BLOCK-BASED DISTRIBUTED ADAPTIVE FILTER FOR ACTIVE NOISE CONTROL IN A COLLABORATIVE NETWORK

Jorge Lorente, Christian Antoñanzas, Miguel Ferrer, Alberto Gonzalez

Instituto de Telecomunicaciones y Aplicaciones Multimedia (iTEAM), Universitat Politècnica de València (UPV)

ABSTRACT

This paper considers the implementation of an Active Noise Control (ANC) system over a network of distributed acoustic nodes. Single-channel nodes composed of one microphone, one loudspeaker, and a processor with communication capabilities have been considered. An equivalent solution to the Multiple Error Filtered-x Least Mean Square algorithm (Me-FxLMS) has been chosen because is a widely used algorithm in ANC systems with centralized processing. The proposed algorithm has been implemented with block-data processing as commonly happens in practical systems. Furthermore, the algorithm works in the frequency domain and with partitioning of the filters for improving its efficiency. Therefore, we present a new formulation to introduce a distributed algorithm based on the Me-FxLMS together with an incremental collaborative strategy in the network. Results demonstrate that the scalable and versatile distributed algorithm exhibits the same performance than the centralized version. Moreover, the computational complexity and some implementation aspects have been analyzed.

Index Terms— Distributed Networks, Active Noise Control, Filtered-x Least Mean Square,

1. INTRODUCTION

A wireless acoustic sensor network (WASN) [1] is a type of wireless sensor network (WSN) [2, 3] whose sensor devices are microphones. It is a cheap, flexible and efficient solution that is generally used for monitoring acoustic fields. A network contains acoustic nodes, which are commonly composed of one or more microphones used to collect signals and a processor with some kind of communication and computation capability. The way the signals are processed in each node depends on the network topology [4].

Some applications that make use of a WASN are presented in [5] and the references therein. There, the acoustics nodes are usually used to record signals through microphones, process them and even share the signals or some local and network parameters. However, in some applications like Active Noise Control (ANC), the nodes have to act on their own environment. To this end, the nodes own loudspeakers. ANC systems [6] are based on the principle of destructive interference between a disturbance sound field called primary noise and a secondary sound field generated by controlled secondary sources called actuators. The target is to cancel, or at least minimize, the primary noise signal. To cancel the primary noise, the ANC system commonly uses adaptive algorithms to generate the secondary sound field from a reference signal that is correlated with the primary noise. For this purpose, the noise is monitored at a specific spatial area by a sensor called error sensor. Therefore, cancelation is only carried out at that specific spatial point and also at a quiet zone around the error sensor. ANC systems can be extended to multichannel ANC systems by overlapping different controlled areas and setting multiple secondary sources [7]. The multichannel ANC systems can be divided into a network with smaller multichannel nodes or single-channel nodes. Figure 1 illustrates it.

This paper presents an ANC system working over a distributed network with an incremental approach in a ring topology [8]. The ANC system is based on the well known filtered-x Least Mean Square (FxLMS) algorithm [9]. The goal is to minimize the sum of the power of the error sensors. In [10], a distributed ANC system based on the Me-FxLMS algorithm was presented in time domain with a sample-by-sample acquisition. However, most of the audio cards work with block-data buffers in practical scenarios. Moreover, a block-based implementation of the FxLMS is more efficient in the frequency domain. On the other hand, if the block size is larger than the adaptive filter, it has to be partitioned. All these reasons have led to the use of the Frequency Partitioned Block Filtered-x LMS (FPBFxLMS) algorithm [11]. Hence, the approach presented in this paper is a distributed ANC system based on the FPBFxLMS algorithm.

The paper is organized as follows: section 2 presents the FPBFxLMS algorithm for a single-channel node and extends it to a distributed ANC system of N nodes. Section 3 analyzes some implementation aspects, while the section 4 and section 5 are devoted to results and conclusions, respectively.

2. DESCRIPTION OF THE ALGORITHM

In [11], the FPBFxLMS algorithm was presented for a generic centralized ANC system with I reference signals, J sec-

This work has been supported by European Union ERDF together with Spanish Government through TEC2012-38142-C04 project, and Generalitat Valenciana through PROMETEOII/2014/003 project.

