The MMFxLMS Algorithm for Active Noise Control with On-line Secondary Path Modelling

Paulo A. C. Lopes, José A. B. Gerald, Moisés Piedade

Instituto de Engenharia de Sistemas e Computadores, Investigação e Desenvolvimento, Instituto Superior Técnico, Universidade de Lisboa, INESC-ID/IST/UL, Rua Alves Redol n.9, 1000-029 Lisbon, Portugal

Abstract

This paper presents an architecture namely the Mirror MFx (MMFx) for adapting adaptive filtering algorithms for Active Noise Control (ANC) with on-line secondary path modelling. The proposed architecture is used in conjunction with the LMS algorithm, resulting in the MMFxLMS algorithm. A time domain analysis of the algorithm is presented, showing that the algorithm converges regardless of secondary path modeling errors. Simulations of the algorithm in different conditions but with the same parameters result in 100% convergence. The algorithm is especially suited to deal with large and sudden changes in the secondary path when the ANC system is in operation. Comparisons with competing algorithms are made, showing that they do not reach the same performance.

Keywords: Active Noise Control, On-line Secondary Path Modelling, LMS, FxLMS, MFxLMS, Sudden Changes.

1. Introduction

In active noise control (ANC) [1, 2, 3, 4, 5, 6, 7, 8] a sound wave (anti-noise) with opposite phase to a noise wave is used to reduce the noise. It works well at low frequencies, acting as a complement to traditional passive techniques. The most used algorithm in ANC is the filtered-x least mean squares (FxLMS) [1, 8] algorithm. The modified FxLMS (MFxLMS) algorithm [9] is also common. Both algorithms require a secondary path model,

Email address: paulo.lopes@tecnico.ulisboa.pt (Paulo A. C. Lopes)

but none is very sensitive to secondary path modelling errors [10, 1]. The secondary path model can be obtained before the starting of the ANC system (off-line). But if the secondary path varies significantly while the ANC system in on then on-line modelling is required. There are several of on-line secondary path modelling algorithms. These can be based on the overall modelling algorithm (OMA) [1, 2, 8] in the simultaneous equations method [11, 12] or with auxiliary noise [13, 14, 15, 16, 17, 18, 19, 20, 21, 22]. However, most of these algorithms are not suitable for dealing with sudden secondary path changes, but only deal with slow changes. They can become unstable after sudden changes. The proposed algorithm however, is stable even with incorrect secondary path models and deals very well with sudden changes.

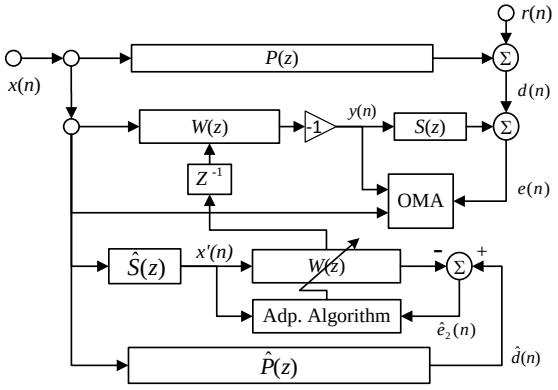

Figure 1: The MMFxLMS algorithm.

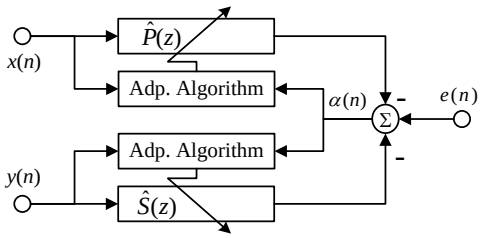

Figure 2: The overall modelling algorithm (OMA).

The proposed algorithm is represented in Figs. 1 and 2 and in Table 1. Fig. 2 is the detail of the block named OMA in Fig. 1. This correspond to equations (5) to (8) in Table 1. Vectors are shown in bold face letters,

Table 1: The MMFxLMS algorithm.

$$\hat{\mathbf{s}}(0) = [1, 0 \dots 0]^{\mathrm{T}} \tag{1}$$
$$\hat{\mathbf{p}}(0) = \mathbf{w}(0) = 0 \tag{2}$$
$$\mathbf{x}(0), \mathbf{x}'(0), \mathbf{y}(0) = 1 \tag{3}$$

