The final publication is available at

http://dx.doi.org/10.1109/fASLP.2016.2583065

Adaptive filtered-x algorithms for room equalization based on block-based combination schemes

Laura Fuster, Maria de Diego, *Member, IEEE,* Luis A. Azpicueta-Ruiz, *Member, IEEE,*
and Miguel Ferrer, *Member, IEEE,*

Abstract—Room equalization has become essential for sound reproduction systems to provide the listener with the desired acoustical sensation. Recently, adaptive filters have been proposed as an effective tool in the core of these systems. In this context, this paper introduces different novel schemes based on the combination of adaptive filters idea: a versatile and flexible approach that permits obtaining adaptive schemes combining the capabilities of several independent adaptive filters.

In this way, we have investigated the advantages of a scheme called combination of block-based adaptive filters which allows a blockwise combination splitting the adaptive filters into non-overlapping blocks. This idea was previously applied to the plant identification problem, but has to be properly modified to obtain a suitable behavior in the equalization application. Moreover, we propose a scheme with the aim of further improving the equalization performance using the a priori knowledge of the energy distribution of the optimal inverse filter, where the block filters are chosen to fit with the coefficients energy distribution. Furthermore, the biased block-based filter is also introduced as a particular case of the combination scheme, especially suited for low signal-to-noise ratios (SNRs) or sparse scenarios. Although the combined schemes can be employed with any kind of adaptive filter, we employ the filtered-x improved proportionate normalized least mean square (Fx-IPNLMS) algorithm as basis of the proposed algorithms, allowing to introduce a novel combination scheme based on partitioned block schemes where different blocks of the adaptive filter use different parameter settings. Several experiments are included to evaluate the proposed algorithms in terms of convergence speed and steady-state behavior for different degrees of sparseness and SNRs.

Index Terms—Convex combination, block-based algorithms, biased filters, filtered-x structures, room adaptive equalization.

I. INTRODUCTION

IN sound reproduction systems, a sound travels through an acoustic space before reaching a listener or a microphone. The behavior of the acoustic system between a source and a listening position could be characterized by an impulse response. Equalization (or room correction) techniques have been used for years to try to compensate for the effect of the audio and acoustic system. They aim at making the global impulse response of the sound reproduction channel as close as possible to the one desired. For real-time applications, this process is accomplished by prefiltering the input audio signal $x(n)$ with the equalization filter before it is propagated through the loudspeaker-enclosure-microphone (LEM) setup [1], represented with the L_h-length FIR filter $\mathbf{h}$, as shown in Fig. 1.

L. Fuster, M. de Diego and M. Ferrer are with the Instituto de Telecomunicaciones y Aplicaciones Multimedia, Universitat Politcnica de Valncia, Valencia, Spain (e-mail: lfuster@iteam.upv.es).

L A. Azpicueta is with the Department of Signal Theory and Communications, Universidad Carlos III, Madrid, Spain.
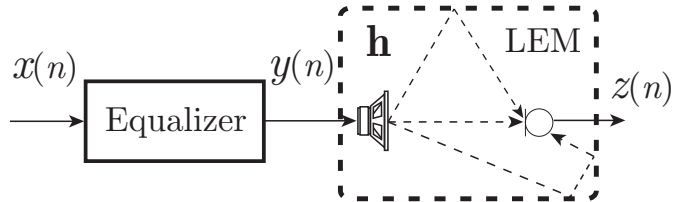

Fig. 1. Sound equalization system.

Thus, the combined effect of the equalizer and the LEM channel will allow to obtain a good approximation of the desired signal at the microphone. Here, although the goal of room equalization is not always to remove all reverberation from an audio signal, we consider that the desired signal is the input signal with a suitable source-microphone delay τ, $x(n - \tau)$. Moreover, while room equalization can also attempt to correct nonlinear distortion as in [1], the focus here is on linear compensation. Bearing this in mind, we need to estimate an L_w-length optimal coefficient vector $\mathbf{w}_o$, so that it corresponds to the inverse of the linear LEM channel. This can be mathematically described as $\mathbf{w}_o * \mathbf{h} = \delta(n - \tau)$, where $*$ denotes the discrete linear convolution. The design of the inverse filter is chosen to obtain a stable inverse using a modeling delay approximately half the length of the inverse filter [2].

Sound equalization systems including a wide number of possibilities, ranging from the simplified case of a single source and a single receiver (see Fig.1), which is the most straightforward to analyze, to more complex configurations considering multiple sources and/or listeners, sparse or non-so-sparse acoustic channels or time-varying characteristics of a given scenario. Those equalization filters can be implemented in two different forms: non-adaptive or adaptive. Several design methods have been presented over the years concerning non-adaptive equalization filters design either in the time or frequency domain, for instance [3] and [4]. These methods compute the inverse filters previously to perform the equalization process. However, practical audio applications involve time-varying scenarios and multiple loudspeakers and microphones (MIMO structure). For that reason, the use of adaptive filtering becomes an excellent tool [5], [6]. In the particular case of room equalization, several contributions have been recently proposed in the time, frequency or wave domain [7]–[15]. The adaptive schemes proposed in the time and frequency domain usually consider the least mean square (LMS) or the normalized LMS (NLMS) algorithms [5], [6], which represent a stable and simple solution.
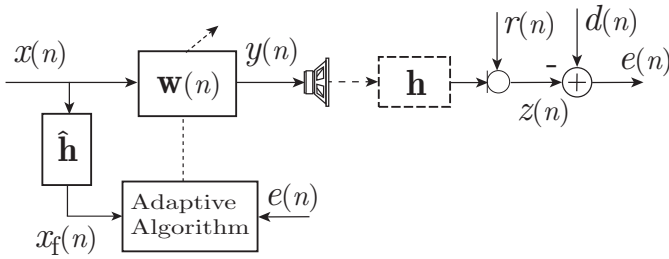
Fig. 2. Block diagram of an AE system.

In this paper, we focus on a single channel adaptive room equalization (AE) system whose general structure is shown in Fig. 2, which can be extended to the MIMO case similarly to that in [7]. The error signal $e(n)$ is obtained by subtracting the acoustic signal measured at the microphone $z(n)$ from the desired signal $d(n) = x(n - \tau)$,

$$e(n) = d(n) - z(n), \tag{1}$$

$$z(n) = \mathbf{h} * y(n) + r(n), \tag{2}$$

where $x(n)$ and $y(n)$ are the input and loudspeaker signals, respectively and $r(n)$ is an additive white Gaussian noise of zero mean and σ_r^2 variance, uncorrelated with the input signal $x(n)$.

It is important to note that a remarkable difference introduced by AE systems (similarly to active noise control (ANC) systems) compared to a system identification problem is due to the presence of the LEM channel $\mathbf{h}$ between the adaptive filter output and the error signal. The usual way to take into account this response and avoid negative effects on the algorithm performance consists in using a particular filtering structure named filtered-x scheme [16], that requires a previous estimation of the acoustic channel. This filtering scheme has been frequently used in ANC systems, [17]–[20]. For room equalization, a brief description of the filtered-x scheme is given for both single-channel and multichannel AE system in [7]. Also for multichannel AE systems, a decoupled version of the filtered-x LMS is presented in [11], and a wave-domain algorithm in [13].

Regarding LMS-type algorithms, one of their main drawbacks is that they distribute the adaptation energy equally among all filter coefficients and thus, they can suffer from slow convergence speed. To overcome this problem, the proportionate adaptive filter (PNLMS) [21] has been introduced to accelerate filter convergence in scenarios where the optimal solution presents a high degree of sparseness. PNLMS spends more energy on adapting the active coefficients, thus it converges faster than the NLMS. However, since it assumes that the filter solution is sparse, its performance degrades significantly when the optimal solution is not so sparse. Thus, the improved proportionate NLMS (IPNLMS) [22] tries to alleviate this problem improving filter convergence for different degrees of sparseness. However, its major drawback is that it requires to *a priori* know the degree of sparseness of the optimal solution, which rarely occurs in practical systems. The filtered-x version of the IPNLMS is called Fx-IPNLMS algorithm and it was introduced by the authors in [23]. Some

successful applications of the proportionated adaptive filters include system identification [24], acoustic echo cancellation (AEC) [21], ANC [23], as well as AE [7].

Moreover, in the last decade, a novel approach in adaptive filtering has received a great interest: the adaptive combination of adaptive filters. This strategy is based on the adaptive combination of several adaptive filters running in parallel. Each component adapts its coefficients independently in order to minimize its own cost function. If the combination is properly adapted, the combined filter obtains an improved behavior, performing at each moment as the best component filter and, under certain circumstances, even better [25]. These combined schemes are introduced to improve robustness when several kinds of adverse scenario conditions can impair the filter performance, and to facilitate the selection of filter parameters, alleviating the different trade-offs inherit to adaptive filters, for instance the well-know speed of convergence vs steady-state misadjustment compromise [26]. Combination of adaptive filters have been successfully employed in different signal processing applications, including system identification [26], [27], signal modality characterization [28], array beamforming [29], [30], adaptive line enhancement [31], and acoustic applications, such as AEC [32]–[35] and ANC [23], [36], [37].

Recently, a new scheme that can be considered as a particular case of the combination of filters, has been proposed to reduce the error of the adaptive filters [38] by means of inclusion of a bias. This method uses a scaling factor $\alpha(n)$ that multiplies the estimator, providing a biased estimator of the optimal solution that can outperform the unbiased one, especially when the SNR is low. The suitable selection of $\alpha(n)$ to bias the weights is a key issue of this scheme, and $\alpha(n)$ can be effectively adapted by considering this scheme as the convex combination of the output of a standard filter and that of a virtual zero filter. In [7], a biased version of the filtered-x IPNLMS algorithm for AE is proposed, which outperforms its unbiased version for low SNR values, even in multichannel conditions.

An important characteristic of many acoustic signal processing applications is the fact that the energy distribution of the impulse response is not uniform. Taking advantage of this feature, different schemes propose a block-based processing, highlighting:

- Block-based combination schemes, where instead of combining complete adaptive filters, each individual filter is split into M non-overlapping blocks, giving rise to M combinations. The mth combination mixes the outputs of the mth block of each adaptive filter. For instance, the block-based algorithm presented in [39] employs a different scaling factor for each block when the energy of the unknown plant is not uniformly distributed, and enhances the performance of the biased scheme that considers the same adaptive filter but only a shrinkage factor.
- Block-based adaptive filters, where different blocks of an adaptive filter adapt using different settings. For instance, a partitioned two-block IPNLMS algorithm is used with two different parameter settings per each block, thus

TABLE I
NOTATION

$\hat{\mathbf{h}}$	L_h-taps FIR estimated impulse response of $\mathbf{h}$
$x_\mathrm{f}(n)$	Input signal $x(n)$ filtered by $\hat{\mathbf{h}}$
$y_{m\mathrm{f}}(n)$	Output signal $y_m(n)$ of the m-block filter, filtered by $\hat{\mathbf{h}}$
$y_\mathrm{f}(n)$	Output signal $y(n)$ filtered by $\hat{\mathbf{h}}$
$\mathbf{x}_\mathrm{f}(n)$	$[x_\mathrm{f}(n), x_\mathrm{f}(n-1), \ldots, x_\mathrm{f}(n-L_w+1)]^T$
$\mathbf{x}_h(n)$	$[x(n), x(n-1), \ldots, x(n-L_h+1)]^T$
$\mathbf{y}(n)$	$[y(n), y(n-1), \ldots, y(n-L_h+1)]^T$
$\mathbf{x}_m(n)$	$[x(n-[m-1]Q), \ldots, x(n-mQ+1)]^T$
$\mathbf{x}_{m\mathrm{f}}(n)$	$[x_\mathrm{f}(n-[m-1]Q), \ldots x_\mathrm{f}(n-mQ+1)]^T$
$\mathbf{w}_m(n)$	$[w_{1+(m-1)Q}(n), \ldots, w_{mQ}(n)]^T$
$\mathbf{w}_{\mathrm{o},m}$	mth optimal inverse filter block

requiring some a priori knowledge of the sparseness about the channel response.

In this paper, we are interested in analyzing how different approaches derived from the combination of adaptive filters can be used in the AE application context. To this end, we proposed novel adaptive schemes that improve different aspects of previous algorithms and produce a good performance in terms of convergence speed and steady-state behavior, when the optimal solution can present different degrees of sparseness or the SNR is unknown.