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

(a) (b)

Fig. 1. Schemes of (a) a centralized ANC system, (b) a distributed ANC system with single-channel nodes.

I	Number of reference signals	J	Number of secondary sources (actuators)
K	Number of error signals (monitoring sensors)	B	Block size
L	Length of the adaptive filters	F	L/B, number of partitions of the adaptive filters
M	Length of the FIR filters that model the acoustic paths	P	M/B, number of partitions of the estimated acoustic paths

$\mathbf{s}jk$	M-length estimation of the acoustic path that links the jth secondary source with the kth monitoring sensor
$\mathbf{S}jk^p$	FFT of size $2B$ of the pth partition of the acoustic path $\mathbf{s}jk$
$\mathbf{w}[n]$	Coefficients of the adaptive filter of length L during the nth block iteration
$\mathbf{W}[n]^f$	FFT of size $2B$ of the fth partition of the coefficients of the adaptive filter $\mathbf{w}$ during the nth block iteration

Table 1. Notation of the description of the algorithms

ondary sources, and K error sensors (I:J:K configuration). Here, we derive the centralized FPBFxLMS algorithm presented in [11] to a distributed ANC system using a ring topology with an incremental approach. The ring topology with an incremental approach means that the nodes collaborate by transmitting information to an adjacent node in a consecutive order. For the sake of simplicity, we consider one disturbance noise (I=1) and single-channel nodes (J=K=1). Therefore, each node is composed of a processor, a microphone and a loudspeaker. For a better understanding, the FPBFxLMS is first adapted to a single channel node, and then, it is extended to a network with N nodes.

2.1. The FPBFxLMS for a single-channel node

The algorithm processes samples by blocks of size B. L is the length of the adaptive filter $\mathbf{w}$, and M is the length of the FIR filters that model the estimated secondary paths $\mathbf{s}$. If L and M are larger than B, both $\mathbf{w}$ and $\mathbf{s}$ are partitioned into F and P partitions, respectively. Thus, the superscript of the following notation denotes the number of partition, and the index between brackets denotes the block iteration. The notation in Table 1 is used to describe the algorithm. According to the notation, the adaptive filter output is calculated as follows

$$\mathbf{Y}[n] = \sum\nolimits_{f=1}^{F} \mathbf{W}^f[n] \circ \mathbf{X}[n-f+1], \tag{1}$$

where $\mathbf{X}[n] = \mathrm{FFT}\{[\mathbf{x}_B[n-1] \quad \mathbf{x}_B[n]]\}$, and $\mathbf{x}_B[n] = [x(Bn) \quad x(Bn-1) \quad \dots \quad x(Bn-B+1)]$. Vector $\mathbf{W}^f[n]$ is the FFT of size $2B$ of the fth partition of $\mathbf{w}$ at the nth block iteration, and $\circ$ denotes the element-wise product of two vectors. The valid samples of the adaptive filter output $\mathbf{y}_B[n]$ are the last B samples of $\mathrm{IFFT}\{\mathbf{Y}[n]\}$.

The filter coefficients are updated in the frequency domain by calculating the correlations between the reference signal ($\mathbf{X}[n]$) that is filtered through the estimated secondary path ($\mathbf{S}^p$), $\mathbf{V}[n]$, and the error signal, $\mathbf{e}_B[n]$. To this end, the following operations are performed

$$\mathbf{V}[n] = \sum\nolimits_{p=1}^{P} \mathbf{S}^p \circ \mathbf{X}[n-p+1], \tag{2}$$

$$\widetilde{\boldsymbol{\mu}}^f[n] = \mathbf{E}[n] \circ \mathbf{V}[n-f+1]^*, \tag{3}$$

where

$$\mathbf{E}[n] = \mathrm{FFT}[\mathbf{0}_B \quad \mathbf{e}_B[n]]. \tag{4}$$

The update of the coefficients of each partition of the adaptive filter at the nth block iteration is calculated as follows

$$\mathbf{W}^f[n+1] = \mathbf{W}^f[n] - \mu \mathrm{FFT}\{[\boldsymbol{\phi}^f[n] \quad \mathbf{0}_B]\}, \tag{5}$$

where μ is the step-size parameter, and the vector $\boldsymbol{\phi}^f[n]$ corresponds to the first B samples of the 2B-IFFT of the partition $\widetilde{\boldsymbol{\mu}}^f[n]$

$$\mathrm{IFFT}\{\widetilde{\boldsymbol{\mu}}^f[n]\} = [\boldsymbol{\phi}^f[n] \quad \bar{\boldsymbol{\phi}}^f[n]]. \tag{6}$$

Equations (3)-(6) are performed for each partition (f=1,...,F).

