Distributed Affine Projection Algorithm Over Acoustically Coupled Sensor Networks

Miguel Ferrer, *Member, IEEE*, Alberto Gonzalez, *Senior Member, IEEE*, Maria de Diego, *Senior Member, IEEE* and Gema Piñero, *Senior Member, IEEE*

Abstract—In this paper we present a distributed affine projection (AP) algorithm for an acoustic sensor network where the nodes are acoustically coupled. Every acoustic node is composed of a microphone, a processor and an actuator to control the sound field. This type of networks can use distributed adaptive algorithms to deal with the active noise control (ANC) problem in a cooperative manner, providing more flexible and scalable ANC systems. In this regard, we introduce here a distributed version of the multichannel filtered-x AP algorithm over an acoustic sensor network that it is called distributed filtered-x AP (DFxAP) algorithm. The analysis of the mean and the mean-square deviation performance of the algorithm at each node is given for a network with a ring topology and without constraints in the communication layer. The theoretical results are validated through several simulations. Moreover, simulations show that the proposed DFxAP outperforms the previously reported distributed multiple error filtered-x least mean square (DMEFxLMS) algorithm.

Index Terms—Acoustic sensor networks, affine projection algorithm, distributed adaptive filters, active noise control.

I. INTRODUCTION

ACOUSTIC sensor networks are a particular case of the well-known wireless sensor networks [1] where the sensors are devoted to monitor acoustic signals. They are usually referred as wireless acoustic sensor networks (WASN) assuming that the sensors can communicate among them through a wireless network. One of the main tasks of the use of WASNs is to monitor and locate acoustic sources in outdoor [2] and indoor spaces [3], [4], or even to jointly locate the sources and the sensors as in [5]. Regarding their use inside rooms, another important application is to enhance the recorded speech signals via beamforming [6]–[8]. Generally speaking, the WASNs use the acoustic signals recorded at the node sensors to perform some kind of signal processing algorithm, usually in a distributed way. However, other type of acoustic networks have been proposed whose nodes are not only equipped with microphones, but that they can manage one or more loudspeakers or actuators as well. This new type of WASNs can perform all the tasks described above, but they can also carry out other applications related to sound field

control, as active noise control (ANC) [9], [10] or personal sound zones [11], [12].

The emergence of the WASNs has forced a review of the classical signal processing algorithms from different points of view. First, the algorithms have to deal with the distributed processing inherent to these networks, and the unavailability of the entire system information at each node, that is, each node knows its own signals and parameters, and has access only to partial information provided by the network. Second, the WASNs can present a wide variety of topologies, which can greatly affect the distributed algorithms and their performance. Third, WASNs need a certain degree of synchronization to perform the sound control. On the other hand, the WASNs present great advantages for acoustic applications as scalability, flexibility in their deployment, and, in general, much lower cost. Moreover, there exist scenarios where the acquisition and generation of signals must be inherently distributed, such as devices with low processing capabilities or transducers far away among them. Furthermore, the majority of multimedia user devices could serve as acoustic nodes or at least perform the processing tasks. Indeed, commercial mobile devices have been recently used as nodes to develop different audio applications, like an immersive audio system [13], the identification of audio sources [14] or the acoustic-based localization of multiple persons within a room [15].

In the particular case of ANC, it is worth highlighting the high potential of the WASNs for the practical deployment of the distributed ANC systems, mainly due to their versatility and scalability. The WASNs can be composed by simple single (few) channel(s) systems and can perform like a complex multichannel one. Nevertheless, despite the great interest for WASNs, very few publications deal with their application in ANC systems. Two distributed ANC systems were first introduced in [16] [17]. These systems are called decentralized since the processors do not collaborate or interchange any local information, although the computational burden is shared among them. Therefore they cannot overcome the centralized system performance if there exists acoustic coupling among the actuators and microphones. In a previous work [9], we showed how the cooperation provided by a WASN helped ANC systems to achieve similar performance to centralized solutions and how they could benefit from the advantages of distributed systems. That approach assumed perfect synchronization among the nodes and was based on the distributed multiple error filtered-x least mean square (DMEFxLMS) algorithm, where the adaptive filters update was carried out in a distributed way over a ring topology with incremental com-

munication [18] [19]. This type of ideal networks allows the study of the performance of the distributed algorithms in order to be compared to the centralized ones. It can be shown that for WASNs with no constraints regarding the communication among the nodes, it is possible to obtain the same performance than in a centralized system, as long as the proper distributed algorithm is implemented. These distributed algorithms bring many benefits, as the acquisition and generation of signals at every node, and the distribution of the computational burden among the nodes, contributing to a more versatile system. The present work is focused on the well known affine projection (AP) algorithm [20], a robust and stable algorithm that can be a good alternative to LMS [21] when higher convergence speed is desired. The efficiency of AP algorithm has been reported in a variety of applications, such as active noise control [22]–[25], acoustic equalization [26] and echo cancellation [27]. Adaptive distributed networks based on the AP have also been presented in some recent works such as [28], [29] where a distributed network with a ring topology is also implemented. Specifically, an incremental collaboration strategy is used in [28], while diffusion collaboration schemes are considered in [29]. However, they solve a distributed system identification problem in which all the nodes in the network try to estimate the same unknown vector $\mathbf{w_0}$. On the contrary, the present paper addresses a different problem in which all the nodes are acoustically coupled and therefore collaboration is essential to reach a solution similar to the one provided by the corresponding centralized system. Indeed, the solution reached at each node also depends on the local measurement of the other nodes of the network due to the acoustic interaction among them. In this regard, signals involved in an ANC system suffer from the acoustic propagation effects, unlike signals used in [28] and [29]. This fact leads to the use of a particular adaptive filtering structure, such as the filtered-x scheme. To further clarify why traditional filtering approaches cannot be directly applied to the ANC problem, it should also be highlighted that the measurement signals such as that used in [28] and [29] are not available in ANC problems. The acoustically generated error signals picked up at the microphones are used instead.

In this paper, and motivated by the good tradeoff between convergence speed and computational cost of the AP algorithms, we focus on a distributed version of the multichannel filtered-x AP (FxAP) algorithm [22] [23] over a WASN with incremental communication among the nodes. In this regard, and on the basis of the metodology introduced in [30], we develop an approximated version of the FxAP for WASNs where the computational burden is efficiently shared among all the processors and every node collaborates to obtain the overall solution, in a similar way to [9]. This means that each node contributes to the network solution by computing a portion of it that depends only on its local data. Thus, the overall solution is obtained from the contributions of all the nodes. The proposed algorithm is called distributed filtered-x AP (DFxAP) algorithm, and we demonstrate through simulations with real acoustic channel responses that, though the steady-state solution of this algorithm is different to the multichannel FxAP, it can outperform the DMEFxLMS in terms of convergence speed. Another key contribution of this
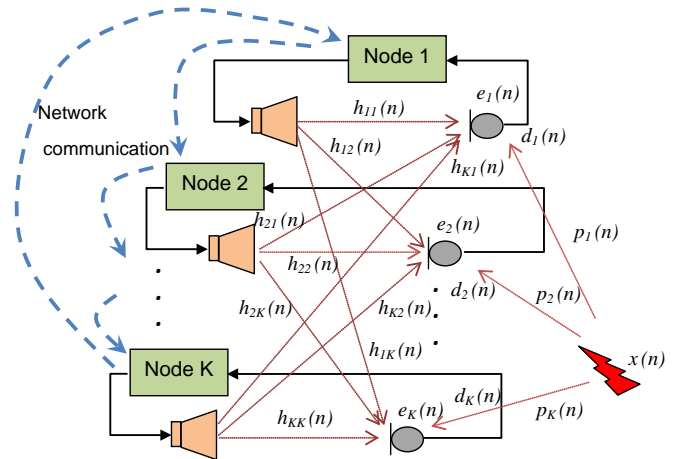

Fig. 1. Acoustic network using incremental ring topology for active noise control.

paper is the analysis of the mean and mean-square deviation of the network weights in order to evaluate the performance of the resulting WASN. The theoretical results have been validated through simulations for ring networks of different sizes, and also for different projection orders and different noise variance at the microphones.

The rest of the paper is organized as follows. In Section II we derive the DFxAP algorithm for WASNs by introducing an optimization problem with constraints. In Section III, we analyze the performance of the distributed algorithm in terms of mean weight and mean-square deviation. Section IV presents a comprehensive set of simulations supporting the validity of the results. Finally, Section V highlights the main conclusions of the paper.

Notation: Throughout the paper the following notation is used: boldface upper-case letters denote matrices (e.g. $\mathbf{A}$), boldface lower-case letters denote vectors (e.g., $\mathbf{a}$), and italics denote scalars, (e.g. a or A). The Euclidean norm and the mathematical expectation are denoted by $\| \cdot \|$ and $\mathrm{E}(\cdot)$, respectively. $\mathrm{Tr}\{\cdot\}$ is the trace of a matrix, $(\cdot)^T$ stands for matrix or vector transpose, and $\mathbf{I}_a$ is the a-size identity matrix.

II. DISTRIBUTED FILTERED-X AFFINE PROJECTION ALGORITHM

We consider a distributed acoustic network of K single-channel nodes that will support a multichannel ANC system composed of K error sensors and K secondary sources, see Fig. 1. In order to clarify the exposition, all the nodes of the network are equipped with a single microphone and a single loudspeaker. An unwanted noise reaches the microphones, being $d_k(n)$ the undesired signal picked up at the microphone of the kth node and $p_k(n)$ the impulse response characterizing the propagation of the noise up to the microphone. The aim of the ANC system is to minimize the noise signal at the microphone locations. For this purpose, each node should update its own adaptive filter relying only on local data and on some available network information, and computational burden should be distributed among the different nodes. Taking

into account this requirements, we introduce in the following the distributed AP algorithm, the DFxAP.

In order to cancel the undesired signals $d_k(n)$, the following equation must be fullfiled at the kth node, with $k = 1, ..., K$

$$d_k(n) = \sum_{j=1}^{K} [-y_j(n) * h_{jk}(n)], \quad (1)$$

where $*$ denotes the discrete linear convolution, $h_{jk}(n)$ is the impulse response of the M-lenght FIR filter that models the acoustic channel between the loudspeaker of the jth node and the microphone of the kth node, and $y_j(n)$ is the signal generated by the loudspeaker of the jth node. The response $h_{jk}(n)$ can be written in vector form as $\mathbf{h}_{jk} = [h_{jk}(0), h_{jk}(1), ..., h_{jk}(M-1)]^T$.

