# Active Noise Control Over Adaptive Distributed Networks

M. Ferrer, M. de Diego, G. Piñero*, A. Gonzalez

*Inst. Telecommunications and Multimedia Applications*
*Universitat Politecnica de Valencia*
*Camino de Vera s/n; 46022 Valencia, Spain*

**Abstract**

This paper presents the implementation of Active Noise Control (ANC) systems over a network of distributed acoustic nodes. For this purpose we define a general acoustic node consisting on one or several microphones and one or several loudspeakers together with a unique processor with communication capabilities. ANC systems can use a wide range of adaptive algorithms, but we have considered specifically the Multiple Error Filtered-x Least Mean Square (MEFxLMS), which has been proved to perform very well for ANC systems with multiple microphones and loudspeakers, and centralized processing. We present a new formulation to introduce the distributed version of the MEFxLMS together with an incremental collaborative strategy in the network. We demonstrate that the distributed MEFxLMS exhibits the same performance as the centralized one when there are no communication constraints in the network. Then, we re-formulate the distributed MEFxLMS to include parameters related to its implementation on an acoustic sensor network: latency of the network, computational capacity of the nodes, and trustworthiness of the signals measured at each node. Simulation results in realistic scenarios show the ability of the proposed distributed algorithms to achieve good performance when proper

*Corresponding author
*Email addresses:* mferrer@dcom.upv.es (M. Ferrer), mdediego@dcom.upv.es (M. de Diego), gpinyero@iteam.upv.es (G. Piñero), agonzal@dcom.upv.es (A. Gonzalez)

values of these parameters are chosen.

*Keywords:* Active noise control, Distributed networks, Acoustic control, Adaptive filters, Acoustic sensor networks.

## 1. Introduction

It has been more than a decade since the wireless sensor networks (WSN) were considered as a cheap, flexible and efficient solution for environmental and habitat monitoring, as well as for monitoring and maintenance of industrial equipment [1–3]. From the very first moment different acoustic applications were proposed [4, 5], which paved the way for the specific wireless acoustic sensor networks (WASN) whose sensor devices are microphones. These microphones are usually connected to a processor with some kind of communication and computation capability [6]. Applications that make use of these kind of acoustic nodes are numerous, see [7] and references therein, but they focus on the estimation of a common signal or parameter that can be measured by all the nodes [8], or on the estimation of node-specific signals sharing some common properties or parameters [9, 10]. Another typical feature of a node relates to its configuration: the acoustic node is usually composed of a microphone plus a processor, where the processing unit is dedicated to recording, control and transmission tasks, and can eventually perform some signal processing algorithms before the transmission. However, for applications involving sound control in general, and particularly for active noise control (ANC) systems, this typical node structure needs to be modified in two aspects. First, the node should have the capacity of acting on the environment to control the sound rendering, that is, the node should be able to emit sounds through a loudspeaker or actuator. Secondly, the network should focus not only on the estimation of a particular signal or some related parameter, but on the design of the signals that will feed the loudspeakers and will control the sound field. To our knowledge, no WASN has been proposed where nodes have the capacity of control and modify their own environment.

Therefore, we will consider a generic acoustic node as a node with a certain computation capability to process signals, that can communicate to other nodes to exchange local and network status information, and which is also able to act on its own environment. The node can record signals through one or more microphones (sensors) and can emit sound signals via one or more loudspeakers (actuators). Moreover, nodes should make use of the network topology to process their own signals in a proper way. Some common topologies are: the total diffusion networks, where all nodes are interconnected with the rest of the nodes; the mesh networks, where each node can communicate with a certain set of nodes; the tree networks, where communication between nodes is hierarchical; and the ring networks, where communication between nodes follows an incremental ordering along the network [11].

The specific application described in this paper is an active noise canceller or active noise control (ANC) system [12]. ANC systems try to reduce some unwanted noise by the addition of one or several secondary sounds specifically designed to cancel the first. In particular the system is intended to reduce the unwanted, also called primary, noise at the microphones' location. Fig. B.1 shows an ANC system with $K$ microphones and $J$ loudspeakers. The signals recorded at the microphones are called *error signals* and denoted $e_k(n)$, the loudspeakers, called *secondary sources*, emit the filter output signals $y_j(n)$, and the acoustic channel impulse response between loudspeaker $j$ and microphone $k$ is modelled as a FIR filter. The unwanted noise is not depicted in Fig. B.1, but the *reference signals* $x_i(n)$ entering the *multichannel adaptive controller* are correlated with it, and they will be used by the adaptive controller to appropriately design the output signals $y_j(n)$.

The algorithmic approach proposed in this paper is based on well known multichannel adaptive filters originally stated for a centralized system [12, 13], where all signals $e_k(n)$ and $x_i(n)$ are available at the multichannel controller. Particularly, we have implemented the Multiple Error Filtered-x Least Mean Square (MEFxLMS) algorithm [14] over a distributed network, which in turn is based on the commonly used Least Mean Square (LMS) algorithm [15]. Regarding previous works on the implementation of the LMS algorithm on distributed systems, the first thorough studies were presented in [8, 16], where distributed LMS is used on networks connected by incremental and diffusion strategies respectively. The authors show that the distributed LMS approach achieves good performance allied with low communication and computational requirements for linear estimation tasks. Although the algorithm proposed herein is based on distributed LMS-type algorithms [8, 17], it has been extended to consider that our acoustic nodes do not only sense the environment but also modify it through its own actuators.

In the particular case of sound control systems, a distributed ANC system was first introduced in [18]. This system, called decentralized since their processors did not collaborate or interchange any local information, was based on a filtered-x scheme of the LMS algorithm [19]. Its main advantage is the scalability and the ability of distributing the computational burden, but it cannot overcome the centralized system performance except for uncoupled actuators and microphones. A similar decentralized ANC system is considered in [20] using adaptive non-linear filters. In both decentralized systems [18, 20], each error signal $e_k(n)$ is only used by the corresponding processor, whereas reference signal $x(n)$ is common to all of them. For the herein proposed network of acoustic nodes, we consider a common reference signal is available.

Consequently, the cooperation provided by a WASN would help ANC systems to achieve similar performance to centralized solutions, whereas they would also benefit from the advantages of distributed systems as scalability and low computational cost. The main contributions of the work herein presented can be summarized as:

- The MEFxLMS algorithm is formulated for WASN's as the distributed MEFxLMS (DMEFxLMS) where the calculation of the adaptive filters is carried out in a distributed way over a ring topology with incremental communication [8]. The computational burden is then shared among all the processors.

- We have extended the DMEFxLMS to a network whose communication is affected by a constant latency. To deal with this latency, the DMEFxLMS has been re-formulated introducing two new parameters: the first one acts in the meantime between two network information arrivals, deciding if the node adapts itself based on its local measurement or just waits for the new network information. The second parameter only acts when the network information arrives at each node, providing different combinations of both network and local information.

- We have carried out a set of simulations using real acoustic channel responses in order to evaluate the influence of the new defined parameters on the ANC system performance. We propose proper values of both parameters depending on the network latency, the node computational capacity, and the level of coupling of the acoustic channels.

The paper is organized as follows: in Section 2 we develop the distributed MEFxLMS algorithm for WASN's without communication constraints. In Section 3 a constant latency in the network is considered and a re-formulation of the distributed MEFxLMS is provided and discussed. Simulated results for unconstrained and constrained communication networks are shown in Section 4 including a discussion on the parameters affecting the algorithm behaviour. Finally Section 5 outlines the main conclusions of the present work.

**Notation:** The following notation is used throughout the paper: 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 $A$). Superscript $(\cdot)^T$ stands for matrix or vector transpose. The expression $[\mathbf{A}]_{(i:j,l:k)}$ stands for a new matrix formed by selecting the rows $i$ to $j$ and columns $l$ to $k$ of $\mathbf{A}$. If all the rows or all the columns are selected, then the subscript changes to $[\mathbf{A}]_{(:,l:k)}$ or $[\mathbf{A}]_{(i:j,:)}$ respectively. $\mathbf{0}_{I\times J}$ is an all-zero matrix of $I$ rows and $J$ columns.

## 2. Distributed adaptive algorithms for active noise control

In this section we will develop the MEFxLMS adaptive algorithm for distributed active noise control over a WASN without communication restrictions. Regarding the communication within the network, we consider an incremental collaborative strategy where each node transmits information to an adjacent node in a consecutive order. For the sake of clarity, we consider a distributed network of single channel acoustic nodes in a homogeneous network. That means that all the nodes are equipped with a single microphone and a single loudspeaker as depicted in Fig. B.2, and they have the same computational capacity and run the same algorithm. In Appendix A we will extend the obtained distributed algorithm to a network whose nodes can handle different number of microphones and loudspeakers. Examples of commercial devices that could be used as nodes for ANC applications are tablets, smartphones, notebooks, hearing aids, etc.

### *2.1. Centralized MEFxLMS algorithm for ANC*

Consider a generic multichannel control system comprised of $I$ reference signals, $J$ secondary sources and $K$ error sensors as the one depicted in Fig. B.1, which is devoted to minimize a function of the measures at the error sensors. This function is usually called *cost function* and is related to the acoustic field in the controlled zone by

$$C(n) = \sum_{k=1}^{K} f[e_k(n)] \ , \qquad (1)$$

where $f[\cdot]$ is a time-invariant function of its argument and $e_k(n)$ is the error signal recorded at the $k$th microphone.

Although there are many centralized adaptive strategies that the system of Fig. B.1 can use, as the LMS, the Affine Projection (AP) [21] or the Recursive Least Squares (RLS) [13] algorithms, we consider in this work the most commonly used algorithm for ANC applications, the multiple error filtered-x LMS (MEFxLMS) [14, 22] based on the LMS strategy. There are other adaptive algorithms usually applied in multichannel ANC scenarios that could also be considered in the developed distributed strategies. Some examples are the Least Maximum Mean Squares (LMMS) [23] and the scanning error LMS [24]. Regarding the MEFxLMS algorithm, it is devoted to minimize the sum of the squares of the measured signals at the $K$ error sensors. Thus the corresponding cost function is derived from (1) and given by

$$C(n) = \sum_{k=1}^{K} e_k^2(n)\,, \tag{2}$$

where the number of sensors is such that $K \geq J$.

The MEFxLMS algorithm needs to know all the acoustic channel responses since it is based on the filtered-x scheme [19]. Therefore, the acoustic channels that link each secondary source with each error sensor must be estimated in a previous stage. A specific multichannel ANC system based on the MEFxLMS algorithm is illustrated in Fig. B.3 where the estimated acoustic channel between the $j$th source and the $k$th sensor is denoted by $\hat{\mathbf{h}}_{jk}$, and $\mathbf{w}_{ij}$ stands for the adaptive filter that links the reference signal $x_i(n)$ with the $j$th secondary source. The centralized controller recursively computes a solution for the $IJ$ adaptive filters as follows

$$\mathbf{w}_{ij}(n) = \mathbf{w}_{ij}(n-1) - \mu \sum_{k=1}^{K} \mathbf{v}_{ijk}(n) e_k(n)\,, \tag{3}$$

where the $[L \times 1]$ vector $\mathbf{w}_{ij}(n)$ is used in Fig. B.3 to filter the $i$th reference signal and obtain the corresponding signal contribution to secondary source $y_j(n)$. Constant $\mu$ is the step-size parameter and $\mathbf{v}_{ijk}(n)$ denotes a $[L \times 1]$ vector obtained by filtering the $i$th reference signal $x_i(n)$ with the $M$-length estimated acoustic channel $\hat{\mathbf{h}}_{jk}$:

$$\mathbf{v}_{ijk}(n) = \mathbf{X}_i(n)\hat{\mathbf{h}}_{jk}\,, \tag{4}$$

where $\mathbf{X}_i(n)$ is a circularly arranged matrix of the last $M+L$ samples of $x_i(n)$:

$$\mathbf{X}_i(n) = \begin{bmatrix} x_i(n) & x_i(n-1) & \cdots & x_i(n-M+1) \\ x_i(n-1) & x_i(n-2) & \cdots & x_i(n-M+2) \\ \vdots & \vdots & \cdots & \vdots \\ x_i(n-L+1) & x_i(n-L+2) & \cdots & x_i(n-(L+M)+2) \end{bmatrix}. \tag{5}$$

Once the filter is calculated in (3), the $j$th output signal that feeds the correspondent actuator is obtained as

$$y_j(n) = \sum_{i=1}^{I} \mathbf{w}_{ij}^T(n) \left[\mathbf{X}_i(n)\right]_{(:,1)} , \tag{6}$$

where $[\mathbf{X}_i(n)]_{(:,1)}$ is the $[L \times 1]$ vector formed by the first column of $\mathbf{X}_i(n)$.