2.2. The FPBFxLMS for a distributed ANC system

The proposed distributed ANC system is composed of N single-channel nodes, and therefore, N error sensors and N secondary sources. Now, there exists a global state network, which is defined by N adaptive filters, one of each node. The global network adaptive filter, $\underline{\mathbf{W}}[n]$, can be defined as

$$\underline{\mathbf{W}}[n] = [\mathbf{W}_1[n], \mathbf{W}_2[n], \dots, \mathbf{W}_N[n]], \tag{7}$$

$$\mathbf{W}_k[n] = [\mathbf{W}_k^1[n], \mathbf{W}_k^2[n], \ldots, \mathbf{W}_k^F[n]], \qquad (8)$$

where $\underline{\mathbf{W}}[n]$ is a $[2B \times FN]$ matrix composed of the concatenation of the adaptive filter of each node. Matrix $\mathbf{W}_k[n]$ of size $[2B \times F]$ is the adaptive filter of the *k*th node, and vector $\mathbf{W}_k^f[n]$ of size $2B$, is the *f*th partition of the adaptive filter in frequency domain of the *k*th node. For calculating the output signal, each node takes its filters from the global filters when they are completely adapted. Hence, the *k*th node uses $\underline{\mathbf{W}}_N[n]_{(:,1+(k-1)F:kF)}$ to calculate its output signal like in (1). Moreover, we define

$$\underline{\mathbf{V}}_k[n] = [\mathbf{V}1_k[n], \mathbf{V}2_k[n], \ldots, \mathbf{V}N_k[n]], \qquad (9)$$

where $\underline{\mathbf{V}}_k[n]$ is a matrix of size $[2B \times FN]$ of the *k*th node. It is composed of the concatenation of the reference signal filtered through all the secondary paths that links the *j*th loudspeaker with the *k*th microphone ($\mathbf{S}jk$ for $j = 1, \ldots, N$). Each node knows the estimation of the secondary paths that links all the loudspeakers of the other nodes with its sensor. This means that the *k*th node knows vectors $\mathbf{S}jk$ for $j = 1, \ldots, N$. Matrix $\mathbf{V}j_k[n]$ of size $[2B \times F]$ is calculated as stated in (2) for each secondary path and each partition

$$\mathbf{V}j_k[n] = [\mathbf{V}j_k^1[n], \mathbf{V}j_k^2[n], \ldots, \mathbf{V}j_k^F[n]]. \qquad (10)$$

Furthermore, each node takes its error signal from its sensor to form its error vector as in (4). Then, each node replicates its error vector FN times forming the matrix $\underline{\mathbf{E}}_k[n]$ of size $[2B \times FN]$. Equation (3) is redefined for the *k*th node as

$$\underline{\widetilde{\boldsymbol{\mu}}}_k[n] = \underline{\mathbf{E}}_k[n] \circ \underline{\mathbf{V}}_k[n]^*. \qquad (11)$$

Matrix $\underline{\widetilde{\boldsymbol{\mu}}}_k[n]$ of size $[2B \times FN]$ is used at each node to calculate the adaptation matrix of the node, $\underline{\boldsymbol{\Psi}}_k[n]$, as

$$\mathrm{IFFT}\{\underline{\widetilde{\boldsymbol{\mu}}}_k[n]\} = [\underline{\boldsymbol{\phi}}_k[n] \quad \underline{\bar{\boldsymbol{\phi}}}_k[n]], \qquad (12)$$

$$\underline{\boldsymbol{\Psi}}_k[n] = \mathrm{FFT}\{[\underline{\boldsymbol{\phi}}_k[n] \ ; \ \mathbf{0}_{[B\times FN]}]\} \qquad (13)$$

