}-1}\sum_{j=1}^{M_3-1} \hat{s}_i s_{3,j}\mathbf{R}_{j-i} \lim_{n\to\infty} E[\boldsymbol{\alpha}_j(n)]. \end{aligned} \qquad (24)$$

Now, assuming algorithm convergence, the steady-state value of the weight vector is obtained from the following condition:

$$\begin{aligned} \lim_{n\to\infty} E[\mathbf{w}(n+1)] & = \lim_{n\to\infty} E[\mathbf{w}(n-j)] \\ & = \lim_{n\to\infty} E[\mathbf{w}(n)] = \mathbf{w}_\infty. \end{aligned} \qquad (25)$$

Then, using the result of (25) in (24) and considering a system identification problem as illustrated in Figure 1, we get

$$\begin{aligned} \mu\sum_{i=0}^{\hat{M}-1}\sum_{j=0}^{M_3-1} \hat{s}_i s_{3,j}\mathbf{R}_{j-i}\mathbf{w}_\infty = & \mu\sum_{i=0}^{\hat{M}-1}\sum_{j=0}^{M_2-1} \hat{s}_i s_{2,j}\mathbf{R}_{j-i}\mathbf{w}_{\mathrm{o}} \\ & -\mu\sum_{i=0}^{\hat{M}-1}\sum_{j=1}^{M_3-1} \hat{s}_i s_{3,j}\mathbf{R}_{j-i} \lim_{n\to\infty} E[\boldsymbol{\alpha}_j(n)]. \end{aligned} \qquad (26)$$

From (18), note that $\lim_{n\to\infty}\{E[\mathbf{w}^{\mathrm{T}}(n)] - E[\mathbf{w}^{\mathrm{T}}(n-j)]\} = \lim_{n\to\infty} E[\boldsymbol{\alpha}_j^{\mathrm{T}}(n)] = 0$, thereby canceling the effect of the compensating term $\Lambda_{\mathrm{f}}(n)$. Then, using such a relation in (26), one obtains

$$\mathbf{w}_\infty = \left(\sum_{i=0}^{\hat{M}-1}\sum_{j=0}^{M_3-1} \hat{s}_i s_{3,j}\mathbf{R}_{j-i}\right)^{-1} \sum_{i=0}^{\hat{M}-1}\sum_{j=0}^{M_2-1} \hat{s}_i s_{2,j}\mathbf{R}_{j-i}\mathbf{w}_{\mathrm{o}}. \quad (27)$$

3.3 Learning Curve

To determine the model expression for the learning curve of the MFxFeLMS algorithm, we substitute (12) into (10), and after simple mathematics, we find

$$\hat{e}_{\mathrm{f}}(n) = \sum_{i=0}^{M_2-1} s_{2,i}\mathbf{w}_{\mathrm{o}}^{\mathrm{T}}\mathbf{x}(n-i) - \sum_{i=0}^{M_3-1} s_{3,i}\mathbf{w}^{\mathrm{T}}(n)\mathbf{x}(n-i) + z_{\mathrm{f}}(n). \qquad (28)$$

Now, squaring (28), taking the expected value of both sides of the resulting expression, and according to A1, we obtain

$$E[\hat{e}_{\mathrm{f}}^2(n)] = \mathbf{w}_{\mathrm{o}}^{\mathrm{T}} \sum_{i=0}^{M_2-1}\sum_{j=0}^{M_2-1} s_{2,i}s_{2,j}\mathbf{R}_{j-i}\mathbf{w}_{\mathrm{o}}$$
$$-2\mathbf{w}_{\mathrm{o}}^{\mathrm{T}} \sum_{i=0}^{M_2-1}\sum_{j=0}^{M_3-1} s_{2,i}s_{3,j}\mathbf{R}_{j-i}E[\mathbf{w}(n)]$$
$$+\sum_{i=0}^{M_3-1}\sum_{j=0}^{M_3-1} s_{3,i}s_{3,j}\mathrm{tr}\{\mathbf{R}_{j-i}E[\mathbf{w}(n)\mathbf{w}^{\mathrm{T}}(n)]\} \quad (29)$$
$$+\sum_{i=0}^{M_2-1} s_{2,i}^2\sigma_z^2.$$