It should be noted that in (3) all the error signals $e_k(n)$ are necessary for the computation of each filter $\mathbf{w}_{ij}(n)$, hence the requirement of a centralized processor. In the following section we will discuss how to implement the MEFxLMS algorithm stated in (3)-(6) in a network of distributed wireless acoustic sensors.

### 2.2. Distributed MEFxLMS (DMEFxLMS) algorithm for ANC

Consider now a WASN of $N$ single-channel nodes that will support an ANC system composed by $N$ error sensors and $N$ secondary sources. The objective of each node is to obtain its own adaptive filters such that they approach the minimization of (2) but relying only on local data and some proper network information, and distributing the computational burden among the different nodes. For this purpose we introduce the distributed version of the adaptive algorithm stated in (3)-(6), which we call distributed MEFxLMS (DMEFxLMS).

Let us define a global $[ILN \times 1]$ filter vector $\mathbf{w}(n)$ as the ordered concatenation of all the filter vectors implemented at each node

$$\mathbf{w}(n) = \left[\mathbf{w}_1^T(n), \mathbf{w}_2^T(n), \ldots, \mathbf{w}_N^T(n)\right]^T , \tag{7}$$

where $\mathbf{w}_k(n) = \left[\mathbf{w}_{1k}^T(n), \mathbf{w}_{2k}^T(n), \ldots, \mathbf{w}_{Ik}^T(n)\right]^T$ contains the $IL$ filter coefficients that will be used at node $k$, and $\mathbf{w}_{ik}(n)$ was introduced in (3). Consequently we define the $[ILN \times 1]$ vector $\mathbf{v}_k(n)$ similarly to $\mathbf{w}(n)$ in (7) as

$$\mathbf{v}_k(n) = \left[\mathbf{v}_{1k}^T(n), \mathbf{v}_{2k}^T(n), \ldots, \mathbf{v}_{Nk}^T(n)\right]^T , \tag{8}$$

where $\mathbf{v}_{jk}(n)$ denotes a $[IL \times 1]$ vector obtained as

$$\mathbf{v}_{jk}(n) = \begin{bmatrix} \mathbf{v}_{1jk}(n) \\ \mathbf{v}_{2jk}(n) \\ \vdots \\ \mathbf{v}_{Ijk}(n) \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1(n) \\ \mathbf{X}_2(n) \\ \vdots \\ \mathbf{X}_I(n) \end{bmatrix} \hat{\mathbf{h}}_{jk} = \mathbf{X}(n)\hat{\mathbf{h}}_{jk} , \tag{9}$$

following the definition of $\mathbf{v}_{ijk}(n)$ given in (4). Matrix $\mathbf{X}(n)$ is the vertical concatenation of matrices $\mathbf{X}_i(n)$ defined in (4) which contain the last $L+M$ samples of all the reference signals $x_i(n)$ properly arranged to perform the filtering.

Once the previous notation is stated, the filter updating equation in (3) can be extended to the whole network as

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \mu \sum_{k=1}^{N} \mathbf{v}_k(n) e_k(n) \,. \tag{10}$$

In distributed networks only local data, $e_k(n)$ and $\hat{\mathbf{h}}_{jk}$ for $j = 1, \ldots, N$, are available at each node, thus the $k$th term in the sum of (10) can only be calculated by the $k$th node. The error signal $e_k(n)$ is picked up at the microphone of the $k$th node, and the secondary paths that link all the loudspeakers with the $k$th microphone, $\hat{\mathbf{h}}_{jk}$ for $j = 1, \ldots, N$, can be estimated in a set-up stage. Regarding the reference signals, $x_i(n)$, needed to calculate (5) and (9), we will assume they arrive to the nodes through a different channel from that used by the network. For example, in Fig. B.2 the reference signals are transmitted through a broadcast radio channel to the WASN. In other cases, as for narrowband noise whose fundamental frequency is known, reference signals correlated with the noise could be self-generated by each node.

To deal with a distributed processing, let us remember that the local updating is performed following an incremental strategy [8]: for a given time instant $n$, a complete round is performed along the network where each node computes its term of the summation in (10), aggregates it to the given filter vector and passes it to the following node in a incremental order. To develop the formulation for this strategy, we start with equation (10) but with all the terms explicitly expressed as:

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \mu \mathbf{v}_1(n) e_1(n) - \mu \mathbf{v}_2(n) e_2(n) - \cdots - \mu \mathbf{v}_N(n) e_N(n) \,. \tag{11}$$

Let us define the local version of the filter coefficient vector $\mathbf{w}(n)$ of (7) at node $k$ as

$$\mathbf{w}^k(n) = \begin{bmatrix} \mathbf{w}_1^k(n) \\ \mathbf{w}_2^k(n) \\ \vdots \\ \mathbf{w}_N^k(n) \end{bmatrix}, \tag{12}$$

and 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:

$$\mathbf{w}^1(n) = \mathbf{w}(n-1) - \mu \mathbf{v}_1(n) e_1(n) \,. \tag{13}$$

Then, node 1 passes its local version of the global vector to node 2 and this node updates its own local version as:

$$\mathbf{w}^2(n) = \mathbf{w}^1(n) - \mu \mathbf{v}_2(n) e_2(n) \,. \tag{14}$$

Afterwards node 2 passes its local version of the global vector to node 3 and so on, till a whole round is done and at node $k = N$ we obtain

$$\mathbf{w}^N(n) = \mathbf{w}^{N-1}(n) - \mu \mathbf{v}_N(n) e_N(n) \,, \tag{15}$$

which is equal to the expression of $\mathbf{w}(n)$ in (10), that is, we have obtained the updated global filter as the local version of the vector at the last node $\mathbf{w}(n) = \mathbf{w}^N(n)$. Therefore, from (13)-(15), we can state the general form of the filter updating at each node as

$$\mathbf{w}^k(n) = \mathbf{w}^{k-1}(n) - \mu \mathbf{v}_k(n) e_k(n) \;,\; 1 \leq k \leq N \,, \tag{16}$$

assuming that the local version of the first node vector is given by $\mathbf{w}^0(n) = \mathbf{w}^N(n-1) = \mathbf{w}(n-1)$.

Finally, once the global updated vector at time instant $n$ has been obtained as $\mathbf{w}(n) = \mathbf{w}^N(n)$, their values are disseminated to the rest of the nodes. Notice that only the local vector that corresponds to the $IL$ coefficients from $IL(k-1)+1$ to $ILk$ of $\mathbf{w}(n)$ defined as

$$\mathbf{w}_k(n) = [\mathbf{w}(n-1)]_{(IL(k-1)+1:ILk)} \,, \tag{17}$$

is needed to generate the $k$th node output signal $y_k(n)$:

$$y_k(n) = \mathbf{w}_k^T(n)\,[\mathbf{X}(n)]_{(:,1)}\,, \tag{18}$$

where $[\mathbf{X}(n)]_{(:,1)}$ is the $[IL \times 1]$ vector corresponding to the first column of $\mathbf{X}(n)$ defined in (9).

It should be noted that the proposed cooperation requires a high data transfer speed and a properly synchronization among the nodes. A data stream of $2ILN(N-1)$ samples should propagate through the nodes in an incremental manner for each sample collected at the error sensors. For example, if the sampling rate is $f_s = 16$ kHz the collaborative tasks should allow a data stream of $32ILN(N-1)$ Ksamples/s.

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 but a remark on the **for** loop of lines 8-10 is needed: The number of multiplications to obtain the $L$ samples of $\mathbf{v}_{jk}(n)$ is $ILM$, which would make a total of $ILMN$ multiplications. However, the oldest $L-1$ samples of $\mathbf{v}_{jk}(n)$ have been already calculated in the previous iterations, so only $IM$ new values are computed, resulting in a total number of $IMN$ multiplications. Consequently, the computational cost of the DMEFxLMS algorithm per iteration and per node is $(IL(N+1)+IMN+1)$ multiplications. It can be easily verified that the DMEFxLMS algorithm has exactly the same computational complexity as the centralized algorithm (MEFxLMS).

The main advantage of this strategy is that the computation of the DMEFxLMs is distributed among the nodes, reducing the computational requirement of the processing units. Moreover, the performance of the DMEFxLMS algorithm is identical to the centralized MEFxLMS algorithm in terms of convergence speed and final residual noise under an incremental strategy and an ideal network, as it can be noticed comparing (10) and (16). As said before, the extension of the DMEFxLMS algorithm to the case of a WASN with multichannel

**Algorithm 1** DMEFxLMS algorithm.

**Initialize:** $\mathbf{w}(0) = \mathbf{w}^k(0) = [0, \ldots, 0]^T \;,\; \forall k \;;\; \mathbf{X}(0) = \mathbf{0}_{IL \times (M+L)}$

1: $n = 1$ % Start sample time

2: **repeat**

3: $\mathbf{w}^0(n) = \mathbf{w}(n-1)$ % Needed at node $k = 1$ in line 12

4: Obtain reference signals $x_i(n)$, $i = 1, \ldots, I$

5: **for all** Node $1 \leq k \leq N$ **do**

6: $\mathbf{w}_k(n) = [\mathbf{w}^k(n-1)]_{(IL(k-1)+1:ILk)}$

7: $y_k(n) = \mathbf{w}_k^T(n)\,[\mathbf{X}(n)]_{(:,1)}$ (Multipl.: $IL$)

8: **for all** $1 \leq j \leq N$ **do**

9: $\mathbf{v}_{jk}(n) = \mathbf{X}(n)\hat{\mathbf{h}}_{jk}$ (Multipl.: $IMN$)

10: **end for**

11: $\mathbf{v}_k(n) = \begin{bmatrix} \mathbf{v}_{1k}^T(n) & \mathbf{v}_{2k}^T(n) & \cdots & \mathbf{v}_{Nk}^T(n) \end{bmatrix}^T$

12: $\mathbf{w}^k(n) = \mathbf{w}^{k-1}(n) - \mu\mathbf{v}_k(n)e_k(n)$ (Multipl.: $ILN + 1$)

13: **end for**

14: $\mathbf{w}(n) = \mathbf{w}^N(n)$ % Updated vector

15: **for all** Node $1 \leq k \leq (N-1)$ **do**

16: $\mathbf{w}^k(n) = \mathbf{w}(n)$ % Disseminate updated vector

17: **end for**

18: $n = n + 1$ % Update sample time

19: **until** ANC system stops

nodes is given in Appendix A.

In the next section we will address some issues regarding a non ideal network, such as limited rate communication and limited computational load at the nodes, which can make the WASN cooperation difficult in practice. For this purpose, we will introduce new parameters in the formulation of the distributed adaptive algorithm and evaluate the performance loss based on these parameters.