where $\underline{\boldsymbol{\phi}}_k[n]$ and $\mathbf{0}_{[B\times FN]}$ are matrices of size $[B \times FN]$. Moreover, the operators FFT and IFFT perform direct and inverse fast fourier transforms of size 2B of each column of the matrices involved. Finally, each node calculates its own estimate of the global adaptive filters using the global estimate of the previous node, and its own adaptation matrix $\underline{\boldsymbol{\Psi}}_k[n]$. The global adaptive filters are adapted at the *k*th node as

$$\underline{\mathbf{W}}_k[n+1] = \underline{\mathbf{W}}_{k-1}[n] - \mu \underline{\boldsymbol{\Psi}}_k[n]. \qquad (14)$$

Once all the nodes have finished the actualization of the filters, the global updated vector $\underline{\mathbf{W}}_N[n]$ is disseminated to the rest of the nodes for the next iteration. Moreover, note that $\underline{\mathbf{W}}_0[n] = \underline{\mathbf{W}}_N[n-1]$.

3. IMPLEMENTATION ASPECTS

The sampling rate (f_s) and the block size (B) are two important parameters when thinking about a real-time implementation of a distributed ANC system. B describes the number of transferred discrete-time samples per iteration and thereby determines the latency of the algorithm. The latency is the time spent from when the input-data buffer is filled up until this data buffer is processed and sent back to the output-data buffer. We refer to the time spent to fill up the input-data buffers as t_{buff}, and is defined as B/f_s. The choice of these parameters is crucial for the performance of the system because there are two conditions that must be satisfied:

- **The real-time condition**. The application works in real time if $t_{proc} < t_{buff}$, where t_{proc} is the execution delay. In a centralized ANC system t_{proc} is the processing delay of the algorithm. In a distributed ANC system, it also includes the delays of transmitting the global network state between the nodes. However, each node process the algorithm simultaneously except the addition of the global network state of the previous node. Moreover, as we consider single-channel nodes, each node has to process less operations than a multichannel centralized system.
- **The causality condition**. The algorithm has to satisfy the condition $t_{buff} + \tau_s < \tau_n$ [12], where τ_s is the maximum delay of the secondary paths that join the actuators with the error sensors, and τ_n is the minimum delay of the paths that join the noise source with the error sensors. This condition guarantees the causality of the system.

It is obvious that the time t_{proc} increases with the number of nodes. If the time t_{proc} increases, the time t_{buff} must increase in order to satisfy the real-time condition. It means that, with a fixed f_s, the block size must be increased. Section 4.1 will show that when B increases, the convergence performance of the algorithm gets worse, until the causality condition is no longer satisfied. Therefore, it seems that some kind of trade off between these parameters must be considered in order to satisfy both conditions.

4. RESULTS

Some experiments were performed to validate the distributed ANC system. In a first stage, both the noise reduction and the convergence performance of the distributed ANC system are evaluated and compared with the centralized ANC system. In a second stage, we evaluate and compare the computational complexity of both ANC systems.

4.1. Simulation Results

In this section, some simulation results are presented to validate the performance of the FPBFxLMS algorithm in a distributed network with a ring topology and an incremental approach. The simulations have been carried out using real acoustic channels between microphones and loudspeakers sampled at 2kHz. This channels have been measured in a listening room. Some examples of the impulse responses of this listening room are available at [13]. We have considered a zero-mean Gaussian random noise with unit variance as the

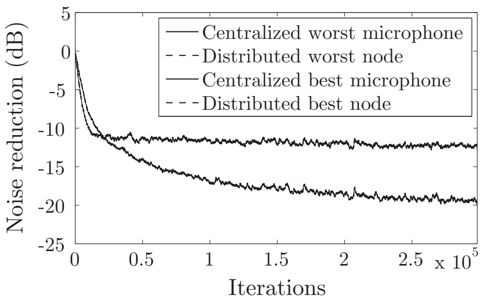

Fig. 2. Noise reduction of the distributed system with 4 single-channel nodes and the centralized system with a 1:4:4 configuration represented for the best and worst microphone.

disturbance noise. Furthermore, a block-size of $B = 512$ and a filter length of $L = 1024$ have been considered. This means that two partitions are carried out. In order to evaluate the performance of the algorithm, we define the instantaneous Noise Reduction ratio at the *k*th node as

$$NR_k[n] = 10 \log_{10} \left(\frac{e_k^2[n]}{d_k^2[n]} \right), \tag{15}$$

where $e_k[n]$ and $d_k[n]$ are the signals measured at the *k*th microphone with and without the ANC operation, respectively. Moreover, the power of these signals have been estimated using an exponential windowing.