$$y(n) = -\mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n) \tag{4}$$
$$\hat{e}_1(n) = \hat{\mathbf{p}}^{\mathrm{T}}(n)\mathbf{x}(n) + \hat{\mathbf{s}}^{\mathrm{T}}(n)\mathbf{y}(n) \tag{5}$$
$$\alpha(n) = e(n) - \hat{e}_1(n) \tag{6}$$
$$\hat{\mathbf{p}}(n+1) = \hat{\mathbf{p}}(n) + \frac{\mu_{sp}\mathbf{x}(n)\alpha(n)}{\mathbf{x}^{\mathrm{T}}(n)\mathbf{x}(n) + \mathbf{y}^{\mathrm{T}}(n)\mathbf{y}(n)} \tag{7}$$
$$\hat{\mathbf{s}}(n+1) = \hat{\mathbf{s}}(n) + \frac{\mu_{sp}\mathbf{y}(n)\alpha(n)}{\mathbf{x}^{\mathrm{T}}(n)\mathbf{x}(n) + \mathbf{y}^{\mathrm{T}}(n)\mathbf{y}(n)} \tag{8}$$
$$x'(n) = \hat{\mathbf{s}}^{\mathrm{T}}(n)\mathbf{x}(n) \tag{9}$$
$$\hat{d}(n) = \hat{\mathbf{p}}^{\mathrm{T}}(n)\mathbf{x}(n) \tag{10}$$
$$\hat{e}_2(n) = \hat{d}(n) - \mathbf{w}^{\mathrm{T}}(n)\mathbf{x}'(n) \tag{11}$$
$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu\,\mathbf{x}'(n)\hat{e}_2(n)}{\mathbf{x}'^{\mathrm{T}}(n)\mathbf{x}'(n)} \tag{12}$$

and $\mathbf{x}^{\mathrm{T}}$ represents the transpose of vector $\mathbf{x}$. The vectors $\mathbf{x}(n)$, $\mathbf{x}'(n)$ and $\mathbf{y}(n)$ are formed by past samples of the corresponding scalar signals, as in $\mathbf{x}(n) = [x(n) \dots x(n-N+1)]^{\mathrm{T}}$. This vectors should be carefully initialized.

The idea is to model the primary and secondary paths using the OMA, and use the model of the primary path to estimate the desired signal, $d(n)$. This is the desired output of the concatenation of the secondary path, $S(z)$, and controller filter, $W(z)$. Then the controller and secondary path filters are exchanged resulting in the algorithm as presented. This can also be derived by creating a mirror copy of the acoustics paths using the obtained models, and then using the MFxLMS algorithm. Due to the mirror operation the algorithm is called Mirror MFxLMS or MMFxLMS. The adaptive algorithm

used is the NLMS [23] as presented in Table 1. Note that in the proposed algorithm there is no need for prior knowledge of the primary and secondary paths.

2. State of the Art

In [13] Eriksson proposes the original on-line secondary path modelling technique using auxiliary noise. In [24] a comparison of the overall modelling technique, with the auxiliary noise technique, is presented. The primary noise is removed from the modelling of the secondary path using an auxiliary filter in [14]. [25] also reduces the disturbance in the modelling of the secondary path by removing the delayed residual error signal. [15] extends [14] by removing the additive noise signal from the adaptation of the auxiliary filter. [16] introduces auxiliary noise power scheduling and imposes a constraint on the norms of the adaptive filters. [17] compares [15] with [14]. [18] uses a variable step size that increases with the convergence of the ANC system and in [19] a convergence measure is used to control the auxiliary noise power. [20] makes the ratio of the auxiliary noise at the error microphone to the residual noise a constant, and uses optimum values for the step size of the controller and secondary path filters. [21] proposes two algorithms, an on-off algorithm that turns the additive noise off when the current estimate performs worse than the best so far, and on when the noise level increases; and an auxiliary noise power scheduling algorithm where the auxiliary noise signal level is proportional to the power of the residual noise signal. [22] proposes a two-stage auxiliary noise power scheduling algorithm. At stage one the auxiliary noise signal level is similar to [20] and in stage two it is proportional to the square of the residual noise signal power. In [26] the auxiliary noise level is similar to [20], but the ratio is variable. The ratio is greater when noise reduction is high and low when noise reduction is lower. This allows for faster convergence, resulting in better performance when dealing with sudden changes in the secondary path. The proposed algorithm does not use auxiliary noise, and is based on the overall modelling technique.

3. Time Domain Analysis

Take the algorithm in Table 1. Most of the signals depend on n, so we dropped the dependence on n on some signals, in order to simplify the notation. Let,

$$\mathbf{A}(n) = \begin{pmatrix} \mu'_{sp}\mathbf{x}\mathbf{x}^{\mathrm{T}} & \mu'_{sp}\mathbf{x}\mathbf{y}^{\mathrm{T}} & 0 \\ \mu'_{sp}\mathbf{y}\mathbf{x}^{\mathrm{T}} & \mu'_{sp}\mathbf{y}\mathbf{y}^{\mathrm{T}} & 0 \\ 0 & 0 & \mu'\mathbf{x}'\mathbf{x}'^{\mathrm{T}} \end{pmatrix} \tag{13}$$

where $\mu'_{sp} = \mu_{sp}/(\mathbf{x}^{\mathrm{T}}(n)\mathbf{x}(n) + \mathbf{y}^{\mathrm{T}}(n)\mathbf{y}(n))$ and $\mu' = \mu/(\mathbf{x}'^{\mathrm{T}}(n)\mathbf{x}'(n))$. This is a block diagonal matrix, formed by two blocks, $\mathbf{A}_1(n)$ and $\mathbf{A}_2(n)$, with $\mathbf{A}_2(n) = \mu'\mathbf{x}'\mathbf{x}'^{\mathrm{T}}$ and

$$\mathbf{A}_1(n) = \begin{bmatrix} \mu'_{sp}\mathbf{I} & 0 \\ 0 & \mu'_{sp}\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} [\mathbf{x}^{\mathrm{T}}\mathbf{y}^{\mathrm{T}}]. \tag{14}$$

