UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Distributed and Collaborative Processing of Audio Signals: Algorithms, Tools and Applications

Doctoral Thesis

by

Christian Antoñanzas Manuel

Supervisors:

Dr. Miguel Ferrer Contreras
Prof. Alberto González Salvador

Valencia, Spain
July 2019

To my wife, Eva,
my daughter, Noa
and my parents

Abstract

This thesis fits into the field of Information and Communications Technology (ICT), especially in the area of digital signal processing. Nowadays and due in part to the rise of the Internet of Things (IoT), there is a growing interest in wireless sensor networks (WSN), that is, networks composed of different types of devices specifically distributed in some area to perform different signal processing tasks. These devices, also referred to as nodes, are usually equipped with electroacoustic transducers, such as sensors or actuators, as well as powerful and efficient processors with communication capability. In the particular case of acoustic sensor networks (ASN), nodes are dedicated to solving different acoustic signal processing tasks, such as environmental sound monitoring, immersive audio, binaural hearing aids, noise-cancelling systems as well as audio teleconferencing. These audio signal processing applications have been undergone a major development in recent years due in part to the advances made in computer hardware and software. This has led to the development of powerful centralized processing systems that allow the number of audio channels to be increased, the control area to be extended or more complex algorithmms to be implemented, thereby improving audio quality or creating independent control over several personal sound zones. In most cases, a distributed ASN topology can be desirable due to several factors such as the limited number of channels used by the sound acquisition and reproduction devices, the convenience of a scalable system or the high computational demands of a centralized fashion. All these aspects may lead to the use of novel distributed signal processing techniques with the aim to be applied over ASNs. To this end, one of the main contributions of this dissertation is the development of adaptive filtering algorithms for multichannel sound systems over distributed networks.

Note that, for sound field control (SFC) applications, such as active noise control (ANC) or active noise equalization (ANE), acoustic nodes must be not only equipped with sensors but also with actuators in order to control and modify the sound field. However, most of the adaptive distributed networks approaches used to solve sound-field control problems do not take into account that the nodes may interfere or modify the behaviour of the rest. This is an important issue which is tackled throughout this thesis. Therefore, other important contribution of this thesis is focused on analyzing

how the acoustic system affects the behavior of the nodes within an ASN.

In cases where the acoustic environment adversely affects the system stability, several distributed strategies have been proposed for solving the acoustic interference problem with the aim to stabilize ANC control systems. These strategies are based on both collaborative and non-collaborative approaches. Implementation aspects such as hardware constraints, sensor locations, convergenge rate or computational and communication burden, have been also considered on the design of the distributed algorithms.

Moreover and with the aim to create independent-zone equalization profiles in the presence of multi-tonal noises, distributed narrowband and broadband ANE algorithms over an ASN with a collaborative learning and composed of acoustic nodes have been presented.

Experimental results are presented to validate the use of the distributed algorithms proposed in the work for practical applications. For this purpose, an acoustic simulation software has been specifically designed to analyze the performance of the developed algorithms. In this way, this simulation tool allows the transition between the initial formulation of any algorithm and its final programming on any digital signal processing platform.

Finally, the performance of the proposed distributed algorithms for multichannel SFC applications has been evaluated by means of a real practical implementation. To this end, a real-time prototype that controls both ANC and ANE applications by using collaborative acoustic nodes has been developed. The prototype consists of two personal audio control (PAC) systems composed of a car seat and an acoustic node, which is equipped with two loudspeakers, two microphones and a processor with communications capability. In this way, it is possible to create two independent noise control zones improving the acoustic comfort of the user without the use of headphones.

Keywords: multichannel soundfield control, distributed networks, acoustic sensor networks, adaptive noise control, adaptive noise equalizer, personal audio.

Resumen

Esta tesis se enmarca en el campo de las Tecnologías de la Información y las Comunicaciones (TIC), especialmente en el área del procesado digital de la señal. En la actualidad, y debido en parte al auge del Internet de los cosas (*Internet of Things*, IoT), existe un creciente interés por las redes de sensores inalámbricos (*Wireless Sensor Networks*, WSN), es decir, redes compuestas de diferentes tipos de dispositivos específicamente distribuidos en una determinada zona para realizar diferentes tareas de procesado de señal. Estos dispositivos, también conocidos como nodos, suelen estar equipados con transductores electroacústicos, como sensores o actuadores, así como con potentes y eficientes procesadores con capacidad de comunicación. En el caso particular de las redes de sensores acústicos (*Acoustic Sensor Networks*, ASN), los nodos se dedican a resolver diferentes tareas de procesado de señales acústicas, como por ejemplo, monitorización de sonido ambiental, audio inmersivo, audífonos binaurales, sistemas de cancelación de ruido o sistemas de teleconferencia, entre otros. Estas aplicaciones de procesado de audio han experimentado un importante desarrollo en los últimos años debido en parte, a los avances realizados en el campo del hardware y software informático. Esto ha llevado al desarrollo de potentes sistemas de procesado centralizado que permiten aumentar el número de canales de audio, ampliar el área de control o implementar algoritmos más complejos, mejorando así la calidad del audio o creando un control independiente sobre varias zonas personales de escucha. En la mayoría de los casos, una topología de ASN distribuida puede ser deseable debido a varios factores tales como el número limitado de canales utilizados por los dispositivos de adquisición y reproducción de audio, la conveniencia de un sistema escalable o las altas exigencias computacionales de los sistemas centralizados. Todos estos aspectos pueden llevar a la utilización de nuevas técnicas de procesado distribuido de señales con el fin de aplicarlas en ASNs. Para ello, una de las principales aportaciones de esta tesis es el desarrollo de algoritmos de filtrado adaptativo para sistemas de audio multicanal en redes distribuidas.