## 3. Cooperation-constrained distributed adaptive algorithms

We now examine the case in which the algorithm DMEFxLMS is used in a non ideal network, which we have called a constrained network in the sense it works under a limited data rate. This limited rate introduces a constant latency between two network information exchanges at each node with respect to the acquisition sample time $T_s$ seconds. As stated in previous equations, $n$ stands for the discrete time, which in turn represents a continuous time of $t = nT_s$ seconds. It has to be noticed that the adaptive algorithms are referred to $n$ as the updating time.

From this point on we assume a first change in the algorithm due to communication constraints: the dissemination of the global filter $\mathbf{w}(n) = \mathbf{w}^N(n)$ from node $N$ to the rest of the nodes (lines 15-17 of **Algorithm 1**) will not be carried out. Therefore, the local version of the global filter, $\mathbf{w}^k(n)$, is considered equal to the global filter $\mathbf{w}(n)$ only at node $N$. For the rest of the nodes, their local version of the global vector, $\mathbf{w}^k(n)$, will not be, in general, equal to $\mathbf{w}(n)$. This consideration will particularly affect their local filters since their values are taken from the corresponding elements of their local versions as

$$\mathbf{w}_k(n) = [\mathbf{w}^k(n-1)]_{(IL(k-1)+1:ILk)} . \tag{19}$$

Moreover, the nodes at the first positions according to the incremental strategy will be more affected by the removal of the dissemination step. One possible solution to alleviate this effect could be to cyclically change the order of the nodes in the network, but this solution has not been considered in this paper.

### *3.1. Constrained DMEFxLMS algorithm with partial updates (P-DMEFxLMS)*

As stated before, the limited rate of the communication network introduces a constant latency between two network information exchanges at each node with respect to the acquisition sample time $T_s$ seconds. For homogeneous nodes, this latency can be modelled through a constant parameter $p$ such that data communication between two consecutive nodes takes $pT_s$ seconds to be carried out, being $p$ a positive integer. Therefore, in our incremental network, a node receives the information from its precedent node every $NpT_s$ seconds, the time it takes a round to be completed. Since the adaptive algorithm works at sample rate $T_s$, the filter updating in (16) must be modified to take into account what to do in the meantime. Notice that $p$ should be a positive integer since **Algorithm 1**) iterates at every discrete-time instant $n$.

We present here the distributed MEFxLMS algorithm with partial updates (P-DMEFxLMS). Its basic idea is similar to the conventional partial update algorithms [25, 26] where, generally speaking, only a subset of coefficients is updated at each iteration. As described before, in this collaborative scheme each node aggregates the estimations from its precedent node every $Np$ iterations, while it uses local information to update its filter coefficients during the remainder $Np-1$ iterations.

The coefficients update equations are stated as follows,

- If $[(n-1) \bmod Np]$ is equal to $p(k-1)$, where mod denotes modulo operation, then node $k$ combines its own local information with the information from the precedent node $k-1$:

  $$\mathbf{w}^k(n) = \alpha_k \mathbf{w}^k(n-1) + (1-\alpha_k)\mathbf{w}^{k-1}(n) - \mu \mathbf{v}_k(n) e_k(n)\,, \tag{20}$$

  where $0 \leq \alpha_k \leq 1$ is a constant that weights the local estimate in node $k$, $\mathbf{w}^k(n-1)$, and the data received from node $k-1$, $\mathbf{w}^{k-1}(n)$. It can be considered as a measure of the trustworthiness [28] that node $k$ assigns to its interaction with node $k-1$ .

- Otherwise node $k$ uses its local data

  $$\mathbf{w}^k(n) = \mathbf{w}^k(n-1) - \beta_k \mu \mathbf{v}_k(n) e_k(n)\,, \tag{21}$$

  where $\beta_k \in [0,1]$ is a binary weight that decides if node $k$ performs a local updating ($\beta_k = 1$) or just waits for the arrival of new network information ($\beta_k = 0$).

The complete P-DMEFxLMS algorithm is summarized in **Algorithm 2**. Notice that we have modelled parameter $p$ as a constant for homogeneous networks. For heterogeneous networks, $p$ could be modelled as a different constant for each node, $p_k$, provided that the statement at line 12 was accordingly modified.

**Algorithm 2** P-DMEFxLMS algorithm.

**Initialize:** $\mathbf{w}(0) = \mathbf{w}^k(0) = [0, \ldots, 0]^T$ , $\forall k$ ; $\mathbf{X}(0) = \mathbf{0}_{IL \times (M+L)}$

1: $n = 1$ % Start sample time

2: **repeat**

3: $\mathbf{w}^0(n) = \mathbf{w}^N(n-1)$ % Needed at node $k = 1$ in line 13

4: Obtain reference signals $x_i(n)$, $i = 1, \ldots, I$

5: **for all** Node $1 \leq k \leq N$ **do**

6: $\mathbf{w}_k(n) = [\mathbf{w}^k(n-1)]_{(IL(k-1)+1:ILk)}$

7: $y_k(n) = \mathbf{w}_k^T(n)\,[\mathbf{X}(n)]_{(:,1)}$ (Multipl.: $IL$)

8: **for all** $1 \leq j \leq N$ **do**

9: $\mathbf{v}_{jk}(n) = \mathbf{X}(n)\hat{\mathbf{h}}_{jk}$ (Multipl.: $IMN$)

10: **end for**

11: $\mathbf{v}_k(n) = \begin{bmatrix} \mathbf{v}_{1k}^T(n) & \mathbf{v}_{2k}^T(n) & \cdots & \mathbf{v}_{Nk}^T(n) \end{bmatrix}^T$

12: **if** $[(n-1) \bmod Np] == p(k-1)$ **then**

13: $\mathbf{w}^k(n) = \alpha_k \mathbf{w}^k(n-1) + (1-\alpha_k)\mathbf{w}^{k-1}(n) - \mu \mathbf{v}_k(n) e_k(n)$ (Multipl.: $3ILN + 1$)

14: **else**

15: $\mathbf{w}^k(n) = \mathbf{w}^k(n-1) - \beta_k \mu \mathbf{v}_k(n) e_k(n)$ (Multipl.: $\beta_k(ILN + 1)$)

16: **end if**

17: **end for**

18: $n = n + 1$ % Update sample time

19: **until** ANC system stops

*3.1.1. On the use of $\alpha_k$ and $\beta_k$ in the P-DMEFxLMS algorithm*

A detailed discussion on the use of $\alpha_k$ and $\beta_k$ for the ANC system is needed since the grade of collaboration among the nodes and their computational capacity will determine the accuracy of the global filter $\mathbf{w}(n)$ obtained by P-DMEFxLMS with respect to the DMEFxLMS solution of (11).

- The value of $\alpha_k = 1$ in (20) means that no network information is available at node $k$, similar to the decentralized approach described in [29, 30]. As stated in [30], this strategy can outperform collaborative implementations in case the interaction caused by secondary sources different from that of node $k$ is not significant. Assuming a dynamic network, the value of $\alpha_k = 1$ can be occasionally assigned to those nodes that are not interacting with the rest of the network from an acoustical point of view. Note that in that case, the value assigned to $\beta_k$ will mainly affect the speed of convergence of the algorithm: for $\beta_k = 0$ the local updating will be done every $Np$ iterations in (20), whereas for $\beta_k = 1$ it will be done at sample rate by means of (20)-(21). We call non-collaborative distributed MEFxLMS (NC-DMEFxLMS) the adaptive algorithm corresponding to a value of $\beta_k = 1$ and $\alpha_k = 1$ , $\forall k$, and its description is given in **Algorithm 3**. Notice that although the NC-DMEFxLMS algorithm corresponds to the extreme case such that nodes do not collaborate at all, it can be useful for the sake of comparison to any other level of collaboration in the network.

- The value of $\alpha_k = 0$ in (20) means that local estimate $\mathbf{w}^k(n-1)$ is discarded every $Np$ iterations because the network information is totally trustworthy with respect to the local estimate. Moreover, $\alpha_k = 0$ makes even more sense in combination with $\beta_k = 0$ (the node just waits for the network data) for WASN's with limited power that should be devoted to save as much energy as possible.

- For values within $0 < \alpha_k < 1$, the P-DMEFxLMS behaves in a collaborative way, with a level of collaboration managed by $\alpha_k$. For a given $\alpha_k$, the parameter $\beta_k$ will mainly affect the speed of convergence, together with the step size $\mu$, and the node power consumption as well. The solution obtained will differ to a greater or lesser extent from that of the unconstrained distributed algorithm (DMEFxLMS): firstly because the network takes $Np$ iterations to achieve a global solution at node $N$, and secondly, because the incremental strategy of communication implies that the $k$th node will use a more updated global filter version than $(k-1)$th node to calculate its own secondary source.

A complete theoretical analysis of the performance of the P-DMEFxLMS algorithm is out of the scope of this work. However, a theoretical analysis

**Algorithm 3** : NC-DMEFxLMS algorithm.

**Initialize:** $\mathbf{w}_k(0) = [0, \ldots, 0]^T$ , $\forall k$ ; $\mathbf{X}(0) = \mathbf{0}_{IL\times(M+L)}$

$n = 1$ % Start sample time

**repeat**

  Obtain reference signals $x_i(n)$, $i = 1, \ldots, I$

  **for all** Node $1 \le k \le N$ **do**

    $y_k(n) = \mathbf{w}_k^T(n-1)\,[\mathbf{X}(n)]_{(:,1)}$ (Multipl.: $IL$)

    $\mathbf{v}_{kk}(n) = \mathbf{X}(n)\hat{\mathbf{h}}_{kk}$ (Multipl.: $IM$)

    $\mathbf{w}_k(n) = \mathbf{w}_k(n-1) - \mu \mathbf{v}_{kk}(n) e_k(n)$ (Multipl.: $IL + 1$)

  **end for**

  $n = n + 1$ % Update sample time

**until** ANC system stops

that provides the mean steady-state behaviour of the adaptive weights at each node has been developed in Appendix B. The analysis relies on the statistical characteristics of both reference and noise signals, and on the algorithm free parameters as well $(\alpha_k, \beta_k, \mu, p)$.

We end this section with a remark on the use of distributed constrained strategies based on diffusion implementation [27] instead of incremental ones. In diffusion networks all the nodes simultaneously exchange information with its neighbours, thus the latency is considered null and the problem is mainly focused on how to combine the local and network information [28]. Depending on the value of $\alpha_k$, its global solution would be similar to the unconstrained DMEFxLMS algorithm of Section 2.2. Diffusion implementation can obtain the non collaborative NC-DMEFxLMS solution for $\alpha_k = 1$ , $\forall k$. Nevertheless, we have only considered a network with an incremental communication in the simulations of Section 4.

## 4. Simulation results

In this section we present a set of simulations carried out to evaluate the performance of the distributed algorithms introduced in Sections 2.2 and 3.1 for unconstrained and constrained networks respectively. All the simulated WASN's use real acoustic responses measured inside a listening room of 9,36 meters long by 4,78 meters wide by 2,63 meters high located at the Audio Processing Laboratory of the Polytechnic University of Valencia, and modelled as FIR filters of $M = 256$ coefficients. This room has an array of 96 independent-driven loudspeakers deployed in an octagonal shape. A photograph of the listening room can be seen in Fig. B.4. We have simulated two homogeneous acoustic networks of eight and four nodes each considering two different settings of loudspeakers and microphones:

**System 1:** Eight loudspeakers were selected from one of the lines with a uniform separation of 20 cm between adjacent loudspeakers. A sketch of the simulated WASN is depicted in Fig. B.5(a). The eight 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 separated 27 cm away from them. Each node was formed by one loudspeaker and the corresponding microphone opposite to it.