Let us define a network $[KL \times 1]$ adaptive filter vector $\mathbf{w}(n)$ as the ordered concatenation of all the adaptive filter vectors of each node.

$$\mathbf{w}(n) = \left[\mathbf{w}_1(n)^T, \mathbf{w}_2(n)^T, \ldots, \mathbf{w}_K(n)^T \right]^T, \quad (2)$$

where $\mathbf{w}_k(n)$ vector contains the L-coefficient filter that operates at node k.

Thus, output signals are obtained as

$$y_j(n) = \mathbf{w}_j(n)^T [\mathbf{X}(n)]_{(:,1)} \quad (3)$$

where $\mathbf{X}(n)$ is a toeplitz matrix of $L \times M$ size defined as

$$\mathbf{X}(n) = \left[\begin{array}{cccc} \mathbf{x}(n) & \mathbf{x}(n-1) & \cdots & \mathbf{x}(n-M+1) \end{array} \right], \quad (4)$$

$\mathbf{x}(n)$ is a $[L \times 1]$ vector with the most recent L samples of the reference signal $x(n)$, which is common to all nodes and is correlated with the unwanted noise. Furthermore $[\mathbf{X}(n)]_{(:,1)}$ is the $[L \times 1]$ vector corresponding to the first column of $\mathbf{X}(n)$.

We consider the $[L \times N]$ matrix $\mathbf{X}_{jk}(n)$ that contains the reference signal $x(n)$ filtered through $\mathbf{h}_{jk}$

$$\mathbf{X}_{jk}(n) = \left[\begin{array}{cccc} \mathbf{x}_{jk}(n) & \mathbf{x}_{jk}(n-1) & \cdots & \mathbf{x}_{jk}(n-N+1) \end{array} \right] \quad (5)$$

where N is the projection order and $\mathbf{x}_{jk}(n)$ denotes an $[L \times 1]$ vector obtained by filtering the most recent L samples of the reference signal $x(n)$ through $\mathbf{h}_{jk}$:

$$\mathbf{x}_{jk}(n) = \mathbf{X}(n) \mathbf{h}_{jk}. \quad (6)$$

If we consider slow convergence of the adaptive filters, by applying (3) and (6) in (1) we get

$$d_k(n) = \sum_{j=1}^{K} [-\mathbf{x}_{jk}^T(n) \mathbf{w}_j(n)]. \quad (7)$$

We consider that the distributed algorithm converges to a specific solution $\mathbf{w_0}$. This allows us to rewrite (7) in vector form as

$$\mathbf{d}_k(n) = \left[\begin{array}{c} d_k(n) \\ d_k(n-1) \\ \vdots \\ d_k(n-N+1) \end{array} \right] = -\mathbf{A}_k^T(n) \mathbf{w_0} \quad (8)$$

with the KL-length vector defined as

$$\mathbf{w_0} = \left[\begin{array}{c} \mathbf{w}_{01} \\ \mathbf{w}_{02} \\ \vdots \\ \mathbf{w}_{0K} \end{array} \right], \quad (9)$$

and the $KL \times N$ matrix defined as

$$\mathbf{A}_k(n) = \left[\begin{array}{c} \mathbf{X}_{1k}(n) \\ \mathbf{X}_{2k}(n) \\ \vdots \\ \mathbf{X}_{Kk}(n) \end{array} \right]. \quad (10)$$

In order to obtain the network coefficients $\mathbf{w_0}$, an optimization problem with constraints based on the minimum disturbance rule [31] must be solved. Thus

$$\Delta \mathbf{w}(n) = \mathbf{w}(n) - \mathbf{w}(n-1) \quad (11)$$

is minimized in the mean-square sense, that is

$$\|\Delta \mathbf{w}(n)\|^2 = [\Delta \mathbf{w}(n)]^T \Delta \mathbf{w}(n) \quad (12)$$

is minimized subject to the constraints

$$\mathbf{d}_k(n) + \mathbf{A}_k^T(n) \mathbf{w}(n) = \mathbf{0} \quad \text{for} \quad k = 1, ..., K. \quad (13)$$

According to (12) and (13), and following the method of Lagrange multipliers [32], the following cost function is built

$$J(n) = \|\Delta \mathbf{w}(n)\|^2 + \sum_{k=1}^{K} [\mathbf{d}_k(n) + \mathbf{A}_k^T(n) \mathbf{w}(n)]^T \boldsymbol{\lambda}_k(n) \quad (14)$$

where $\boldsymbol{\lambda}_k(n)$ is the $N \times 1$ vector that comprises the N Lagrange multipliers corresponding to the N constraints at the kth node. To solve (14), we obtain the gradient of $J(n)$ with respect to the weight vector $\mathbf{w}(n)$ as

$$\nabla_w J(n) = \frac{\partial J(n)}{\partial \mathbf{w}(n)} = 2[\mathbf{w}(n) - \mathbf{w}(n-1)] + \sum_{k=1}^{K} \mathbf{A}_k(n) \boldsymbol{\lambda}_k. \quad (15)$$

The minimum of $J(n)$ is obtained by setting (15) to zero. This condition leads to

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \frac{1}{2} \sum_{k=1}^{K} \mathbf{A}_k(n) \boldsymbol{\lambda}_k. \quad (16)$$

Since the objective is to calculate the adaptive filter coefficients $\mathbf{w}(n)$ in a distributed way over a ring topology with incremental learning, every node will add one term to the summation in (16). We assume that at time n, node $k = 1$ has available the updated global vector obtained at time $n-1$, $\mathbf{w}(n-1)$. Then at node $k = 1$ the following equation can be computed:

$$\mathbf{w}^1(n) = \mathbf{w}(n-1) - \frac{1}{2} \mathbf{A}_1(n) \boldsymbol{\lambda}_1. \quad (17)$$

Afterwards node 1 transmits its local version of the global vector to node 2, and this node updates its local version as:

$$\mathbf{w}^2(n) = \mathbf{w}^1(n) - \frac{1}{2} \mathbf{A}_2(n) \boldsymbol{\lambda}_2. \quad (18)$$

Then, by induction, the update equation at the kth node becomes

$$\mathbf{w}^k(n) = \mathbf{w}^{k-1}(n) - \frac{1}{2}\mathbf{A}_k(n)\boldsymbol{\lambda}_k. \tag{19}$$

In order to supress the Lagrange multipliers in (19), and bearing in mind (13), we assume that at each iteration the coefficients estimated at every node tend to be the coefficients vector, namely $\mathbf{w}^k(n) \approx \mathbf{w}(n)$. By substituting (19) in the constraint relation (13) yields

$$\mathbf{d}_k(n) = -\mathbf{A}_k^T(n)\mathbf{w}^{k-1}(n) + \frac{1}{2}[\mathbf{A}_k^T(n)\mathbf{A}_k(n)]\boldsymbol{\lambda}_k. \tag{20}$$

It should be noted that (20) could also be derived as a particular solution of the minimization problem in (11) subject to the constraints $\sum_{k=1}^{K}[\mathbf{d}_k(n) + \mathbf{A}_k^T(n)\mathbf{w}(n)] = \mathbf{0}$ when we assume that $\mathbf{w}^k(n) = \mathbf{w}(n)$ for all k, as well as each of the terms of the summation in the previous equation is zero. Solving (20) for the kth Lagrange vector, we have

$$\boldsymbol{\lambda}_k = 2[\mathbf{A}_k^T(n)\mathbf{A}_k(n)]^{-1}\mathbf{e}_k^{\mathrm{a}}(n), \tag{21}$$

where

$$\mathbf{e}_k^{\mathrm{a}}(n) = \mathbf{d}_k(n) + \mathbf{A}_k^T(n)\mathbf{w}^{k-1}(n) \tag{22}$$

is the *a priori* error vector. Finally, substituting (21) in (19) and introducing both a step-size parameter μ to control changes in the weight vector from one iteration to the other, and a regularization factor δ, the following update equation for the DFxAP algorithm is given

$$\mathbf{w}^k(n) = \mathbf{w}^{k-1}(n) - \mu\mathbf{A}_k(n)[\mathbf{A}_k^T(n)\mathbf{A}_k(n) + \delta\mathbf{I}_N]^{-1}\mathbf{e}_k^{\mathrm{a}}(n). \tag{23}$$

Finally, once the global updated vector at time instant n has been obtained as $\mathbf{w}(n) = \mathbf{w}^K(n)$, their values are disseminated to the rest of the nodes. The kth node output signal is given by

$$y_k(n) = \mathbf{w}_k^T(n)\,[\mathbf{X}(n)]_{(:,1)}, \tag{24}$$

where $\mathbf{w}_k(n)$ is the $[L \times 1]$ vector that contais the coefficients $L(k-1)+1$ to Lk of $\mathbf{w}(n)$. It should be noted that every node only uses a part of the estimated global solution to generate the signals that will feed the corresponding secondary sources. Taking into account the practical implementation of this algorithm, two considerations should be highlighted. First, matrices $\mathbf{A}_k(n)$ defined in (10) require the real acoustic channels but only an estimate of them can be available ($\hat{\mathbf{h}}_{jk}$). Secondly, ANC systems based on the conventional filtered-x structure [23] do not provide the desired signal $d_k(n)$ to calculate the required *a priori* error vector. Therefore, the DFxAP algorithm can use past samples of the error signal $e(n)$ to estimate $\mathbf{e}_k^{\mathrm{a}}(n)$, $\mathbf{e}_k^{\mathrm{a}}(n) \approx \mathbf{e}_k(n) = [e_k(n)\ e_k(n-1)\ ...\ e_k(n-N+1)]^T$. This approximation provides a significant computational saving since it avoids to compute (22) at each iteration and at each node. Moreover, it provides a fairly accurate guess in the usual case of slow filter variations.

A summary of the algorithm instructions executed per sample time n, including the required multiplications involved in each operation is given in **Algorithm 1**. The number of multiplications can be directly calculated from the equations, obtaining a computational cost of the DFxAP algorithm per

---

Algorithm 1 DFxAP algorithm.