Es importante tener en cuenta que, para aplicaciones de control del campo sonoro (S*ound Field Control*, SFC), como el control activo de ruido (*Active Noise Control*, ANC) o la ecualización activa de ruido (*Active Noise Equalization*, ANE), los nodos

acústicos deben estar equipados, no sólo con sensores, sino también con actuadores con el fin de controlar y modificar el campo sonoro. Sin embargo, la mayoría de las propuestas de redes distribuidas adaptativas utilizadas para resolver problemas de control del campo sonoro no tienen en cuenta que los nodos pueden interferir o modificar el comportamiento del resto. Este es un tema importante que se aborda a lo largo de esta tesis. Por lo tanto, otra contribución destacable de esta tesis se centra en el análisis de cómo el sistema acústico afecta el comportamiento de los nodos dentro de una ASN. En los casos en que el entorno acústico afecta negativamente a la estabilidad del sistema, se han propuesto varias estrategias distribuidas para resolver el problema de interferencia acústica con el objetivo de estabilizar los sistemas de ANC. Estas estrategias se basan tanto en métodos colaborativos como no colaborativos. En el diseño de los algoritmos distribuidos también se han tenido en cuenta aspectos de implementación práctica tales como ciertas restricciones en el hardware, la ubicación de los sensores, la velocidad de convergencia, el coste computacional o el coste en las comunicaciones. Además, y con el objetivo de crear perfiles de ecualización diferentes en zonas de escucha independientes en presencia de ruidos multitonales, se han presentado varios algoritmos distribuidos de ANE en banda estrecha y banda ancha sobre una ASN con una comunicación colaborativa y compuesta por nodos acústicos.

Se presentan además resultados experimentales para validar el uso de los algoritmos distribuidos propuestos en el trabajo para aplicaciones prácticas. Para ello, se ha diseñado un software de simulación acústica que permite analizar el rendimiento de los algoritmos desarrollados en la tesis. De esta manera, esta herramienta de simulación permite la transición entre la formulación inicial de cualquier algoritmo y su programación final en cualquier procesador digital de señales o DSP (Digital Signal Processor). Finalmente, se ha evaluado el rendimiento de los algoritmos distribuidos propuestos mediante una implementación práctica que permite ejecutar aplicaciones multicanal de SFC. Para ello, se ha desarrollado un prototipo en tiempo real que controla las aplicaciones de ANC y ANE utilizando nodos acústicos colaborativos. El prototipo consiste en dos sistemas de control de audio personalizado (*Personal Audio Control*, PAC) compuestos por un asiento de coche y un nodo acústico, el cual está equipado con dos altavoces, dos micrófonos y un procesador con capacidad de comunicación entre los dos nodos. De esta manera, es posible crear dos zonas independientes de control de ruido que mejoran el confort acústico del usuario sin necesidad de utilizar auriculares.

Palabras clave: control multicanal del campo sonoro, redes distribuidas, redes de sensores acústicos, control adaptativo de ruido, ecualizador adaptativo de ruido, audio personal.

Resum

Esta tesi s'emmarca en el camp de les Tecnologies de la Informació i les Comunicacions (TIC), especialment en l'àrea del processat digital del senyal. En l'actualitat, i degut en part a l'auge de l'Internet dels coses (Internet of Things, IoT) , hi ha un creixent interés per les xarxes de sensors sense fil (Wireless Sensor Networks, WSN), és a dir, xarxes compostes de diferents tipus de dispositius específicament distribuïts en una determinada zona per a realitzar diferents tasques de processat de senyal. Estos dispositius, també coneguts com a nodes, solen estar equipats amb transductores electroacústics, com a sensors o actuadors, així com amb potents i eficients processadors amb capacitat de comunicació. En el cas particular de les xarxes de sensors acústics (Acoustic Sensor Networks, ASN), els nodes es dediquen a resoldre diferents tasques de processat de senyals acústics, com per exemple, monitorització de so ambiental, àudio inmersivo, audiòfons binaurales, sistemes de cancel·lació de soroll o sistemes de teleconferència, entre altres. Estes aplicacions de processat d'àudio han experimentat un important desenrotllament en els últims anys degut en part, als avanços realitzats en el camp del maquinari i programari informàtic. Açò ha portat al desenrotllament de potents sistemes de processat centralitzat que permeten augmentar el nombre de canals d'àudio, ampliar l'àrea de control o implementar algoritmes més complexos, millorant així la qualitat de l'àudio o creant un control independent sobre unes quantes zones personals d'escolta. En la majoria dels casos, una topologia d'ASN distribuïda pot ser desitjable a causa de diversos factors com ara el número limitat de canals utilitzats pels dispositius d'adquisició i reproducció d'àudio, la conveniència d'un sistema escalable o les altes exigències computacionals dels sistemes centralitzats. Tots estos aspectes poden portar a la utilització de noves tècniques de processat distribuït de senyals a fi d'aplicar-les en ASNs. Per a això, una de les principals aportacions d'esta tesi és el desenrotllament d'algoritmes de filtrat adaptatiu per a sistemes d'àudio multicanal en xarxes distribuïdes.