First, we compare the noise reduction of a square centralized ANC system with a 1:4:4 configuration and a distributed ANC system with 4 single channel nodes. Fig.2 shows the noise reduction of both the centralized and the distributed implementations of the FPBFxLMS algorithm. Fig.2 illustrates the results for the microphone with best and worst performance in the centralized implementation, and the node with the best and worst performance in the distributed implementation. As expected, the distributed implementation has exactly the same results than the centralized implementation in terms of convergence speed and final residual noise.

Another important property related with the causality condition is the stability limit. In the literature, some contributions have studied the convergence behavior of the block filtered-x LMS algorithm (BFxLMS). In [14], the maximum μ parameter that leads to the fastest convergence rate was derived as

$$0 \leq \mu < \frac{1}{B\lambda_{max}} \tag{16}$$

where λ_{max} is the maximum eigenvalue of the filtered reference signal autocorrelation matrix $\mathbf{R}_{vv}$ defined as $\mathbf{R}_{vv} = E[\mathbf{V}\mathbf{V}^T]$. Therefore, the convergence performance of the algorithm depends on the statistics of the reference signal, the acoustic paths, and the block length B. For the same reference signal, the step-size parameter μ depends on B, so the maximum μ value increases by reducing the size of B, and, consequently, the convergence speed is improved by reducing B. However, as commented in section 3, the size of B is also limited by the real-time condition. Therefore, there is a minimum value of B for a given configuration that assures the real-time condition and maximum convergence speed.

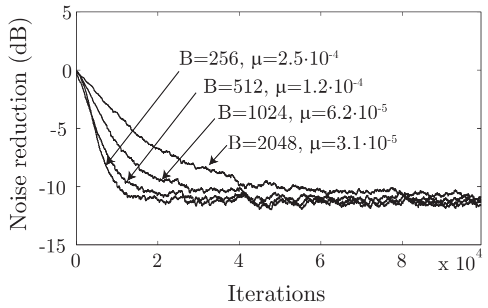

Fig. 3. Noise reduction of the distributed algorithm for different size of B.

		Generic	N=1	N=4	N=8
(1)	MUX	$4LN + 4LN^2$	$8L$	$80L$	$288L$
	ADD	$LN + 3LN^2$	$4L$	$52L$	$200L$
	FFTs	$2 + 6N$	8	26	50
(2)	MUX	$2L + 6LN$	$8L$	$26L$	$50L$
	ADD	$L + 3LN$	$4L$	$13L$	$25L$
	FFTs	$3 + 5N$	8	23	43

Table 2. Total number of multiplications (MUX), additions (ADD), and FFTs per iteration of the implementation of the FPBFxLMS algorithm in (1) a centralized ANC system and (2) a distributed ANC system.

Fig. 3 illustrates the convergence behavior of the worst node in a distributed network of four nodes when the size of B changes between 256 and 2048. As expected, it shows that the algorithms converge faster with a smaller block size, B. As these results show, the maximum μ is more or less doubled when B is halved. This fact can be explained from (16), where, for the same reference signal, the maximum μ is doubled by reducing the size of B by half.

4.2. Computational complexity

Table 2 compares the computational complexity in terms of multiplications, additions, and FFTs per iteration of the FPBFxLMS algorithm implemented for a centralized and a distributed ANC system. For the centralized implementation, we consider a multichannel ANC system with one disturbance noise and the same number (N) of microphones and loudspeakers (1:N:N configuration). For the distributed implementation, we consider a network of N single-channel nodes. However, we only compute the operations of one single-channel node because each node could perform all the operations independently, except the last addition of the global adaptive filters calculated by the previous node. Since we use a value of $M = L$, and $B = L/2$ (two partitions) the computational complexity only depends on L and N.