**System 2:** A sketch of the simulated eight-node WASN for System 2 is depicted in Fig. B.5(b). Regarding the loudspeakers, four were selected from the same line used in System 1, but with a minimum separation of 80 cm between them. Then, the remaining loudspeakers were taken from the lines perpendicular to the first linear sector, two from the right and two from the left, with a minimum separation of 80 cm between them. Each microphone was placed opposite to one loudspeaker, separated 27 cm away from it, to form a node. Notice that the larger separation between loudspeakers in System 2 will affect the acoustic coupling between nodes, which in turn will affect the performance of the algorithms.

For both System 1 and System 2, the simulated four-node WASN was formed by selecting nodes 1 to 4 (with gray microphones and loudspeakers) in Fig. B.5(a) and Fig. B.5(b) respectively. All microphones and loudspeakers involved were located at a height of 1,47 m.

The ANC system tries to cancel an unwanted noise by means of $N$ secondary sources. In this case, the reference signal ($I = 1$) is a Gaussian random noise of zero mean and unit variance and is provided to all the nodes of the WASN as well as to the primary signal loudspeaker. This reference signal is emitted by one loudspeaker of the array shown in Fig. B.4, but located at the opposite side of the line of loudspeakers forming the WASN. Therefore, the unwanted noise recorded at the nodes is the reference signal filtered through the acoustic plants between the primary loudspeaker and their corresponding microphones. The adaptive filters to be designed have a length of $L = 150$ coefficients. A fixed step size of $\mu = 0.001$ has been used in all the algorithms for the sake of fair comparison. All the signals and filters work at a sample rate of 16000 Hz.

In order to evaluate the performance of different algorithms and parameters, we define the instantaneous *Noise Reduction* at node $k$, $\mathrm{NR}_k(n)$, as the ratio in dB between the instantaneous estimated error power with and without the application of the active noise controller,

$$\mathrm{NR}_k(n) = 10 \log_{10} \left[ \frac{e_k^2(n)}{d_k^2(n)} \right] , \tag{22}$$

where $d_k(n)$ is the signal that would be measured by the $k$th microphone if the ANC system was inactive. The noise reduction (NR) can be depicted versus the number of iterations providing the learning curves for each sensor and each algorithm. In all the figures presented in this section, the curves represent the averaged NR over 100 independent runs.

In the first scenario the performances of the DMEFxLMS and the NC-DMEFxLMS algorithms compared to the centralized MEFxLMS have been evaluated. Fig. B.6 presents the NR obtained by a WASN of four nodes for the three algorithms. Fig. B.6.(a) shows the NR's for the node with the best performance (denoted as best node), whereas Fig. B.6.(b) shows the results for the node with the poorest performance (denoted as worst node). In both cases DMEFxLMS and centralized MEFxLMS exhibit equal performance as it has been theoretically stated in section 2.2 so both curves are labelled as DME-FxLMS. Regarding System 1, it can be observed that NC-DMEFxLMS starts cancelling the noise but at a certain point it turns unstable and does not converge. However DMEFxLMS and MEFxLMS algorithms are stable, providing 30 dB of noise reduction for the best node and more than 10 dB for the worst. Results for System 2 are quite different. Both distributed and non-cooperative approaches achieve similar results in both Fig. B.6.(a) and Fig. B.6.(b), with a slight improvement in the convergence of NC-DMEFxLMS compared to DMEFxLMS.

The results for an eight-node WASN are depicted in Fig. B.7.(a) for the best node and Fig. B.7.(b) for the worst one. The NR curves present a similar behaviour to the four-node case, although the unstable tendency of the NC-DMEFxLMS algorithm in System 1 is further stressed. Notice that the NR achieved in the worst node, Fig. B.7.(b), is very poor for both configurations, particularly for System 2 whose NR fluctuates around 0 dB. This is due to the physical location of the microphone belonging to the worst-performance node. The physical acoustic paths between the loudspeakers and the referred microphone presented much lower average energy values than the best-performance node, limiting the attenuation achieved by the ANC system at that node.

From the results obtained in Fig. B.6 and B.7 for this first scenario, we can conclude that the proposed distributed DMEFxLMS algorithm has the same performance as the centralized MEFxLMS. Both show a robust behaviour regarding stability, although the achieved NR depends on the WASN settings (number of nodes and location of microphones and loudspeakers). On the contrary, the non-cooperative NC-DMEFxLMS appears to be very sensitive to WASN settings, as it can be seen from the unstable behaviour obtained for System 1 in both four and eight-node networks.

A second simulation using only the four-node WASN of System 1 has been carried out to evaluate the partial update algorithm (P-DMEFxLMS) for different latency values of the network, $p$, and different choices of $\alpha_k$ and $\beta_k$. For the sake of simplicity we consider $\alpha_k = \alpha\,, \forall k$ and $\beta_k = \beta\,, \forall k$, i.e., we assume the nodes are homogeneous in their behaviour. We will compare P-DMEFxLMS with the unconstrained DMEFxLMS and the non collaborative

NC-DMEFxLMS algorithms. First we study the effect of the $\alpha$ parameter and assume $\alpha = 0$ in (20) so all nodes discard their local estimates every $Np$ iterations. The NR curves (22) of the best node in the network for $\alpha = 0$ and different latency values of $p$ are presented in Fig. B.8.(a) for $\beta = 0$, and Fig. B.8.(b) for $\beta = 1$. For both values of $\beta$, the performance of the P-DMEFxLMS algorithm worsens as $p$ increases, but the deterioration due to the latency is much more emphasized when $\beta = 0$. However, for $\beta = 1$ the behaviour of the P-DMEFxLMS presents steady-state values close to the DMEFxLMs for all $p$. These preliminary results have been confirmed for $\alpha = 0.5$, which equally combines local estimate and network information at each node. The NR for different latency values of $p$ is presented in Fig. B.9.(a) for $\beta = 0$, and Fig. B.9.(b) for $\beta = 1$. Results show a similar behaviour to Fig. B.8 suggesting the following conclusion: for networks with a limited data rate, convergence performance can be greatly improved if the nodes are allowed to update their filter coefficients by setting $\beta = 1$ in (21). This statement is true for networks with an acoustic interaction among the nodes similar to the one analysed.

A second scenario consisting on a two-node WASN has been considered to study the behaviour of the ANC system regarding the weighting value $\alpha_k$. For this purpose we have built a WASN with 2 nodes with three different relationships between their acoustic paths. These acoustic paths have been selected from the real acoustic responses measured inside the listening room of Fig. B.4, such that they fulfil the conditions described below. Regarding the unwanted noise and reference signals of the ANC system, they are the same signals used in the first scenario.

To identify the type of interaction between nodes, let us define the Level of Interaction (LI) of node $j$ over node $i$ as the ratio between the energy of acoustic paths $\mathbf{h}_{ji}$ and $\mathbf{h}_{ii}$:

$$\mathrm{LI}_{ji} = \frac{\sum_{m=1}^{M} h_{ji}^2(m)}{\sum_{m=1}^{M} h_{ii}^2(m)} = \frac{E_{h_{ji}}}{E_{h_{ii}}}\,, \tag{23}$$

where $h_{ji}(m)$ is the $m$th coefficient of the acoustic path between the $j$th loudspeaker and the $i$th microphone and $E_{h_{ji}}$ is the energy of acoustic path $\mathbf{h}_{ji}$.

To clarify the meaning of the proposed ratio (23), assume all the secondary sources were fed with the same energy. Then for small values of $\mathrm{LI}_{ji}$, the secondary source of node $j$ would barely contribute to the error signal measured at node $i$ with respect to its own secondary source $i$. On the contrary, big values of $\mathrm{LI}_{ji}$ would indicate a great contribution from node $j$ to node $i$, that is, generally speaking they would be acoustically coupled.

We have designed three different settings to show different levels of interaction $\mathrm{LI}_{ji}$ between the nodes. For the sake of fair comparison, we have simulated in the three cases a symmetric WASN where $\mathbf{h}_{12} = \mathbf{h}_{21}$ and $\mathbf{h}_{11} = \mathbf{h}_{22}$, consequently $\mathrm{LI}_{12} = \mathrm{LI}_{21}$. Table B.1 shows the value of LI for each type of ANC system. Depending on the value of $\mathrm{LI}_{12}$, we have called the ANC system uncoupled for a $\mathrm{LI}_{12}$ near to 0, coupled for $\mathrm{LI}_{12}$ near to 1, and forced coupled for $\mathrm{LI}_{12}$ much larger than 1. Table B.1 also includes the value of the normalized cross-covariance coefficient $\rho_{12}$ defined as

$$\rho_{ji} = \frac{\sum_{m=1}^{M} h_{ji}(m) h_{ii}(m)}{\sqrt{\sum_{m=1}^{M} h_{ji}^2(m)}\sqrt{\sum_{m=1}^{M} h_{ii}^2(m)}}\,. \tag{24}$$

We have run the simulations with $0 \leq \alpha_k \leq 1$ in steps of 0.1, and $\beta_k \in [0, 1]$. We consider homogeneous nodes and we have stated $\alpha_1 = \alpha_2 = \alpha$ and $\beta_1 = \beta_2 = \beta$. The average steady-state NR for both nodes is plotted in Fig. B.10 versus parameter $\alpha$ for latency values of $p = 2, 4, 10$ and for (a) $\beta = 0$ and (b) $\beta = 1$. Notice that a value of $\beta = 0$ in (21) means that the filter coefficients are updated only every $Np$ samples, see also line 15 in **Algorithm 2**, whereas for $\beta = 1$ the filters are updated every sample.

For the case of $\beta = 0$, we can see in Fig. B.10.(a) that the network latency determines the steady-state NR achieved by the WASN, except for the case of coupled systems that does not vary so much. The performance of uncoupled systems is better for $\alpha = 1$ (no network information) where the node updates its filter coefficient every $Np$ iterations using its local information (third term in (20)). A good performance is also obtained for $\alpha = 0$ where every $Np$ iterations the node discards its own estimate and uses the network information together with its local information. Any other combination provided by $\alpha$ worsens its behaviour. It seems that relying on its own information ($\alpha = 1$) or totally discarding it ($\alpha = 0$) is the best option, while any other combination interferes to achieve the best performance. In the case of forced coupled systems, NR increases with $\alpha$, which means that the nodes should totally trust on the estimate provided by the network ($\alpha = 0$). Finally, the performance of the coupled system does not vary with the latency, neither with $\alpha$.

For $\beta = 1$ shown in Fig. B.10.(b), it can be seen that the uncoupled system keeps the same tendency to work well for any value of $\alpha$, the best being $\alpha = 1$, whereas the forced coupled system stresses the need for any kind of collaboration, $0 \leq \alpha < 1$, obtaining no NR when $\alpha = 1$. However, the coupled system is affected by neither $\alpha$ nor $p$, and it achieves the worst performance. A possible explanation of this behaviour is given observing the cross-covariance between the nodes in Table B.1. The value of $\rho_{12}$ is close to 1 in the coupled system which means that acoustic path $\mathbf{h}_{12}$ is very correlated to $\mathbf{h}_{22}$, and due to the symmetry of the WASN, acoustic path $\mathbf{h}_{21}$ has the same correlation with $\mathbf{h}_{11}$. As a consequence the four acoustic paths are very correlated, thus they have quite similar responses. Therefore the poor NR achieved can be explained since the two-node ANC system presents a unique secondary path (all $\mathbf{h}_{ij}$ are fully correlated), but it is trying to cancel two unwanted noises, one recorded at node 1 and other recorded at node 2. Regarding the other two ANC systems, notice that their cross-covariance terms in Table B.1 are close to 0. Consequently their acoustic paths are uncorrelated and their secondary sources can be independently driven.