És important tindre en compte que, per a aplicacions de control del camp sonor (Sound Field Control, SFC) , com el control actiu de soroll (Active Noise Control, ANC) o l'ecualización activa de soroll (Active Noise Equalization, ANE) , els nodes acústics han d'estar equipats, no sols amb sensors, sinó també amb actuadors a fi de controlar i modificar el camp sonor. No obstant això, la majoria

de les propostes de xarxes distribuïdes adaptatives utilitzades per a resoldre problemes de control del camp sonor no tenen en compte que els nodes poden interferir o modificar el comportament de la resta. Este és un tema important que s'aborda al llarg d'esta tesi. Per tant, una altra contribució destacable d'esta tesi se centra en l'anàlisi de com el sistema acústic afecta el comportament dels nodes dins d'una ASN. En els casos en què l'entorn acústic afecta negativament l'estabilitat del sistema, s'han proposat diverses estratègies distribuïdes per a resoldre el problema d'interferència acústica amb l'objectiu d'estabilitzar els sistemes d'ANC. Estes estratègies es basen tant en mètodes col·laboratius com no col·laboratius. En el disseny dels algoritmes distribuïts també s'han tingut en compte aspectes d'implementació pràctica com ara certes restriccions en el maquinari, la ubicació dels sensors, la velocitat de convergència, el cost computacional o el cost en les comunicacions. A més, i amb l'objectiu de crear perfils d'ecualización diferents en zones d'escolta independents en presència de sorolls multitonales, s'han presentat diversos algoritmes distribuïts d'ANE en banda estreta i banda ampla sobre una ASN amb una comunicació col·laborativa i composta per nodes acústics.

Es presenten a més resultats experimentals per a validar l'ús dels quelcom ritmes distribuïts proposats en el treball per a aplicacions pràctiques. Per a això, s'ha dissenyat un programari de simulació acústica que permet analitzar el rendiment dels algoritmes desenrotllats en la tesi. D'esta manera, esta ferramenta de simulació permet la transició entre la formulació inicial de qualsevol algoritme i la seua programació final en qualsevol processador digital de senyals o DSP (Digital Signal Processor) . Finalment, s'ha avaluat el rendiment dels algoritmes distribuïts proposats per mitjà d'una implementació pràctica que permet executar aplicacions multicanal de SFC. Per a això, s'ha desenrotllat un prototip en temps real que controla les aplicacions d'ANC i ANE utilitzant nodes acústics col·laboratius. El prototip consistix en dos sistemes de control d'àudio personalitzat (Personal àudio Control, PAC) compostos per un assentisc de cotxe i un node acústic, el qual estaÂ´ equipat amb dos altaveus, dos micròfons i un processador amb capacitat de comunicació entre els dos nodes. D'esta manera, és possible crear dos zones independents de control de soroll que milloren el confort acústic de l'usuari sense necessitat d'utilitzar auriculars.

Paraules clau: control multicanal del camp sonor, xarxes distribuïdes, xarxes de sensors acústics, control adaptatiu de soroll, ecualizador adaptatiu de soroll, àudio personal.

Acknowledgements

It is a pleasure for me to thank those who made this thesis possible. First and foremost, I offer my sincerest gratitude to my supervisors, Dr. Miguel Ferrer and Prof. Alberto González, for supporting me throughout this thesis with their knowledge and advice, for the numerous hours of proofreading and for allowing me to work in my own way.

I am very grateful to the reviewers and members of the committee of this thesis, Dr. Máximo Cobos Serrano from Universidad de Valencia, Dr. Efrén Fernández Grande from Technical University of Denmark, Dr. Jorge P., Prof. Carlos León de Mora from Universidad de Sevilla and Prof. Antonio M. Vidal Macia from Universitat Politècnica de València. Many thanks for having provided me very useful comments that helped to improve the quality of the thesis. I deeply appreciate the time they dedicated to review this thesis.

I owe my deepest gratitude to Dr. María de Diego for their continuous support and collaboration, which have contributed in a very important way to the development of this thesis. Likewise, I would like to thank to Prof. Gema Piñero for her support. I would like to thank Prof. Marc Moonen for providing me the possibility to work as a visiting researcher in KU Leuven during some months. I really appreciate the opportunity he gave me to work within his team. I wish also to thank the Spanish Ministry of Education for the received financial support under the FPI program.

I would like to show my gratitude to all the people of the Universitat Politècnica de València that shared my daily work at the Telecommunications and Multimedia Applications Institute (iTEAM). Thanks to Prof. Jose J. López, Dr. Paco Martínez, Dr. Laura Fuster, Dr. Jose A. Belloch and Dr. Marian Simarro. Very special thanks go to my colleagues of the Audio and Communications Signal Processing Group (GTAC). In particular, thank you to Dr. Laura Fuster, Dr. Marian Simarro, Emmanuel Aguilera, Pablo Gutierrez, Juan Estreder, Vicent Molés, Fabián Aguirre and Jorge Lorente.