This paper is organized as follows. The next section introduces two different strategies for AE based on the convex combination of filters: the convex combination of two filtered-x (CFx) adaptive filters and the convex combination of block-based filtered-x (CBFx) filters. Section III provides particular implementations of the general CBFx scheme. Firstly, in Subsection III-A, the biased block-based filtered-x (BBFx) scheme is presented. This algorithm allows to reduce the final excess mean-square error (EMSE) for sparse inverse responses, which usually occurs in equalization applications, or low SNRs conditions. To better understand the behavior of the BBFx scheme for AE, a theoretical analysis of its optimal steady-state performance is included in the appendix. Subsection III-B copes with utilization of the Fx-IPNLMS algorithm to update the adaptive filters of the CBFx scheme. In Subsection III-C, the partitioned block-based filtered-x (PBFx) algorithm and its combination scheme are also presented, whose novelty is that each block is independently modeled using different parameter settings. Simulation results that show the performance and advantages of the different schemes are given in Section IV, where the computational cost of the different proposed approaches is also detailed. Finally, the main conclusions of this work are summarized in Section V.

Throughout the paper, we denote vectors and matrices by boldface lower case and boldface capital letters, respectively, whereas (n) refers to the n-time sample. Notation in Table I will be used to describe the proposed schemes.

II. CONVEX COMBINATION OF BLOCK-BASED FILTERED-X ADAPTIVE FILTERS FOR AE

In this section, we start by presenting the CFx scheme for AE. Then, and as evolution of the convex combination of two adaptive filters, the CBFx scheme is proposed.
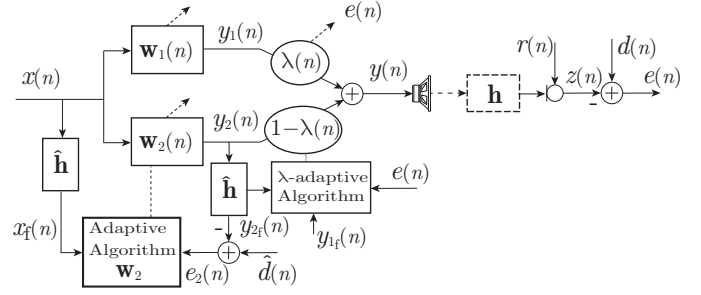

Fig. 3. Block diagram of the convex combination of filtered-x adaptive filters for AE (CFx scheme).

A. Convex combination of filtered-x adaptive filters

One of the simplest combination schemes is based on the combination of two parallel adaptive filters that work separately. Fig. 3 depicts the convex combination of two filtered-x filters (CFx) which could be employed to circumvent the compromise convergence speed vs final residual error. In a convex combination scheme, two adaptive filters are suitably combined in order to obtain the output of the parallel filter $y(n)$ as the weighted sum of the single outputs $y_1(n)$ and $y_2(n)$,

$$y(n) = \lambda(n)y_1(n) + \left[1 - \lambda(n)\right]y_2(n), \qquad (3)$$

being $\lambda(n) \in [0, 1]$ the combination factor. This parameter is adapted in order to minimize the power of the error of the overall filter, i.e., $J(n) = E\{e^2(n)\}$ [25], [32], following a gradient descent method. However, instead of updating directly $\lambda(n)$, we adapt an auxiliary mixing parameter $a(n)$ univocally related with $\lambda(n)$ via a modified sigmoid activation function:

$$\lambda(n) = \frac{\mathrm{sgm}[a(n)] - \mathrm{sgm}[-a^+]}{\mathrm{sgm}[a^+] - \mathrm{sgm}[-a^+]}, \qquad (4)$$

where $\mathrm{sgm}[a(n)] = \frac{1}{1+e^{[-a(n)]}}$. Following an NLMS adaptation rule with power normalization [40], the mixing parameter $a(n)$ is updated as[1]

$$
\begin{aligned}
a(n+1) = a(n) &+ \frac{\mu_a}{p(n)\{\mathrm{sgm}[a^+] - \mathrm{sgm}[-a^+]\}}e(n) \\
&\times \left[y_{1\mathrm{f}}(n) - y_{2\mathrm{f}}(n)\right]\mathrm{sgm}[a(n)]\left\{1 - \mathrm{sgm}[a(n)]\right\},
\end{aligned}
\qquad (5)
$$

where μ_a is a step size parameter and $y_{i\mathrm{f}}(n)$ ($i = 1, 2$ for each component adaptive filter) corresponds to the filter output signal $y_i(n)$ filtered through the estimated impulse response $\hat{\mathbf{h}}$. Moreover $p(n)$ is the normalization factor obtained as

$$p(n) = \beta p(n-1) + (1-\beta)\left[y_{1\mathrm{f}}(n) - y_{2\mathrm{f}}(n)\right]^2, \qquad (6)$$

being $0 \ll \beta < 1$ a forgetting factor. It should be remarked that, unlike other applications where combinations have been employed, in AE the output signal $y_i(n)$ (for $i = 1, 2$) of the adaptive filter $\mathbf{w}_i(n)$ has to be filtered through the estimated impulse response $\hat{\mathbf{h}}$ to obtain $y_{i\mathrm{f}}(n)$ following a filtered-x

---

[1]To avoid that the adaptation of $a(n)$ stops if $\mathrm{sgm}[a(n)]$ is close to 0 or 1, the range of values of $a(n)$ has to be restricted to $[a^+, -a^+]$, being $a^+ = 4$ a common choice [25].

TABLE II
CBFx STRUCTURE DESCRIPTION

$$y(n) = \sum_{m=1}^{M} \left\{ \lambda_m(n) y_{1,m}(n) + [1 - \lambda_m(n)] y_{2,m}(n) \right\}$$

$$\lambda_m(n) = \frac{\text{sgm}[a_m(n)] - \text{sgm}[-a^+]}{\text{sgm}[a^+] - \text{sgm}[-a^+]}$$

$$a_m(n+1) = a_m(n) + \frac{\mu_{a_m}}{p_m(n)\left\{\text{sgm}[a^+] - \text{sgm}[-a^+]\right\}} e(n)$$
$$\times \left[y_{1,mf}(n) - y_{2,mf}(n) \right] \text{sgm}[a_m(n)] \left\{ 1 - \text{sgm}[a_m(n)] \right\}$$

$$p_m(n) = \beta p_m(n-1) + (1 - \beta)\left[y_{1,mf}(n) - y_{2,mf}(n) \right]^2$$

$$e_i(n) = \hat{d}(n) - \sum_{m=1}^{M} y_{i,mf}(n) = \hat{d}(n) - y_{if}(n)$$

Fig. 4. Block diagram of a CBFx scheme, where for the sake of clarity, the adaptive algorithms $\mathbf{w}_{2,M}(n)$ and λ_M should be repeated with their respective inputs for each adaptive block combination.

fashion. Furthermore, the error signal $e_i(n)$ used to adapt each adaptive filter $\mathbf{w}_i(n)$, for $i = 1, 2$ is given by

$$e_i(n) = \hat{d}(n) - y_{if}(n), \tag{7}$$

where $\hat{d}(n)$ is estimated from $e(n)$ as follows

$$\hat{d}(n) = e(n) + y_f(n). \tag{8}$$

B. CBFx adaptive filters

In order to overcome the limitations of CFx scheme when the energy of the optimal inverse filter is not uniformly distributed, we introduce the CBFx strategy. This novel approach follows a block-based strategy combining each block of both component filters independently and where, furthermore, the different combination factors are computed to minimize the power of the overall error, $e(n)$. See the CBFx scheme depicted in Fig. 4, where for the sake of clarity, only the blocks required to update the second adaptive filter of block M ($\mathbf{w}_{2,M}(n)$) are shown. This structure should be repeated with their respective inputs for each adaptive block combination, $\mathbf{w}_{2,m}(n)$, where $i = 1, 2$ represents the adaptive filter component, and $m = 1, ..., M$ is the block index, being M the number of blocks.

In the particular case of AE, the structure of the inverse channel presents a nonuniform energy distribution, thus different degrees of sparseness can be found on different sections of the adaptive filter. With this in mind, and similarly to [32], one combination parameter is used for each group of coefficients in order to reduce the computational complexity rather than using a different parameter for each coefficient. The goal of this approach is to optimize the combination of each pair of blocks of coefficients instead of combining all the coefficients in order to minimize the quadratic error of the overall filter. Moreover, when used with a proper combination of algorithms it will combine their behavior for each block, so that the whole structure behaves at least as the best component filter [25].

The CBFx scheme can be considered as an extended version of the CFx in which each of the two adaptive filters is split into M non-overlapping blocks. Thus, the CBFx may be described using very similar expressions to those provided in Subsection II-A ((3)-(7)) as it is shown in Table II. The output of the combination filter $y(n)$ is obtained as the weighted sum of the single outputs $y_{1,m}(n)$ and $y_{2,m}(n)$ for $m = 1, \ldots, M$.
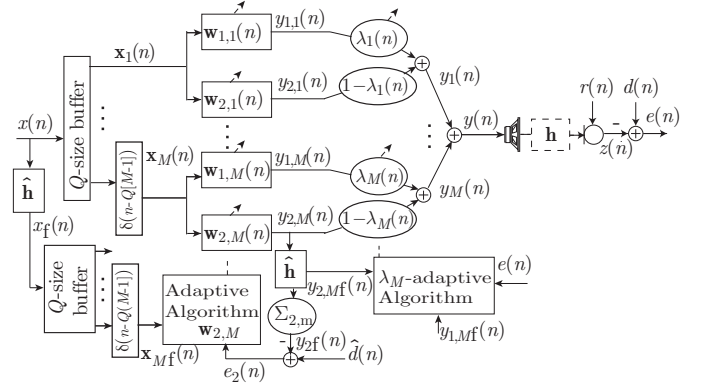
$\lambda_m(n) \in [0, 1]$ is the combination factor of the mth block, which is updated through the auxiliary mixing parameter $a_m(n)$ where μ_{a_m} is the mth step size parameter, and $p_m(n)$ the mth-block normalization factor. The error signal $e_i(n)$ used to adapt each adaptive block filter $\mathbf{w}_{i,m}(n)$, for $i = 1, 2$ and $m = 1, \ldots, M$, involves the computation of $y_{if}(n)$, that is represented in Fig. 4 by the $\sum_{i,m}$ term.

The proposed CBFx scheme allows to properly combine the different block filters to outperform the individual algorithms. The main drawback is that it implies a high computational cost, which significantly increases with the number of blocks due to the filtered-x structure. For that reason, a trade-off between number of blocks and computational burden should be sought.

Moreover, we have investigated the optimal inverse filters in the context of room equalization. See as an example the optimal inverse responses depicted in Fig. 7 (d-e-f), which have been computed by using the least squares error method (LSE) [3]. Given that they are designed with a proper modeling delay to satisfy causality and stability requirements, as a result and for different degrees of sparseness, the optimal filter consists in a central part that is almost always sparse with active coefficients while the remaining early and late parts are normally dispersive and with low energy coefficients. Thereby, the *a priori* knowledge of the inverse channel coefficient distribution can be exploited to improve the performance of block-based filters and unlike standard block-based combination schemes, we propose the design of the block filter location depending on the energy distribution of the adaptive filter. With this in mind and to provide an efficient block-based approach, the block scheme must be designed to locate the central coefficients in the same block filter. In Fig. 5 (a) these coefficients are located in a central block and in a second block that groups the remaining coefficients although they are not contiguous. Other distributions with different block sizes and even more blocks can be also considered. See, for example, Fig. 5 (b), where the central coefficients are located in block 3.

The simplest block-based filter consists in a filter with only two blocks (see Fig. 5 (a)). The first block is comprised of the L_c central coefficients and the second block that includes the remaining coefficients at both sides with a total length of L_l coefficients. Thus, the whole adaptive filter has a length of
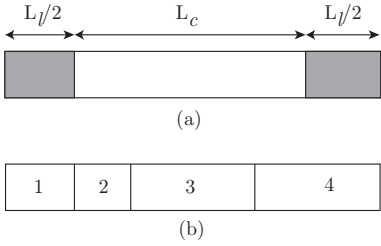
Fig. 5. Examples of coefficient distribution of a block-based algorithm: (a) two blocks and non-contiguous coefficients (plotted as white and shaded areas), (b) four blocks of different sizes.

$L_w = L_c + L_l$ taps.

It is important to note that although we have used a block strategy based on the energy distribution of the coefficients, the CBFx filter can be design with other considerations such as the sparseness degree of the block.

III. Particular implementations of the CBFx scheme for AE

In this section, different approaches are presented as particular implementations of the CBFx scheme for AE applications. Firstly, when one of the branches of the combination scheme is an all-zero filter [39], it leads to the BBFx structure that can provide an improved performance, reducing the final EMSE for low SNRs or for sparse inverse responses. Furthermore, the Fx-IPNLMS algorithm is proposed to update the adaptive filters of the presented schemes. This algorithm can achieve a good performance for inverse filters with different sparseness degrees using a proper configuration. Finally, the PBFx scheme is presented, where each block is modeled using different parameter settings (μ or κ values) based on the block filter characteristics. This structure could be combined using the proposed approach, providing the combination PBFx (CPBFx) scheme.