To complete the derivation of (29), we should obtain the second moment of $\mathbf{w}(n)$. For such, allowing for the slow adaptation condition, the following approximation can be used [14]:

$$E[\mathbf{w}(n)\mathbf{w}^{\mathrm{T}}(n)] \cong E[\mathbf{w}(n)]E[\mathbf{w}^{\mathrm{T}}(n)]. \quad (30)$$

4. SIMULATION RESULTS

In this section, two examples are presented; the first compares the performance of the FxFeLMS algorithm with the proposed MFxFeLMS algorithm and its model, for white Gaussian input data, under two situations: the same convergence speed and the same minimum steady–state error. In the second example, the proposed model is also assessed for colored Gaussian input data. For the two presented examples, a system identification problem is used. Also, we assume that the input signal path filter estimate is given by $\hat{\mathbf{s}} = \mathbf{s}_3 = \mathbf{s}_1 * \mathbf{s}_2$.

4.1 Example 1

In this example, the unknown plant is the length-17 vector $\mathbf{w}_{\mathrm{o}} = [-0.457\ -0.086\ -0.100\ -0.071\ 0.107\ 0.231\ 0.347\ -0.429\ 0.458\ -0.429\ 0.347\ 0.231\ 0.107\ -0.071\ -0.100\ -0.086\ -0.457]^{\mathrm{T}}$. The input data is white with variance $\sigma_x^2 = 1$, and $\mathbf{s}_1 = [0.801\ 0.535\ 0.266]^{\mathrm{T}}$, $\mathbf{s}_2 = [0.122\ 0.468\ 0.623\ 0.519\ 0.274\ 0.169\ 0.055]^{\mathrm{T}}$, and $\hat{\mathbf{s}} = [0.977\ 0.441\ 0.783\ 0.874\ 0.664\ 0.421\ 0.208\ 0.074\ 0.014]^{\mathrm{T}}$. The (experimentally determined) maximum step-size values $\mu_{\max}$ for algorithm stability are 0.006 and 0.014 for the FxFeLMS and MFxFeLMS, respectively. Monte Carlo (MC) simulations are obtained from averaging 400 independent runs. Figure 2 shows the learning curves for both the algorithms (FxFeLMS and MFxFeLMS) enforcing the same convergence speed $(\mu = 0.003)$, as well as the proposed model (29). From these curves, we observe that the FxFeLMS presents larger mean-square error (MSE) in steady state than the MFxFeLMS. On the other hand, now enforcing the two algorithms to have the same steady-state MSE (see Figure 3), the MFxFeLMS exhibits faster convergence than the FxFeLMS. This latter converges only after 10^5 iterations. Moreover in Figures 2 and 3, we observe very good match between MC simulation and the proposed model of the MFxFeLMS algorithm using white input data.

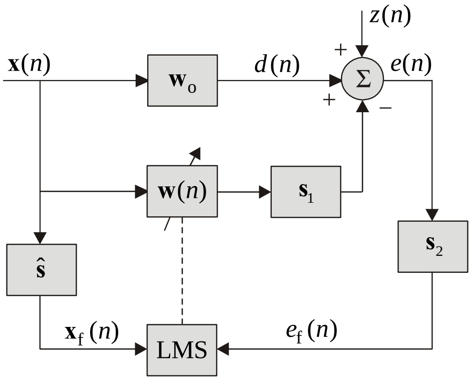


Figure 1 – Block diagram of the FxFeLMS algorithm.

4.2 Example 2

In this case, colored input data is used, obtained from an AR(2) process, given by

$$x(n) = a_1x(n-1) + a_2x(n-2) + u(n) \quad (31)$$

where $u(n)$ is white noise with $\sigma_u^2 = 0.39$ and the coefficients are $a_1 = 0.5833$ and $a_2 = -0.75$ with an eigenvalue spread of the input autocorrelation matrix equal to 63. The plant vector and path filter parameters are the same as in the previous example. The (experimentally determined) maximum step-size values for this case are $\mu_{\max} = 0.013$ and 0.023 for the FxFeLMS and MFxFeLMS algorithms, respectively. For this example, $\mu = 0.0006$ and 0.006 are, respectively, used for the FxFeLMS and MFxFeLMS. Figure 4 shows the mean weight behavior and learning curve (MSE) obtained from MC simulations and the theoretical model. From these figures, we observe very good agreement between MC simulations and the proposed model of the MFxFeLMS algorithm. In addition, comparing the curves in Figure 4(b), we notice that, for the same steady-state MSE, the MFxFeLMS algorithm achieves the fastest convergence.