Finalmente y sin lugar a dudas, mi más profunda gratitud va hacia mi familia. A mis queridos padres, Pilar y Jóse Ignacio por su apoyo incondicional y por su amor. Soy lo que soy gracias a vuestro esfuerzo y vuestro cariño. A mi hermano Iván, a

mi cuñada Rebeca y a mi sobrino Marcos, por ayudarme siempre que lo necesito y hacerme la vida más feliz. A mi abuelos Lali y Vitín por su apoyo y ternura desde la distancia. Y por último, mi más sincero amor y gratitud a mi mujer Eva. He crecido contigo durante estos años tanto en lo personal como en lo profesional y este trabajo es en gran parte tuyo. Sin tu ayuda, paciencia y cariño, no lo hubiera conseguido. Tú y nuestra hija Noa, sois lo mejor que me ha pasado en la vida. Os quiero.

Christian Antoñanzas
July 2019

Contents

List of Figures

List of Tables

List of symbols

$\mathbf{X}$	Matrix
$\mathbf{x}$	Vector
x	Scalar
$(\cdot)^{\mathrm{T}}$	Transpose
$(\cdot)^{*}$	Complex conjugation
$(\cdot)^{\mathrm{H}}$	Hermitian or Conjugate transpose
$(\cdot)^{-1}$	Matrix inverse
$\lvert\cdot\rvert$	Absolute value
$\lVert\cdot\rVert_2$	ℓ_2 norm
$E\{.\}$	Expectation operation
$\mathrm{Tr}\{.\}$	Trace of a matrix
$\mathbb{R}e$	Real part of a complex number
$\mathbb{I}m$	Imaginary part of a complex number
λ	Eigenvalue
δ	Regularization factor
Σ	Summation
$*$	Linear convolution
$\circ$	Hadamard (element-wise) product
(n)	Discrete time index
$x(n)$	Reference signal
$y_j(n)$	Control signal at the actuator j
$e_k(n)$	Error signal at the sensor k
$d_k(n)$	Acoustic noise signal at the sensor k
$\mathbf{w}_k(n)$	Adaptive filter at node k
$\mathbf{p}_k$	Real acoustic channel between the noise source and sensor k modeled as a FIR filter
$\mathbf{h}_{jk}$	Real acoustic channel between actuator j and sensor k modeled as a FIR filter
$\tilde{\mathbf{h}}_{jk}$	Estimated acoustic channel impulse response which is defined as a FIR filter that models $\mathbf{h}_{jk}$
N	Number of single-channel nodes = number of actuators

	= number of sensors
L	Length of $\mathbf{w}_k(n)$
M	Length of $\tilde{\mathbf{h}}_{jk}$
Q	Projection order related to the affine projection algorithm
$\mathbf{I}_{A\times 1}, \mathbf{0}_{A\times 1}$	Vector of ones/zeros with size $[A \times 1]$
$\mathbf{I}_A, \mathbf{0}_A$	Identity/null matrix of size $[A \times A]$
$\mathbf{I}_{A\times B}, \mathbf{0}_{A\times B}$	Identity/null matrix of size $[A \times B]$
$\underline{x}$	Frequency domain notation
∇	Gradient operator
$\lim_{n\to\infty}$	Limit function as n approaches ∞
μ	Step-size parameter
$1{:}N{:}N$	Multichannel configuration of one noise sourcel, N actuators and N sensors
$[.]_{[1:B]}$	First B elements of the vector between brackets
$[.]_{(:,1)}$	Vector formed by the first column of the matrix between brackets
$NR_k(n)$	Noise Reduction at node k
$NR(n)$	Noise Reduction of the network
$NR_{f_m}(n)$	Steady-state Noise Reduction of the network at excitation frequency f_m

Partitioned block implementation:

$[n]$	Block iteration index
B	Block size
F	L/B, number of partitions of the adaptive filters
P	M/B, Number of partitions of the estimated secondary paths
f, p	super-indexes that denote partition number
$\underline{\mathbf{H}}_{jk}^{p}$	FFT of size $2B$ of the p-th partition of the acoustic path h_{jk}
α	Collaboration parameter

Clustering implementation:

α_{jk}	Collaboration parameter between the node j and the node k
C_j	Set of nodes related with the j-th node
N_j	Size of C_j
$\in$	is member of
$\notin$	is not member of
col[.]	Column vector
$C_k(l)$	l-th node within the set C_k
$\emptyset$	Empty set
max{.}	Maximum of a set or vector

Remote microphone technique:

$e_{\mathrm{m,k_m}}(n)$	Error signal at the monitoring sensor k_{m}
$e_{\mathrm{v,k_v}}(n)$	Error signal at the virtual sensor k_{v}
$d_{\mathrm{m,k_m}}(n)$	Acoustic noise signal at the monitoring sensor k_{m}
$d_{\mathrm{v,k_v}}(n)$	Acoustic noise signal at the virtual sensor k_{v}
$\mathbf{p}_{\mathrm{m,k_m}}, \mathbf{p}_{\mathrm{v,k_v}}$	Acoustic channels between the noise source and sensors k_{m} and k_{v}, respectively
$\mathbf{h}_{\mathrm{m,jk_m}}, \mathbf{h}_{\mathrm{v,jk_v}}$	Acoustic channels between actuator j and sensors k_{m} and k_{v}, respectively
$\tilde{\mathbf{h}}_{\mathrm{m,jk_m}}, \tilde{\mathbf{h}}_{\mathrm{v,jk_v}}$	Impulse responses defined as FIR filters that estimates $\mathbf{h}_{\mathrm{m,jk_m}}$ and $\mathbf{h}_{\mathrm{v,jk_v}}$, respectively
K_{m}	Number of monitoring sensors
K_{v}	Number of virtual sensors