A. Biased filtered-x adaptive filter

The performance of adaptive filters can be severely deteriorated when the SNR is low. For sparse inverse filters in AE applications, the output of each block filter presents a different energy. Thus and even with high SNR conditions, the output signal level of some blocks can be lower than the system noise, that can be due to the gradient noise associated with the identification of these coefficients or to the ambient noise. This special feature of the BBFx scheme is showed in the theoretical analysis of the filter steady-state behavior included in the appendix.

As it has been said above, the BBFx scheme is derived from the CBFx structure by setting all the coefficients of one of the filters to zero and never adapting them [38]. Thus, only one branch of the block convex combination remains, allowing different scaling factors for the block filters. By doing so, the output of each block filter $\mathbf{w}_m(n)$ for $m = 1, \ldots, M$, is independently biased by the scaling factor $\alpha_m(n)$, which is defined as $\lambda_m(n)$ in Table II, where $a_m(n)$ is adapted similarly to that in Table II, being $y_{2,\mathrm{mf}}(n) = 0$.

B. Convex Combination of Fx-IPNLMS algorithms

The proposed CBFx scheme introduced in Section II can be implemented with different kinds of adaptive filters depending on both the application and performance requirements. With this in mind, this section focuses on efficiently increasing the robustness of the Fx-IPNLMS algorithm for AE applications by using the CBFx scheme. This fact implies to extend our previous contributions [23], [41] in several ways that depend on the performance requirements and scenario conditions (such as energy distribution and variability of the optimal solution, SNR values, etc.).

The Fx-IPNLMS assigns a different adaptation speed $\mu_l(n)$ to each coefficient according to,

$$w_l(n) = w_l(n-1) + \mu_l(n)e(n)x_{\mathrm{f}}(n-l), \qquad (9)$$

$$\mu_l(n) = \frac{\mu g_l(n-1)}{\delta + \sum_{k=0}^{L_w-1} g_k(n-1)x_{\mathrm{f}}^2(n-k)}, \qquad (10)$$

$$g_l(n-1) = (1-\kappa)\frac{1}{2L_w} + (1+\kappa)\frac{|w_l(n-1)|}{\varepsilon + 2\sum_k |w_k(n-1)|}, \quad (11)$$

for $l = 0, \ldots, L_w - 1$ and where $w_l(n)$ is the lth coefficient of the L_w-length vector $\mathbf{w}(n)$ and $x_{\mathrm{f}}(n)$ corresponds to the input signal $x(n)$ filtered through the L_h-length estimated LEM impulse response $\hat{\mathbf{h}}$. $g_l(n)$ is the adaptation gain factor of the lth filter coefficient, ε is a small value to avoid division by zero and $\kappa \in [-1, 1]$ arranges from the NLMS algorithm ($\kappa = -1$), especially suited for dispersive scenarios, to the PNLMS filter ($\kappa = 1$), optimally designed for sparse cases.

The adaptive scheme proposed will be called throughout the paper as CBFx-IP algorithm. The benefits of this approach are twofold. On the one hand, it allows to improve the robustness of the Fx-IPNLMS mainly in terms of convergence speed for different degrees of sparseness of the optimal filter when the adaptive components only differ in the value of κ. On the other hand, and as any other combination of gradient-base adaptive filters, it achieves a good trade-off between convergence speed and steady-state behavior when used with different step sizes.

The sequence of operations of the CBFx-IP scheme, where the computation of the M combination parameters and the update of the Q coefficients of the mth block of every component filter, is illustrated in **Algorithm 1**. For simplicity, we take $Q = L_w/M$ coefficients for each filter block.

Also in order to provide a particular implementation of the BBFx scheme for AE applications, in **Algorithm 2** we list the pseudocode for the BBFx-IP algorithm according to notation in Table I.

C. Combination of partitioned block schemes

Based on both the *a priori* knowledge of the coefficient energy distribution and the algorithm goal, each block of the PBFx filter can be modeled using a different parameter setting. For instance, the partitioned algorithm with two blocks, which was introduced in [42] for AEC, uses a different κ value for each block depending on the degree of sparseness of each block. This idea can be also exploited in an AE context, by

Algorithm 1 CBFx-IP algorithm.

Input: Reference signal $x(n)$, desired signal $d(n)$ and microphone signal $z(n)$

Output: Output of the parallel filter $y(n)$

1: Update the vectors $\mathbf{x}_m(n)$ (for $m = 1, \ldots, M$), $\mathbf{x}_h(n)$ and $\mathbf{y}(n)$
2: $x_{\mathrm{f}}(n) = \mathbf{x}_h^T(n)\widehat{\mathbf{h}}$
3: $y_{i,m}(n) = \mathbf{w}_{i,m}^T(n-1)\mathbf{x}_m(n)$, for $i = 1, 2$ and $m = 1, \ldots, M$
4: $y_{\mathrm{f}}(n) = \mathbf{y}^T(n)\widehat{\mathbf{h}}$
5: Update the vectors $\mathbf{x}_{m\mathrm{f}}(n)$, $\mathbf{y}_{1,m}(n)$ and $\mathbf{y}_{2,m}(n)$ for $m = 1, \ldots, M$
6: $y_{i,m\mathrm{f}}(n) = \mathbf{y}_{i,m}^T(n)\widehat{\mathbf{h}}$, for $i = 1, 2$ and $\quad m = 1, \ldots, M$
7: $e(n) = d(n) - z(n)$
8: $\hat{d}(n) = e(n) + y_{\mathrm{f}}(n)$
9: $e_i(n) = \hat{d}(n) - \displaystyle\sum_{m=1}^{M} y_{i,m\mathrm{f}}(n), \quad i = 1, 2$
10: $p_m(n) = \beta p_m(n-1) + (1-\beta)[y_{1,m\mathrm{f}}(n) - y_{2,m\mathrm{f}}(n)]^2$, for $m = 1, \ldots, M$
11: $a_m(n) = a_m(n-1) + \dfrac{\mu_{a_m}}{p_m(n)\left\{\mathrm{sgm}[a^+] - \mathrm{sgm}[-a^+]\right\}} e(n)$ $\times [y_{1,m\mathrm{f}}(n) - y_{2,m\mathrm{f}}(n)] \, \mathrm{sgm}[a_m(n)]\{1 - \mathrm{sgm}[a_m(n)]\}, \quad$ for $m = 1, \ldots, M$
12: $\lambda_m(n) = \dfrac{\mathrm{sgm}[a_m(n)] - \mathrm{sgm}[-a^+]}{\mathrm{sgm}[a^+] - \mathrm{sgm}[-a^+]}$, for $m = 1, \ldots, M$
13: $y(n) = \displaystyle\sum_{m=1}^{M} \left\{ \lambda_m(n) y_{1,m}(n) + [1 - \lambda_m(n)] y_{2,m}(n) \right\}$
14: $g_{i,l}(n-1) = (1 - \kappa_{i,m})\dfrac{1}{2L_w} + (1 + \kappa_{i,m})\dfrac{|w_{i,l}(n-1)|}{\varepsilon + 2\displaystyle\sum_k |w_{i,k}(n-1)|}$, for $l = 1, \ldots, L_w$, $i = 1, 2$ and $m = 1, \ldots, M$
15: $\mu_{i,m_q}(n) = \dfrac{\mu_i g_{i,m_q}(n-1)}{\delta + \displaystyle\sum_{k=0}^{L_w - 1} g_{i,k}(n-1)x_{\mathrm{f}}^2(n-k)}$, for $m = 1, \ldots, M$, $q = 1, \ldots, Q$ and $i = 1, 2$
16: $\mathbf{G}_{i,m}(n) = diag(\mu_{i,m_1}(n), \mu_{i,m_2}(n), \ldots, \mu_{i,m_Q}(n))$, for $i = 1, 2, \quad m = 1, \ldots, M$
17: $\mathbf{w}_{i,m}(n) = \mathbf{w}_{i,m}(n-1) + \mathbf{G}_{i,m}(n)\mathbf{x}_{m\mathrm{f}}(n)e_i(n)$, for $i = 1, 2$, $m = 1, \ldots, M$

Algorithm 2 BBFx-IP algorithm.

Input: Reference signal $x(n)$, desired signal $d(n)$ and microphone signal $z(n)$

Output: Output of the parallel filter $y(n)$

1: Update the vectors $\mathbf{x}_m(n)$ (for $m = 1, \ldots, M$), $\mathbf{x}_h(n)$ and $\mathbf{y}(n)$
2: $x_{\mathrm{f}}(n) = \mathbf{x}_h^T(n)\widehat{\mathbf{h}}$
3: $y_m(n) = \mathbf{w}_m^T(n-1)\mathbf{x}_m(n)$, for $m = 1, \ldots, M$
4: $y_{\mathrm{f}}(n) = \mathbf{y}^T(n)\widehat{\mathbf{h}}$
5: Update the vectors $\mathbf{x}_{m\mathrm{f}}(n)$, $\mathbf{y}_m(n)$ (for $m = 1, \ldots, M$)
6: $y_{m\mathrm{f}}(n) = \mathbf{y}_m^T(n)\widehat{\mathbf{h}}$, for $m = 1, \ldots, M$
7: $e(n) = d(n) - z(n)$
8: $\hat{d}(n) = e(n) + y_{\mathrm{f}}(n)$
9: $e_1(n) = \hat{d}(n) - \displaystyle\sum_{m=1}^{M} y_{m\mathrm{f}}(n)$
10: $p_m(n) = \beta p_m(n-1) + (1-\beta)y_{m\mathrm{f}}^2(n)$, for $m = 1, \ldots, M$
11: $a_m(n) = a_m(n-1) + \dfrac{\mu_{a_m}}{p_m(n)\left\{\mathrm{sgm}[a^+] - \mathrm{sgm}[-a^+]\right\}} e(n)$ $\times y_{m\mathrm{f}}(n)\mathrm{sgm}[a_m(n)]\{1 - \mathrm{sgm}[a_m(n)]\}$, for $m = 1, \ldots, M$
12: $\alpha_m(n) = \dfrac{\mathrm{sgm}[a_m(n)] - \mathrm{sgm}[-a^+]}{\mathrm{sgm}[a^+] - \mathrm{sgm}[-a^+]}$, for $m = 1, \ldots, M$
13: $y(n) = \displaystyle\sum_{m=1}^{M} \alpha_m(n)y_m(n)$
14: $g_l(n-1) = (1 - \kappa_m)\dfrac{1}{2L_w} + (1 + \kappa_m)\dfrac{|w_l(n-1)|}{\varepsilon + 2\displaystyle\sum_k |w_k(n-1)|}$, for $l = 1, \ldots, L_w$ and $m = 1, \ldots, M$
15: $\mu_{m_q}(n) = \dfrac{\mu g_{m_q}(n-1)}{\delta + \displaystyle\sum_{k=0}^{L_w - 1} g_k(n-1)x_{\mathrm{f}}^2(n-k)}$, for $m = 1, \ldots, M$ and $q = 1, \ldots, Q$
16: $\mathbf{G}_m = diag(\mu_{m_1}, \mu_{m_2}, \ldots, \mu_{m_Q})$, for $m = 1, \ldots, M$
17: $\mathbf{w}_m(n) = \mathbf{w}_m(n-1) + \mathbf{G}_m\mathbf{x}_{m\mathrm{f}}(n)e_1(n)$, for $m = 1, \ldots, M$

---

using the Fx-IPNLMS algorithm with a high κ value for the block with a high degree of sparseness and the Fx-NLMS ($\kappa = 1$) filter for the lateral block with a dispersive response.

In [42], for the partitioned scheme with 2 different κ values, to avoid problems during regularization of the PBFx-IP algorithm, a normalized factor has to be used to meet the constraint $\displaystyle\sum_{l=0}^{L_w - 1} g_l(n) \approx 1$. Results showed that a good convergence speed was achieved for dispersive and sparse responses with a normalized factor close to 0.5, whereas normalized factors close to 0 and 1 got a slower convergence speed, especially for dispersive filters. For AE applications, this has to be adapted in a similar way to the central and lateral blocks with the normalized factors $\dfrac{\|\mathbf{w}_c\|_1}{\|\mathbf{w}\|_1}$ and $\dfrac{\|\mathbf{w}_l\|_1}{\|\mathbf{w}\|_1}$ that multiply $\displaystyle\sum_{l_c=0}^{L_c - 1} g_{l_c}(n)$ and $\displaystyle\sum_{l_l=0}^{L_l - 1} g_{l_l}(n)$, respectively, where $\|\mathbf{w}_c\|_1$, $\|\mathbf{w}_l\|_1$ and $\|\mathbf{w}\|_1$ correspond to the ℓ_1-norm of the central, lateral and overall adaptive filters, respectively.