And let,

$$\boldsymbol{\pi}(n) = [\hat{\mathbf{p}}^{\mathrm{T}}(n), \hat{\mathbf{s}}^{\mathrm{T}}(n), \mathbf{w}^{\mathrm{T}}(n)]^{\mathrm{T}} \tag{15}$$

$$\boldsymbol{\pi}_o(n) = [\mathbf{p}^{\mathrm{T}}(n), \mathbf{s}^{\mathrm{T}}(n), \mathbf{w}_o^{\mathrm{T}}(n)]^{\mathrm{T}} \tag{16}$$

$$\mathbf{r}(n) = [\mu'_{sp}\mathbf{x}^{\mathrm{T}}(n)r(n), \mu'_{sp}\mathbf{y}^{\mathrm{T}}(n)r(n), 0]^{\mathrm{T}}. \tag{17}$$

The proposed algorithm can be written as,

$$\boldsymbol{\pi}(n+1) = (\mathbf{I} - \mathbf{A}(n))\boldsymbol{\pi}(n) + \mathbf{A}(n)\boldsymbol{\pi}_o(n) + \mathbf{r}(n), \tag{18}$$

where $\mathbf{w}_o(n)$ was defined so that,

$$\mathbf{x}^{\mathrm{T}}(n)\hat{\mathbf{p}}(n) = \mathbf{x}'^{\mathrm{T}}(n)\mathbf{w}_o(n). \tag{19}$$

Taking the z-transform, this corresponds to having $\hat{S}(z)W_o(z) \approx \hat{P}(z)$. Note, however, that in general $\mathbf{w}_o(n)$ will be time varying. This will degrade the performance of the algorithm, because it makes $\boldsymbol{\pi}_o(n)$ also time varying. Defining $\Delta\boldsymbol{\pi}(n) = \boldsymbol{\pi}(n) - \boldsymbol{\pi}_o(n)$ and $d\boldsymbol{\pi}_o(n+1) = \boldsymbol{\pi}_o(n+1) - \boldsymbol{\pi}_o(n)$ we can rewrite (18) as,

$$\Delta\boldsymbol{\pi}(n+1) = (\mathbf{I} - \mathbf{A}(n))\Delta\boldsymbol{\pi}(n) - d\boldsymbol{\pi}_o(n+1) + \mathbf{r}(n) \tag{20}$$

As long as the absolute value of eigenvalues of the matrix $\mathbf{I} - \mathbf{A}(n)$ are less than one, $\boldsymbol{\pi}(n+1)$ will converge to $\boldsymbol{\pi}_o(n)$, as soon as this stabilizes so that $d\boldsymbol{\pi}_o(n+1)$ is zero, apart from a noise term due to $r(n)$. This is equivalent to having the eigenvalues of $\mathbf{A}$ between zero and two. Note, that the eigenvalues of $\mathbf{A}$ are equal to the union of the eigenvalues of $\mathbf{A}_1$ and $\mathbf{A}_2$. Both matrixes have characteristic one. The eigenvalues of $\mathbf{A}_1$ are $\lambda =$

$\mu'_{sp}\mathbf{x}^{\mathrm{T}}\mathbf{x} + \mu'_{sp}\mathbf{y}^{\mathrm{T}}\mathbf{y}$, corresponding to eigenvector $u = [\mu'_{sp}\mathbf{x}^{\mathrm{T}}, \mu'_{sp}\mathbf{y}^{\mathrm{T}}]^{\mathrm{T}}$, and $\lambda = 0$. The eigenvalues of $\mathbf{A}_2$ are $\lambda = \mu'\mathbf{x}'^{\mathrm{T}}\mathbf{x}'$, corresponding to eigenvector $u = \mathbf{x}'^{\mathrm{T}}$, and $\lambda = 0$. So for convergence one should have $\mu'_{sp}(\mathbf{x}^{\mathrm{T}}\mathbf{x} + \mathbf{y}^{\mathrm{T}}\mathbf{y}) < 2$ or $\mu_{sp} < 2$; and $\mu' < 2/(\mathbf{x}'^{\mathrm{T}}\mathbf{x}')$ or $\mu < 2$. The zero eigenvalues require special attention. This will result in eigenvalues of $\mathbf{I} - \mathbf{A}(n)$ equal to one. So one gets decreasing modes and constant modes at each iteration of the algorithm. Still, as long as there are persistent excitation signals, the decreasing modes will spread through all the modes of the state, resulting in global convergence. However, if some modes are not excited then they will get stuck on their initial values. Note that for constant modes $\mathbf{r}(n)$ is zero, as shown in the following.