Finally, it should be noted that all the conclusions stated in this paper are derived from the particular settings and parameter configuration. For instance, the use of a specific step size for each algorithm would improve the NR obtained and would also affect their speed of convergence.

## 5. Conclusions

In this paper an active noise canceller has been implemented over an acoustic sensor network that makes use of incremental communication strategies. For this purpose several distributed versions of the centralized adaptive algorithm MEFxLMS have been introduced to deal with unconstrained and constrained networks. In the case of constrained networks, the distributed algorithm can be configured by means of two parameters whose values can be adjusted to control the collaboration in the network. Indeed, these parameters affect the performance of the ANC system as well. We have provided simulations to demonstrate the performance of the proposed algorithms in different scenarios where the number of nodes, the acoustical interactions between them, and the network constraints have been varied.

### Acknowledgements

This work has been supported by European Union ERDF and Spanish Government through TEC2012-38142-C04 project, and Generalitat Valenciana through PROMETEO/2009/013 project.

## Appendix A. Generalization of the DMEFxLMS algorithm to multichannel nodes

In this section we will extend the algorithm provided in Section 2.2 to the general case of a WASN with multichannel nodes just considering that each node is equipped with a maximum of $K'$ sensors and a maximum of $J'$ loudspeakers. The problem formulation and notation is derived from the single channel nodes case, and the algorithm is straightforwardly extended from the corresponding equations simply redefining some matrices and vectors.

Firstly, the local version of the global filter vector $\mathbf{w}(n)$ in every node is comprised of $INJ'$ filters of size $L$ unlike the single-channel nodes case that uses $IN$ filters. From (12) we can define the local version of the global filter in node $k$, $\mathbf{w}^k(n)$ as comprised of $N$ vectors $\mathbf{w}_q^k(n)$, $q = 1, \ldots, N$, which correspond to the $k$th local version of the $IJ'L$ filter coefficients of the $q$th node, and whose expression is given by

$$\mathbf{w}_q^k(n) = [\mathbf{w}_{11q}^k(n)^T \ldots \mathbf{w}_{I1q}^k(n)^T \ldots \mathbf{w}_{ijq}^T(n) \ldots \mathbf{w}_{1J'q}^k(n)^T \ldots \mathbf{w}_{IJ'q}^k(n)^T]^T, \tag{A.1}$$

where the $\mathbf{w}_{ijq}(n)$ are vectors of size $L$. Considering the case of non homogeneous networks, if node $k$ has a number of loudspeakers $J'_k$ such that $J'_k < J'$, the corresponding coefficients of filters denoted by sub-index $j = J'_k + 1, \ldots, J'$ in (A.1) will be zero.

The $IJ'L$ filter coefficients used to generate the output signals at node $k$, equation (17) for single channel nodes, are taken now as

$$\mathbf{w}_k(n) = [\mathbf{w}^k(n-1)]_{(ILJ'(k-1)+1:ILJ'k)} \,. \tag{A.2}$$

The output signals at time $n$ that feed the actuators of node $k$ form the $J'$-size vector $\mathbf{y}_k(n)$, given by

$$\mathbf{y}_k(n) = (\mathbf{W}_{kJ'}(n))^T [\mathbf{X}(n)]_{(:,1)} \,, \tag{A.3}$$

where $\mathbf{W}_{kJ'}(n)$ is a $IL \times J'$ matrix created rearranging the elements of the $\mathbf{w}_k(n)$ vector defined in (A.2) ,

$$\mathbf{W}_{kJ'}(n) = \begin{bmatrix} \mathbf{w}_k(n)_{(1:IL)} & \mathbf{w}_k(n)_{(IL+1:2IL)} & \ldots & \mathbf{w}_k(n)_{(IL(J'-1)+1:IJ'L)} \end{bmatrix}. \tag{A.4}$$

Notice again that in case a node has less than $J'$ actuators, the corresponding element in vector $\mathbf{y}_k(n)$ will be zero. The $ILN$ vector $\mathbf{v}_k(n)$ introduced in (9) is extended to a matrix $\mathbf{V}_k(n)$ of $ILJ'N \times K'$ dimensions composed of matrices $\mathbf{V}_{jk}(n)$ with dimensions $IL \times K'$ as $\mathbf{V}_k = [\mathbf{V}_{1k}^T(n) \quad \mathbf{V}_{2k}^T(n) \quad \ldots \quad \mathbf{V}_{(J'N)k}^T(n)]^T$, where

$$\mathbf{V}_{jk}(n)) = \mathbf{X}(n)\hat{\mathbf{H}}_{jk}, \quad \forall\, 1 \le j \le J'N \tag{A.5}$$

with

$$\hat{\mathbf{H}}_{jk} = [\hat{\mathbf{h}}_{j[K'(k-1)+1]} \ldots \hat{\mathbf{h}}_{j[K'k]}]. \tag{A.6}$$

Similarly to the previous notation, and assuming again that a node can have less than $J'$ actuators and $K'$ error sensors, only the corresponding secondary paths contained in $\hat{\mathbf{H}}_{jk}$ are different from zero.

Finally, we define the $\mathbf{e}_k(n)$ vector that contains the $K'$ error signals obtained at time $n$ in the $k$th node as

$$\mathbf{e}_k(n) = \left[e_{K'(k-1)+1}(n) \dots e_{K'k}(n)\right], \tag{A.7}$$

and assume that the corresponding position of vector $\mathbf{e}_k(n)$ is zero if node $k$ has less than $K'$ sensors.

In the following we summarize the changes required in **Algorithm 1** of Section 2.2 in order to extend the DMEFxLMS algorithm to the case of a WASN of multichannel nodes. Notice that the elements of the coefficient vector $\mathbf{w}^k(n)$ in (12) have been redefined as in (A.1), and that the global vector $\mathbf{w}(n)$ has the same element ordering than $\mathbf{w}^k(n)$. For **Algorithm 1**:

- Replace line 6 by equation (A.2).
- Replace scalar calculation of line 7 by vector calculation of (A.3).
- Replace vector calculation of line 9 by matrix calculation of (A.5).
- Define matrix $\mathbf{V}_k(n)$ accordingly to line 11 and (A.5).
- Update the filter coefficients of line 12 considering matrix $\mathbf{V}_k(n)$ and error vector (A.7).

## Appendix B. Mean steady-state weight behavior of the P-DMEFxLMS algorithm

In this section the mean weight behaviour of the P-DMEFxLMS algorithm is analysed assuming that all nodes in the WASN are homogeneous in their parameters, thus $\alpha_k = \alpha$, $\forall k$ and $\beta_k = \beta$, $\forall k$. This analysis describes the mean behaviour of the adaptive weights at each node of the network and provides the corresponding steady-state mean weight vector.

The P-DMEFxLMS algorithm updates the adaptive weights at every node according to equations (20) and (21). Particularizing these expressions for a generic node $k$ with $p > 1$, both equations can be merged as

$$\begin{aligned}
&\mathbf{w}^k(n) = (1-\alpha)^N \mathbf{w}^k(n-Np) \\
&-\mu \sum_{m=0}^{N-1} (1-\alpha)^m \mathbf{v}_{(k-m)}(n-mp)\, e_{(k-m)}(n-mp) \\
&+\alpha \sum_{m=0}^{N-1} (1-\alpha)^m \Big[\mathbf{w}^{(k-m)}(n-(N+m)p) \\
&-\beta\mu \sum_{q=1}^{Np-1} \mathbf{v}_{(k-q)}(n-q-mp)\, e_{(k-q)}(n-q-mp)\Big].
\end{aligned} \tag{B.1}$$

It should be noted that for $m \geq k$, $\mathbf{w}^{(k-m)}(n) = \mathbf{w}^{(N+k-m)}(n)$, $\mathbf{v}_{(k-m)}(n) = \mathbf{v}_{(N+k-m)}(n)$ and $e_{(k-m)}(n) = e_{(N+k-m)}(n)$.

Taking expectations of both sides of (B.1), we get

$$\begin{aligned}
&\mathrm{E}[\mathbf{w}^k(n)] = (1-\alpha)^N \mathrm{E}[\mathbf{w}^k(n-Np)] - \\
&\mu \sum_{m=0}^{N-1} (1-\alpha)^m \mathrm{E}[\mathbf{v}_{(k-m)}(n-mp)\, e_{(k-m)}(n-mp)] \\
&+\alpha \sum_{m=0}^{N-1} (1-\alpha)^m \mathrm{E}[\mathbf{w}^{(k-m)}(n-(N+m)p)] \\
&-\beta\mu\alpha \sum_{m=0}^{N-1} (1-\alpha)^m \sum_{q=1}^{Np-1} \mathrm{E}[\mathbf{v}_{(k-q)}(n-q-mp)\, e_{(k-q)}(n-q-mp)].
\end{aligned} \tag{B.2}$$

The error signal at the $k$th node at time $n$ is given by (see Fig. B.1),

$$e_k(n) = d_k(n) + \left[\mathbf{h}_{1k}^T \mathbf{X}(n)^T,\ \mathbf{h}_{2k}^T \mathbf{X}(n)^T, \ldots,\ \mathbf{h}_{Nk}^T \mathbf{X}(n)^T\right] \mathbf{w}(n), \tag{B.3}$$

where $d_k(n)$ is the disturbance or primary signal at the $k$th error sensor. Considering perfect secondary path estimation, $\left[\mathbf{h}_{1k}^T \mathbf{X}(n)^T, \ldots, \mathbf{h}_{Nk}^T \mathbf{X}(n)^T\right]$ can be substituted by $\mathbf{v}_k^T(n)$, which is known by the adaptive controller and it is defined in (8). Substituting (B.3) in (B.2) and taking the limit as $n \to \infty$, we obtain

$$\begin{aligned}
&\mathrm{E}[\mathbf{w}^k(\infty)] = (1-\alpha)^N \mathrm{E}[\mathbf{w}^k(\infty)] - \mu\{\mathbf{r} + \mathbf{R}\,\mathrm{E}[\mathbf{w}(\infty)]\} \\
&+\alpha \sum_{m=0}^{N-1} (1-\alpha)^m \left[\mathrm{E}[\mathbf{w}^{(k-m)}(\infty)] - \beta\mu \sum_{q=1}^{Np-1} \{\mathbf{r}_{(k-q)} + \mathbf{R}_{(k-q)}\mathrm{E}[\mathbf{w}(\infty)]\}\right],
\end{aligned} \tag{B.4}$$

where we have used the steady-state condition $\mathrm{E}[\mathbf{w}^k(n)] = \mathrm{E}[\mathbf{w}^k(n - Np)] = \mathrm{E}[\mathbf{w}^k(\infty)]$. The previous expression has been derived by means of the following considerations. On the one hand both the filtered reference signal vectors $\mathbf{v}_k(n)$ and the disturbance signals $d_k(n)$ are wide-sense stationary. Thus,

$$\mathrm{E}[\mathbf{v}_k(n-m)d_k(n-m)] = \mathrm{E}[\mathbf{v}_k(n)d_k(n)] = \mathbf{r}_k\,, \tag{B.5}$$

where $\mathbf{r}_k$ is the cross correlation vector at the $k$th node between the primary and filtered reference signals, and yields

$$\begin{aligned}&\sum_{m=0}^{N-1} \mathrm{E}(1-\alpha)^m[\mathbf{v}_{(k-m)}(n-mp)\, d_{(k-m)}(n-mp)] \\ &= \sum_{m=0}^{N-1}(1-\alpha)^m \mathrm{E}[\mathbf{v}_{(k-m)}(n)\, d_{(k-m)}(n)] = \sum_{k=0}^{N-1}(1-\alpha)^k \mathbf{r}_k = \mathbf{r}.\end{aligned} \tag{B.6}$$