As it will be shown experimentally, this roughly *a priori* knowledge is not so accurate to determine both the best block sizes and κ values, which depend on the optimal inverse filter. For that reason, a combination of partitioned block (CPBFx) filters has been also proposed, giving rise to a robust scheme that exhibits generally a good performance.
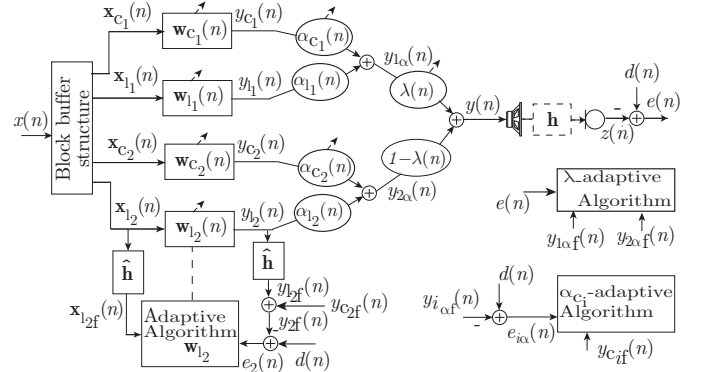

Fig. 6. Block diagram of a CPBFx scheme with 2 blocks.

Fig. 6 shows a CPBFx scheme where, in addition, each block filter has been biased. For simplicity, only two blocks are used in each algorithm, see Fig. 5 (a). As the central (subindex c) and lateral (subindex l) blocks of each adaptive filter have a different number of coefficients, the combination of both algorithms is performed after scaling the weights. For this reason, we consider different input vectors named $\mathbf{x}_{c_i}(n)$ and $\mathbf{x}_{l_i}(n)$ for $i = 1, 2$, with a different number of input signal samples.

IV. EXPERIMENTS

In this section, we present the experimental evaluation of the proposed schemes for room equalization. Moreover, the single Fx-IPNLMS [22] with different κ values will be used
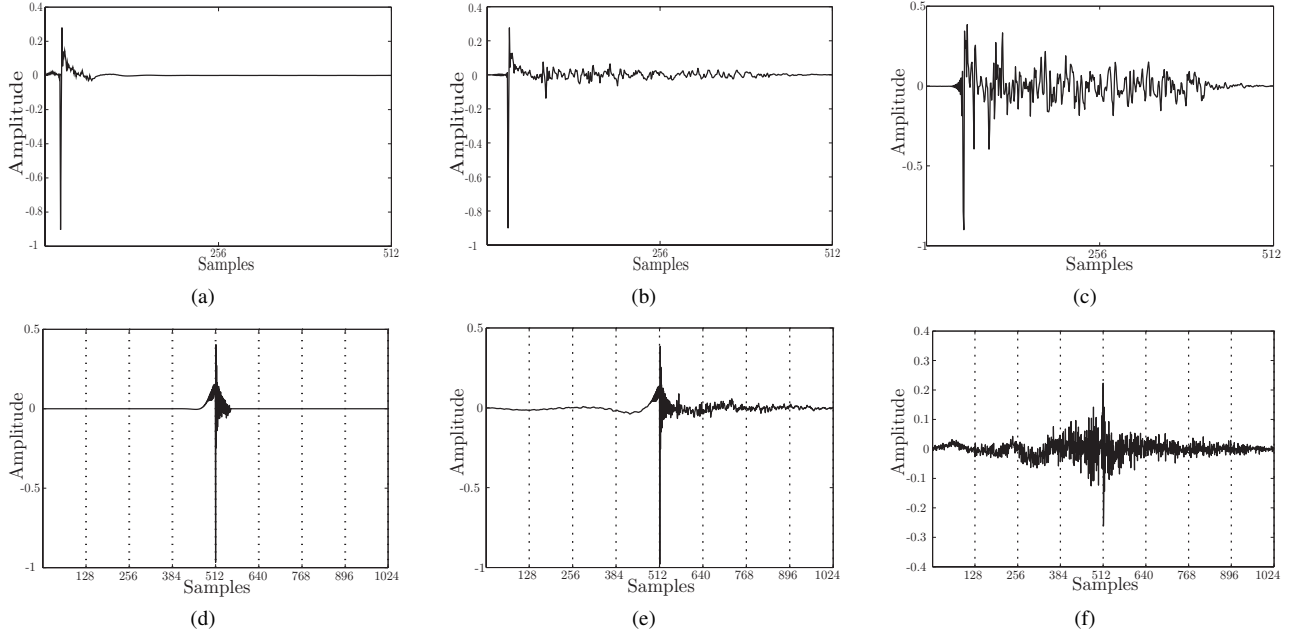
Fig. 7. Impulse responses of the considered acoustic channels (a, b and c) and their respective inverse filters (d, e and f). These responses present different degrees of sparseness. (a): $\xi = 0.89$, (b): $\xi = 0.70$, (c): $\xi = 0.46$, (d): $\xi = 0.86$, (e): $\xi = 0.30$ and (f): $\xi = 0.38$.

for comparison purposes. If another κ value is not specified, we will use $\kappa = -0.5$, as recommended in [22] to achieve a good behavior. Selection of $\kappa = -1$ will be specifically referred to as the Fx-NLMS algorithm.

The evaluation of the algorithms behavior is based on the figure of merit EMSE, $\text{EMSE}(n) = \text{E}\left\{[e(n) + r(n)]^2\right\}$, that has been estimated by averaging over 100 independent runs of the algorithms. It should be noted that as the adaptive filters are initialized to zero, the initial EMSE only depends on the input signal $x(n)$, which corresponds to a Gaussian noise with zero mean and unit variance, while the variance of $r(n)$ is adjusted to obtain different SNRs.

Room channel responses have been measured in a real listening room, with a reverberation time $T_{60} \cong 200$ ms and a sampling rate of $f_s = 8$ kHz. An omnidirectional Earthworks QTC microphone (frequency range 4 Hz - 40 kHz) and an Event PS6 model loudspeaker (frequency range from 45 Hz to 20 kHz) have been used, both with flat frequency responses. Responses of Figs. 7 (b-c) have been decimated to $f_s = 2$ kHz and 512 samples. In order to use a sparser response, the impulse response in Fig. 7 (a) has been artificially obtained from Fig. 7 (b) by taking the first 120 samples and zero-padding to length 512. Thus, three acoustic paths with different degrees of sparseness have been considered (Fig. 7 (a) $\xi = 0.89$, Fig. 7 (b) $\xi = 0.70$ and Fig. 7 (c) $\xi = 0.46$), where ξ is the sparseness measure based on the ℓ_1 and ℓ_2 norms as in [43]:

$$\xi(\mathbf{h}) = \frac{L_h}{L_h - \sqrt{L_h}} \left(1 - \frac{\|\mathbf{h}\|_1}{\sqrt{L_h}\|\mathbf{h}\|_2}\right). \tag{12}$$

The corresponding inverse filters have been modeled with FIR filters of 1024 samples. The length of the inverse filter has been chosen in order to allow a good equalization result without highly increasing the number of samples [3]. It should be noted that the sparseness of the inverse filters is lower

than that of their respective acoustic channels ($\xi = 0.86$, $\xi = 0.30$ and $\xi = 0.38$). The most dispersive impulse response is depicted in Fig. 7 (c) with $\xi = 0.46$, whereas the most dispersive inverse response is that of Fig. 7 (e) with $\xi = 0.30$, that corresponds to the measured acoustic channel of Fig. 7 (b). We assume that the channel paths $\widehat{\mathbf{h}}$ are perfectly estimated, but for Experiment G, where $\text{SNR}(\widehat{\mathbf{h}})$ is introduced[2].

A. Improving the Convergence Speed vs. Steady-state EMSE

For the first experiment, the CBFx scheme for $M = 1$ (see Fig. 4) achieves both a good convergence speed and a low steady-state EMSE. For this purpose, a combination of two filters with different step sizes has been used. The acoustic channel of Fig. 7 (a) has been used for this experiment, but any other acoustics system would show a similar behavior. The SNR has been chosen as 30 dB. Fig. 8 shows the EMSE performance (left axis) of the CBFx-IP algorithm (circle marker line) with different step sizes of $\mu_1 = 0.01$ and $\mu_2 = 0.1$, and of its Fx-IPNLMS component filters. On the one hand, $\mu = 0.1$ allows a fast convergence without getting unstable, meanwhile a lower μ value of 0.01 leads to a lower steady-state error. The step size of the combination factor μ_a has been set to 0.1 to allow a fast shift from one algorithm to the other. As it is expected, the combining parameter (right axis) changes from one to the other algorithm. As it can be seen, the combination outperforms both components around iteration $n = 5.8 \times 10^6$.

B. Improving the EMSE for nonuniform energy distribution in the optimal inverse filter

The inverse filter has normally nonuniform energy distribution among its coefficients. When an Fx-IPNLMS filter is

[2]$\text{SNR}(\widehat{\mathbf{h}})$ is defined as $\text{SNR}(\widehat{\mathbf{h}}) = 10 \log_{10} \frac{\|\mathbf{h}\|_2^2 / L_h}{\sigma^2}$, where $\widehat{\mathbf{h}} = \mathbf{h} + n_o$, being n_o a random vector with zero mean and σ^2 variance.
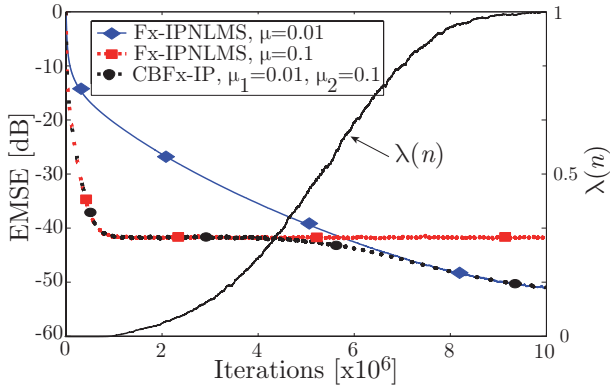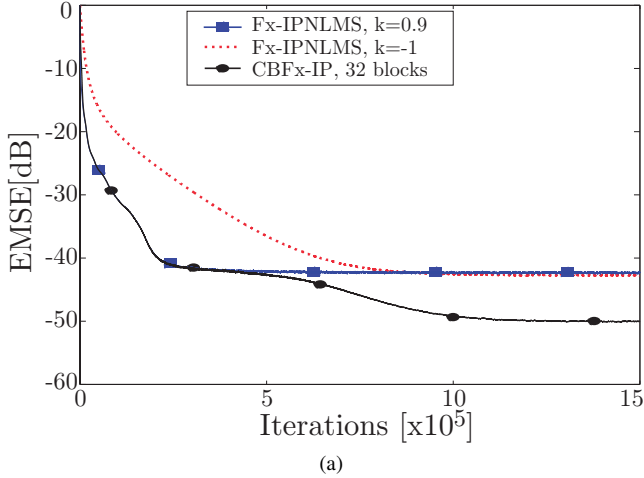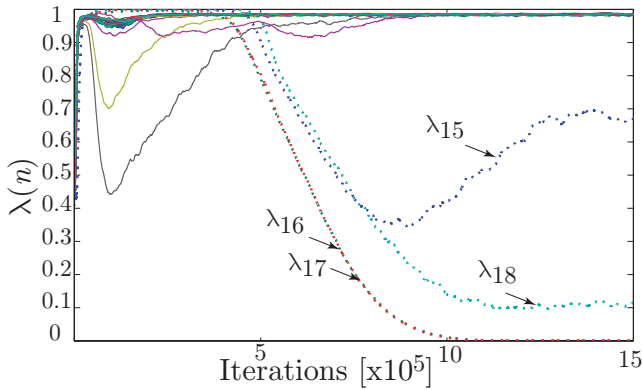
Fig. 8. EMSE(n) evolution (left axis) for the CBFx-IP (for $M = 1$) and their individual algorithms. The right axis shows the combining parameter $\lambda(n)$ for the CBFx-IP.

(a)

(b)

Fig. 9. 32-block CBFx-IP: (a) EMSE(n) of the combination and its component filters. (b) $\lambda_m(n)$ evolution.

employed to identify such inverse filter, the higher the coefficient energy, the larger the step size and the associated gradient noise. However, an Fx-NLMS filter spreads its gradient noise uniformly among all the coefficients. Therefore, we propose a CBFx-IP algorithm where the component filters differ in the κ value. Thus, those blocks with higher energy will behave as the Fx-IPNLMS algorithm with lower κ value, meanwhile blocks with low energy will follow the Fx-IPNLMS algorithm with a higher κ, resulting in a reduced EMSE [32].