ANE systems:

(z)	z-transform domain
β_k	Equalization profile at node k
ω	Angular frequency
$\widehat{x}(n)$	Quadrature component
cos	Cosine trigonometric function
sin	Sine trigonometric function
A	Amplitude
ϕ_0	Initial phase
$\lvert X(z)\rvert$	Magnitude of a complex number
$\angle X(z)$	Phase of a complex number
I	Number of frequencies of a tonal noise

Abbreviations and Acronyms

WSN	Wireless Sensor Network
WMSN	Wireless Multimedia Sensor Networks
WASN	Wireless Acoustic Sensor Network
ASN	Acoustic Sensor Network
SFC	Sound Field Control
ANC	Active Noise Control
ANE	Active Noise Equalization
PAC	Personal Audio Control
SCN	Sound Control Node
CU	Central Unit
CPU	Communication and Processing Unit
MSE	Mean Square Error
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
IFFT	Inverse Fast Fourier Transform
FIR	Finite Impulse Response
FM	Frequency Missmatch
MUX	total number of Multiplications
ADD	Additions
RM	Remote Microphone
HT	Hearing Threshold
1r	one round
SNR	Signal-to-Noise Ratio
SPL	Sound Pressure Level
EQ	Equalization
NB	Narrowband
BB	Broadband
ACE	Array Control Effort
LI	Level of Interaction
EfI	Effective independence
EfI	Effective independence

s	second
dB	decibel
Hz	Hertz
kB	kilobytes
MB	Megabytes
kBps	kB per second
MBps	MB per second
i.e. .	id est, that is
IR	Interference ratio

Algorithms:

LMS	Least Mean Square
RLS	Recursive Least Square
NLMS	Normalized LMS
AP	Affine Projection
BLMS	Block LMS
FBLMS	Fast BLMS
PFBLMS	Partitioned FBLMS
FxLMS	Filtered-x LMS
BFxLMS	Block FxLMS
FeLMS	Filtered-error LMS
MEFxLMS	Multiple-error FxLMS
MEFeLMS	Multiple-error FeLMS
NC-DFxLMS	Non-Collaborative Distributed FxLMS
l-NC-DFxLMS	leaky NC-DFxLMS
icNC-DFxLMS	NC-DFxLMS using interference control
QoS NC-DFxLMS	NC-DFxLMS using quality of service
NC-DFxAP	Non-Collaborative Distributed Filtered-x Affine Projection
DMEFxLMS	Distributed MEFxLMS
l-DMEFxLMS	Leaky DMEFxLMS
APL	Affine-Projection-Like
APL-I	Affine-Projection-Like I
FxAP	Filtered-x AP
FxAPL-I	Filtered-x APL-I
DFxAPL-I	Distributed FxAPL-I
FPBFxLMS	Frequency-domain Partitioned BFxLMS
Incremental DFPBFxLMS	Incremental Distributed FPBFxLMS
Diffusion DFPBFxLMS	Diffusion Distributed FPBFxLMS
NC DFPBFxLMS	Non-Collaborative Distributed FPBFxLMS
cm-MEFxLMS	coupling mask MEFxLMS

cm-DMEFxLMS	coupling mask DMEFxLMS
cl-DMEFxLMS	clustering non-disjoint DMEFxLMS
RM-MEFxLMS	Remote Microphone MEFxLMS
RM-DMEFxLMS	Remote Microphone DMEFxLMS
pnc-RM-DMEFxLMS	partially non-collaborative RM-DMEFxLMS

Chapter 1

Introduction and Scope

1.1 Motivation

This thesis has been developed at the Audio and Communications Signal Processing Group (GTAC) of the Institute for Telecommunications and Multimedia Applications (iTEAM) of the Universitat Politecnica de Valencia (UPV) and it has been supported by the following European Union and Spanish Government projects:

- TEC2012-38142-C04-01 project. DisCoSound: Distributed and Collaborative Sound Signal Processing: algorithms, tools and applications.
- TEC2015-67387-C4-1-R project. Sspressing: Smart Sound Processing for the Digital Living.
- H2020-FETOPEN-4-2016-2017 FET Innovation Launchpad. 754576 project. Dnoise: Distributed Network of Active Noise Equalizers for Multi-User Sound Control.

Taking advantage of GTAC's experience in soundfield control applications, this thesis has focused on the analisys and development of adaptive control algorithms for audio signal processing applications over distributed networks.

Although there exist recent literature on distributed adaptive algorithms for adaptive networks, in most cases they have not been applied to specific problems. The processing and control of sound fields represents a novel challenge if distributed strategies are considered. Sound control systems aim to modify the sound field within a certain area by using loudspeakers and microphones to create local listening zones. Massive multichannel systems arise from the need to expand the area to be controlled.