Regarding matrix $\mathbf{R}$ in (B.4), the autocorrelation matrix for the filtered reference signals at the $k$th node is given by

$$\mathrm{E}\{\mathbf{v}_k(n-m)\mathbf{v}_k^T(n-m)\} = \mathrm{E}\{\mathbf{v}_k(n)\mathbf{v}_k^T(n)\} = \mathbf{R}_k, \tag{B.7}$$

and similar to (B.6), it yields

$$\begin{aligned}&\sum_{m=0}^{N-1} \mathrm{E}[(1-\alpha)^m \mathbf{v}_{(k-m)}(n-mp)\, \mathbf{v}_{(k-m)}^T(n-mp)] \\ &= \sum_{m=0}^{N-1}(1-\alpha)^m \mathrm{E}[\mathbf{v}_{(k-m)}(n)\mathbf{v}_{(k-m)}^T(n)] = \sum_{k=0}^{N-1}(1-\alpha)^k \mathbf{R}_k = \mathbf{R}.\end{aligned} \tag{B.8}$$

Finally, (B.4) can be rewritten as

$$a\mathrm{E}[\mathbf{w}^k(\infty)] = \alpha \sum_{m=1}^{N-1}(1-\alpha)^m \mathrm{E}[\mathbf{w}^{k-m}(\infty)] - \mathbf{B}_k \mathrm{E}[\mathbf{w}(\infty)] - \mathbf{c}_k, \tag{B.9}$$

where

$$a = 1 - \alpha - (1-\alpha)^N, \tag{B.10}$$

$$\mathbf{B}_k = \mu \left[\mathbf{R} + \mathbf{R}_k\, \alpha\, \beta\, (p-1) \sum_{m=0}^{N-1}(1-\alpha)^m + \left\{\sum_{m=1}^{N-1} \mathbf{R}_{(k-m)}\right\} \alpha\, \beta\, p \sum_{m=0}^{N-1}(1-\alpha)^m\right] \tag{B.11}$$

$$\mathbf{c}_k = \mu \left[\mathbf{r} + \mathbf{r}_k\, \alpha\, \beta\, (p-1) \sum_{m=0}^{N-1}(1-\alpha)^m + \left\{\sum_{m=1}^{N-1} \mathbf{r}_{(k-m)}\right\} \alpha\, \beta\, p \sum_{m=0}^{N-1}(1-\alpha)^m\right], \tag{B.12}$$

and for indices such that $m \geq k$, $\mathbf{R}_{(k-m)}(n) = \mathbf{R}_{(N+k-m)}(n)$ and $\mathbf{r}_{(k-m)}(n) = \mathbf{r}_{(N+k-m)}(n)$.

Note that (B.9) is the steady-state mean weight vector at the $k$th node and it would only provide the steady-state behaviour of the global network in some simple cases such as $\alpha = 1$ and $\beta = 0$, where all the nodes converge to the same solution $\big(\mathrm{E}[\mathbf{w}(\infty)] = \mathbf{R}^{-1}\mathbf{r}\big)$. However, to derive a network global solution we should pose a set of vector equations considering (B.9) at every node. For example, for a two-node network ($N = 2$) the steady-state filter vector is represented at each node by

$$\mathbf{w}^1(\infty) = \begin{bmatrix} \mathbf{w}_1^1(\infty) \\ \mathbf{w}_2^1(\infty) \end{bmatrix}, \mathbf{w}^2(\infty) = \begin{bmatrix} \mathbf{w}_1^2(\infty) \\ \mathbf{w}_2^2(\infty) \end{bmatrix}, \tag{B.13}$$

and the global filter vector is given by

$$\mathbf{w}(\infty) = \begin{bmatrix} \mathbf{w}_1^1(\infty) \\ \mathbf{w}_2^2(\infty) \end{bmatrix}. \tag{B.14}$$

The steady-state solution can be derived solving the following system of differential equations

$$a \begin{bmatrix} \mathbf{w}_1^1(\infty) \\ \mathbf{w}_2^1(\infty) \\ \mathbf{w}_1^2(\infty) \\ \mathbf{w}_2^2(\infty) \end{bmatrix} = \alpha(1-\alpha) \begin{bmatrix} \mathbf{w}_1^2(\infty) \\ \mathbf{w}_2^2(\infty) \\ \mathbf{w}_1^1(\infty) \\ \mathbf{w}_2^1(\infty) \end{bmatrix} - \begin{bmatrix} \mathbf{B}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_2 \end{bmatrix} \begin{bmatrix} \mathbf{w}_1^1(\infty) \\ \mathbf{w}_2^2(\infty) \\ \mathbf{w}_1^1(\infty) \\ \mathbf{w}_2^2(\infty) \end{bmatrix} - \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix}, \tag{B.15}$$

that can be rewritten as

$$\begin{bmatrix} \mathbf{w}_1^1(\infty) \\ \mathbf{w}_2^1(\infty) \\ \mathbf{w}_1^2(\infty) \\ \mathbf{w}_2^2(\infty) \end{bmatrix} = - \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} [a\mathbf{I}_{(IL4)} - \alpha(1-\alpha)\mathbf{G} + \mathbf{B}]^{-1} \tag{B.16}$$

where $\mathbf{I}_{(IL4)}$ is a $(IL4 \times IL4)$ identity matrix and $\mathbf{G}$ is a permutation matrix defined as

$$\mathbf{G} = \begin{bmatrix} \mathbf{0}_{(IL2)} & \mathbf{I}_{(IL2)} \\ \mathbf{I}_{(IL2)} & \mathbf{0}_{(IL2)} \end{bmatrix}, \tag{B.17}$$

being $\mathbf{I}_{(IL2)}$ a $(IL2 \times IL2)$ identity matrix. Matrix $\mathbf{B}$ is a block matrix given by

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1^{11} & \mathbf{0}_{(IJ)} & \mathbf{0}_{(IJ)} & \mathbf{B}_1^{12} \\ \mathbf{B}_1^{21} & \mathbf{0}_{(IJ)} & \mathbf{0}_{(IJ)} & \mathbf{B}_1^{22} \\ \mathbf{B}_2^{11} & \mathbf{0}_{(IJ)} & \mathbf{0}_{(IJ)} & \mathbf{B}_2^{12} \\ \mathbf{B}_2^{21} & \mathbf{0}_{(IJ)} & \mathbf{0}_{(IJ)} & \mathbf{B}_2^{22} \end{bmatrix}, \tag{B.18}$$

whose non-zero components are the $[IJ \times IJ]$ blocks resulting from the partition of matrices $\mathbf{B}_1$ and $\mathbf{B}_2$ defined in (B.11) and used in (B.15):

$$\mathbf{B}_1 = \begin{bmatrix} \mathbf{B}_1^{11} & \mathbf{B}_1^{12} \\ \mathbf{B}_1^{21} & \mathbf{B}_1^{22} \end{bmatrix}, \qquad \mathbf{B}_1 = \begin{bmatrix} \mathbf{B}_2^{11} & \mathbf{B}_2^{12} \\ \mathbf{B}_2^{21} & \mathbf{B}_2^{22} \end{bmatrix}. \tag{B.19}$$

To conclude this section it should be noted that (B.16) provides an estimation of the steady-state weight vector at every node depending on both the algorithm configuration parameters ($\alpha$, $\beta$, $p$, $\mu$ and $N$ values), and the statistical characteristics of the filtered reference signals and primary signals.

## References


[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, Communications Magazine, IEEE 40 (8) (2002) 102–114.

[2] K. Martinez, J. Hart, R. Ong, Environmental sensor networks, Computer 37 (8) (2004) 50–56.

[3] D. Puccinelli, M. Haenggi, Wireless sensor networks: applications and challenges of ubiquitous sensing, Circuits and Systems Magazine, IEEE 5 (3) (2005) 19–31.

[4] M. Maroti, G. Simon, A. Ledeczi, J. Sztipanovits, Shooter localization in urban terrain, Computer 37 (8) (2004) 60–61.

[5] W.-P. Chen, J. Hou, L. Sha, Dynamic clustering for acoustic target tracking in wireless sensor networks, Mobile Computing, IEEE Transactions on 3 (3) (2004) 258–271.

[6] H. Kwon, V. Berisha, A. Spanias, Real-time sensing and acoustic scene characterization for security applications, in: Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on, 2008, pp. 755–758.

[7] A. Bertrand, Applications and trends in wireless acoustic sensor networks: A signal processing perspective, in: Communications and Vehicular Technology in the Benelux (SCVT), 2011 18th IEEE Symposium on, 2011, pp. 1–6.

[8] C. Lopes, A. Sayed, Incremental adaptive strategies over distributed networks, Signal Processing, IEEE Transactions on 55 (8) (2007) 4064–4077.

[9] A. Bertrand, M. Moonen, Distributed Adaptive Node-Specific Signal Estimation in Fully Connected Sensor Networks - Part I: Sequential Node Updating, Signal Processing, IEEE Transactions on 58 (10) (2010) 5277–5291.

[10] N. Bogdanovic, J. Plata-Chaves, K. Berberidis, Distributed incremental-based LMS for node-specific parameter estimation over adaptive networks, Proc. 2013 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), (2013) 5425–5429.

[11] A. Tanenbaum, D. Wetherall, Computer Networks, Pearson Education, Limited, 2011.

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

[13] S. Haykin, Adaptive Filter Theory, 4th Edition, Prentice-Hall information and system sciences series, Prentice-Hall, Ed., Upper Saddle River, NJ, 2002.

[14] S. Elliott, I. Stothers, P. Nelson, A multiple error LMS algorithm and its application to the active control of sound and vibration, Acoustics, Speech and Signal Processing, IEEE Transactions on 35 (10) (1987) 1423–1434.

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

[16] C. Lopes, A. Sayed, Diffusion least-mean squares over adaptive networks: Formulation and performance analysis, Signal Processing, IEEE Transactions on 56 (7) (2008) 3122–3136.

[17] J. Predd, S. Kulkarni, H. Poor, Distributed learning in wireless sensor networks, Signal Processing Magazine, IEEE 23 (4) (2006) 56–69.

[18] S. Elliott, C. Boucher, Interaction between multiple feedforward active control systems, Speech and Audio Processing, IEEE Transactions on 2 (4) (1994) 521–530.

[19] E. Bjarnason, Analysis of the filtered-x LMS algorithm, Speech and Audio Processing, IEEE Transactions on 3 (6) (1995) 504–514.

[20] N. George, G. Panda, A particle-swarm-optimization-based decentralized nonlinear active noise control system, Instrumentation and Measurement, IEEE Transactions on 61 (12) (2012) 3378–3386.

[21] K. Ozeki, 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 J67-A (1984) 126–132.

[22] M. de Diego, A. Gonzalez, G. Piñero, M. Ferrer, J. Garcia-Bonito, Subjective evaluation of actively controlled interior car noise, ICASSP 2001, Proc. of the Int. Conference on Acoustics, Speech and Signal Proces. 5 (2001) 3225–3228.

[23] A. Gonzalez, A. Albiol, S. Elliott, Minimisation of the maximum error signal in active control, IEEE Transactions on Acous. Speech and signal Processing 6 (3) (1998) 268–281.

[24] H. Hamada, Signal processing for active noise control: adaptive signal processing, Proceeding 1991 Int. Symposium on Active Control of Sound and Vibration 1 (33-44).

[25] S. Douglas, Adaptive filters employing partial updates, Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on 44 (3) (1997) 209–216.