The acoustic channel of Fig. 7 (a), whose inverse channel

shows a high nonuniform energy distribution, has been used with an SNR= 30 dB. We combine Fx-IPNLMS filter with $\kappa_1 = 0.9$ and Fx-NLMS algorithm, both split in 32 blocks of $Q = 32$ consecutive coefficients, where, for simplicity, the first block starts with the first coefficient. Both component filters have the same step size $\mu_1 = \mu_2 = 0.1$ and $\mu_{am} = 0.01$ to adapt the 32 mixing parameters. As can be seen in Fig. 9 (a), the combined filter behaves always as the best component filter and, in steady state, it reaches a better performance because each block combination converges to the block component with lower EMSE. Fig. 9 (b) shows the combination factors of the 32-block CBFx-IP algorithm, where a $\lambda_m(n)$ value close to 1 means a behavior similar to a block with $\kappa = 0.9$, whereas a value close to 0 means an Fx-NLMS behavior. In steady state, the central blocks (dotted line) with high energy exhibit a factor closer to the Fx-IPNLMS with $\kappa = -1$, meanwhile the blocks with lower energy behave mainly as the Fx-IPNLMS with $\kappa = 0.9$. Combination factors of blocks 15 and 18 get intermediate values as they have mixed energy values.

C. Improving the steady-state EMSE for unknown SNRs

In this subsection we illustrate how the BBFx-IP filter outperforms the standard Fx-IPNLMS algorithm, specially for low SNR. For this purpose, Fig. 10 shows a comparison between the simulation results (with circle markers) and the theoretical results obtained by means of the analysis in the appendix (with asterisk markers and dotted line). We compare the steady-state EMSE of the BBFx-IP filter with different M-block sizes (EMSE$_M$) with respect to that of the standard filtered-x algorithm, defining $\Delta\text{EMSE}(\infty) = \text{EMSE}(\infty) - \text{EMSE}_M(\infty)$. For inverse filters with a high degree of sparseness, the BBFx-IP adaptive filter allows to improve the steady-state EMSE value when increasing the number of blocks.

The results are represented for different SNRs and number of blocks M, with $\mu = 0.3$, $\mu_a = 0.01$ and $\kappa = -0.5$, considering the impulse response of Fig. 7 (a), whose corresponding inverse filter has a high number of coefficients close to zero [Fig. 7 (d)]. In this context, the BBFx scheme leads to a performance improvement because it allows to manage the variance vs bias tradeoff independently at each block. Thus, the gradient noise related with the identification of blocks with coefficients close to zero is removed, consequence of the bias introduced by scaling factors that are zero, as shown in Fig. 10 (b). Meanwhile, for lower SNRs the improvement is more evident since the gradient noise is inversely proportional to the SNR. Also, the block-based algorithm improves the EMSE results when increasing the number of blocks. However, when the block size is too small, as that for $M = 256$ and a block size of $Q = 4$, this improvement is limited to almost 4 dB for the represented SNR range. This is due to the gradient noise associated with the update of the scaling factors $\alpha_m(n)$, for $m = 1, \ldots, M$ [33].

As shown in Fig. 10 (a), a good performance is achieved with $M = 16$, which corresponds to $Q = 64$ samples. For this case, Fig. 10 (b) shows the 16 steady-state scaling factors considering different SNRs. For low SNRs, the factors of the adaptive blocks with low energy bias these outputs almost to
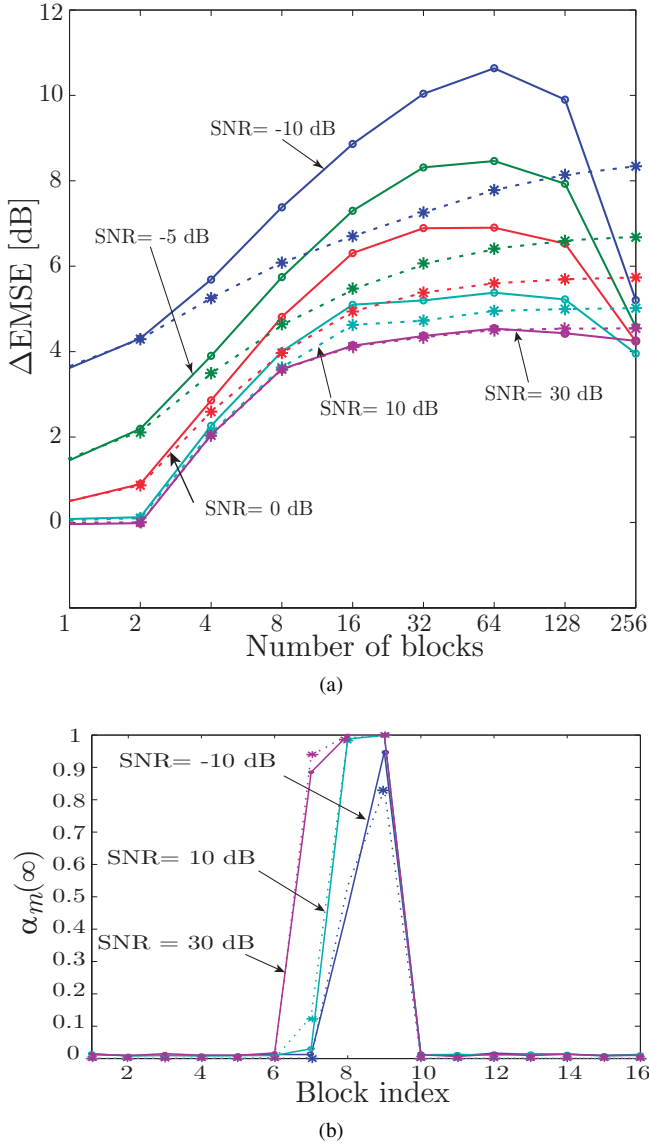
Fig. 10. (a): $\Delta \text{EMSE}(\infty) = \text{EMSE}(\infty) - \text{EMSE}_M(\infty)$ for different number of blocks (x-axis) and SNRs. (b) Scaling factors at steady state, $\alpha_m(\infty)$, for $M = 16$ and different SNRs. Solid and dotted lines represent experimental and theoretical results, respectively.

0. Meanwhile, the blocks with high energy (the central ones for room equalization) exhibit a lower bias. For higher SNRs, these central scaling factors converge to a higher value.

With regard to the influence of κ, lower values (such as, $\kappa = -1$ for the Fx-NLMS algorithm) allow a higher improvement of the block-based scheme. For that reason, if the steady-state performance of the algorithm is crucial, a block-based Fx-NLMS filter can be employed, improving the EMSE at steady state and reducing also the computational cost.

The theoretical and simulated values have a similar tendency, although they are not the same, specially when the number of blocks increases. The explanation might lie in the assumptions of the theoretical analysis and because in the appendix we are considering optimal scaling factors, and the gradient noise in the adaptation of $\alpha_m(n)$ can degrade the performance of the BBFx-IP scheme with small blocks.
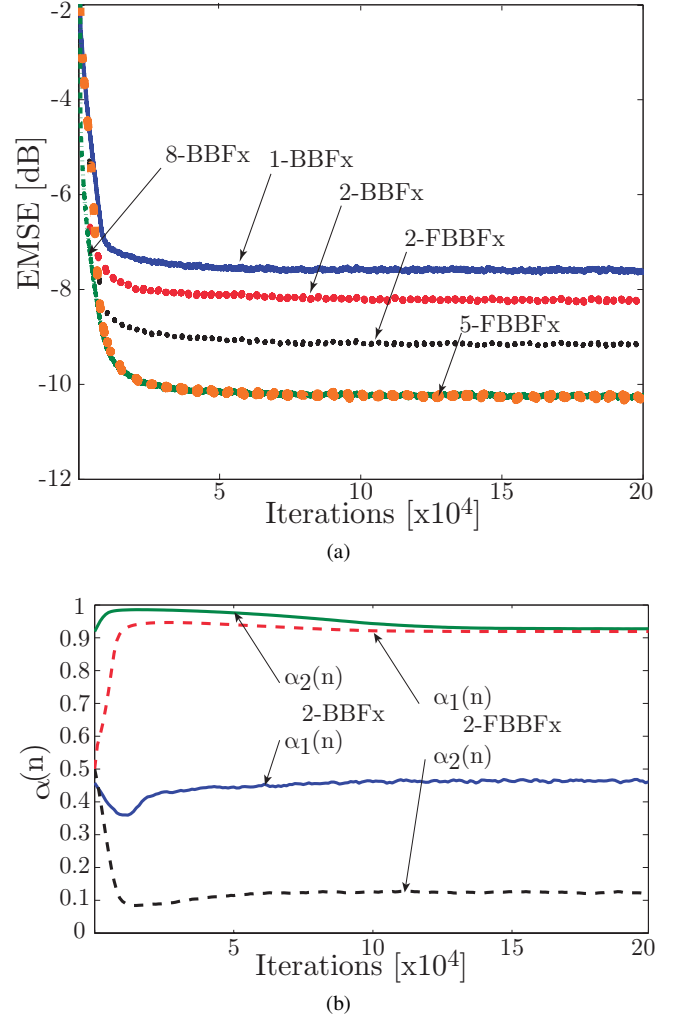

Fig. 11. (a): EMSE evolution of the 1, 2 and 8-BBFx-IPNMS algorithm and the 2 and 5-FBBFx-IP algorithm. (b): Scaling factors $\alpha_i(n)$ for the two-block schemes: 2-BBFx (solid lines) and 2-FBBFx (dashed lines).

D. Block-based filters for AE

As it has been showed in the previous experiments, the CBFx and the BBFx schemes improve their behavior as the number of blocks increases, but also causing the computational cost to increase (see III). It should be remarked that unlike other applications where an increment of number of blocks leads to a negligible increment in computational cost [33], in AE the number of blocks must remain modest, in order to keep the computational cost feasible.

However, in an equalization context is possible to keep the computational cost modest employing a block-based scheme where the block filter coefficients are designed depending on the energy distribution of the expected inverse filter. For instance, Fig. 11 illustrates the EMSE evolution of the BBFx-IP algorithm with a central block of 512 coefficients and another block with the 512 remaining taps, denoted as 2-FBBFx (fitted block scheme), considering $\mu = 0.3$ and $\kappa = -0.5$. The performance of this filter has been compared with the BBFx-IP filter with $M = 1, 2$ and 8 blocks, denoted as M-BBFx, where the coefficients are consecutively selected. Results in Fig. 11 (a) show that the 2-FBBFx-IP algorithm (black dotted line) outperforms the 2-BBFx-IP (red dotted

line), although it does not achieve the stationary EMSE of the 8-BBFx-IP algorithm (green solid line). However, a 5-FBBFx-IP (with 205 coefficients for each filter block) falls close to it (orange dotted line). For this experiment SNR= 0 dB has been used, although higher SNR values obtain a similar behavior provided that the scaling factors are not 1. However, lower SNRs lead to more significant differences.

Fig. 11 (b) shows the scaling parameters of two approaches with same number of blocks, and hence, same computational costs: 2-BBFx-IP and 2-FBBFx-IP. As it can be seen, the block design of the 2-FBBFx-IP algorithm permits a more efficient bias, giving rise to very different value of its scaling factors: $\alpha_1(n) \simeq 0.9$ for the central coefficients with higher energy, and $\alpha_2(n) \simeq 0.1$ for the lateral coefficients with low energy. This allows to obtain a reduced EMSE with respect to that of the 2-BBFx-IP algorithm.

E. Performance evolution of the PBFx scheme for different sparseness degrees

Next, we will like to study the influence of the central-block length on the PBFx-IP performance. Thus, as an example, the notation PBFx-IP$_{256}$N$_{768}$ refers to a central block with an IPNLMS with $\kappa = -0.5$ and $L_c = 256$ samples, and a lateral block with an NLMS (an IPNLMS with $\kappa = -1$) and $L_l = 768$ samples. To meet the constraint $\sum_{l=1}^{L_w} g_l(n) \approx 1$, the terms $\sum_{l_c=1}^{L_c} g_{l_c}(n)$ and $\sum_{l_l=1}^{L_l} g_{l_l}(n)$ have been multiplied by 0.5.

Different block sizes have been considered, $L_c = L_w/2 = 512$, $L_c = 256$, $L_c = 64$ and $L_c = 960$ samples, with $\mu_a = 0.01$ and SNR= 15dB. For comparison purposes, we have also studied the behavior of the Fx-NLMS and Fx-IPNLMS algorithms.

Fig. 12 (a) illustrates the EMSE evolution considering $\mu = 0.1$ and the acoustic channel of Fig. 7 (c), whose inverse filter of Fig. 7 (f) shows a sparseness degree of $\xi = 0.38$. The fastest convergence corresponds to the normalized factors $\frac{\|\mathbf{w}_c\|_1}{\|\mathbf{w}\|_1}$ and $\frac{\|\mathbf{w}_l\|_1}{\|\mathbf{w}\|_1}$ close to 0.5 for the PBFx-IP$_{256}$N$_{768}$ and PBFx-N$_{256}$IP$_{768}$ schemes. Whereas a narrower or wider block size implies a slower convergence, as the normalized factors are closer to 0 or 1.