Since $\mathbf{A}$ is a block diagonal matrix, then (20) can be split into two equations, one related to the update of $\hat{\mathbf{p}}$ and $\hat{\mathbf{s}}$ and the other related to the update of $\mathbf{w}$. The component of $d\boldsymbol{\pi}_o(n+1)$ used in the first equation is zero because $\mathbf{p}^{\mathrm{T}}(n)$ and $\mathbf{s}^{\mathrm{T}}(n)$ are constant, since we are analyzing convergence after the primary path and secondary path stabilize. Note that the same is not true for the component of $d\boldsymbol{\pi}_o(n+1)$ used in the second equation because it depends on $\hat{\mathbf{p}}^{\mathrm{T}}(n)$ and $\hat{\mathbf{s}}^{\mathrm{T}}(n)$ that are time varying. However, this will also become true after $\hat{\mathbf{p}}^{\mathrm{T}}(n)$ and $\hat{\mathbf{s}}^{\mathrm{T}}(n)$ converges.

This results in,

$$\Delta\boldsymbol{\eta}(n+1) = (\mathbf{I} - \mu'_{sp}\mathbf{v}(n)\mathbf{v}^{\mathrm{T}}(n))\Delta\boldsymbol{\eta}(n) + \mu'_{sp}\mathbf{v}(n)r(n) \tag{21}$$

with $\Delta\boldsymbol{\eta}(n) = [\mathbf{p}^{\mathrm{T}}(n) - \hat{\mathbf{p}}^{\mathrm{T}}(n), \mathbf{s}^{\mathrm{T}}(n) - \hat{\mathbf{s}}^{\mathrm{T}}(n)]$, $\mathbf{v}(n) = [\mathbf{x}^{\mathrm{T}}(n)\mathbf{y}^{\mathrm{T}}(n)]^{\mathrm{T}}$ and $\mathbf{A}_1(n) = \mu'_{sp}\mathbf{v}(n)\mathbf{v}^{\mathrm{T}}(n)$. The matrix $\mathbf{v}(n)\mathbf{v}^{\mathrm{T}}(n)$ is hermitian and has eigenvalues $\lambda = ||\mathbf{v}(n)||^2$ corresponding to the normalized eigenvector $\mathbf{v}/||\mathbf{v}||$ and $\lambda = 0$. Since the matrix is hermitian it is diagonalizable by an orthogonal matrix $\mathbf{Q}(n)$, with $\mathbf{v}(n)\mathbf{v}^{\mathrm{T}}(n) = \mathbf{Q}(n)\mathbf{D}(n)\mathbf{Q}^{\mathrm{T}}(n)$. The matrix $\mathbf{Q}(n)$ is given by,

$$\mathbf{Q}(n) = [\mathbf{v}(n)/||\mathbf{v}(n)||, \mathbf{q}_1(n), \ldots, \mathbf{q}_{L_s+L_p-1}(n)] \tag{22}$$

and

$$\mathbf{D}(n) = \mathrm{diag}(||\mathbf{v}(n)||^2, 0, \ldots, 0) \tag{23}$$

where the vectors $\mathbf{q}_i(n)$ are unitary and orthogonal to $\mathbf{v}(n)$ and among them. Let $\Delta\boldsymbol{\eta}'(n) = \mathbf{Q}^{\mathrm{T}}\Delta\boldsymbol{\eta}(n)$ be $\Delta\boldsymbol{\eta}(n)$ in the new coordinate system that diagonalises $\mathbf{v}(n)\mathbf{v}^{\mathrm{T}}(n)$. Then we can write (21) in the new coordinate system as,

$$\Delta\boldsymbol{\eta}'(n+1) = (\mathbf{I} - \mu'_{sp}\mathbf{D}(n))\Delta\boldsymbol{\eta}'(n) + \mu'_{sp}\mathbf{Q}^{\mathrm{T}}(n)\mathbf{v}(n)r(n) \tag{24}$$

where

$$\mathbf{Q}^{\mathrm{T}}(n)\mathbf{v}(n) = [||\mathbf{v}(n)||, 0, \ldots, 0]^{\mathrm{T}} \tag{25}$$

since $\mathbf{q}_i(n)$ are orthogonal to $\mathbf{v}(n)$. This means that as long as $r(n)$ is small the first coordinate of $\Delta\boldsymbol{\eta}'(n+1)$ will decrease exponentially while the other will remain the same. $||\Delta\boldsymbol{\eta}'(n+1)||$ will also decrease. Since $\mathbf{Q}(n)$ is orthogonal and preserves norms $||\Delta\boldsymbol{\eta}(n+1)||$ also decreases. As $||\Delta\boldsymbol{\eta}(n+1)||$ decreases $\hat{\mathbf{p}}(n)$ and $\hat{\mathbf{s}}(n)$ will converge to the actual primary path and secondary path, $\mathbf{s}(n)$ and $\mathbf{p}(n)$.