1: Initialize: $\mathbf{w}(0) = \mathbf{w}^k(0) = [0,\dots,0]^T$, $\forall k$; $\mathbf{X}(0) = \mathbf{0}_{L \times M}$
2: $n = 1$ {Start sample time}
3: **repeat**
4:    $\mathbf{w}^0(n) = \mathbf{w}(n-1)$ {Needed at node $k=1$ in line 12}
5:    {Obtain reference signal $x(n)$}
6:    **for all** Node $1 \le k \le K$ **do**
7:       $\mathbf{w}_k(n) = [\mathbf{w}^k(n-1)]_{(L(k-1)+1:Lk)}$
8:       $y_k(n) = \mathbf{w}_k^T(n)\,[\mathbf{X}(n)]_{(:,1)}$ (Multipl.: L)
9:       **for all** $1 \le j \le K$ **do**
10:          $\mathbf{x}_{jk}(n) = \mathbf{X}(n)\hat{\mathbf{h}}_{jk}$ (Multipl.: KM)
11:          $\mathbf{X}_{jk}(n) = \big[\ \mathbf{x}_{jk}(n),\ \ [\mathbf{X}_{jk}(n-1)]_{(:,1:(N-1))}\ \big]$
12:       **end for**
13:       $\mathbf{e}_k(n) = [e_k(n)\ e_k(n-1)\ \cdots\ e_k(n-N+1)]^T$
14:       $\mathbf{A}_k(n) = \big[\ \mathbf{X}_{1k}^T(n)\ \ \mathbf{X}_{2k}^T(n)\ \ \cdots\ \ \mathbf{X}_{Kk}^T(n)\ \big]^T$
15:       $\boldsymbol{\epsilon}_k(n) = [\mathbf{A}_k^T(n)\mathbf{A}_k(n) + \delta\mathbf{I}]^{-1}\mathbf{e}_k(n)$
         (Multipl.: $[KL+1]N^2 + O(N^3)$)
16:       $\mathbf{w}^k(n) = \mathbf{w}^{k-1}(n) - \mu\mathbf{A}_k(n)\boldsymbol{\epsilon}_k(n)$
         (Multipl.: $N[KL+1]$)
17:    **end for**
18:    $\mathbf{w}(n) = \mathbf{w}^K(n)$ {Updated vector}
19:    **for all** Node $1 \le k \le (K-1)$ **do**
20:       $\mathbf{w}^k(n) = \mathbf{w}(n)$ {Disseminate updated vector}
21:    **end for**
22:    $n = n + 1$ {Update sample time}
23: **until** convergence is achieved

---

iteration and per node of $L + KM + (KL+1)N + (KL+1)N^2 + 2N^3$ multiplications. It should be noted that the matrix multiplication $[\mathbf{A}_k^T(n)\mathbf{A}_k(n)]$ in line 15 involves KLN^2 multiplications. However, this product can be iteratively computed reducing the multiplications to $2N^2$. Thus, the DFxAP algorithm would require only $L + MK + 3N^2 + (KL+1)N + 2N^3$ multiplications per iteration and per node. It can be easily verified that when low N values are used, the DFxAP algorithm exhibits a slightly higher computational cost than the DMEFxLMS algorithm, that involves $L(K+1) + KM + 1$ multiplications [9]. The computational complexity of both algorithms, the DFxAP and the DMEFxLMS is summarized in Table I, where a typical case is also illustrated for $K = 4$, $L = 20$, $M = 256$, and $N = 2$ and 4.

TABLE I
SUMMARY OF THE COMPUTATIONAL COMPLEXITY OF THE DFxAP AND
THE DMEFxLMS. COMPLEXITY IS MEASURED AS THE NUMBER OF
MULTIPLICATIONS PER ITERATION. TYPICAL CASE: $M = 256$, $K = 4$
AND $L = 20$.

Algorithm	Multiplications	Typ. case
DFxAP ($N = 2$)	$\mathbf{L + KM + 3N^2 + (KL+1)N + 2N^3}$	1234
DFxAP ($N = 4$)	$\mathbf{L + KM + 3N^2 + (KL+1)N + 2N^3}$	1544
DMEFxLMS	$\mathbf{L(K+1) + KM + 1}$	1125

III. PERFORMANCE ANALYSIS

The performance of the DFxAP algorithm is studied in terms of the mean and the mean-square deviation (MSD) of the network weights. The closed-form expressions are derived for the last node of the network as this node provides the network global state at time n, see step 18 in **Algorithm 1**.

As we need to deal with the expectation operator, we shall rely on several assumptions in the following analysis. First, we

TABLE II
THE LIST OF ASSUMPTIONS USED IN THE PAPER

$A.1)$	The matrix $\mathbf{A}_k(n)$ (and consequently the matrix $\mathbf{Q}_k(n)$, which depends on the same data as $\mathbf{A}_k(n)$) is uncorrelated with $\mathbf{w}(\infty)$.
$A.2)$	The matrix $\mathbf{A}_k(n)$ (and consequently the matrix $\mathbf{Q}_k(n)$) is uncorrelated with $\mathbf{r}_k(n)$.
$A.3)$	The matrices $\mathbf{\Psi}_K(n)$ and $\mathbf{\Phi}_{KK}(n)$ (which depend on the same data as $\mathbf{A}_k(n)$) are uncorrelated with both $r_k(n)$ and $\widetilde{\mathbf{w}}^K(n)$.
$A.4)$	$r_k(n)$ is uncorrelated with $\widetilde{\mathbf{w}}^K(n)$.
$A.5)$	$r_k(n)$ is uncorrelated with $r_p(n)$ for $p \neq k$.

will introduce some approximations due to statistical considerations between different vectors and matrices. These assumptions are collected in Table II. Secondly, at steady-state when $n \to \infty$, it is found that $\mathrm{E}\left(\mathbf{w}^K(n)\right) = \mathrm{E}\left(\mathbf{w}^K(n-1)\right) = \mathrm{E}\left(\mathbf{w}^{K-p}(n)\right) = \mathrm{E}\left(\mathbf{w}(\infty)\right)$ for $1 \leq p < K$.

A. Mean steady-state weight behaviour

The aim of this section is to analyse the mean value of the network weights (2) of the proposed DFxAP algorithm. This analysis describes the mean weight behaviour of the adaptive weights at each node and provides the corresponding steady-state mean weight vector.

The DFxAP algorithm updates the weights at every node according to (23). Particularizing this equation for node K (the last node of the network), we get

$$\mathbf{w}^K(n)$$
$$= \mathbf{w}^K(n-1)$$
$$\quad -\mu \sum_{p=0}^{K-1} \mathbf{Q}_{K-p}(n)\mathbf{e}_{K-p}^{\mathrm{a}}(n)$$
$$= \mathbf{w}^K(n-1)$$
$$\quad -\mu \sum_{p=0}^{K-1} \mathbf{Q}_{K-p}(n)[\mathbf{d}_{K-p}(n) + \mathbf{A}_{K-p}(n)^T \mathbf{w}^{K-p-1}(n)]$$
$$= \mathbf{w}^K(n-1) - \mu \sum_{p=0}^{K-1} \mathbf{Q}_{K-p}(n)\mathbf{d}_{K-p}(n)$$
$$\quad -\mu \sum_{p=0}^{K-1} \mathbf{Q}_{K-p}(n)\mathbf{A}_{K-p}(n)^T \mathbf{w}^{K-p-1}(n), \tag{25}$$

where $\mathbf{Q}_k(n)$ is a $KL \times N$ matrix defined as $\mathbf{Q}_k(n) = \mathbf{A}_k(n)[\mathbf{A}_k^T(n)\mathbf{A}_k(n) + \delta\mathbf{I}_N]^{-1}$.

By taking expectations of both sides and applying the limit as $n \to \infty$, expression (25) becomes

$$\lim_{n \to \infty} \mathrm{E}\left(\sum_{p=0}^{K-1} \mathbf{Q}_{K-p}(n)\mathbf{d}_{K-p}(n)\right)$$
$$= -\lim_{n \to \infty} \mathrm{E}\left(\sum_{p=0}^{K-1} \mathbf{Q}_{K-p}(n)\mathbf{A}_{K-p}(n)^T \mathbf{w}(\infty)\right). \tag{26}$$

The previous expression has been derived by using the steady-state condition. Moreover, we make use of the following assumption. $A.1)$ *The matrix $\mathbf{A}_k(n)$ (and consequently the*

matrix $\mathbf{Q}_k(n)$, which depends on the same data as $\mathbf{A}_k(n)$) is uncorrelated with $\mathbf{w}(\infty)$. Using $A.1)$ (26) becomes

$$\mathrm{E}(\mathbf{w}(\infty)) = -\lim_{n \to \infty} \left\{ \left[\sum_{p=0}^{K-1} \mathrm{E}(\mathbf{Q}_{K-p}(n)\mathbf{A}_{K-p}^T(n))\right]^{-1} \right.$$
$$\left. \left[\sum_{p=0}^{K-1} \mathrm{E}(\mathbf{Q}_{K-p}(n)\mathbf{d}_{K-p}(n))\right] \right\}. \tag{27}$$

Note that (27) is the steady-state mean weight vector at the kth node but it also provides the steady-state behaviour of the global network, since all the nodes converge to the same solution. Furthermore, in the specific case of a one-node network, the DFxAP algorithm is equivalent to the centralized FxAP and therefore they provide the same steady-state solution. Thus, by setting K=1 in (27), we can derive the steady-state mean weight vector for a one-node network

$$\mathrm{E}(\mathbf{w}(\infty))$$
$$= -\lim_{n \to \infty} \left\{ \left[\mathrm{E}\left(\mathbf{Q}_1(n)\mathbf{A}_1^T(n)\right)\right]^{-1} \mathrm{E}\left(\mathbf{Q}_1(n)\mathbf{d}_1(n)\right) \right\}, \tag{28}$$

which is equivalent to that derived in [22] for the FxAP. Moreover, the elements of the matrices $\mathbf{Q}_k(n)$ and $\mathbf{A}_k(n)$ involved in (27) and (28), are different for both solutions, the distributed and the centralized one, except in the case of $K = 1$. Thus, matrices $\mathbf{Q}_1(n)$ and $\mathbf{A}_1(n)$ in (28) are the same as those used by the FxAP in [22].

B. Mean Square Deviation analysis of the DFxAP

In this section the evolution of the global filter vector with respect to the global optimal solution is analyzed. To this aim, the mean square deviation (MSD) at iteration n is derived as

$$\mathrm{MSD}(n) = \mathrm{E}(\|\widetilde{\mathbf{w}}(n)\|^2) = \mathrm{E}(\widetilde{\mathbf{w}}^T(n)\widetilde{\mathbf{w}}(n)), \tag{29}$$

where $\widetilde{\mathbf{w}}(n) = \mathbf{w_0} - \mathbf{w}(n)$ is the weight-error vector.