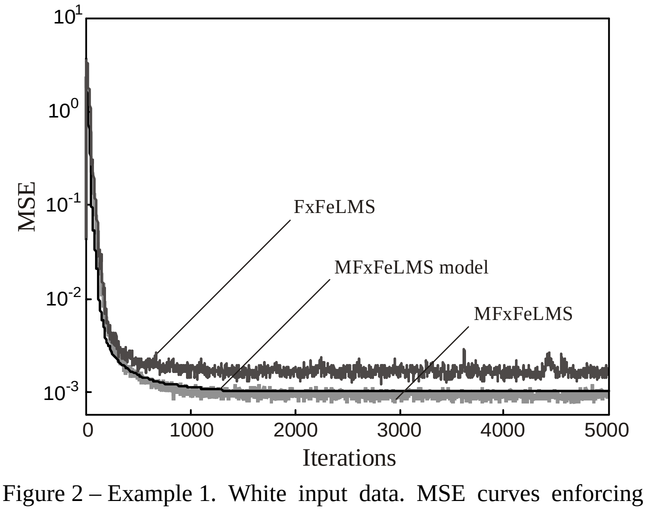


Figure 2 – Example 1. White input data. MSE curves enforcing the same convergence speed for both the FxFeLMS and MFxFeLMS algorithms, and proposed model (29), for $\mu = 0.003$.

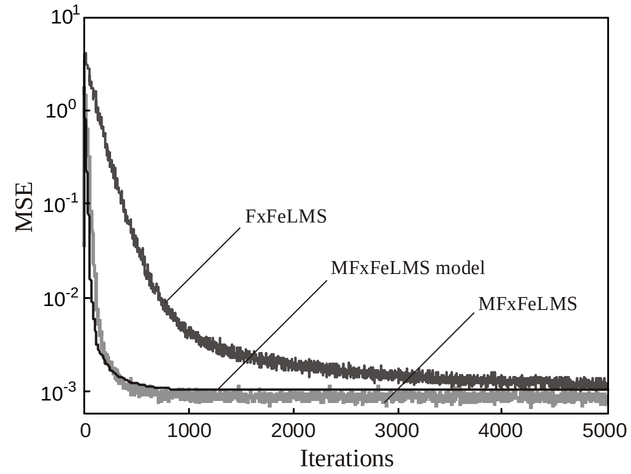
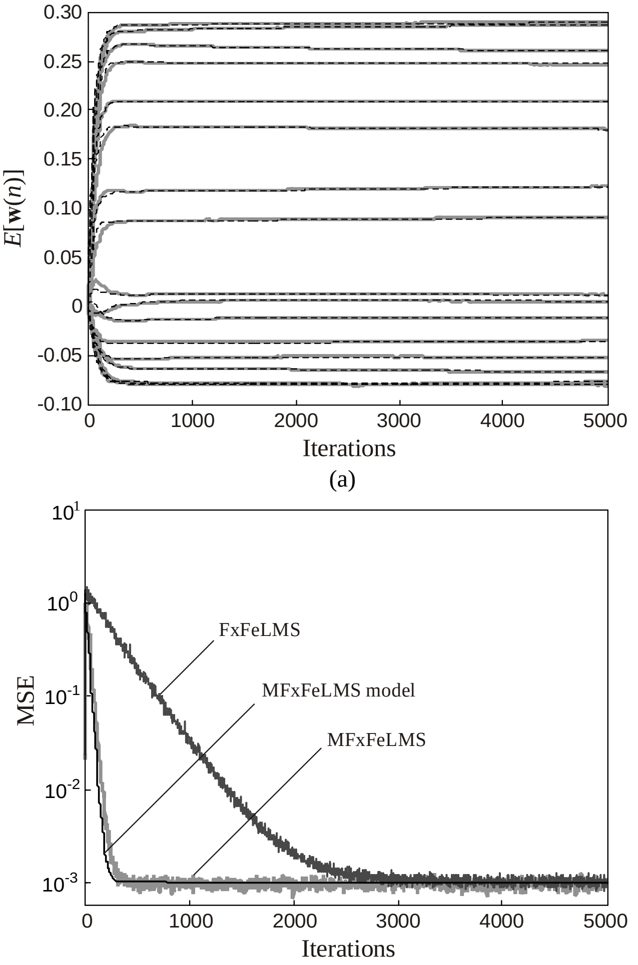


Figure 3 – Example 1. White input data. MSE curves enforcing the same steady-state MSE for both FxFeLMS and MFxFeLMS algorithms, using, respectively, $\mu = 0.0006$ and 0.006, and proposed model (29).