The same type of analysis can be made for the second part of the split of (20), regarding the controller filter. In this case we have $A_2(n) = \mu'\mathbf{x}'(n)\mathbf{x}'^{\mathrm{T}}(n)$. Note that in this case we assume that $\hat{\mathbf{p}}^{\mathrm{T}}(n)$ and $\hat{\mathbf{s}}^{\mathrm{T}}(n)$ have converged so that $\mathbf{w}_o(n)$ is constant resulting that $\boldsymbol{\pi}_o(n+1)$ is also constant. This simply means that $\mathbf{w}(n)$ converges only after $\hat{\mathbf{p}}^{\mathrm{T}}(n)$ and $\hat{\mathbf{s}}^{\mathrm{T}}(n)$. Note that there are no conditions on the value of the secondary path estimate for convergence, so the controller will converge regardless of secondary path modelling errors.

4. Computational complexity

In this section we presented the computational complexity of the proposed and competing algorithm as the number of multiplications ($\times$), divisions (%) and squares roots ($\sqrt{.}$) in table 2. The size of the controller, primary path and secondary path filters are given by N_W, N_P and N_S respectively. The algorithms compared are: the MMFxLMS, the algorithm in [22] named Ahmed2013, the algorithm by Carini in [20], the algorithm in [26] named Lopes and the original algorithm by Eriksson in [13]. This are the ones that obtained the best results in the simulations of the following section plus the Eriksson algorithm. Table 3 presents a numerical example of the computational complexity, namely with $N_W = N_S = N_P = 128$ and $D = 16$.

Algorithm	multiplications ($\times$)	%	$\sqrt{.}$
Proposed (MMFxLMS)	$4N_W + 3N_P + 4N_S + 3$	3	-
Ahmed 2003	$4N_W + 6N_S + 29$	5	1
Carini	$7N_W + 6N_S + 4D + 19$	6	1
Lopes	$4N_W + 6N_S + 14$	5	1
Eriksson	$2N_W + 3N_S + 2$	-	-

Table 2: Computational complexity of proposed and competing algorithms.

Algorithm	multiplications (×)	%	$\sqrt{\cdot}$
Proposed (MMFxLMS)	1411	3	-
Ahmed 2003	1309	5	1
Carini	1747	6	1
Lopes	1294	5	1
Eriksson	642	-	-

Table 3: Example of computational complexity of proposed and competing algorithms with $N_W = N_S = N_P = 128$ and $D = 16$.

5. Simulation

In this section simulations of the proposed algorithm are presented and compared with other related algorithms. The secondary path was modelled as a non-minimum phase multi-path filter formed by three fractional delays implemented by sync filters, namely with delays of 15.7, 23.3 and 18.4 and amplitudes 1.3, -0.3 and 0.4. The primary path was similar but had different values for the delays and amplitudes. The frequency response of the primary and secondary paths was plotted in Fig. 3. The root mean square of the residual noise signal, $r(n)$, was set to -30 dB. The sizes of all the adaptive filters were set to 64.

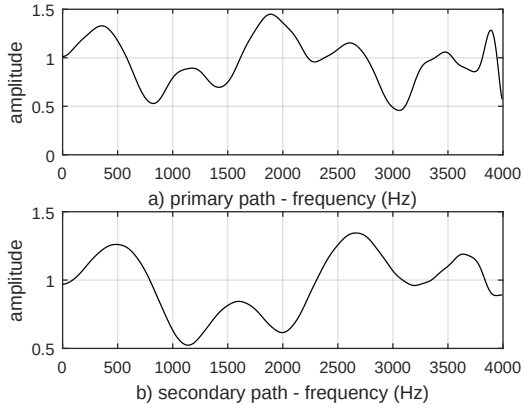

Figure 3: Frequency response of the primary a) and secondary b) paths.

After 50000 samples a delay of 5 samples was added to the secondary path making the FxLMS algorithm unstable unless the new path was re-estimated quickly. The primary path was changed to a new path with different delays

and amplitudes. The compared algorithms were the MMFxLMS, the algorithm by Carini in [20], algorithms 1 and 2 in [21] named Ahmed 1 and Ahmed2, the algorithm in [22] named Ahmed2013 and the algorithm in [26] named Lopes. Some small changes were made to some of the algorithms, which improved their performance in the simulations. In Carini the step size of the controller filter was multiplied by μ_2, to make the algorithm stable after the secondary path changes. In Ahmed2013 the value of $\rho(n)$ was limited to the range from 0 to 1 and the value of $G(n)$ was made always positive.

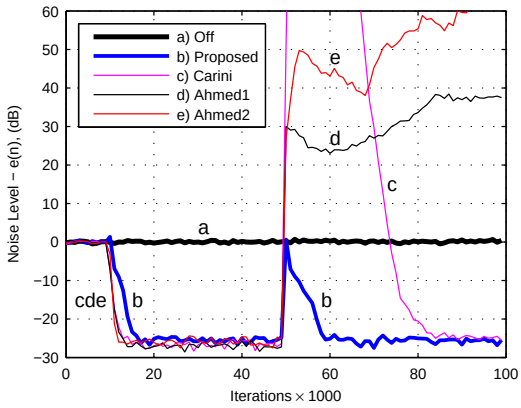

Figure 4: Comparison of algorithm convergence curves for broadband signals with the parameters tuned to maximize convergence rate at the beginning.