Till now we considered that the convergence of the algorithm provides at each node the global solution $\mathbf{w_0}$, such that the kth undesired signal vector is given by (8). However, this result is not achieved in practice and it is more realistic to use,

$$\mathbf{d}_k(n) = -\mathbf{A}_k^T(n)\mathbf{w_0} + \mathbf{r}_k(n), \tag{30}$$

where $\mathbf{r}_k(n)$ is modeled as an $N \times 1$ Gaussian noise vector of zero mean and σ_k^2 variance that is statistically uncorrelated with the reference signal. Therefore, we introduce the following assumption. $A.2)$ *The matrix $\mathbf{A}_k(n)$ (and consequently the matrix $\mathbf{Q}_k(n)$) is uncorrelated with $\mathbf{r}_k(n)$.* In case the real

ANC system fulfils the model in (30), and taking into account the hypothesis in $A.2$), we prove in Appendix A that

$$\mathrm{E}(\mathbf{w}(\infty)) = \mathbf{w_0}. \tag{31}$$

This result is similar to that derived in [22] for the centralized FxAP algorithm.

The general form of the filter updating at each node given in (23) can be compactly written as

$$\mathbf{w}^k(n) = \mathbf{w}^{k-1}(n) - \mathbf{\Psi}_k(n)\mathbf{e}_k^{\mathrm{a}}(n), \tag{32}$$

where the $[KL \times N]$ data matrix has been defined as $\mathbf{\Psi}_k(n) = \mu\mathbf{A}_k(n)[\mathbf{A}_k^T(n)\mathbf{A}_k(n) + \delta\mathbf{I}_N]^{-1}$, and it is assumed that $\mathbf{w}^0(n) = \mathbf{w}^K(n-1) = \mathbf{w}(n-1)$.

If the weight error vector at the kth node is defined as $\widetilde{\mathbf{w}}^k(n) = \mathbf{w_0} - \mathbf{w}^k(n)$, then, taking into account (22), (32) can be rewritten as

$$\widetilde{\mathbf{w}}^k(n) = \mathbf{\Phi}_k(n)\widetilde{\mathbf{w}}^{k-1}(n) + \mathbf{\Psi}_k(n)\mathbf{r}_k(n). \tag{33}$$

where $\mathbf{\Phi}_k(n) = \mathbf{I}_{KL} - \mathbf{\Psi}_k(n)\mathbf{A}_k^T(n)$ is a $KL \times KL$ matrix.

From (33) the updating equation for node K is obtained as,

$$\widetilde{\mathbf{w}}^K(n) = \widetilde{\mathbf{w}}(n) = \mathbf{\Phi}_K(n)\widetilde{\mathbf{w}}^{K-1}(n) + \mathbf{\Psi}_K(n)\mathbf{r}_K(n). \tag{34}$$

Then, in an iterative manner, $\widetilde{\mathbf{w}}^{K-1}(n)$ in (34) is expressed as a function of $\widetilde{\mathbf{w}}^K(n-1)$,

$$\widetilde{\mathbf{w}}^K(n) = \mathbf{\Phi}_{KK}(n)\widetilde{\mathbf{w}}^K(n-1) + \mathbf{\Psi}_K(n)\mathbf{r}_K(n) + \mathbf{\Phi}_K'(n), \tag{35}$$

where $\mathbf{\Phi}_{KK}(n)$ is the $KL \times KL$ matrix defined as

$$\mathbf{\Phi}_{KK}(n) = \prod_{i=K}^{1} \mathbf{\Phi}_i(n), \tag{36}$$

and $\mathbf{\Phi}_K'(n) = \sum_{k=1}^{K-1} \mathbf{\Phi}_{Kk}(n)\mathbf{r}_{K-k}(n)$, being

$$\mathbf{\Phi}_{Kk}(n) = \left[\prod_{i=K}^{K-k+1} \mathbf{\Phi}_i(n)\right]\mathbf{\Psi}_{K-k}(n), \quad 1 < k < K-1 \tag{37}$$

a $KL \times N$ matrix.

Following the methodology introduced in [33], we bring out the following model

$$\begin{bmatrix} \mathbf{r}_K(n+1) \\ \widetilde{\mathbf{w}}^K(n) \end{bmatrix} = \begin{bmatrix} \mathbf{Z} & \mathbf{0} \\ \mathbf{\Psi}_K(n) & \mathbf{\Phi}_{KK}(n) \end{bmatrix} \begin{bmatrix} \mathbf{r}_K(n) \\ \widetilde{\mathbf{w}}^K(n-1) \end{bmatrix} + \begin{bmatrix} \mathbf{r}_K'(n+1) \\ \mathbf{\Phi}_K'(n) \end{bmatrix}, \tag{38}$$

where $\mathbf{r}_K'(n+1)$ is an $N \times 1$ vector whose first component is equal to $r_K(n+1)$ and the rest equal zero. $\mathbf{0}$ is an $N \times KL$ matrix of zeros and

$$\mathbf{Z} = \begin{bmatrix} \mathbf{0}'^T & 0 \\ \mathbf{I}_{(N-1)} & \mathbf{0}' \end{bmatrix} \tag{39}$$

is an $N \times N$ matrix, being $\mathbf{0}'$ an $(N-1)$-length vector of zeros.

Considering that in steady state (29) can be expressed by using the weight-error vector at node K and at iteration n,

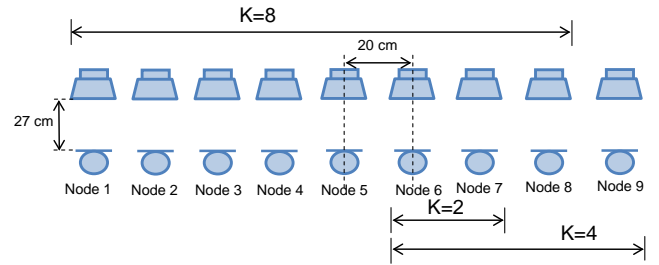$$\mathrm{MSD}(n) = \mathrm{E}\left(\widetilde{\mathbf{w}}^K(n)^T\widetilde{\mathbf{w}}^K(n)\right), \tag{40}$$

Fig. 2. Sketch of the nodes used in the experiments. Nodes selected for each presented WASN are indicated.

we define the covariance matrices of $\begin{bmatrix} \mathbf{r}_K(n+1)^T & \widetilde{\mathbf{w}}^K(n)^T \end{bmatrix}^T$ as

$$\begin{bmatrix} \mathbf{R}(n) & \mathbf{S}(n) \\ \mathbf{S}^T(n) & \mathbf{P}(n) \end{bmatrix} = \mathrm{E}\left\{ \begin{bmatrix} \mathbf{r}_K(n+1) \\ \widetilde{\mathbf{w}}^K(n) \end{bmatrix} \begin{bmatrix} \mathbf{r}_K^T(n+1) & \widetilde{\mathbf{w}}^K(n)^T \end{bmatrix} \right\} \tag{41}$$

Thus, the MSD evolution at node K can be expressed as $\mathrm{MSD}(n) = \mathrm{Tr}\left\{\mathrm{E}\left(\widetilde{\mathbf{w}}^K(n)\widetilde{\mathbf{w}}^K(n)^T\right)\right\} = \mathrm{Tr}\left(\mathbf{P}(n)\right)$. To analyze the matrices defined in (41) we first postmultiply the transpose of (38) to itself followed by the expectation, leading to

$$\begin{aligned} &\begin{bmatrix} \mathbf{R}(n) & \mathbf{S}(n) \\ \mathbf{S}^T(n) & \mathbf{P}(n) \end{bmatrix} \\ &= \mathrm{E}\left(\begin{bmatrix} \mathbf{Z} & \mathbf{0} \\ \mathbf{\Psi}_K(n) & \mathbf{\Phi}_{KK}(n) \end{bmatrix} \begin{bmatrix} \mathbf{r}_K(n) \\ \widetilde{\mathbf{w}}^K(n-1) \end{bmatrix} \right. \\ &\quad \left. \times \begin{bmatrix} \mathbf{r}_K(n)^T & \widetilde{\mathbf{w}}^K(n-1)^T \end{bmatrix} \begin{bmatrix} \mathbf{Z}^T & \mathbf{\Psi}_K(n)^T \\ \mathbf{0}^T & \mathbf{\Phi}_{KK}(n)^T \end{bmatrix} \right) \\ &+ \mathrm{E}\left(\begin{bmatrix} \mathbf{Z} & \mathbf{0} \\ \mathbf{\Psi}_K(n) & \mathbf{\Phi}_{KK}(n) \end{bmatrix} \begin{bmatrix} \mathbf{r}_K(n) \\ \widetilde{\mathbf{w}}^K(n-1) \end{bmatrix} \right. \\ &\quad \left. \times \begin{bmatrix} \mathbf{r}_K'(n+1)^T & \mathbf{\Phi}_K'(n)^T \end{bmatrix} \right) \\ &+ \mathrm{E}\left(\begin{bmatrix} \mathbf{r}_K'(n+1) \\ \mathbf{\Phi}_K'(n) \end{bmatrix} \begin{bmatrix} \mathbf{r}_K^T(n) & \widetilde{\mathbf{w}}^K(n-1)^T \end{bmatrix} \right. \\ &\quad \left. \times \begin{bmatrix} \mathbf{Z}^T & \mathbf{\Psi}_K(n)^T \\ \mathbf{0}^T & \mathbf{\Phi}_{KK}(n)^T \end{bmatrix} \right) \\ &+ \mathrm{E}\left(\begin{bmatrix} \mathbf{r}_K'(n+1) \\ \mathbf{\Phi}_K'(n) \end{bmatrix} \begin{bmatrix} \mathbf{r}_K'(n+1)^T & \mathbf{\Phi}_K'(n)^T \end{bmatrix} \right). \end{aligned} \tag{42}$$