Fig. 12 (b) represents the EMSE evolution for the acoustic response of Fig. 7 (a), with $\mu = 0.3$, whose inverse filter [Fig. 7 (d)] has a sparseness degree of $\xi = 0.86$. For this inverse channel with many coefficients close to zero, the normalized factors near 0.5 correspond to $L_c = 16$, thus the PBFx-N$_{16}$IP$_{1008}$ algorithm exhibits a fast convergence. On the other hand, the PBFx-IP$_{16}$N$_{1008}$ is not so fast, as the Fx-NLMS algorithm uniformly distributes the 0.5 normalized value between a high number of coefficients (1008).

For the steady-state error in Fig. 12 (b), as the coefficients with high energy are included in a narrow block, the lower steady state is obtained when an Fx-NLMS algorithm that contains all these taps for the central block is used, for instance, $L_c = 256$ or 512 samples.

To summarize, the PBFx-IP filter can exhibit a convergence speed faster than the Fx-IPNLMS, when the normalized factors $\frac{\|\mathbf{w}_c\|_1}{\|\mathbf{w}\|_1}$ and $\frac{\|\mathbf{w}_l\|_1}{\|\mathbf{w}\|_1}$ are close to 0.5. Moreover, to reduce the
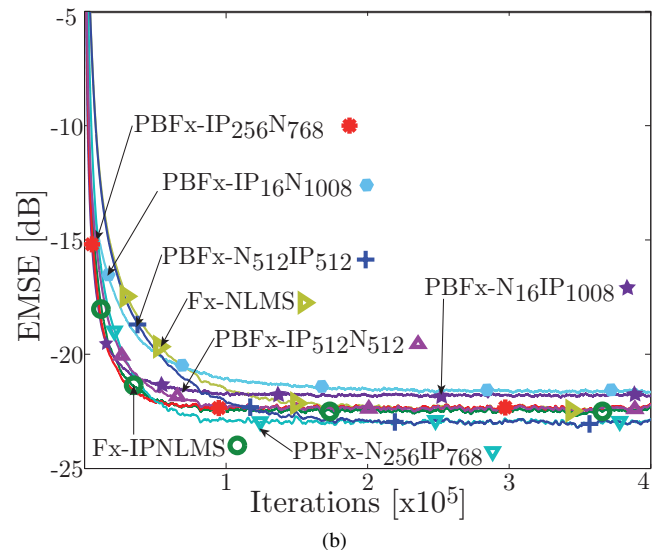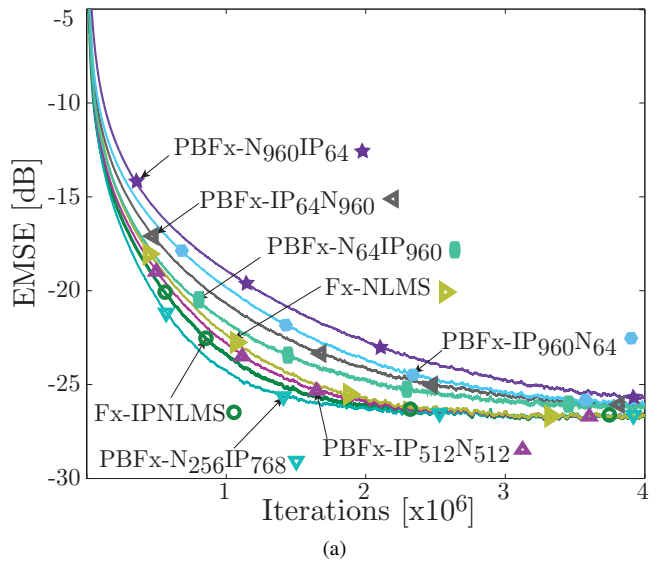

Fig. 12. EMSE evolution for the PBFx scheme with different block sizes and κ values. PBFx-IP$_i$N$_j$ refers to the partitioned fitted-block algorithm with a central block with $\kappa = -0.5$ and $L_c = i$ samples, and a lateral block with $\kappa = -1$ and $L_l = j$ samples. Also the Fx-NLMS and Fx-IPNLMS are shown as reference.(a): for the acoustic channel of Fig. 7 (c). (b): for the acoustic channel of Fig. 7 (a).

EMSE in steady state, the suitable setup is a central block of $L_w/4$ or $Lw/2$ with the Fx-NLMS algorithm, which includes the coefficients that have a higher energy, and the Fx-IPNLMS algorithm for the lateral block (with coefficients with lower values).

F. Convex combination of partitioned block schemes

From the previous experiment, we exploit the convex combination scheme to improve the speed of convergence versus the steady-state performance. To do so and for a general scenario, we combine the PBFx-IP$_{256}$N$_{768}$ and the PBFx-N$_{512}$IP$_{512}$ algorithms, CPBFx-IP$_{256}$N$_{768}$-N$_{512}$IP$_{512}$.

Fig. 13 shows the EMSE evolution for the acoustic channel of Fig. 7 (a), which corresponds to the same filter used in the previous experiment in Fig. 12 (b), but removing some schemes for clarity. The CPBFx-IP$_{256}$N$_{768}$-N$_{512}$IP$_{512}$ exhibits a fast convergence speed and the lowest steady state.
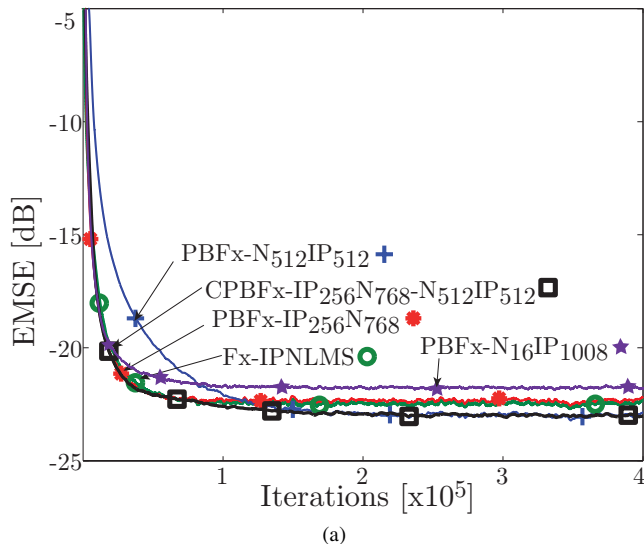
Fig. 13. EMSE evolution for the CPBFx algorithm for the acoustic channel of Fig. 7 (a) .
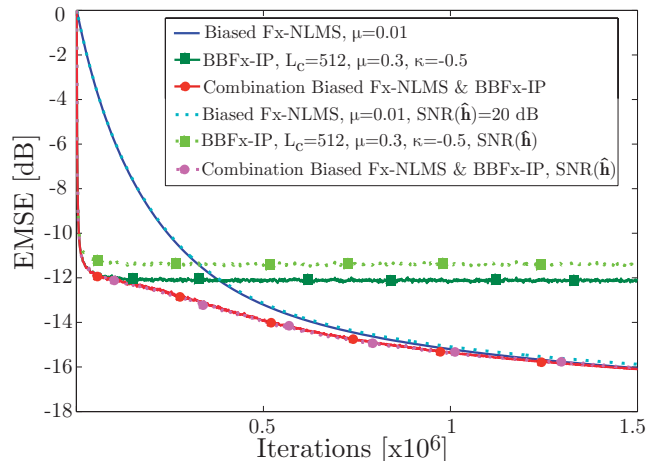

Fig. 14. Combination of a BBFx-IP algorithm with a biased Fx-NLMS algorithm with different μ values (in solid lines) and with path estimation errors (in dotted lines).

G. Convex Combination of BBFx-IP and biased Fx-NLMS Algorithms

The combination can be designed employing any kind of algorithms, even differing in the tuning of several parameters simultaneously. In this experiment, we pursuit to obtain a combination with very fast convergence and very low steady-state error. Then, we combine a 2-block BBFx-IP algorithm with $\mu_1 = \mu_2 = 0.3$, $\kappa_1 = \kappa_2 = -0.5$ with a central-block of 512 coefficients (that exhibits a very fast convergence) with a biased Fx-NLMS algorithm with $\mu = 0.01$, that reaches a very low steady-state EMSE with reduced computational cost. As it can be seen in Fig. 14, where the acoustic channel of Fig. 7 (b) has been considered, the combination achieves both good convergence speed and low steady-state error, even outperforming the individual components in some iterations. It should be noted that the role of the biased version of the Fx-NLMS algorithm is to prevent low SNR situations (for this experiment, SNR= 10 dB). However, if the computational burden is not a constraint, a better performance at steady state

can be achieved using a block-based strategy for the Fx-NLMS algorithm, or even a partitioned block scheme.

The same adaptive algorithms have been considered with a non-perfect estimate of the acoustic path being $\mathrm{SNR}(\hat{\mathbf{h}}) = 20$ dB (in dotted lines), which allows to analyze robustness of the adaptive filters in this scenario. The difference of this imperfect estimation is shown on the BBFx-IP algorithms. As expected, the combination filter combines the performance of the component filters and outperforms them at some points.

H. Computational Cost

In this subsection, we analyze the computational complexity of the proposed algorithms in terms of multiplications per iteration, which is given in Table III. Regarding the Fx-IPNLMS filter, it requires $6L_w+2$ multiplications more than the simplest of the algorithms considered in this paper, the Fx-NLMS. In the case of the combination schemes, the increase in the number of blocks can provide a better performance, but with a larger computational cost. Specifically, the computation of the combination factors involves $(2L_h+11)M$ multiplications more than the single algorithm, including also the additional filtering through the estimated channel and $(L_h + 10)M$ multiplications for the biased schemes.

The computational cost of the CPBFx-IP algorithm of Fig. 6 is computed as explained in Section III-C with an Fx-IPNLMS algorithm for the central block and an Fx-NLMS algorithm for the lateral blocks. In this case, the computational burden of the adaptive filter is of $L_h + 2(2L_w + 6L_c + 5)$ multiplications.

The computational complexity of the different schemes considered is summarized in Table III, where also an example has been given at the last column for $L_h = 512$, $L_w = 1024$ and $M = 16$. For the CPBFx scheme with only 2 blocks, a central block of 512 coefficients has been considered for the typical case.

V. CONCLUSION

In this contribution, we propose the utilization of combination schemes for room equalization applications. In this context, and due to the nonuniform distribution of the inverse optimal filter, the blockwise combination inherent to the CBFx algorithm allows to reach an improved performance with respect to that of combinations of whole adaptive filters. We can obtain additional advantages using particular implementations, as the BBFx scheme, which presents a good performance, specially for low SNRs and/or with sparse optimal filters.

Although combination schemes can be designed employing any kind of adaptive filter as components, we have based our proposal in the Fx-IPNLMS algorithm, which allows to design combination filters that efficiently alleviate the compromise related with the sparseness degree of the inverse filter.

An important difference between the proposed schemes and other blockwise combination schemes employed in other applications is that, in the room equalization context, the number of blocks should remain modest, in order to keep a reasonable computational cost. However, we propose different schemes that fulfill with this premise presenting a good performance. For instance, an improved behavior is obtained if the blocks are designed considering the standard shape of the inverse filter

TABLE III
SUMMARY OF THE COMPUTATIONAL COMPLEXITY OF THE PROPOSED SCHEMES. COMPLEXITY IS MEASURED AS THE NUMBER OF MULTIPLICATIONS PER ITERATION OF THE CONSIDERED ALGORITHMS. TYPICAL CASE: $L_h = 512$, $L_w = 1024$, $M = 16$.

Algorithm	Multiplications	Typical case
Fx-NLMS // Fx-IPNLMS	$\mathbf{L_h + 3L_w + 2}$ // $\mathbf{L_h + 9L_w + 4}$	3586 // 9732
CBFx-N // CBFx-IP ($M = 1$)	$\mathbf{3L_h + 5L_w + 15}$ // $\mathbf{3L_h + 17L_w + 19}$	6671 // 18963
CBFx-N // CBFx-IP	$\mathbf{L_h(2M + 1) + 5L_w + 11M + 4}$ // $\mathbf{L_h(2M + 1) + 17L_w + 11M + 8}$	22196 // 34488
BBFx-N // BBFx-IP	$\mathbf{L_h(M + 1) + 3L_w + 10M + 2}$ // $\mathbf{L_h(M + 1) + 9L_w + 10M + 4}$	11938 // 18084
CPBFx-IP ($M = 2$)	$\mathbf{L_h + 2(2L_w + 6L_c + 5)}$	10762

and the central coefficients are located in the same block. In addition, we present the PBFx scheme for AE applications, where each block is modeled using different parameter settings (μ or κ values) based on the block filter characteristics.