In Fig. 4 the proposed algorithm is compared with Carini, Ahmed1 and Ahmed2 for the case of broadband signals. The reference signal was colored broadband noise obtained by passing a white noise signal through a FIR filter with a frequency response of 1 from 0 to $\pi/2$ rad/s and 0.3 from $\pi/2$ rad/s to π rad/s. The MMFxLMS, Ahmed2013 and Lopes algorithms are compared in Fig. 7. In Fig. 4 the parameters of the algorithms were all optimized by trial and error to maximize the convergence rate at the beginning. In the case of Carini a conservative value of $R = 3$ was used. This results in fast convergence of all algorithms, but after the sudden secondary path changes all algorithms diverged except the proposed one. In Fig. 5 the parameters of the algorithms were set so that they were stable after the sudden secondary path change. The new parameters are shown in Table 4. Now all the algorithms but the proposed one and Carini suffer from slow convergence. In Carini $R = 1$ was used. However, Carini has a higher overshot and residual noise.

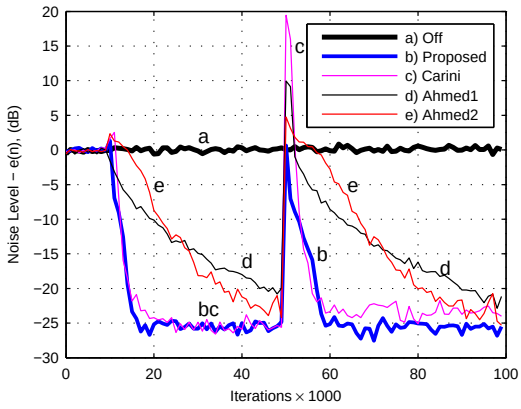

Figure 5: Comparison of algorithm convergence curves for broadband signals with the parameters tuned to ensure stability after sudden changes.

Table 4: Algorithms parameters for broadband signals (Fig. 5 and 7).

MMFxLMS	$\mu = \mu_{sp} = 0.5$
Ahmed1	$\mu = 0.01$, $\mu_S = 0.1$, $\epsilon = 3$, $\xi = 16$,$\lambda = 0.9$,$\sigma_v = 3$
Ahmed2	$\mu = 0.02$, $\mu_s = 0.1$. $\alpha = 0.9$, $\gamma = 1$, $\lambda = 0.9$
Carini	$\mu_2 = 0.5$, $R = 1$, $D = 16$, $\mu_{s\,\min} = 0.02$, $\hat{\lambda} = 0.7$, $\lambda = 0.9$
Ahmed2013	$\mu = 0.5$, $\mu_s = 0.1$, $\alpha = 0.99$, $\gamma_{\min} = 0.3$, $\gamma_{\max} = 0.9$, $\lambda = 0.9$
Lopes	$\mu = 0.2$, $\mu_s = 0.2$, $\lambda = 0.9$, $k_R = 0.1$, $\beta = 5$, $A = 10$

Fig. 6 shows the same simulation as Fig. 5 but here the reference signal is formed by two sinusoids of frequencies 150 Hz and 330 Hz, and amplitudes of 1 and 0.3 while the sampling frequency is 8000 Hz. The parameters of the algorithm used in this simulation are presented in Table 5. Once again the MMFxLMS is much faster than all the other algorithms.

Fig. 7 and 8 compares the MMFxLMS, Ahmed2013 and Lopes algorithms for broadband and narrowband reference signals respectively. All show good performance.

Table 5: Algorithms parameters for narrowband signals (Fig. 6 and 8).

MMFxLMS	$\mu = \mu_{sp} = 0.2$
Ahmed1	$\mu = 0.02$, $\mu_S = 0.1$, $\epsilon = 0.5$, $\xi = 16$,$\lambda = 0.9$,$\sigma_v = 3$
Ahmed2	$\mu = 0.01$, $\mu_s = 0.01$. $\alpha = 0.99$, $\gamma = 0.1$, $\lambda = 0.9$
Carini	$\mu_2 = 0.1$, $R = 1$, $D = 16$, $\mu_{s\,\min} = 0.01$, $\hat{\lambda} = 0.9$, $\lambda = 0.99$
Ahmed2013	$\mu = 0.1$, $\mu_s = 0.3$, $\alpha = 0.99$, $\gamma_{\min} = 0.3$, $\gamma_{\max} = 0.9$, $\lambda = 0.9$
Lopes	$\mu = 0.1$, $\mu_s = 0.1$, $\lambda = 0.9$, $k_R = 0.1$, $\beta = 5$, $A = 10$

The paper proceeds to present a stability analysis of the same algorithms in different physical setups, but without changing the parameters used for the algorithms. We will only compare MMFxLMS, Ahmed2013, Carini and Lopes for the broadband case, and MMFxLMS, Ahmed2013 and Lopes for the narrowband case because these were the ones that had best performance in the previous simulations. For the case of the Ahmed2013 algorithm the value of γ used was very small. This led to the best results, but it almost

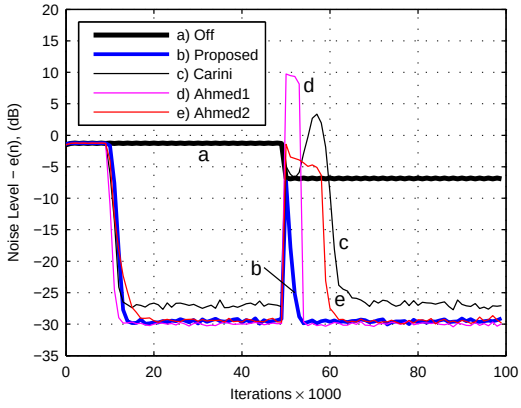

Figure 6: Comparison of algorithm convergence curves for narrowband signals with the parameters tuned to assure stability after sudden changes.