Furthermore, assuming that $A.3$) *the matrices $\mathbf{\Psi}_K(n)$ and $\mathbf{\Phi}_{KK}(n)$ (which depend on the same data as $\mathbf{A}_k(n)$) are uncorrelated with both $r_k(n)$ and $\widetilde{\mathbf{w}}^K(n)$*, as explained in Appendix B, the following recurrence relations can be derived:

$$\mathbf{R}(n) = \mathbf{Z}\mathbf{R}(n-1)\mathbf{Z}^T + \sigma_K^2\mathbf{a}_1\mathbf{a}_1^T \tag{43}$$

$$\mathbf{S}(n) = \mathbf{Z}\mathbf{S}(n-1)\mathbf{\Phi}_{KK}(n)^T + \mathbf{Z}\mathbf{R}(n-1)\mathbf{\Psi}_K(n)^T \tag{44}$$

$$\begin{aligned} \mathbf{P}(n) \;=\; &\mathbf{\Phi}_{KK}(n)\mathbf{P}(n-1)\mathbf{\Phi}_{KK}(n)^T \\ &+\mathbf{\Psi}_K(n)\mathbf{R}(n-1)\mathbf{\Psi}_K(n)^T \\ &+\mathbf{\Psi}_K(n)\mathbf{S}(n-1)\mathbf{\Phi}_{KK}(n)^T \\ &+\mathbf{\Phi}_{KK}(n)\mathbf{S}(n-1)^T\mathbf{\Psi}_K(n)^T + \mathbf{\Omega}_K(n), \end{aligned} \tag{45}$$

where $\mathbf{\Omega}_K(n)$ is a matrix of KL dimensions defined as

$$\mathbf{\Omega}_K(n) = \sum_{k=1}^{K-1} \sigma_{K-k}^2\mathrm{E}(\mathbf{\Phi}_{Kk}(n)\mathbf{\Phi}_{Kk}(n)^T), \tag{46}$$

and $\mathbf{a}_1$ represents an N-length vector whose first element is 1 and the others are zeros.

Since $\mathbf{Z}$ is nilpotent (i.e. $\mathbf{Z}^n = 0$), when $n \to \infty$, $\mathbf{R}(n) = \sigma_K^2 \mathbf{I}_N$. Using $A.3$), matrix $\mathbf{S}(n)$ in (44) becomes zero. Then, (45) can be rewritten as

$$\begin{aligned} \mathbf{P}(n) &= \mathbf{\Phi}_{KK}(n)\mathbf{P}(n-1)\mathbf{\Phi}_{KK}(n)^T \\ &+ \mathbf{\Psi}_K(n)\mathbf{R}(n-1)\mathbf{\Psi}_K(n)^T + \mathbf{\Omega}_K(n). \end{aligned} \tag{47}$$

The initial values of the covariance matrices in (47) have been selected as follows: $\mathbf{P}(0) = \beta^2 \mathbf{I}_{KL}$, with $\beta \gg 1$, and $\mathbf{R}(0) = \sigma_K^2 \mathbf{I}_N$. Finally, the MSD evolution is obtained by computing the trace of $\mathbf{P}(n)$.

IV. SIMULATION RESULTS

In this section, we show the performance of the proposed DFxAP algorithm and compare the theoretical analysis with simulations. All the simulated WASNs use real acoustic responses measured inside a listening room of 9.36 m long by 4.78 m wide and 2.63 m high, with a reverberation time $T_{60} \simeq 200$ ms located at the Audio Processing Laboratory of the Universitat Politecnica de Valencia. These responses have been modelled as FIR filters of $M = 256$ coefficients with a sample rate of 2 kHz. We have simulated several acoustic networks of two, four or eight nodes considering the same setting of microphones and loudspeakers. The K loudspeakers, $K = \{2, 4, 8\}$, were selected from a linear loudspeaker array with a uniform separation of 20 cm between adjacent lodus-peakers. The corresponding K microphones were mounted on a linear platform with an equal separation of 20 cm between adjacent microphones. The microphones were placed opposite to the loudspeakers and separate 27 cm away from them. A sketch of the system is depicted in Fig. 2. The nodes selected for each presented WASN are: nodes 6 and 7 for $K = 2$, nodes 6 to 9 for $K = 4$, and nodes 1 to 8 for $K = 8$. The WASN uses a ring topology with incremental communication. Moreover, the communication among the network nodes does not experiment any constraint that could affect data rate or computational load. The WASN tries to cancel an unwanted noise at the microphones by means of K sources. The reference signal (unwanted noise) is a Gaussian random noise of zero mean and unit variance that is provided to all the nodes of the WASN as well as to the primary signal loudspeaker. This reference signal has been generated by a primary loudspeaker located 2 m away from both the secondary loudspeakers and error microphones.

A. Convergence speed performance

In the first set of experiments we evaluate the convergence speed of the proposed algorithm for an incremental network of two, four and eight nodes (K=2, 4 and 8), compared to the DMEFxLMS. The evaluation of the algorithms behavior is based on the instantaneous relative residual sound level at node k, $\mathrm{SL}_k(n)$, defined as the ratio in dB between the instantaneous estimated error power with and without the application of the active noise controller,

$$\mathrm{SL}_k(n) = 10\log_{10}\left[\frac{e_k^2(n)}{d_k^2(n)}\right], \tag{48}$$

where $d_k(n)$ is the signal that would be measured by the microphone of the kth node if the ANC system was inactive, and $e_k(n)$ is the signal that is measured by the same microphone when ANC is active. Moreover, we define

$$\mathrm{SL}(n) = 10\log_{10}\left[\frac{\displaystyle\sum_{k=1}^{N} e_k^2(n)}{\displaystyle\sum_{k=1}^{N} d_k^2(n)}\right] \tag{49}$$

as the instantaneous relative residual sound level in the whole network. The relative residual sound level (SL) from (49) or (48) can be depicted versus the number of iterations providing the learning curves for each node or for the overall network, respectively.

In order to generate the performance curves, 30 independent experiments were performed and averaged. The adaptive filters to be designed have a length of $L = 100$ coefficients. The step size has been chosen in order to provide the fastest stable convergence speed for each algorithm. For the DFxAP algorithm, the regularization factor is set to $\delta = 10^{-8}$ and a projection order from $N = 1$ up to $N = 6$ is used. Figs. 3-5 show the learning behavior of both algorithms in terms of SL for a WASN with $K = 2$ (Fig. 3), $K = 4$ (Fig. 4) and $K = 8$ (Fig. 5), in the network (Figures 3-5.(a)) and in a specific node ((Figures 3-5.(b)). As it was expected, the proposed algorihm outperforms the DMEFxLMS at the transient state and a similar behavior is exhibited at the different nodes of the network. Moreover, the convergence speed of the DFxAP increases with the projection order at the expense of higher computational requirements, though in this particular scenario for $N > 6$ the convergence speed does not increase.

B. Mean steady-state weight behaviour

The aim of this subsection is to evaluate the mean weight behaviour at steady-state. We have compared the theoretical expression derived in (27) with the averaged value of the filter weights obtained after filter convergence for the simulations carried out. We have used the same examples than in the previous subsection. For the simulations, the value obtained after 500,000 iterations has been considered. For the theoretical results, the expectations involved in evaluating the mean behaviour have been computed by ensemble averaging at every node the matrices $\mathbf{Q}_k(n)$, $\mathbf{A}_k^T(n)$, and vector $\mathbf{d}_k(n)$. Figure 6 illustrates the magnitude spectrum in dB of the simulated weights after filter convergence and their predictions by using (27) in a two-node WASN with $N = 2$ (Figure 6.(a) and (b)) and with $N = 5$ (Figure 6.(c) and (d)). The results show a close match between the theoretical and the simulated values.

C. Mean Square Deviation performance

In the last set of experiments, we demonstrate how the MSD steady-state values of the DFXAP are predicted by the analysis proposed in subsection III-B. Consider two networks with $K = 2$ and 4 nodes, and a global adaptive filter of
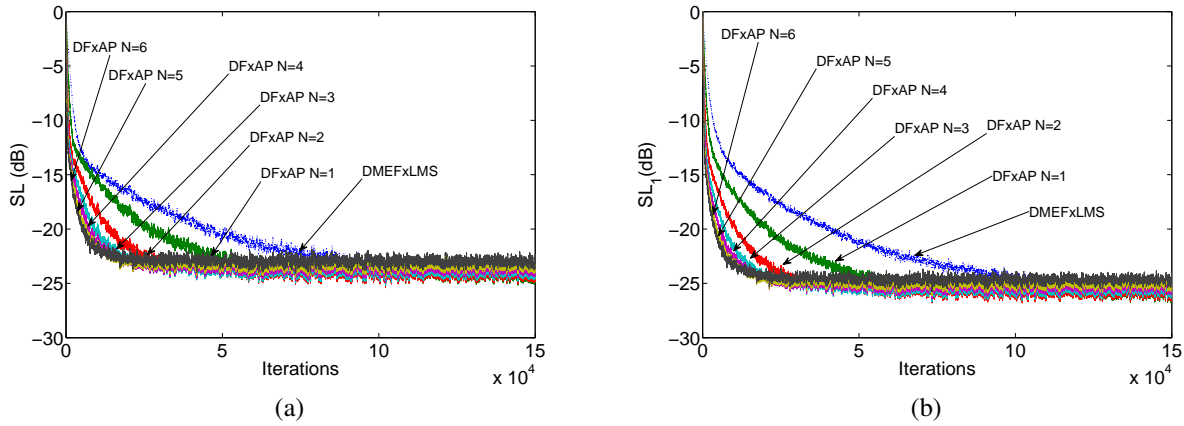
Fig. 3. Relative residual sound level obtained using a two-node WASN for the DFxAP with different projection orders and the DMEFxLMS algorithm. SL in the network (a), and in the first node (b).
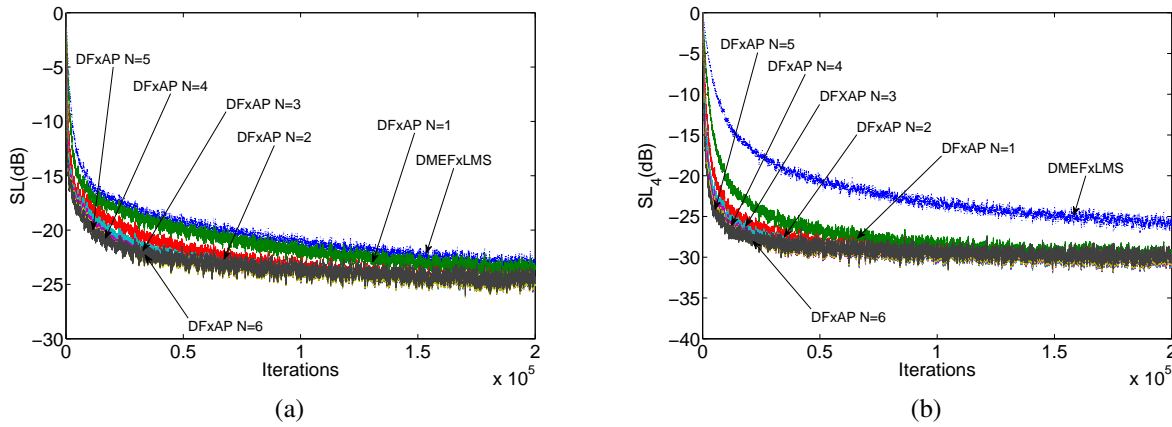