Simulation results show the benefits of the proposed algorithms for room equalization applications. The use of combination schemes in this context permits to obtain a robust behavior for different sparseness degrees and different SNR conditions, allowing to efficiently alleviate several trade-offs inherent to the adaptive filters, highlighting the compromise between convergence speed and steady-state error.

Future work includes its extension to the multichannel case, and to study the performance of other kind of adaptive combinations.

APPENDIX
STEADY-STATE ANALYSIS OF THE BBFx-IP ALGORITHM: EMSE PERFORMANCE AND OPTIMAL SCALING FACTORS

The BBFx-IP algorithm provides an improvement with respect to the Fx-IPNLMS algorithm for low SNRs but also when the optimal adaptive filter has a sparse response, which normally occurs in AE applications. To better understand the behavior of the BBFx scheme, this appendix presents the theoretical analysis of its steady state. The following development can be straightforwardly extended to the CBFx scheme. Nevertheless, due to the limited space, it has not been included in this contribution.

In order to compute the EMSE of the BBFx-IP algorithm introduced in Section III-A, the output error in **Algorithm 2** can be expressed as a function of the *a priori* error $e_a(n)$.

For the sake of simplicity, we consider each coefficient separately and constant scaling factors. Thus, we use the filter coefficients $w_l(n)$ for $l = 1, \ldots, L_w$, and the scaling factors that multiplies each block, α_m for $m = 1, \ldots, M$ can be rewritten as $\alpha_{\lceil \frac{l}{Q} \rceil}$. That means, m is assigned to each filter coefficient $w_l(n)$ depending on the filter block it belongs to, rounding up the division i/Q towards the nearest integer. Moreover, for low variations of the weight vector, $\mathbf{h}$ and $\mathbf{w}^T(n)$ are considered linear systems that can be interchanged (associative law of linear time-invariant systems, [44]). Thus, the microphone signal $z(n)$ and the error signal $e(n)$ have been rewritten respectively as

$$z(n) = \sum_{l=1}^{L_w} \alpha_{\lceil \frac{l}{Q} \rceil} w_l^T(n) x_f(n - l) + r(n). \quad (13)$$

$$e(n) = d(n) - z(n) = \sum_{l=1}^{L_w} \left[w_{o,l} - \alpha_{\lceil \frac{l}{Q} \rceil}(n) w_l(n) \right]^T x_f(n - l) - r(n)$$

$$= \Psi^T(n) \mathbf{x}_f(n) - r(n) = \Psi^T(n) \widehat{\mathbf{H}}^T \mathbf{x}_{wh}(n) - r(n) = e_a(n) - r(n), \quad (14)$$

being $\Psi(n)$ a vector containing the $\psi_l(n)$ terms for $l = 1, \ldots, L_w$, $\Psi(n) = [\psi_1(n), \ldots, \psi_{L_w}(n)]^T$, where $\psi_l(n) = w_{o,l} - \alpha_{\lceil \frac{l}{Q} \rceil} w_l(n)$ has been used for compactness. $\mathbf{x}_{wh}(n)$ is a column vector containing the last $L_w + L_h - 1$ samples of the input signal $x(n)$ that multiplies the convolution matrix $\widehat{\mathbf{H}}^T$ of size $L_w \times L_w + L_h - 1$ to provide the filter vector $\mathbf{x}_f(n)$, being

$$\widehat{\mathbf{H}}^T = \begin{bmatrix} \hat{h}(0) & \hat{h}(1) & \ldots & \hat{h}(L_h - 1) & \ldots & 0 \\ 0 & \hat{h}(0) & h(1) & \ldots & \ldots & 0 \\ & & \ddots & \ddots & \ddots & \\ 0 & 0 & 0 & \hat{h}(0) & \ldots & \hat{h}(L_h - 1) \end{bmatrix}. \quad (15)$$

Following the analysis in [39], the EMSE of the filter is therefore given by

$$J_{ex}(n) = E\{e_a^2(n)\} = E\left\{ \Psi^T(n)^T \widehat{\mathbf{H}}^T \mathbf{x}_{wh}(n) \mathbf{x}_{wh}^T(n) \widehat{\mathbf{H}} \Psi(n) \right\}$$

$$\approx E\left\{ \Psi^T(n) \widehat{\mathbf{H}}^T E\left\{ \mathbf{x}_{wh}(n) \mathbf{x}_{wh}^T(n) \right\} \widehat{\mathbf{H}} \Psi(n) \right\}, \quad (16)$$

where $E\{\cdot\}$ denotes the mathematical expectation.

Assuming the step size is small enough to allow the filter coefficients to follow the average statistics of the input signal (see the Direct Averaging Method applied in [5]), the term $\mathbf{x}_{wh}(n) \mathbf{x}_{wh}^T(n)$ can be approximated by its expected value. If we consider the statistics of the input signal, for a white noise signal of zero mean and σ_x^2 variance, the following term in (16) can be simplified, $E\left\{ \mathbf{x}_{wh}(n) \mathbf{x}_{wh}^T(n) \right\} = \sigma_x^2 \mathbf{I}_{(L_w + L_h - 1)}$, where $\mathbf{I}_{(L_w + L_h - 1)}$ is the identity matrix of size $(L_w + L_h - 1)$. Then, (16) can be rewritten as

$$J_{ex}(n) = \sigma_x^2 E\left\{ \Psi^T(n) \widehat{\mathbf{H}}^T \widehat{\mathbf{H}} \Psi(n) \right\}. \quad (17)$$

After some manipulations and using the symmetrical property of the correlation matrix, (17) can be rewritten as,

$$J_{ex}(n) = \sigma_x^2 E\left[\Psi^T(n) \Phi(n) \right] \mathbf{D} \widehat{\mathbf{H}}_{L_h}^T \widehat{\mathbf{h}}, \quad (18)$$

where $\mathbf{D}$ is a diagonal matrix of L_h length, whose first entry is equal to 1, and the rest of the elements of the main diagonal are 2 to implement the symmetric factor. Furthermore, $\widehat{\mathbf{H}}_{L_h}^T \widehat{\mathbf{h}}$ implements the correlation of the channel response, where $\widehat{\mathbf{H}}_{L_h}^T$ is a matrix composed with the first L_h columns of $\widehat{\mathbf{H}}^T$. Moreover, matrix $\Phi(n)$ of size $L_w \times L_h$ is a convolution matrix built with the $\psi_l(n)$ coefficients, whose first column is $\Psi(n)$ and the following columns are built by shifting down this vector and zero-padding at the beginning, i.e.,

$$\Psi^T(n) \Phi(n) = \begin{bmatrix} \psi_1(n) & \psi_2(n) & \ldots & \psi_{L_w}(n) \end{bmatrix}$$

$$\cdot \begin{bmatrix} \psi_1(n) & 0 & \ldots & 0 \\ \psi_2(n) & \psi_1(n) & \ldots & \vdots \\ \vdots & \vdots & \ddots & \psi_1(n) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{L_w}(n) & \psi_{L_w-1}(n) & \ldots & \psi_{L_w-L_h+1}(n) \end{bmatrix}. \quad (19)$$

For $\mathbf{h} = \delta(n - \tau)$ (acoustic propagation in free field conditions), $\widehat{\mathbf{H}}^T\widehat{\mathbf{h}} = [1, 0, \dots, 0]^T$, and the multiplication by matrix $\mathbf{D}$ gives also the L_h length vector $[1, 0, \dots, 0]^T$. This vector activates only the first column of $\Phi(n)$ in (19), and thus expression (20) reduces to the one obtained in [39], when the filtered-x structure is not considered.

$$J_{ex}(n) = \sigma_x^2 E\left[\Psi^T(n)\Psi(n)\right]. \quad (20)$$

Expression (19) shows the influence of the channel filtering among the $\Psi_l(n)$ coefficients. Thus, (18) can be rewritten as

$$J_{ex}(n) = \sigma_x^2\|\widehat{\mathbf{h}}\|_2^2 E\left[\Psi^T(n)\Psi(n)\right]$$
$$+2\sigma_x^2\sum_{\tau=1}^{L_h-1} r(\tau)\sum_{l=1}^{L_w-\tau} E\left[\psi_l(n)\psi_{l+\tau}(n)\right], \quad (21)$$

where $r(\tau) = \sum_{k=\tau}^{L_h-1}\widehat{h}(k)\widehat{h}(k-\tau)$. For $\tau = 0$, $r(0) = \|\widehat{\mathbf{h}}\|_2^2$ which appears as the first term in (21).

In steady state ($n \longrightarrow \infty$), the EMSE of the block-biased filtered-x scheme can be developed as in [39]. After some manipulations, using the coefficient weight error $\widetilde{w}_l(n)$, defined as $\widetilde{w}_l(n) = w_{o,l} - w_l(n)$, in $\psi_l(n) = w_{o,l} - \alpha_m w_l(n) = \alpha_m\widetilde{w}_l(n) + [1 - \alpha_m]w_{o,l}$ and assuming independence of the filter coefficients and that they tend to their optimal solution, $E\{\widetilde{w}_l(n)\widetilde{w}_{l+\tau}(n)\} = E\{\widetilde{w}_l(n)\}E\{\widetilde{w}_{l+\tau}(n)\} = 0$, where $E\{w_l(\infty)\} = w_{o,l}$, we get

$$J_{ex}(\infty) = \sigma_x^2\|\widehat{\mathbf{h}}\|_2^2\left[\sum_{m=1}^{M}\alpha_m^2 E\left\{\|\widetilde{\mathbf{w}}_m(\infty)\|_2^2\right\}\right.$$
$$+\sum_{m=1}^{M}\left(1 - \alpha_m\right)^2\|\mathbf{w}_{o,m}\|_2^2\right] + 2\sigma_x^2\sum_{\tau=1}^{L_h-1} r(\tau)$$
$$\left[\sum_{l=1}^{L_w-\tau}\left(1 - \alpha_{\lceil\frac{l}{Q}\rceil}\right)\left(1 - \alpha_{\lceil\frac{l+\tau}{Q}\rceil}\right) w_{o,l}w_{o,l+\tau}\right], \quad (22)$$

where the first term is similar to that of (12) in [39] but including the effect of the channel energy $\|\widehat{\mathbf{h}}\|_2^2$, and there is an additional second term dependent on the correlation of the channel response $r(\tau)$ for $\tau = 1, ..., L_h - 1$.

If only one block is considered ($\alpha_m = \alpha$) as in [41], the coefficients in (14) can be rewritten as $\psi_l(n) = \alpha\widetilde{w}_l(n) + [1 - \alpha]w_{o,l}$. Leading to,

$$J_{ex}(\infty) = \alpha^2\sigma_x^2\|\widehat{\mathbf{h}}\|_2^2\sum_{l=1}^{L_w} E\left\{\|\widetilde{w}_l(\infty)\|_2^2\right\}$$
$$+\sigma_x^2\left(1 - \alpha\right)^2\mathbf{w}_o^T\widehat{\mathbf{H}}^T\widehat{\mathbf{H}}\mathbf{w}_o. \quad (23)$$

In order to compute the EMSE in steady state, we have to obtain the terms $E\left\{\|\widetilde{\mathbf{w}}_m(\infty)\|_2^2\right\}$ and α_m. For simplicity, we assume that the channel correlation terms $r(\tau)$ for $\tau = 1, \dots, L_h - 1$ can be discarded respect to the $r(0) = \|\widehat{\mathbf{h}}\|_2^2$ coefficient.

Following the development as in [32], the weight error coefficients can be obtained as,

$$E\left\{\|\widetilde{w}_l(\infty)\|_2^2\right\} = \frac{\mu\sigma_r^2}{\sigma_x^2\|\widehat{\mathbf{h}}\|_2^2}\frac{g_l(\infty)}{2 - \mu g_l(\infty)}, \quad l = 1, \dots, L_w \quad (24)$$

$$g_l(\infty) = (1 - \kappa)\frac{1}{2L_w} + (1 + \kappa)\frac{|w_{o,l}|}{\varepsilon + 2\sum_k|w_{o,k}|}, \quad l = 1, \dots, L_w \quad (25)$$

Finally,

$$E\left\{\|\widetilde{\mathbf{w}}_m(\infty)\|_2^2\right\} = \sum_{l=1}^{Q} E\left\{\|\widetilde{w}_{(m-1)Q+l}(\infty)\|_2^2\right\}. \quad (26)$$

In steady state, if the scaling factors converge to the optimal solution α_m^*, we can obtain this solution from (22) with $\frac{\partial J_{ex}(\infty)}{\partial\alpha_m} = 0$. The scaling terms can be obtained as

$$\alpha_m^* = \frac{1}{1 + \frac{E\left\{\|\widetilde{\mathbf{w}}_m(\infty)\|_2^2\right\}}{\|\mathbf{w}_{o,m}\|_2^2}}. \quad (27)$$

REFERENCES

[1] L. Fuster., M. D. Diego, M. Ferrer, G. Piñero, and A. González, "Nonlinear filtered-x second-order adaptive Volterra filters for listening-room compensation," in 23rd Proc. EUSIPCO Conf., 2015.