[26] M. Godavarti, A. Hero, Partial update LMS algorithms, Signal Processing, IEEE Transactions on 53 (7) (2005) 2382–2399.

[27] N. Takahashi, I. Yamada, A. Sayed, Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis, Signal Processing, IEEE Transactions on 58 (9) (2010) 4795–4810.

[28] A. Sayed, S.-Y. Tu, J. Chen, X. Zhao, Z. Towfic, Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior, Signal Processing Magazine, IEEE 30 (3) (2013) 155–171.

[29] P. Grosdidier, M. Morari, Interaction measures for systems under decentralized control, Automatica 22 (3) (1986) 309–319.

[30] P. Joseph, S. Elliot, P. A. Nelson, Near field zones of quiet, Journal of Sound and Vibration 172 (5) (1994) 605–627.

| | **Type of ANC system** | | |
|---|---|---|---|
| | UNCOUPLED | COUPLED | FORCE COUPLED |
| | $E_{h_{21}} \ll E_{h_{11}}$ | $E_{h_{21}} \simeq E_{h_{11}}$ | $E_{h_{21}} \gg E_{h_{11}}$ |
| $\text{LI}_{21}$ | 0.029 | 0.911 | 16.160 |
| $\rho_{21}$ | -0.001 | 0.993 | 0.018 |

Table B.1: Levels of Interaction and normalized cross-covariance values of the coupled, uncoupled and force coupled ANC systems for a two-node WASN.

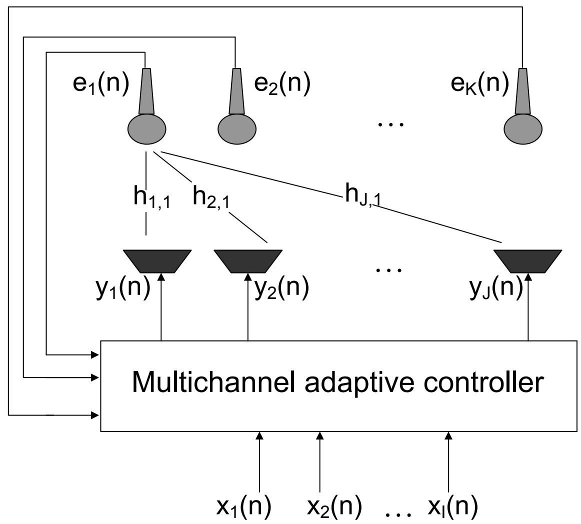



Figure B.1: Multichannel active noise controller with $K$ microphones and $J$ loudspeakers.

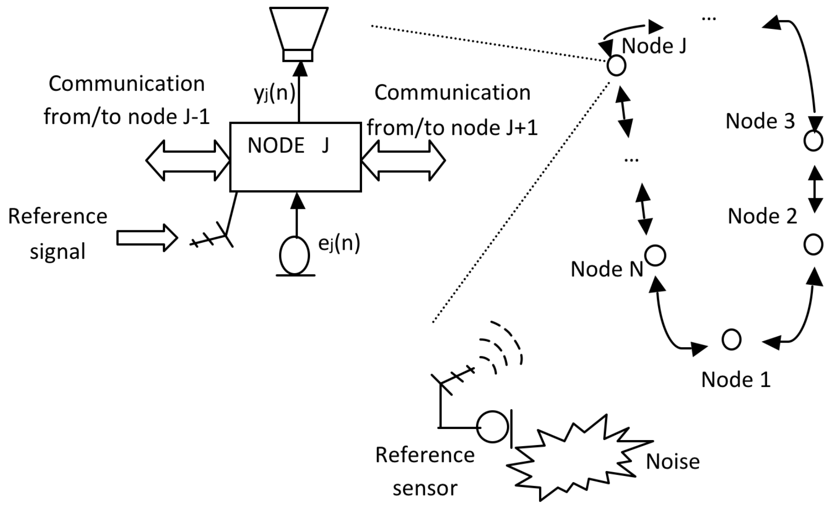



Figure B.2: Single-channel acoustic node within a ring topology network.

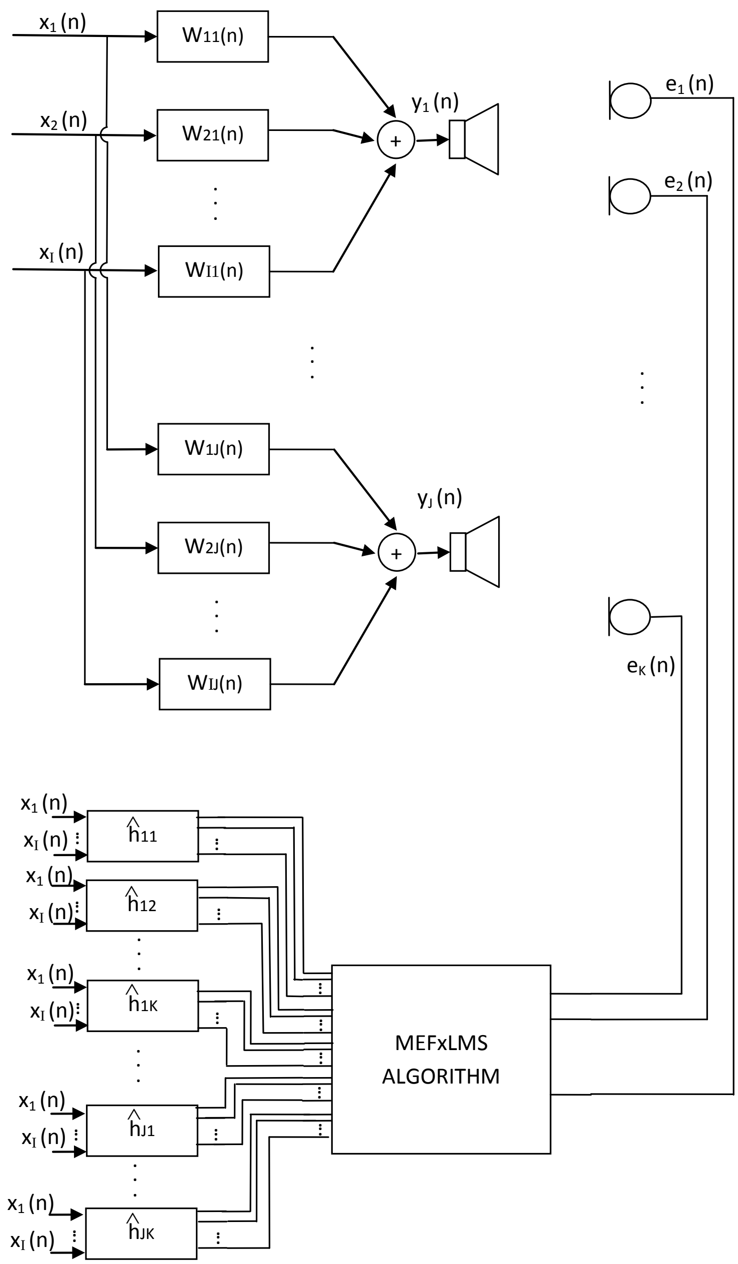
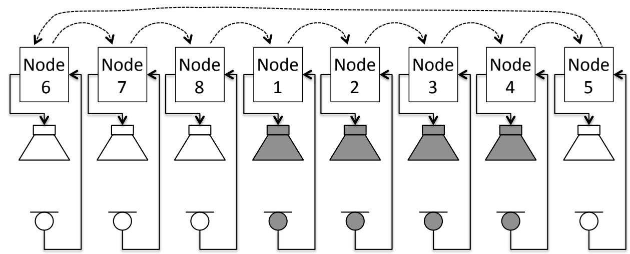
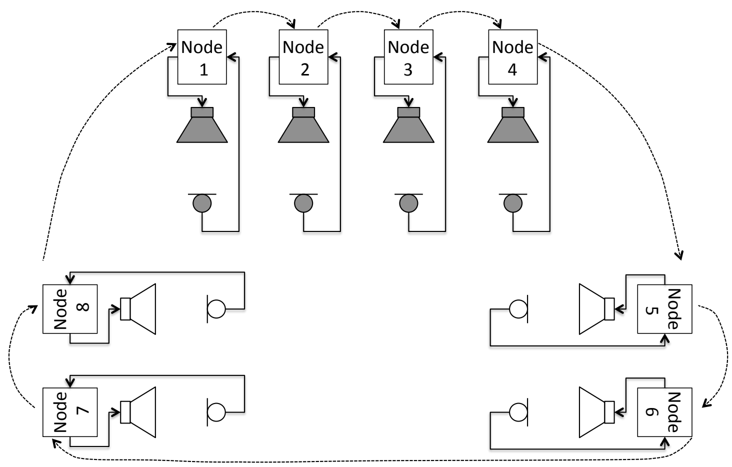



Figure B.3: Centralized multichannel ANC system based on the MEFxLMS algorithm. Filter coefficients are calculated by the MEFxLMS algorithm and then copied to the corresponding $\mathbf{w}_{ij}$ to obtain output signals $y_j(n)$.

Figure B.4: Picture of the listening room at the Audio Processing Laboratory of the Polytechnic University of Valencia.

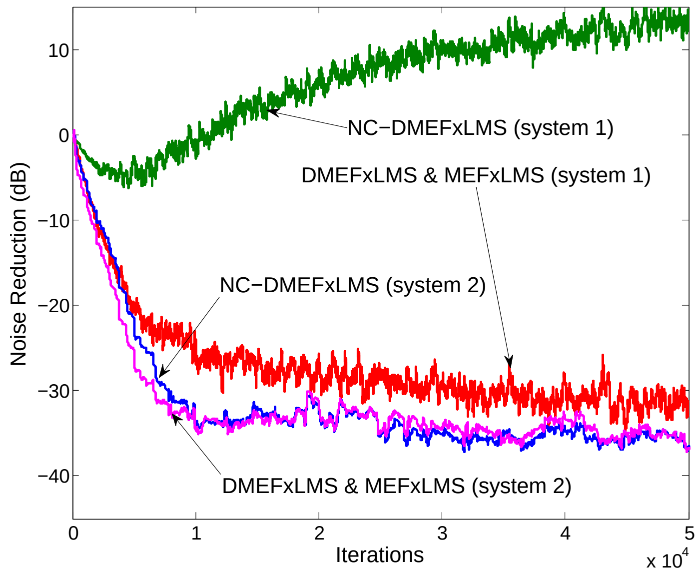



(a)

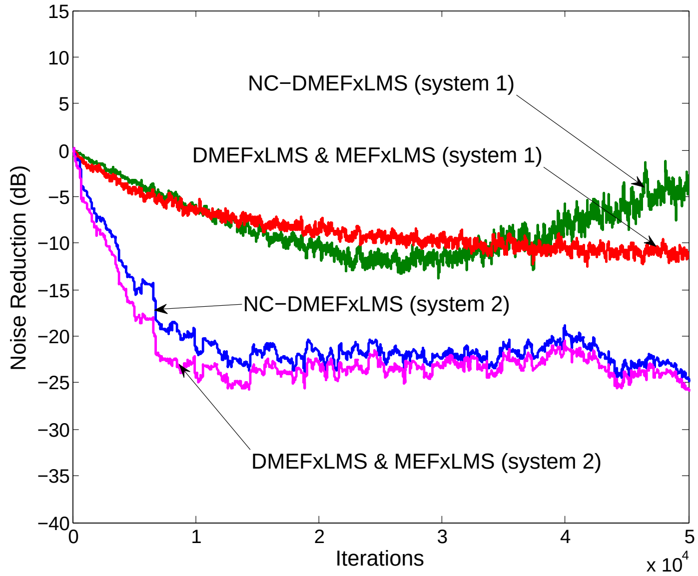



(b)

Figure B.5: Sketch of the simulated eight-node WASN described in (a) System 1 and (b) System 2. The corresponding four-node WASN was formed by nodes 1 to 4 in gray. The incremental communication strategy is represented by the dashed lines.