First, the third column of table 2 shows the computational complexity of both algorithms related to L and N. Then, the computational complexity is particularized for $N = 1$, $N = 4$ and $N = 8$. As expected, when $N = 1$, both implementations make the same operations. This is because both the distributed and the centralized ANC system become a single-channel system. When $N = 4$, we compare the operations of the centralized ANC system with a 1:4:4 configuration (16 channels) with the operations of a single-channel node of a network of 4 nodes. The same is done for $N = 8$. Results show that in a centralized ANC system, the computational complexity increases significatively with the number of channels. This fact constitute a bottleneck in massive multichannel ANC systems. Otherwise, the increase of computational complexity at each node in a distributed ANC system is not so significant. However, in a distributed ANC system we also have to consider the delay in transmitting the global network filters between nodes. This involves the transmission of $2L \times N$ coefficients between N nodes. As the transmission of data is done in an incremental mode, there are $(N-1)$ transmissions in each direction. Therefore, each iteration, $2L \times N$ coefficients are transmitted $2(N-1)$ times. This fact implies that the transmission speed of the network has to be considered.

5. CONCLUSIONS

An scalable and versatile distributed implementation of the FPBFxLMS algorithm for an ANC system using an incremental strategy in the network has been presented. It has been demonstrated that the proposed algorithm has the same performance than the centralized version when there are no communication constraints in the network. Moreover, some implementation aspects have been studied regarding the block size of the algorithm. In a real implementation, the value of B has to be chosen to satisfy both the real-time and the causality conditions. On the one hand, if B increases, the system has more time for processing, allowing a better exchange of information between nodes or the possibility to add more nodes to extend the quite zone. On the other hand, if B decreases, it has been proved that the algorithm converges faster.

Moreover, the computational complexity of the distributed algorithm has been studied and compared with the centralized version. Since in the distributed algorithm, each node can perform almost all the operations independently, the computational complexity is significantly reduced at each node. However, in a real implementation, the time used to transfer the network information between nodes would has to be considered. Therefore, in practical implementations, a trade-off between some aspects of the implementation like the size of B, the number of nodes (N), and the network data transfer rate have to be considered.

REFERENCES

[1] W. P. Chen, J. C. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," *Mobile Computing, IEEE Trans. on*, vol. 3, no. 3, pp. 258–271, 2004.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *Communications magazine, IEEE*, vol. 40, pp. 102–114, 2002.

[3] D. Puccinelli and M. Haenggi, "Wireless sensor networks: applications and challenges of ubiquitous sensing," *Circuits and Systems Magazine, IEEE*, vol. 5, no. 3, pp. 19–31, 2005.

[4] A. Tanenbaum and M. Van Steen, *Distributed systems*, Pearson Prentice Hall, 2007.

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

[6] S. J. Elliot and P. A. Nelson, "Active noise control," *Signal Processing Magazine, IEEE*, vol. 10, no. 4, pp. 12–35, October 1994.

[7] S. J. Elliott, I. M. Stothers, and P. A. Nelson, "A multiple error LMS algorithm and its application to the active control of sound and vibration," *Acoustics, Speech and Signal Processing, IEEE Trans. on*, vol. 35, no. 10, pp. 1423–1434, October 1987.

[8] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *Signal Processing, IEEE Trans. on*, vol. 55, pp. 4064–4077, 2007.

[9] B. Farhang-Boroujeny, *Adaptive Filters Theory and Applications*, John Wiley & Sons, Inc., 1998, New York.

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

[11] J. Lorente, M. Ferrer, M. De Diego, and A. Gonzalez, "GPU implementation of multichannel adaptive algorithms for local active noise control," *Audio, Speech and Language Processing, IEEE Trans. on*, vol. 22, no. 11, pp. 1624–1635, 2014.

[12] RA Burdisso, JS Vipperman, and CR Fuller, "Causality analysis of feedforward-controlled systems with broadband inputs," *The Journal of the Acoustical Society of America*, vol. 94, no. 1, pp. 234–242, 1993.

[13] "Audio and communications signal processing group (gtac)," *online at: http://www.gtac.upv.es.*

[14] D. Das, G. Panda, and S. Kuo, "New block filtered-X LMS algorithms for active noise control systems," *IET Signal Processing*, vol. 1, no. 2, pp. 73–81, 2007.