(a)

(b)

Figure 4 – Example 2. Colored input data. (a) Mean weight behavior of the MFxFeLMS algorithm using $\mu = 0.006$: (gray lines) simulation, (dashed-dark lines) proposed model (21). (b) MSE curves: FxFeLMS algorithm simulation with $\mu = 0.0006$, and MFxFeLMS algorithm simulation and proposed model (29) using $\mu = 0.006$.

5. CONCLUSIONS

In this paper, the modified fitered-reference/fitered-error LMS (MFxFeLMS) algorithm is proposed. In addition, analytical expressions for the first moment and the learning curves have been derived considering slow adaptation and without invoking IT. Through numerical simulations, the performance of the proposed algorithm was compared with that of the FxFeLMS. Also, curves of simulation and the proposed model of the MFxFeLMS algorithm were shown, attesting very good accuracy achieved by the analytical model.

REFERENCES

[1] R. D. Poltmann, “Conversion of the delayed LMS algorithm into the LMS algorithm,” *IEEE Signal Process. Lett.*, vol. 2, no. 12, pp. 223, Dec. 1995.

[2] S. C. Douglas, “An efficient implementation of the modified filtered-x LMS algorithm,” *IEEE Signal Process. Lett.*, vol. 4, no. 10, pp. 286-288, Oct. 1997.

[3] S. M. Kuo and D. R. Morgan, *Active Noise Control Systems*: *Algorithms and DSP Implementations*. New York: John Wiley, 1996.

[4] S. J. Eliott and P. A. Nelson, “Active noise control,” *IEEE Signal Processing Mag.*, vol. 10, no. 4, pp. 12-35, Oct. 1993.

[5] E. A. Wan, “Adjoint LMS: An efficient alternative to the filtered-x LMS and multiple error LMS algorithms,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Atlanta, USA, May 1996, vol. 3, pp. 1842-1845.

[6] S. Shaffer and C. S. Williams, “The filtered error LMS algorithm,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Boston, USA, Apr. 1983, pp. 41-44.

[7] J. R. V. Lopez, O. J. Tobias, and R. Seara, “Stochastic model for the modified filtered-error LMS algorithm,” in *Proc. 16th European Signal Process. Conf.*, Lausanne, Switzerland, Sep. 2008, pp. 1-5.

[8] L. Sujbert, “A new fitered LMS algorithm for active noise control,” in *Proc. Int. EAA Symp. Active Control of Sound and Vibration*, Fort Lauderdale, Florida, USA, Dec. 1999, pp. 1101-1110.

[9] S. Miyagi and H. Sakai, “Mean-square performance of the filtered-reference/filtered-error LMS algorithm,” *IEEE Trans. Circuits Syst.*, vol. 52, no. 5, pp. 2454-2463, Nov. 2005.

[10] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1998.

[11] E. Bjarnason, “Analysis of the filtered-X LMS algorithm,” *IEEE Trans. Speech and Audio Process.*, vol. 3, no. 6, pp. 504-514, Nov. 1995.

[12] O. J. Tobias, J. C. M. Bermudez, N. J. Bershad, and R. Seara, “Mean weight behavior of the filtered-X LMS algorithm,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Seattle, USA, May 1998, pp. 3545-3548.

[13] —— “Second moment analysis of the filtered-X LMS algorithm,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Phoenix, USA, Mar. 1999, pp. 1873-1876.

[14] N. J. Bershad, P. Celka, and J. M. Vesin, “Stochastic analysis of gradient adaptive identification of nonlinear systems with memory for Gaussian data and noisy input and output measurements,” *IEEE Trans. Signal Process.*, vol. 47, no. 3, pp. 675-689, Mar. 1999.