Most of these systems are executed in a centralized controller ruled by algorithms which have access to all the system signals. This requires high-performance computing hardware capable of managing, processing and generating multiple signals which can lead to not very flexible systems as well as very sensitive ones to failures of the central processor. For this reason, one of the objectives of this thesis is to develop multichannel sound control systems working in a distributed way. This can be done by adapting centralized control algorithms, by using current distributed algorithms applied in other disciplines or by proposing new strategies. In this way, multichannel sound control systems would be provided with higher flexibility, facilitating the use of low-cost devices for their implementation.

Although distributed sensor networks have been widely used in acoustic applications such as monitoring of environmental parameters [1] or industrial monitoring and control [2] among others, their use for multichannel soundfield control systems has not been addressed up to the beginning of this thesis. Traditional distributed sensor networks are usually composed of passive devices or nodes equipped with microphones connected to a processor, with some kind of communication and computation capability, plus a transmitter-receiver device. These networks focus on the estimation of a signal or parameter that can be measured by all the nodes. However, for applications involving sound control, nodes should have the capacity of acting on the environment by emitting sounds through a loudspeaker or actuator (see Figure 1.1). On the other hand, distributed systems can be particularly suitable for massive multichannel control systems where the total computational cost becomes unapproachable by a centralized system (as long as the required algorithms can be implemented in a distributed manner). In addition, distributed systems are more flexible, more versatile and easier scalable than the centralized ones.

In this thesis we focus on the development of multichannel control systems for application in distributed networks. Initially, the multiple-error least-mean-square algorithm with the filtered-x structure (MEFxLMS) [3] has been adapted in a distributed network, deriving its original centralized approach into a distributed adaptive algorithm. It should be noted that the least-mean-square (LMS) algorithm [4], on which the MEFxLMS algorithm is based, has been already proposed for its application in distributed systems [5, 6]. However, it had not been already studied for sound field control (SFC) applications over these type of networks. Note that the objective of traditional distributed networks (usually the estimation of a measurement) is independent of the number of nodes or their location, since a single node could provide by itself a good solution to the global system. However, in distributed systems devoted to acoustic control by means of active elements, the final solution of the system depends on the number of nodes that intervene in it. When nodes working independently are not capable of obtaining such a solution, collaboration among them is required with the aim to reach the best common solution for a given configuration (solution which each node separately could not achieve).

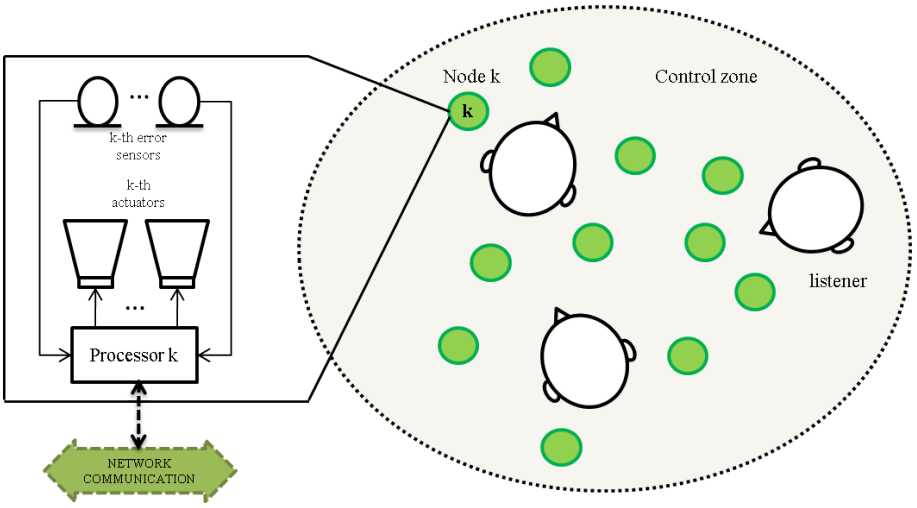


Figure 1.1: Distributed network composed of multiple acoustic nodes designed for a soundfield control application.

Consequently, our objective is to take advantage of the benefits of the acoustic sensor networks to implement multichannel soundfield control systems in real environments. The use of adaptive distributed control algorithms will improve the behaviour of several multichannel audio applications which have been until now implemented by means of centralized control algorithms, such as local sound field control systems or real-time auralization systems. Moreover, distributed control will be able to overcome centralized systems limitations related to high computational burden and management of high amount of data in multichannel scenarios. In this way, distributed processing allows to add certain data redundancy avoiding the exclusive dependence of a central processing unit.

1.2 Objectives

Considering these aspects, this thesis focuses on the development and analysis of different massive SFC applications over distributed networks composed of acoustic nodes capable of interacting with the enviroment. To this end, the following particular objectives should be met:

- To compile and analyze the centralized algorithms proposed for SFC applications and study of their feasibility to apply them to distributed sensor networks.

- To analyze the acoustical interaction effects on soundfield control systems over distributed networks. This includes providing methods to be able to determine when acoustic enviroment will negatively influence on the control system stability. In the cases when this happens, to contribute with experimental techniques to try to solve this problem from both collaborative and non-collaborative points of view.

- To study the collaborative strategies between the different nodes of the network. The problem of choosing which nodes are devoted to exchange information in acoustically coupled networks should be analysed. The optimal selection of these nodes depends on the acoustic enviroment in which they are located. Therefore, the previous acoustic interaction analysis must be taken into account when selecting collaborative nodes in order to save on computational cost and data transfer in the network.