[2] B. Widrow and M. Bilello, "Adaptive inverse control," in Intelligent Control, Process of the IEEE Int Symp. on. IEEE, 1993, pp. 1–6.

[3] M. Miyoshi and Y. Kaneda, "Inverse filtering of room acoustics," IEEE Trans. Acoust., Speech, Signal Process., vol. 36, pp. 145–152, 1988.

[4] O. Kirkeby, P. Nelson, H. Hamada, and F. Orduna-Bustamante, "Fast deconvolution of multichannel systems using regularization," IEEE Trans. Speech Audio Process., vol. 6 (2), pp. 189–194, 1998.

[5] S. Haykin, Adaptive Filter Theory. Prentice Hall, 4th ed., 2002.

[6] B. Widrow and S. Stearns, Adaptive Signal Processing. Prentice Hall, 1985.

[7] L. Fuster, M. De Diego, M. Ferrer, A. González, and G. Piñero, "A biased multichannel adaptive algorithm for room equalization," in Proc. EUSIPCO Conf., 2012, pp. 1344–1348.

[8] S. Cecchi, A. Primavera, F. Piazza, and A. Carini, "An adaptive multiple position room response equalizer," in Proc. EUSIPCO Conf., September 2011.

[9] D. S. Talagala, W. Zhang, and T. D. Abhayapala, "Efficient multi-channel adaptive room compensation for spatial soundfield reproduction using a modal decomposition," Audio, Speech, and Language Process., IEEE/ACM Trans., vol. 22, no. 10, pp. 1522–1532, 2014.

[10] M. Kolundzija, C. Faller, and M. Vetterli, "Multi-channel low-frequency room equalization using perceptually motivated constrained optimization," in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Process. (ICASSP), 2012, pp. 533–536.

[11] S. Goetze, M. Kallinger, A. Mertins, and K.-D. Kammeyer, "Multi-channel listening-room compensation using a decoupled filtered-x LMS algorithm," in 42nd Asilomar Conf. on Signals, Systems and Computers, oct. 2008, pp. 811–815.

[12] S. Spors, H. Buchner, and R. Rabenstein, "Efficient active listening room compensation for wave field synthesis," 116th AES Conv, 2004.

[13] S. Spors, H. Buchner, R. Rabenstein, and W. Herbordt, "Active listening room compensation for massive multichannel sound reproduction systems using wave-domain adaptive filtering," J Acoust Soc Am, vol. 122, no. 1, pp. 354–369, July 2007.

[14] M. Schneider and W. Kellermann, "Adaptive listening room equalization using a scalable filtering structure in the wave domain," in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Process., 2012, pp. 13–16.

[15] S. Cecchi, L. Romoli, A. Carini, and F. Piazza, "A multichannel and multiple position adaptive room response equalizer in warped domain: Real-time implementation and performance evaluation," Applied Acoustics, vol. 82, pp. 28 – 37, 2014.

[16] E. Bjarnason, "Analysis of the filtered-x LMS algorithm," Speech and Audio Process., IEEE Trans., vol. 3, no. 6, pp. 504–514, Nov 1995.

[17] M. de Diego, A. Gonzalez, M. Ferrer, and G. Piñero, "Multichannel active noise control system for local spectral reshaping of multifrequency noise," *J Sound Vib*, vol. 274, no. 1, pp. 249–271, 2004.

[18] M. de Diego, A. Gonzalez, and C. Garcia., "On the performance of a local active noise control system," *ICASSP99, Proc. of the Int. Conf on Acoustics, Speech and Signal Process.*, vol. 2, pp. 885–888, 1999.

[19] M. Ferrer, A. González, M. De Diego, and G. Piñero, "Mean-square analysis of a fast filtered-x affine projection algorithm," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Process.*, 2008, pp. 349–352.

[20] M. Ferrer, M. de Diego, G. Piñero, and A. González, "Active noise control over adaptive distributed networks," *Signal Processing*, vol. 107, no. 0, pp. 82 – 95, 2015.

[21] D. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *Speech and Audio Process., IEEE Trans.*, vol. 8, no. 5, pp. 508–518, Sep 2000.

[22] J. Benesty and S. Gay, "An improved PNLMS algorithm," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Process. (ICASSP)*, vol. 2, 2002, pp. II–1881–II–1884.

[23] J. Arenas-García, M. de Diego, L. A. Azpicueta-Ruiz, M. Ferrer, and A. González, "Combinations of proportionate adaptive filters in acoustics: An application to active noise control," in *Proc. EUSIPCO Conf.*, 2011.

[24] F. C. de Souza, O. J. Tobias, R. Seara, and D. R. Morgan, "A PNLMS algorithm with individual activation factors," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2036–2047, Apr. 2010.

[25] J. Arenas-García, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *Signal Process., IEEE Trans.*, vol. 54, no. 3, pp. 1078–1090, 2006.

[26] M. Martinez-Ramon, J. Arenas-Garcia, A. Navia-Vazquez, and A. Figueiras-Vidal, "An adaptive combination of adaptive filters for plant identification," in *Digital Signal Process., 2002. DSP 2002. 2002 14th Int Conf on*, vol. 2, 2002, pp. 1195–1198 vol.2.

[27] J. Arenas-García, M. Martínez-Ramón, Á. Navia-Vázquez, and A. R. Figueiras-Vidal, "Plant identification via adaptive combination of transversal filters," *Signal Processing*, vol. 86, no. 9, pp. 2430–2438, 2006.

[28] B. Jelfs, S. Javidi, P. Vayanos, and D. Mandic, "Characterisation of signal modality: Exploiting signal nonlinearity in machine learning and signal processing," *Journal of Signal Processing Systems*, vol. 61, no. 1, pp. 105–115, 2010.

[29] S. Lu, J. Sun, G. Wang, and J. Tian, "A novel GSC beamformer using a combination of two adaptive filters for smart antenna array," *Antennas and Wireless Propagation Letters, IEEE*, vol. 11, pp. 377–380, 2012.

[30] D. Comminiello, M. Scarpiniti, R. Parisi, and A. Uncini, "Combined adaptive beamforming schemes for nonstationary interfering noise reduction," *Signal Processing*, vol. 93, no. 12, pp. 3306–3318, 2013.

[31] T. Trump, "A combination of two NLMS filters in an adaptive line enhancer," in *Digital Signal Processing (DSP), 2011 17th International Conference on*. IEEE, 2011, pp. 1–6.

[32] J. Arenas-Garcia and A. Figueiras-Vidal, "Adaptive combination of proportionate filters for sparse echo cancellation," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 17, no. 6, pp. 1087–1098, 2009.

[33] L. A. Azpicueta-Ruiz, M. Zeller, A. R. Figueiras-Vidal, J. Arenas-García, and W. Kellermann, "Improved acoustic echo cancellation for low SNR based on blockwise combination of filters," in *Proceedings of 20th Int Congress on Acoustics, ICA*, 2010.

[34] J. Ni and F. Li, "Adaptive combination of subband adaptive filters for acoustic echo cancellation," *Consumer Electronics, IEEE Transactions on*, vol. 56, no. 3, pp. 1549–1555, 2010.

[35] L. A. Azpicueta-Ruiz, M. Zeller, A. R. Figueiras-Vidal, J. Arenas-García, and W. Kellermann, "Adaptive combination of Volterra kernels and its application to nonlinear acoustic echo cancellation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 1, pp. 97–110, 2011.

[36] M. Ferrer, M. De Diego, A. González, and G. Piñero, "Steady-state mean square performance of the multichannel filtered-x affine projection algorithm," *IEEE Trans. Signal Process.*, vol. 60 (6), pp. 2771–2785, 2012.

[37] M. Ferrer, A. González, M. De Diego, and G. Piñero, "Convex combination filtered-x algorithms for active noise control systems," *IEEE Trans. Audio, Speech, and Language Process.*, vol. 21, no. 1, pp. 156–167, 2013.

[38] M. Lazaro-Gredilla, L. Azpicueta-Ruiz, A. Figueiras-Vidal, and J. Arenas-Garcia, "Adaptively biasing the weights of adaptive filters," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3890–3895, 2010.

[39] L. A. Azpicueta-Ruiz, M. Lazaro-Gredilla, A. R. Figueiras-Vidal, and J. Arenas-Garcia, "A block-based approach to adaptively bias the weights of adaptive filters," in *Mach. Learn. for Signal Process. (MLSP), IEEE Int Workshop on*. IEEE, 2011, pp. 1–6.

[40] L. A. Azpicueta-Ruiz, A. R. Figueiras-Vidal, and J. Arenas-Garcia, "A normalized adaptation scheme for the convex combination of two adaptive filters," in *Acoustics, Speech and Signal Process., 2008. ICASSP 2008. IEEE Int. Conf on*. IEEE, 2008, pp. 3301–3304.

[41] L. Fuster, M. de Diego, M. Ferrer, and A. González, "Steady-state analysis of biased filtered-x algorithms for adaptive room equalization," in *Acoustics, Speech and Signal Process. (ICASSP), 2014 IEEE Int Conf on*, May 2014, pp. 6647–6651.

[42] P. Loganathan, E. A. P. Habets, and P. Naylor, "A proportionate adaptive algorithm with variable partitioned block length for acoustic echo cancellation," in *Acoustics, Speech and Signal Process. (ICASSP), 2011 IEEE Int Conf on*, May 2011, pp. 73–76.

[43] C. Paleologu, J. Benesty, and S. Ciochina, "Sparse adaptive filters for echo cancellation," *Synthesis Lectures on Speech and Audio Process.*, vol. 6, no. 1, pp. 1–124, 2010.

[44] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*. Prentice-Hall, Inc., 1996.

Laura Fuster received the Telecommunication Engineer degree from the Universitat Politcnica de Valencia (UPV), Spain, in 2002, and the Ph.D. degree in 2015. Since 2001, she has been working in the Audio and Communications Signal Processing Group (GTAC) of the Institute of Telecommunications and Multimedia Applications (iTEAM) of the UPV, where at present she is a Technical Researcher in charge of the experimental laboratory of audio and communications. Her current research interests include multichannel signal processing for audio applications, adaptive filtering, room equalization, spatial sound reproduction and psychoacoustics.

Maria de Diego (S'99-M'00-SM'14) received the M.Sc. degree in Telecommunications Engineering from the Universitat Politcnica de Valencia, Spain, in 1994, and the Ph.D degree from the same university in 2003. Her dissertation was on active noise conformation of enclosed acoustic fields. She is currently working as Associate Professor in digital signal processing and communications and a Researcher with the GTAC Group that belongs to the iTEAM Institute of the Universitat Politcnica de Valencia. Dr. de Diego has been involved in different research projects including acoustic sensor networks, active noise control, fast adaptive filtering algorithms, sound quality evaluation, and 3-D sound reproduction. She has published more than 70 papers in international technical journals and conferences in the fields of signal processing and applied acoustics. Her current research interests include smart sound processing for reproduction and control, multichannel signal processing and distributed adaptive filtering.

Luis A. Azpicueta-Ruiz (azpicueta@tsc.uc3m.es) received the telecommunication engineer degree in 2004 from Universidad Politcnica de Madrid, Madrid, Spain, and the Ph.D. degree in telecommunication technologies from Universidad Carlos III de Madrid, Legans, Spain, in 2011. He is a lecturer of electroacoustics and acoustic engineering with the Department of Signal Theory and Communications at Universidad Carlos III de Madrid. His research interests are focused in the fields of adaptive signal processing and their applications, mainly in audio and acoustic processing. His main publications are related with nonlinear adaptive filtering and its application to acoustic signal processing. He is a Member of the IEEE.

Miguel Ferrer graduated in Telecommunications Engineering in the year 2000 at the Universidad Politcnica de Valencia (UPV), Spain. He was collaborating with the Audio and Communications Signal Processing Group (GTAC) since a year before, performing a six months research stay in the Instituto de Investigacin Aplicada al Automvil, Tarragona, Spain (Automobile Applied Research Institute). Subsequently, he was awarded several grants offered both by the Communications Department of the UPV and by the Research, Development and Innovation Vice-Chancellery of the same university, enabling him to start his Doctorate studies and to collaborate in different research projects within the GTAC. During this period he has authored or co-authored over sixty papers related with signal processing in renowned journals and conferences. Since 2005 he works as an assistant lecturer in the Communications Department of the Polytechnic University of Valencia (UPV). His research activity is focused on the study of adaptive algorithms and its application to audio digital processing and noise active control, a subject about which he developed his doctoral thesis.