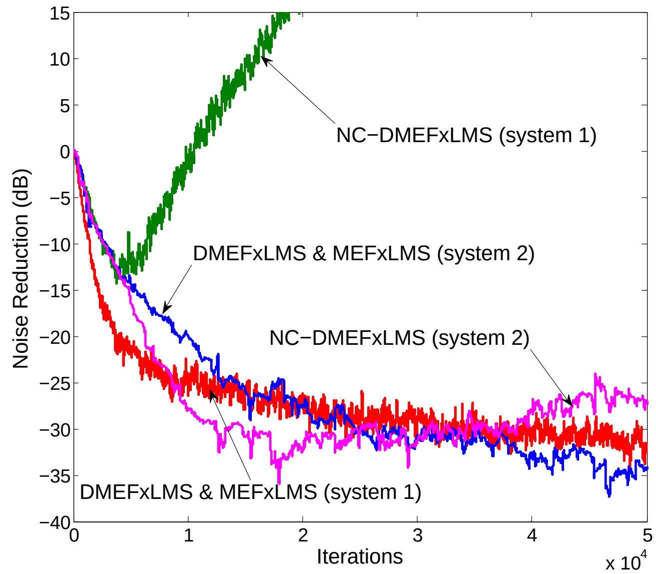



(a)

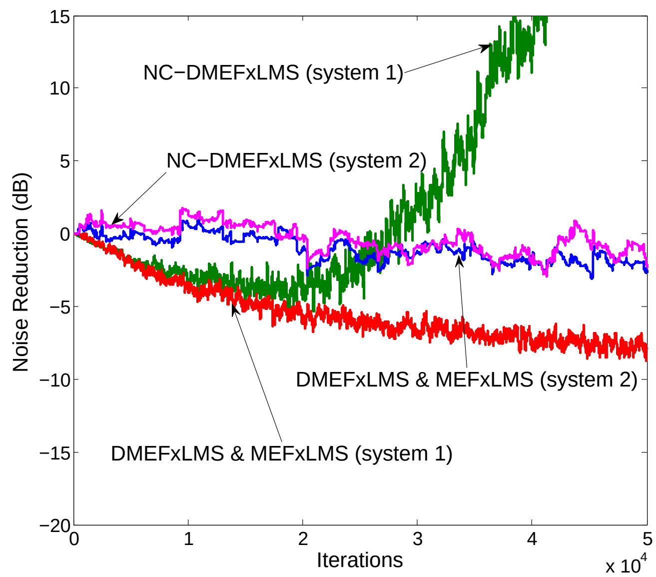



(b)

Figure B.6: Noise reduction obtained by the simulated four-node WASN's in System 1 and System 2 using the MEFxLMS, the DMEFxLMS, and the NC-DMEFxLMS algorithms. NR at the node (a) with the best performance, and (b) with the worst performance.

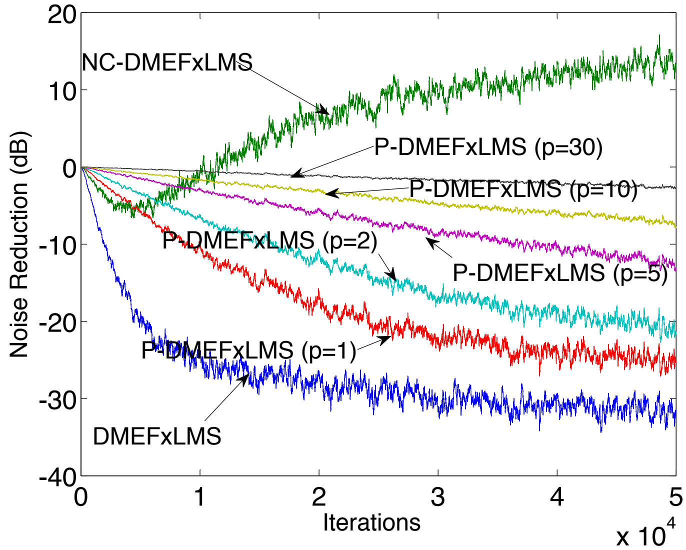



(a)

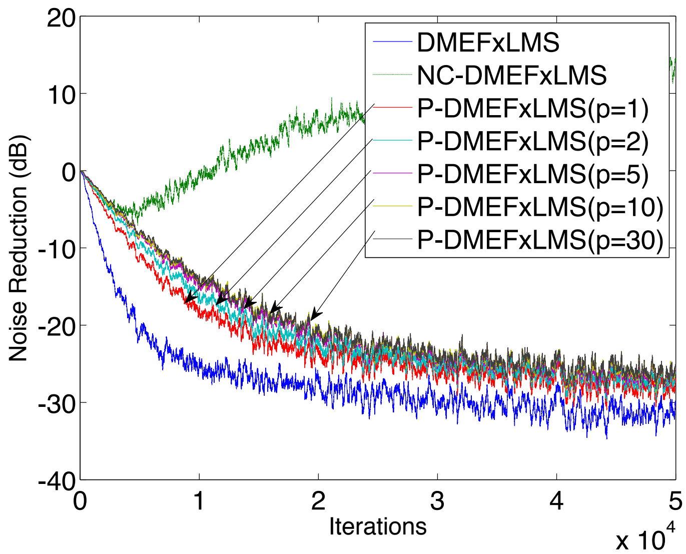



(b)

Figure B.7: Noise reduction obtained by the simulated eight-node WASN's in System 1 and System 2 using the MEFxLMS, the DMEFxLMS, and the NC-DMEFxLMS algorithms. NR at the node (a) with the best performance, and (b) with the worst performance.

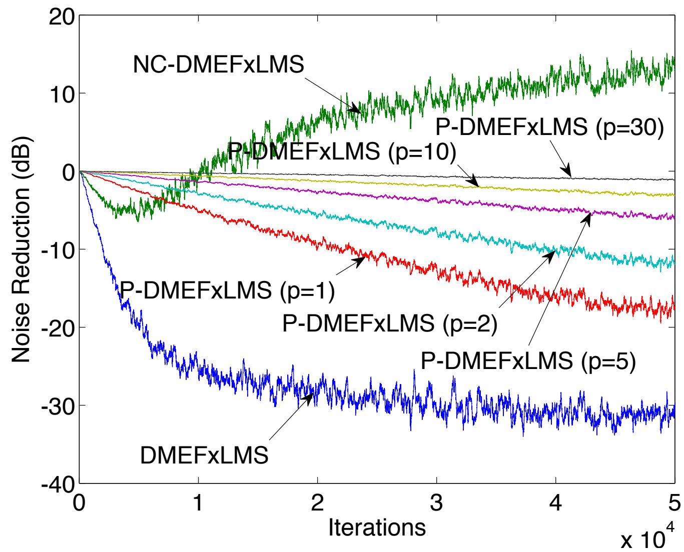



(a)

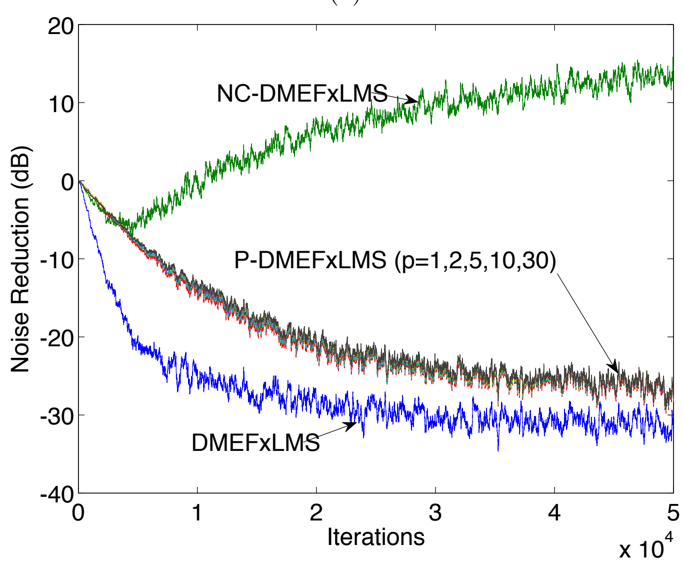



(b)

Figure B.8: Noise reduction obtained at the best node using the P-DMEFxLMS, the DMEFxLMS, and the NC-DMEFxLMS algorithms in the four-node WASN of System 1. P-DMEFxLMS algorithm uses $\alpha = 0$ and (a) $\beta = 0$ or (b) $\beta = 1$.

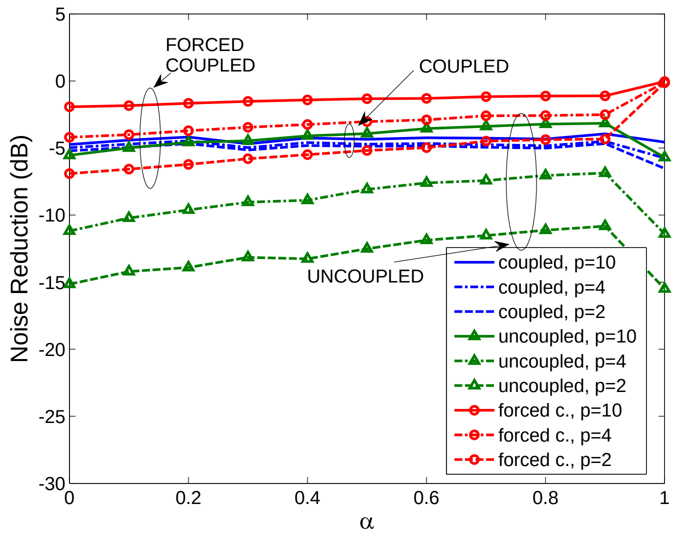



(a)

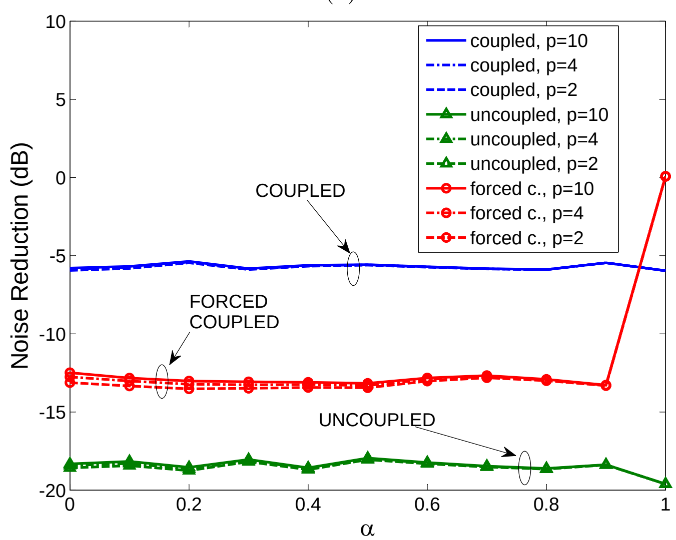



(b)

Figure B.9: Noise reduction obtained at the best node using the P-DMEFxLMS, the DME-FxLMS, and the NC-DMEFxLMS algorithms in the four-node WASN of System 1. P-DMEFxLMS algorithm uses $\alpha = 0.5$ and (a) $\beta = 0$ or (b) $\beta = 1$.




(a)




(b)

Figure B.10: Steady-state noise reduction for the P-DMEFxLMS algorithm versus $\alpha$ for the three types of ANC systems of Table B.1 over a two-node WASN. (a) $\beta = 0$ and (b) $\beta = 1$.