Fig. 4. Relative residual sound level obtained using a four-node WASN for the DFxAP with different projection orders and the DMEFxLMS algorithm. SL in the network (a), and in the fourth node (b).
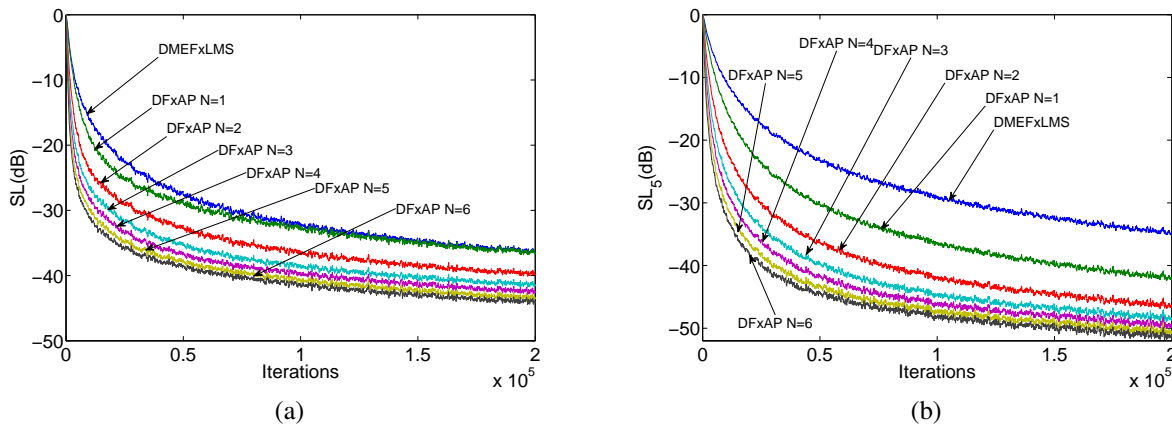

Fig. 5. Relative residual sound level obtained using an eight-node WASN for the DFxAP with different projection orders and the DMEFxLMS algorithm. SL in the network (a), and in the fifth node (b).

$L = 20$ coefficients. For the simulations, the MSD is obtained as the difference between the simulated coefficient vector and the optimal coefficient vector $\mathbf{w_0}$ following expression (29). Furthermore, the undesired signal at every node is generated from (30) where the noise term $\mathbf{r}_k(n)$ is a Gaussian random

signal of zero mean. The simulated curves are obtained by averaging 20 independent runs. The theoretical results are calculated by using expression (45) and its approximated version (47) as well. Note that the last term in both expressions involves the computation of the expectation of matrices
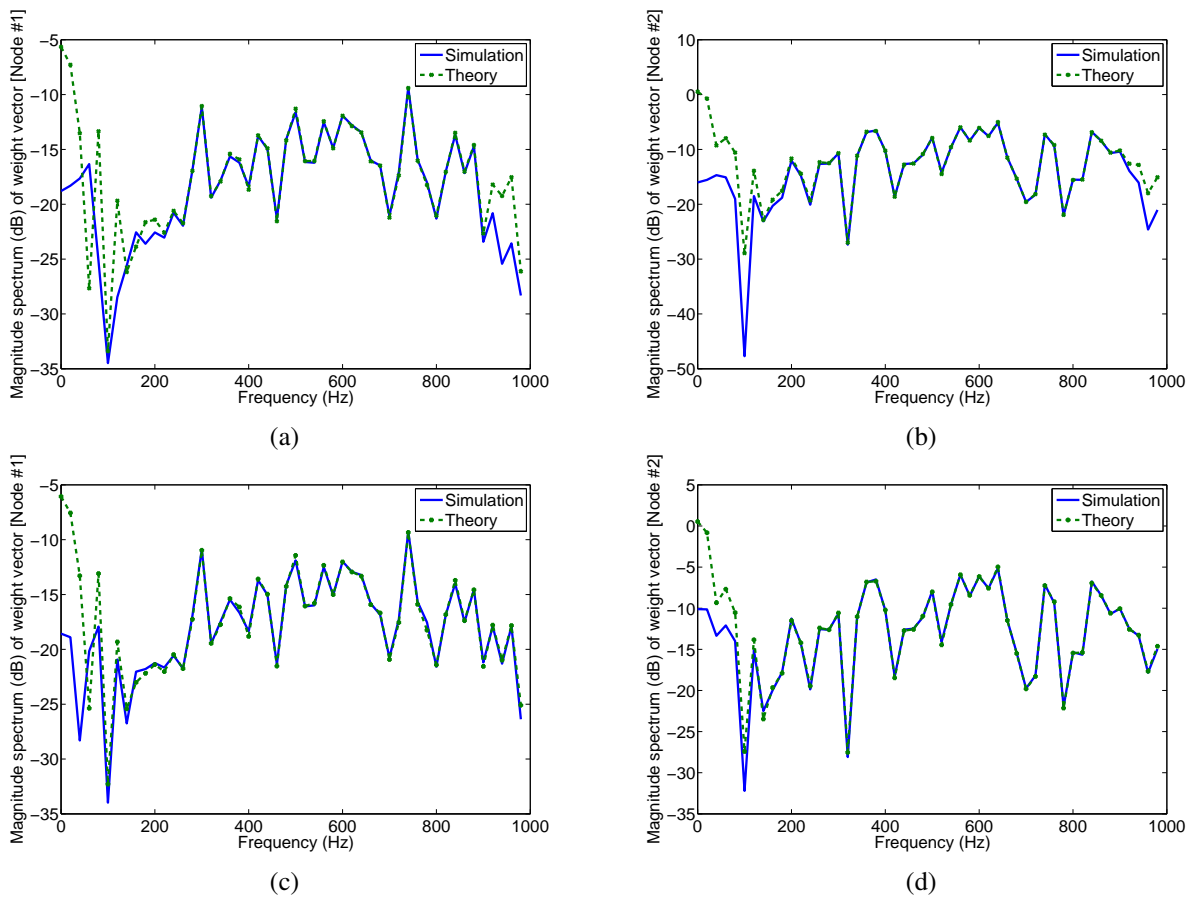
Fig. 6. Magnitude spectrum of the simulated mean steady-state weight values and their predictions using (27) for DFxAP in a two-node WASN with: (a) and (b) $N = 2$, and (c) and (d) $N = 5$.
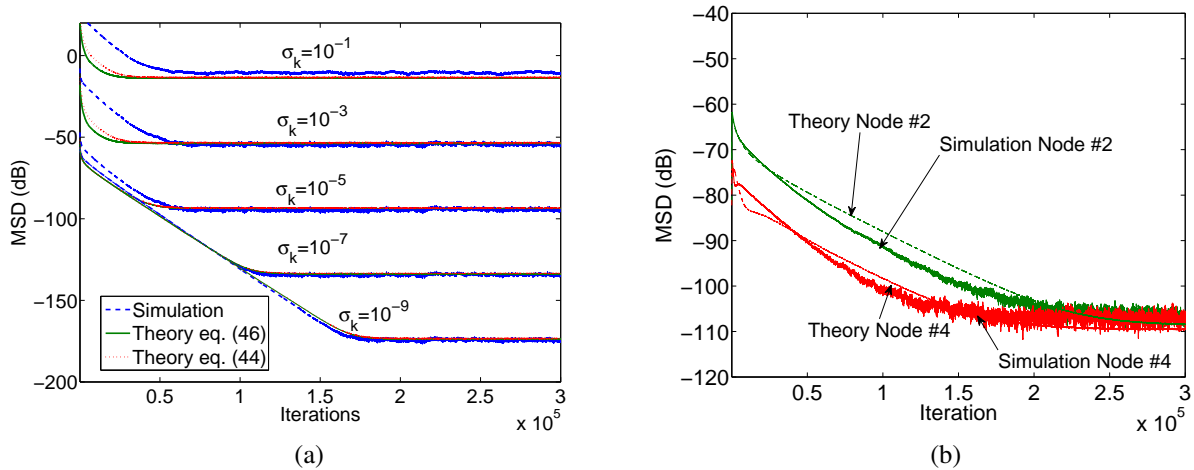

Fig. 7. Simulated MSD values and their predictions for DFxAP in a four-node WASN (a) with $N = 4$ and (b) with $N = 2$ and $\sigma_k = 10^{-6}$ at node 2 and at node 4.

dependent on the reference signal as it can be seen in (46). Figure 7.(a) depicts the MSD curves for different σ_k values using the same value for all the nodes in the network and $N = 4$. The theoretical values are in good agrement with simulated value. Moreover, increasing the value of σ_k leads to a degradation of the prediction. In addition, it has been

analyzed the MSD performance at each node defined as follows

$$\text{MSD}_k(n) = \text{E}\left(\|\widetilde{\mathbf{w}_k}(n)\|^2\right) = \text{E}\left(\widetilde{\mathbf{w}_\mathbf{k}}^T(n)\widetilde{\mathbf{w}_\mathbf{k}}(n)\right), \quad (50)$$

where $\widetilde{\mathbf{w}}_k(n) = \mathbf{w_0} - \mathbf{w}_k(n)$ is the weight-error vector at the kth node. Figure 7.(b) illustrates the comparison between the theoretical and the simulated values of MSD at two different

nodes of a four-node WASN with $N = 2$ and $\sigma_k = 10^{-6}$. In this case only (45) has been used to depict the MSD curves since the results of both models are very similar. Again, the predicted values show a good match with the simulation results in steady state.