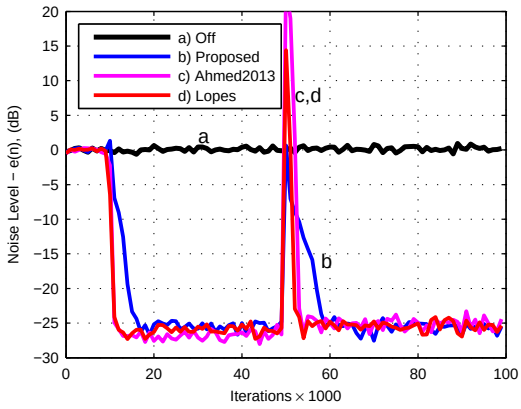

Figure 7: Comparison of algorithm convergence curves for broadband signals.

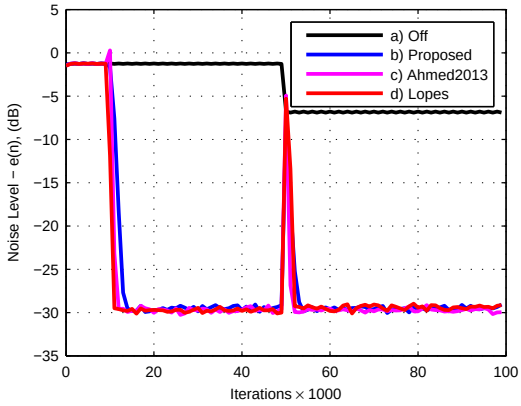

Figure 8: Comparison of algorithm convergence curves for narrowband signals.

turns off the noise in stage two, making it difficult for the algorithm to track slow changes in the secondary path. Tables 6 and 7 show the number of convergent runs and the median of the overshot in dB. There were 100 trials and seven cases for the physical setup. In each trial the fractional delays and amplitudes of the primary path and secondary multi-path models were random, resulting in random but physically reasonable models; and in the narrowband case the amplitudes and frequencies of the reference signal were also random with frequencies ranging between 50 Hz to 350 Hz. The different physical setups were: normal, the normal case for which the parameters were tuned; $x/10$ and $10x$, the level of the reference signal was decreased

Table 6: Number of convergent runs and the median of the overshot (db) in 100 trials, for broadband reference signals.

	MMFx	Ahmed2013	Carini	Lopes
Normal	100 / 4.3	100 / 22.8	94 / 45.7	100 / 12.9
$x/10$	100 / 4.6	97 / 21.8	88 / 50.6	100 / 11.1
$10x$	100 / 4.5	96 / 19.9	87 / 43.3	100 / 11.9
$S/10$	100 / 3.0	86 / 16.3	90 / 46.8	100 / 10.7
$10S$	100 / 16.0	100 / 41.6	89 / 50.8	100 / 13.3
$P/10$	100 / 17.2	93 / 43.7	86 / 54.6	100 / 10.4
$10P$	100 / 2.5	95 / 8.8	93 / 41.5	100 / 10.2

Table 7: Number of convergent runs and the median of the overshot (db) in 100 trials, for narrowband reference signals.

	MMFx	Ahmed2013	Lopes
Normal	100 / -1.8	100 / 3.7	100 / 5.9
$x/10$	100 / 0.2	100 / 1.8	99 / 6.2
$10x$	100 / -0.7	100 / 21.2	99 / 4.7
$S/10$	100 / 2.3	98 / 3.8	100 / 4.6
$10S$	100 / 1.2	99 / 0.1	97 / 5.3
$P/10$	100 / -0.3	97 / 0.5	83 / 6.8
$10P$	100 / 2.3	100 / 5.1	100 / 4.5

or increased 10 times; $S/10$ and $10S$, the amplitude of the secondary path was decreased or increased 10 times; $P/10$ and $10P$, the amplitude of the primary path was decreased or increased 10 times. Table 6 presents the results for broadband signals. It can be seen that the MMFxLMS and Lopes always converges, while that is not the case of the other algorithms. Table 7 presents the results for narrowband signals. In this case only the MMFxLMS always converges. Also, the overshot of the MMFxLMS is much lower than on the other algorithms.