- To provide a generic formulation for the active noise control problem over distributed networks. Thus, both centralized and distributed (non-collaborative and collaborative) strategies can be derived from a generic point of view of the problem.

- To develop distributed control algorithms for SFC applications. This implies the design and choice of the algorithms executed in each node of the network for personal sound control considering possible practical constraints in real scenarios, such as hardware limitations or comfort problems, as well as with the aim to improve the processing efficiency or the convergence behavior of practical noise control system.

- To design and implement distributed systems for massive SFC applications such as active noise control (ANC) or active noise equalization (ANE). In particular, it is aimed to develop a practical implementation of personal sound control systems using collaborative acoustic nodes. This includes the design of the prototype and its integration to the system by selecting and designing the hardware and software to be used and identifying their limitations.

1.3 Organization of the thesis

This thesis describes the research that has been undertaken to develop the previous aims. The chapters are organized and presented as follows:

- Chapter 2. This chapter presents the basic knowledge of necessary concepts which will be required to understand this dissertation. It includes, among other concepts, acoustic sensor networks, distributed processing, adaptive filtering

and finally, a brief state of the art of SFC applications focusing mainly in active noise control and active noise equalization.

- Chapter 3. This chapter deals with some previous considerations to take into account before the implementation of ANC systems over distributed networks. It analyzes the influence of the acoustic system on the stability of adaptive control algorithms. In particular, the objective is to try to identify the conditions under which nodes must collaborate in a distributed network in order to avoid the SFC system unstability due to the acoustic enviroment. In addition, a generic formulation of the distributed ANC problem based on the LMS algorithm is also proposed.

- Chapter 4. This chapter develops multichannel active noise control systems over distributed networks where the adaptive control algorithms rely on LMS strategies and non-collaborative strategies. It includes several techniques which consider practical aspects in realistic scenarios.

- Chapter 5. This chapter presents the implementation of ANC systems over a network of distributed collaborative acoustic nodes. Considering the LMS method and some of its variants, several collaborative approaches are proposed with the aim tyo consider implementation constraints on real-time applications.

- Chapter 6. This chapter focuses on LMS-based distributed algorithms for acoustic sensor networs which support multichannel active noise equalization systems. Narrowband and broadband approaches are considered with the aim to create independent-zone equalization profiles.

- Chapter 7. This chapter summarizes the results obtained in the practical implementation of a personal audio protoype located whitin an enclosure of a distributed sound control system used for ANC and ANE applications. Particularly, the description and configuration of the prototype, its hardware and software integration, the practical implementation aspects to be considered and the selected distributed algorithms are presented.

- Chapter 8. Finally, the conclusions obtained throughout this thesis are presented, including some guidelines for future research lines. Also, a list of published work related to this thesis is given.

- Appendix A. The first appendix shows several methods based on the analysis of the acoustic system presented in Chapter 3 with the aim to define when nodes are acoustically coupled and then, to determine the collaborative condition.

- Appendix B. In the second appendix, an acoustic simulation software is presented as an useful tool to model sound control systems over distributed net-

works and to evaluate the performance of the several algorithms proposed in this thesis.

Chapter 2

State of the art

This chapter describes some necessary concepts for the understanding of this dissertation. Throughout the first section, acoustic sensor networks have been presented. Next, different schemes for processing data in sensor networks are stated. It also contains an introduction to the topic of adaptive filters. Finally, a brief state of the art of sound field control applications focusing mainly in active noise control and active noise equalization systems are given in the last section.

2.1 Acoustic Sensor Networks (ASN)

Recent advances in electronics are enabling the development of high performance devices increasingly smaller, less expensive and with less power requirements. These electronic devices are usually equipped with electroacoustic transducers, such as sensors and actuators, as well as powerful and efficient processors with communication capability. As wireless communication technologies become affordable, the use of this kind of devices over wireless sensor networks (WSN) [1] has been growing during the last years. A WSN consists of a set of low-power, low-cost, and small-size wireless devices, called *nodes*, specifically distributed in some area to perform a certain task. Some advantages of the WSNs compared to the traditional wired networks are scalability and low computational cost [7, 8] among others. In addition, many more sensors can be used to cover larger sound zones in order to get more information from the signals of interest. WSN are in the scope of research since the beginning of this century, although their commercial used is not as spread as it was expected. Different types of WSNs were developed for various applications, including military and security monitoring [9] or healthcare applications [10]. For the purpose of monitoring and transmission of multimedia content, Wireless Multimedia Sensor Networks (WMSN) [11] are available, which require increased computational cost,

synchronization, data transmission and energy consumption because of certain characteristics of the multimedia signals [12]. A subclass of WMSN are the *Wireless Acoustic Sensor Networks (WASN)* [13, 14]. The WASN are specifically designed for acoustic signal processing tasks and they are a popular and efficient solution for different applications in multiple acoustic areas, such as environmental audio monitoring [15,16], binaural hearing aids [17], audio surveillance [18] as well as industrial monitoring and control [2]. It seems this kind of networks will be essential for future audio signal acquisition, control and monitoring since they present great advantages such as scalability, flexibility and low computational cost [13]. For the sake of clarity, the wireless communication field is not addresed throughout this thesis. We focus on acoustic sensor networks (ASN) with communication capabilities but the type of communication (wired or wireless) is not relevant to the proper understanding of this dissertation. For these reasons, the term wireless will be omitted from now. An ASN usually consists of a set of sensor nodes specifically distributed in some area and connected to a processing unit with some kind of communication and computation capability [19]. These passive nodes are interested in the estimation of the same network signal or parameter [5] or in solving node-especific estimation problems [20] [21]. The acoustic signals captured by the sensors are recorded and transmitted by the processing unit, doing eventually some processing before the transmission.