V. CONCLUSION

In this paper, we have proposed a distributed affine projection algorithm suitable to be used over acoustically coupled networks with incremental communication among the nodes. Specifically, the distributed filtered-x AP (DFxAP) has been used to control the sound field in an active noise control system. While the DFxAP is an approach of the multichannel AP that distributes the computation among the nodes, we have verified through numerical simulations that it exhibits good convergence speed and low steady-state residual error. It can also be observed that it outperforms the DMEFxLMS algorithm, previously introduced as the distributed version of the centralized MEFxLMS, in terms of convergence speed. An analytical model has been derived for the mean and mean-square deviation behavior of the network weights. Numerical simulations have showed that both models accurately predict the mean and mean-square behavior, even for high values of the noise variance.

We have considered a distributed network without communication constraints and the DFxAP algorithm is derived under this assumption. Indeed, non-ideal scenarios with a network with communication constraints will adversely affect the algorithm performance. However, the performance will worsen similarly to the performance of other collaborative distributed algorithms.

APPENDIX A
DERIVATION OF (31)

Substituting (30) in (27) leads to

$$
E(\mathbf{w}(\infty)) = \lim_{n \to \infty} \left\{ \left[\sum_{p=0}^{K-1} E\left(\mathbf{Q}_{K-p}(n) \mathbf{A}_{K-p}^T(n) \right) \right]^{-1} \right.
$$
$$
\times \left[\sum_{p=0}^{K-1} E\left(\mathbf{Q}_{K-p}(n) \mathbf{A}_{K-p}^T(n) \mathbf{w_0} \right) \right]
$$
$$
- \left[\sum_{p=0}^{K-1} E\left(\mathbf{Q}_{K-p}(n) \mathbf{A}_{K-p}^T(n) \right) \right]^{-1}
$$
$$
\left. \times \left[\sum_{p=0}^{K-1} E\left(\mathbf{Q}_{K-p}(n) \mathbf{r}_{K-p}(n) \right) \right] \right\}. \tag{51}
$$

That can be rewritten as

$$
E(\mathbf{w}(\infty)) = \lim_{n \to \infty} \left\{ \mathbf{w_0} - \left[\sum_{p=0}^{K-1} E\left(\mathbf{Q}_{K-p}(n) \mathbf{A}_{K-p}^T(n) \right) \right]^{-1} \right.
$$
$$
\left. \times \left[\sum_{p=0}^{K-1} E\left(\mathbf{Q}_{K-p}(n) \mathbf{r}_{K-p}(n) \right) \right] \right\}. \tag{52}
$$

Under Assumption A.2), (52) results in (31).

APPENDIX B
DERIVATION OF (43)-(44)

If we assume that $A.4$) $r_k(n)$ *is uncorrelated with* $\widetilde{\mathbf{w}}^K(n)$, and considering $A.3$), thus the second and third term on the right hand side of (42) are zero. Regarding the first term of (42), we consider the conditioned expectation instead of only the expectation. To do this, we define the conditioned covariance matrices by using the covariance matrices in (41) as in equation (13) of [34]:

$$
\begin{bmatrix} \mathbf{R}(n) & \mathbf{S}(n) \\ \mathbf{S}^T(n) & \mathbf{P}(n) \end{bmatrix}
$$
$$
= E\left\{ E\left(\begin{bmatrix} \mathbf{r}_K(n+1) \\ \widetilde{\mathbf{w}}^K(n) \end{bmatrix} \begin{bmatrix} \mathbf{r}_K^T(n+1) & \widetilde{\mathbf{w}}^K(n)^T \end{bmatrix} \right) \right.
$$
$$
\left. |\chi(n-1)) \right\}, \tag{53}
$$

for a given set $\chi(n-1) = \{\mathbf{x}(m) | 0 \le m \le n-1\}$. Then

$$
E\left\{ E\left(\begin{bmatrix} \mathbf{Z} & \mathbf{0} \\ \mathbf{\Psi}_K(n) & \mathbf{\Phi}_{KK}(n) \end{bmatrix} \begin{bmatrix} \mathbf{r}_K(n) \\ \widetilde{\mathbf{w}}^K(n-1) \end{bmatrix} \right. \right.
$$
$$
\times \begin{bmatrix} \mathbf{r}_K^T(n) & \widetilde{\mathbf{w}}^K(n-1)^T \end{bmatrix} \begin{bmatrix} \mathbf{Z}^T & \mathbf{\Psi}_K^T(n) \\ \mathbf{0}^T & \mathbf{\Phi}_{KK}^T(n) \end{bmatrix}
$$
$$
|\chi(n-1)) \}
$$
$$
= \begin{bmatrix} \mathbf{Z} & \mathbf{0} \\ \mathbf{\Psi}_K(n) & \mathbf{\Phi}_{KK}(n) \end{bmatrix} E\left\{ E\left(\begin{bmatrix} \mathbf{r}_K(n) \\ \widetilde{\mathbf{w}}^K(n-1) \end{bmatrix} \right. \right.
$$
$$
\times \begin{bmatrix} \mathbf{r}_K^T(n) & \widetilde{\mathbf{w}}^K(n-1)^T \end{bmatrix} |\chi(n-1)) \}
$$
$$
\times \begin{bmatrix} \mathbf{Z}^T & \mathbf{\Psi}_K^T(n) \\ \mathbf{0}^T & \mathbf{\Phi}_{KK}^T(n) \end{bmatrix}
$$
$$
= \begin{bmatrix} \mathbf{Z} & \mathbf{0} \\ \mathbf{\Psi}_K(n) & \mathbf{\Phi}_{KK}(n) \end{bmatrix} \begin{bmatrix} \mathbf{R}(n-1) & \mathbf{S}(n-1) \\ \mathbf{S}^T(n-1) & \mathbf{P}(n-1) \end{bmatrix}
$$
$$
\times \begin{bmatrix} \mathbf{Z}^T & \mathbf{\Psi}_K^T(n) \\ \mathbf{0} & \mathbf{\Phi}_{KK}^T(n) \end{bmatrix}
$$
$$
= \begin{bmatrix} \mathbf{Z}\mathbf{R}(n-1)\mathbf{Z}^T & \mathbf{Z}\mathbf{R}(n-1)\mathbf{\Psi}_K^T(n) \\ & +\mathbf{Z}\mathbf{S}(n-1)\mathbf{\Phi}_{KK}(n)^T \\ & \mathbf{\Psi}_K(n)\mathbf{R}(n-1)\mathbf{\Psi}_K^T(n) \\ \mathbf{\Psi}_K(n)\mathbf{R}(n-1)\mathbf{Z}^T & +\mathbf{\Phi}_{KK}(n)\mathbf{S}^T(n-1)\mathbf{\Psi}_K^T(n) \\ +\mathbf{\Phi}_{KK}(n)\mathbf{S}^T(n-1)\mathbf{Z}^T & +\mathbf{\Psi}_K(n)\mathbf{S}(n-1)\mathbf{\Phi}_{KK}^T(n) \\ & +\mathbf{\Phi}_{KK}(n)\mathbf{K}(n-1)\mathbf{\Phi}_{KK}^T(n) \end{bmatrix}. \tag{54}
$$

On the other hand, the fourth term in (42) is given by

$$
E\left(\begin{bmatrix} \mathbf{r}'_K(n+1) \\ \mathbf{\Phi}'_K(n) \end{bmatrix} \begin{bmatrix} \mathbf{r}'_K(n+1) \\ \mathbf{\Phi}'_K(n) \end{bmatrix}^T \right)
$$
$$
= E\left(\begin{bmatrix} \mathbf{r}'_K(n+1)\mathbf{r}'_K(n+1)^T & \mathbf{r}'_K(n+1)\mathbf{\Phi}'_K(n)^T \\ \mathbf{\Phi}'_K(n)\mathbf{r}'_K(n+1)^T & \mathbf{\Phi}'_K(n)\mathbf{\Phi}'_K(n)^T \end{bmatrix} \right). \tag{55}
$$

The first element in (55) becomes

$$
E\left(\mathbf{r}'_K(n+1)\mathbf{r}'_K(n+1)^T \right)
$$
$$
= E\left(r_K^2(n+1) \right) \mathbf{a}_1 \mathbf{a}_1^T = \sigma_K^2 \mathbf{a}_1 \mathbf{a}_1^T, \tag{56}
$$

where $\mathbf{a}_1$ is an $N \times 1$ vector whose first element is 1 and the others are 0's.

Assuming that $A.5)$ $r_k(n)$ *is uncorrelated with* $r_p(n)$ *for* $p \neq k$ and using $A.3)$ into the fourth term in (42), we get

$$
\mathrm{E}\left(\mathbf{\Phi}'_K(n)\mathbf{r}'_K(n+1)^T\right)
$$
$$
= \mathrm{E}\left(\sum_{k=1}^{K-1} \mathbf{\Phi}_{Kk}(n)\mathbf{r}_{K-k}(n)\mathbf{r}'_K(n+1)^T\right) = 0. \tag{57}
$$

After some manipulations and applying $A.3)$ in the second diagonal term in (55), we get

$$
\mathrm{E}\left(\mathbf{\Phi}'_K(n)\mathbf{\Phi}'_K(n)^T\right)
$$
$$
= \mathrm{E}\left(\sum_{k=1}^{K-1}\mathbf{\Phi}_{Kk}(n)\mathbf{r}_{K-k}(n)\sum_{l=1}^{K-1}\mathbf{r}_{K-l}^T(n)\mathbf{\Phi}_{Kl}^T(n)\right) \tag{58}
$$
$$
= \sum_{k=1}^{K-1}\sigma_{K-k}^2\mathrm{E}\left(\mathbf{\Phi}_{Kk}(n)\mathbf{\Phi}_{Kk}^T(n)\right) = \mathbf{\Omega}_K(n),
$$

being $\mathbf{\Omega}_K(n)$ a $KL \times KL$ matrix. Then, (55) can be rewritten as

$$
\mathrm{E}\left(\begin{bmatrix} \mathbf{r}'_K(n+1) \\ \mathbf{\Phi}'_K(n) \end{bmatrix}\begin{bmatrix} \mathbf{r}'_K(n+1)^T & \mathbf{\Phi}'_K(n)^T \end{bmatrix}\right)
$$
$$
= \begin{bmatrix} \sigma_K^2\mathbf{a}_1\mathbf{a}_1^T & \mathbf{0} \\ \mathbf{0} & \mathbf{\Omega}_K(n) \end{bmatrix}. \tag{59}
$$

Finally, from (54) and (59) we can obtain expressions from (43) to (45).