6. Conclusion

This paper presents the Mirror Modified FxLMS algorithm (MMFxLMS). The algorithm is especially suited for active noise control with on-line secondary path modeling. It deals very well with sudden changes of the secondary path, because it is stable even with incorrect secondary path mod-

els. The algorithm is compared with algorithms with auxiliary noise power scheduling, showing better performance. The compared algorithms fail to converge after sudden secondary path changes if the physical setup is not the same as the one the algorithm was tuned for, while the MMFxLMS converges in all cases. Also, the overshot after the secondary path changes is much smaller.

7. Acknowledgements

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013.

[1] S. M. Kuo, D. R. Morgan, Active Noise Control Systems, Algorithms and DSP implementations, John Wiley and Sons, 1996.

[2] S. M. Kuo, D. R. Morgan, Active Noise Control: A tutorial review, Proc. IEEE 87 (6) (1999) 943–973.

[3] P. Nelson, S. Elliott, Active Control of Sound, Academic Press, 1996.

[4] S. Elliott, Signal Processing for Active Control, Academic Press, 2001.

[5] S. Elliott, Down with noise, IEEE Spectrum (1999) 54–61.

[6] Y. Kajikawa, W.-S. Gan, S. M. Kuo, Recent advances on active noise control: open issues and innovative applications, APSIPA Transactions on Signal and Information Processing 1 (2012) e3.

[7] S. D. Snyder, Active noise control primer, Springer Science & Business, 2012.

[8] C. H. Hansen, Active control of noise and vibration, CRC Press, 2013.

[9] M. Rupp, A. H. Sayed, Modified Fx-LMS algorithms with improved convergence performance, in: 29th Asilomar Conf. Signals, Systems and Computers, Vol. 2, Los Alamitos, CA, 1995, pp. 1255–1259.

[10] P. A. C. Lopes, M. S. Piedade, The behavior of the modified FX-LMS algorithm with secondary path modeling errors, IEEE Signal Process. Lett. 11 (2) (2004) 148–151.

[11] K. Fujii, J. Ohga, Method to update the coefficients of the secondary path filter under active noise control, Signal processing 81 (2) (2001) 381–387.

[12] K. Fujii, M. Muneyasu, J. Ohga, Active noise control system using the simultaneous equation method without the estimation of error path filter coefficients, Electronics and Communications in Japan (Part III: Fundamental Electronic Science) 85 (12) (2002) 101–108.

[13] L. J. Eriksson, M. C. Allie, Use of random noise for on-line transducer modeling in an adaptive active attenuation system, The Journal of the Acoustical Society of America 85 (2) (1989) 797–802.

[14] C. Bao, P. Sas, H. Van Brussel, Adaptive active control of noise in 3-d reverberant enclosures, Journal of sound and vibration 161 (3) (1993) 501–514.

[15] M. Zhang, H. Lan, W. Ser, Cross-updated active noise control system with online secondary path modeling, IEEE Trans. Speech Audio Process. 9 (5) (2001) 598–602.

[16] M. Zhang, H. Lan, W. Ser, A robust online secondary path modeling method with auxiliary noise power scheduling strategy and norm constraint manipulation, IEEE Trans. Speech Audio Process. 11 (1) (2003) 45–53.

[17] M. Zhang, H. Lan, W. Ser, On comparison of online secondary path modeling methods with auxiliary noise, IEEE Trans. Speech Audio Process. 13 (4) (2005) 618–628.

[18] M. T. Akhtar, M. Abe, M. Kawamata, A new variable step size LMS algorithm-based method for improved online secondary path modeling in active noise control systems, IEEE Transactions on Audio, Speech, and Language Processing 14 (2) (2006) 720–726.

[19] M. T. Akhtar, M. Abe, M. Kawamata, Noise power scheduling in active noise control systems with online secondary path modeling, IEICE Electronics Express 4 (2) (2007) 66–71.

[20] A. Carini, S. Malatini, Optimal variable step-size NLMS algorithms with auxiliary noise power scheduling for feedforward active noise control,

IEEE Transactions on Audio, Speech, and Language Processing 16 (8) (2008) 1383–1395.

[21] S. Ahmed, A. Oishi, M. T. Akhtar, W. Mitsuhashi, Auxiliary noise power scheduling for on-line secondary path modeling in single channel feedforward active noise control systems, in: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2012, pp. 317–320.

[22] S. Ahmed, M. T. Akhtar, X. Zhang, Robust auxiliary-noise-power scheduling in active noise control systems with online secondary path modeling, IEEE Transactions on Audio, Speech, and Language Processing 21 (4) (2013) 749–761.

[23] S. Haykin, Adaptive Filter Theory, Prentice-Hall, 1996.

[24] C. Bao, P. Sas, H. V. Brussel, Comparison of two on-line identification algorithms for Active Noise Control, in: Proc. Recent Advances in Active Control of Sound Vibration, 1993, pp. 38–51.

[25] S. M. Kuo, D. Vijayan, A secondary path modeling technique for active noise control systems, IEEE Trans. Speech Audio Process. 5 (4) (1997) 375.

[26] P. Lopes, J. Gerald, et al., Auxiliary noise power scheduling algorithm for active noise control with online secondary path modeling and sudden changes, IEEE Signal Processing Letters 22 (10) (2015) 1590–1594.