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] X. Xiaohong Sheng and Y.-H. Yu-Hen Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 53, no. 1, pp. 44–53, 2005.

[3] M. Cobos, J. J. Perez-Solano, S. Felici-Castell, J. Segura, and J. M. Navarro, "Cumulative-Sum-Based Localization of Sound Events in Low-Cost Wireless Acoustic Sensor Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1792–1802, 2014.

[4] Y. Dorfan and S. Gannot, "Tree-Based Recursive Expectation-Maximization Algorithm for Localization of Acoustic Sources," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 10, pp. 1692–1703, 2015.

[5] L. Wang, T.-K. Hon, J. D. Reiss, and A. Cavallaro, "Self-Localization of Ad-Hoc Arrays Using Time Difference of Arrivals," *IEEE Transactions on Signal Processing*, vol. 64, no. 4, pp. 1018–1033, 2016.

[6] S. Markovich-Golan, S. Gannot, and I. Cohen, "Distributed Multiple Constraints Generalized Sidelobe Canceler for Fully Connected Wireless Acoustic Sensor Networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 2, pp. 343–356, 2013.

[7] S. Markovich-Golan, A. Bertrand, M. Moonen, and S. Gannot, "Optimal distributed minimum-variance beamforming approaches for speech enhancement in wireless acoustic sensor networks," *Signal Processing*, vol. 107, pp. 4–20, 2015.

[8] A. Hassani, A. Bertrand, and M. Moonen, "Gevd-based low-rank approximation for distributed adaptive node-specific signal estimation in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 64, no. 10, pp. 2557–2572, May 2016.

[9] M. Ferrer, M. de Diego, G. Piñero, and A. Gonzalez, "Active noise control over adaptive distributed networks," *Signal Processing*, vol. 107, pp. 82–95, feb 2015.

[10] C. Antoñanzas, M. Ferrer, M. de Diego, and A. Gonzalez, "Blockwise Frequency Domain Active Noise Controller Over Distributed Networks," *Applied Sciences*, vol. 6, no. 5, p. 124, 2016.

[11] T. Betlehem, W. Zhang, M. A. Poletti, and T. D. Abhayapala, "Personal sound zones: Delivering interface-free audio to multiple listeners," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 81–91, March 2015.

[12] G. Piñero, J. Estreder, F. Martinez-Zaldivar, M. De Diego, and M. Ferrer, "Reshaping of room impulse responses over wireless acoustic networks," in *2016 AES Int. Conf. on Sound Field Control*, Jul 2016. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=18320

[13] H. Kim, S. Lee, J.-W. J.-w. Choi, H. Bae, J. Lee, J. Song, and I. Shin, "Mobile Maestro: Enabling Immersive Multi-speaker Audio Applications on Commodity Mobile Devices," in *Proc. ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing (UbiComp '14)*, 2014, pp. 277–288.

[14] S. Chouvardas, M. Muma, K. Hamaidi, S. Theodoridis, and A. Zoubir, "Distributed robust labeling of audio sources in heterogeneous wireless sensor networks," in *2015 IEEE Int. Conf. on Acoustics, Speech and Signal Proc. (ICASSP)*, 2015, pp. 5783–5787.

[15] D. Haddad, W. Martins, M. da Costa, L. Biscainho, L. Nunes, and B. Lee, "Robust Acoustic Self-Localization of Mobile Devices," *IEEE Trans. Mobile Computing*, vol. 15, no. 4, pp. 982–995, 2016.

[16] S. J. Elliott and C. C. Boucher, "Interaction between multiple feedforward active control systems," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 521–530, Oct 1994.

[17] N. George and G. Panda, "A particle-swarm-optimization-based decentralized nonlinear active noise control system," *Instrumentation and Measurement, IEEE Transactions on*, vol. 61, no. 12, pp. 3378–3386, 2012.

[18] C. Lopes and A. Sayed, "Incremental adaptive strategies over distributed networks," *Signal Processing, IEEE Transactions on*, vol. 55, no. 8, pp. 4064–4077, 2007.

[19] F. S. Cattivelli and A. H. Sayed, "Analysis of spatial and incremental LMS processing for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1465–1480, April 2011.

[20] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Proc. of the Electron. and Communic. in Japan*, vol. J67-A, pp. 126–132, Feb. 1984.

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

[22] A. Carini and G. Sicuranza, "Transient and steady-state analysis of filtered-x affine projection algorithms," *Signal Processing, IEEE Transactions on*, vol. 54, no. 2, pp. 665–678, Feb 2006.

[23] M. Ferrer, A. Gonzalez, M. de Diego, and G.Piñero, "Fast affine projection algorithms for filtered-x multichannel active noise control," *IEEE Trans. on Audio, Speech and Language Proces.*, vol. 16, no. 8, pp. 1396–1408, November 2008.

[24] M. Ferrer, A. Gonzalez, M. de Diego, and G. Piñero, "Transient analysis of the conventional filtered-x affine projection algorithm for active noise control," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 3, pp. 652–657, March 2011.

[25] M. Ferrer, M. de Diego, A. Gonzalez, and G. Piñero, "Steady-state mean square performance of the multichannel filtered-x affine projection algorithm," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2771–2785, June 2012.

[26] M. Bouchard, "Multichannel affine and fast affine projection algorithms for active noise control and acoustic equalization systems," *IEEE Trans. on Speech and Audio Proces.*, vol. 11, no. 1, pp. 54–60, Enero 2003.

[27] J. Benesty, P. Duhamel, and Y. Grenier, "A multichannel affine projection algorithm with applications to multichannel acoustic echo cancellation," *IEEE Signal Processing Letters*, vol. 3, no. 2, pp. 35–37, Feb 1996.

[28] L. Li, J. A. Chambers, C. G. Lopes, and A. H. Sayed, "Distributed estimation over an adaptive incremental network based on the affine projection algorithm," *IEEE Transactions on Signal Processing*, vol. 58, no. 1, pp. 151–164, Jan 2010.

[29] M. S. E. Abadi and Z. Saffari, "Distributed estimation over an adaptive diffusion network based on the family of affine projection algorithms," in *Telecommunications (IST), 2012 Sixth International Symposium on*, Nov 2012, pp. 607–611.

[30] A. Khalili, A. Rastegarnia, and W. M. Bazzi, "Incremental augmented affine projection algorithm for collaborative processing of complex signals," in *Information and Communication Technology Research (IC-TRC), 2015 International Conference on*, May 2015, pp. 60–63.

[31] A. H. Sayed, *Adaptive Filters*.   NJ: John Wiley & Sons, 2008.

[32] S. Haykin, *Adaptive Filter Theory*, 4th ed.   Prentice-Hall, Ed., Upper Saddle River, NJ, 2002.

[33] P. Park, C. H. Lee, and J. W. Ko, "Mean-square deviation analysis of affine projection algorithm," *Signal Processing, IEEE Transactions on*, vol. 59, no. 12, pp. 5789–5799, Dec 2011.

[34] A. Papoulis, *Probability, Random Variables, and Stochastic Processes.*, 4th ed.   New York: McGraw-Hill, 1991.

Miguel Ferrer received the Graduate degree in telecommunications engineering from the Universitat Politècnica de València (UPV), Valencia, Spain, in 2000. 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 Investigacion Aplicada al Automovil, Tarragona, Spain (Automobile Applied Research Institute). Subsequently, he was awarded several grants offered both by the Communications Department, 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 more than 60 papers related with signal processing in renowned journals and conferences. Since 2005, he has been an Assistant Lecturer in the Communications Department, UPV. His research interests include the study of adaptive algorithms and its application to audio digital processing and active noise control, a subject about which he developed his doctoral thesis.

Alberto Gonzalez received the Graduate degree (highest Hons.) in telecommunication engineering from the Universitat Politecnica Catalunya, Barcelona, Spain, and the Ph.D. degree (magna cum laude) from Universitat Politècnica de València (UPV), Valencia, Spain, in 1997.

During 1995, Professor Gonzalez worked as a Visiting Researcher at the Institute of Sound and Vibration Research, University of Southampton, Southampton, U.K., and is currently heading a research group in audio and communications digital signal processing. He has published more than 100 papers in international technical journals and renowned conferences in the fields of signal processing and applied acoustics and serves as the Dean of the Telecommunication Engineering School since June 2012. His research interests include optimization of computation methods for detection and decoding in digital communications and distributed sound signal processing. Professor Gonzalez belongs to the EURASIP Special Area Team on Acoustic, Speech and Music Signal Processing.

Maria de Diego (S'99-M'00-SM'14) received the M.Sc. degree in telecommunications engineering and the Ph.D degree from the Universitat Politècnica de València, Valencia, Spain, in 1994 and 2003, respectively. Her dissertation was on active noise conformation of enclosed acoustic fields. She is currently working as an Associate Professor in digital signal processing and communications and a Researcher with the GTAC Group, Institute of Telecommunications and Multimedia Applications, Universitat Politècnica de València. She 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 research interests include smart sound processing for reproduction and control, and adaptive filtering for distributed sensors in audio applications. She is a Senior Member of the Institute of Electrical and Electronic Engineers (IEEE) and an active committee Member of the International Institute of Acoustics and Vibration where she currently serves as a Director.

Gema Piñero received the M.Sc. degree from the Polytechnic University of Madrid, Spain, in 1990, and the Ph.D. degree from the Polytechnic University of Valencia, Spain, in 1997, both in telecommunication engineering. She is currently an Associate Professor in the Department of Communications of the Polytechnic University of Valencia and a researcher within the Audio and Communications Signal Processing group (GTAC) of the Institute of Telecommunications and Multimedia Applications (iTEAM) of the same university. Since 1990 she has participated in several research projects in the areas of array signal processing, multi-channel sound systems, wireless communications and sound quality analysis. She has leaded two research projects on sound quality analysis for the automotive industry, and has participated in several projects on wireless communications and sound processing applications supported by public and private funding (Spanish Government, Regional Government, Telefonica, NVIDIA, AIJU). She has published more than 90 contributions in international journals and conferences. She has enjoyed two scholarships as a visiting professor at the University of Illinois at Urbana-Champaign (2011) and the Imperial College London (2016). Her current research interests include distributed algorithms for wireless acoustic sensor networks and multi-user techniques in coordinated communication systems. She is currently a member of the editorial board of the Digital Signal Processing Journal. She is also a senior member of the IEEE and a founding member of the Spanish Association for Research and Teaching UNIVERSITAS.