However, if the ASN has to support an application that involves the use of both sensors and actuators to perform an acoustic signal processing task, nodes capable of measuring and generating signals are required. Furthermore, the ASN should focus not only on the estimation of a certain parameter or signal, but also on the generation of the signals that will feed the actuators in order to control and modify the sound field [22]. Therefore, it is necessary to redefine the concept of *acoustic node* in order to use it in these type of acoustic applications. Thus, we define an acoustic node as a device capable of measuring, processing, and generating signals individually as well as capable of exchanging the necessary information with other nodes using a suitable communication network. From a practical point of view, an ASN could be implemented using smartphones or tablets as acoustic nodes [23]. They can also communicate to other devices via Wi-Fi IEEE 802.11 or Bluetooth connections. In this sense, the number of audio processing applications on mobile devices has been increasing during the last years [24,25]. Nevertheless, the electroacoustic transducers of these electronic devices have limited capability, which may affect their frequency response and the power level of the captured and emitted sounds, and hence, the audio application performance.

On the implementation and design of algorithms for ASNs, many aspects require to be considered [1, 26]. Some of them are commented as follows:

- *Data stream processing*: Since data is captured from multiple locations, different data managements could be applied in sensor networks. A first approach

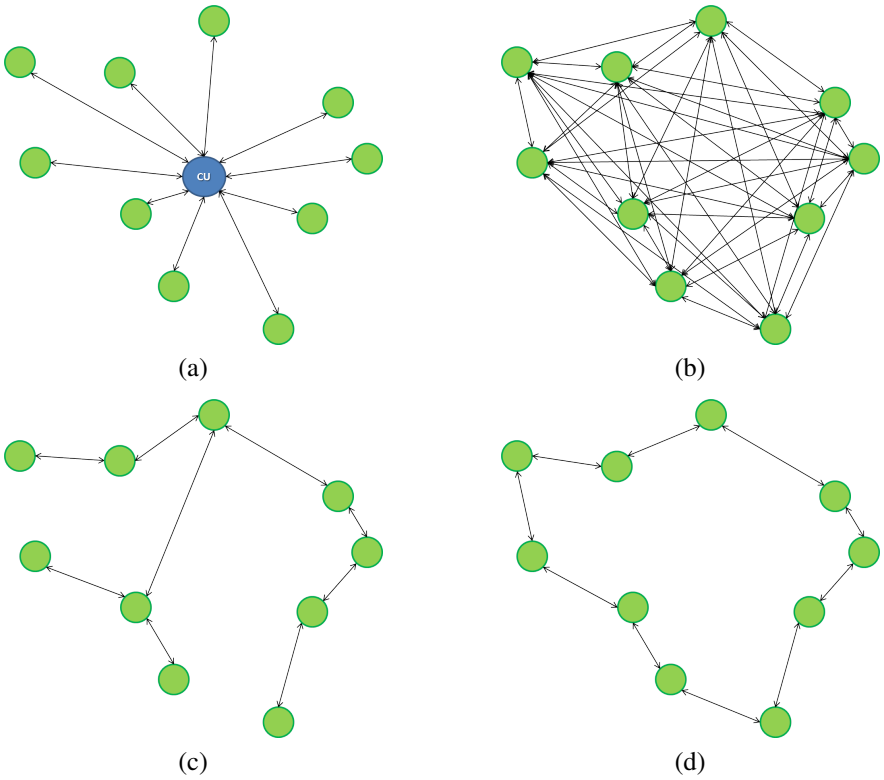


Figure 2.1: Different topologies of ASN: (a) star, (b) fully-connected, (c) mesh and (d) ring. CU is referred to central unit.

consist of a centralized process where a unit central gather data from the all nodes, analyzing it and processing it in a centralized manner. However, distributed processing is often preferred since the processing is splitted among the nodes of the network, and therefore, the computational burden at each node is lower than centralized processing. This type of processing is more efficient since the number of parameters or signals to be processed by each node is low. In addition, it can be reduced if, for example, nodes only exchange information with neighboring nodes. Both types of data processing will be discussed in Section 2.2.

- *Topology*. It should be noted that the selection of the network topology will affect how data is processed by each node. Some common topology architectures are mesh, star, ring and fully-connected topologies [27] (see Figure 2.1). The proper election of the topology depends on the communication constraints dictated by the network such as the amount and frequency of the transmitted data, transmission distance, battery life of the node, *etc*.

- *Scalability*. Algorithms should provide the same behavior and efficiency regardless of the size of the ASN. So that adding new nodes has no impact (or at least as low as possible) on the computational cost or data transfer between the remaining nodes of the network.

- *Communication bandwidth*. Some new strategies may be required to be found in order to reduce data transmission and thus to reduce the network bandwidth requirements.

- *Network lifetime*. Defined as the time until the first sensor's energy runs out, energy conservation to prolong the network lifetime is one of the major issues in sensor networks [28, 29]. Nodes can typically use batteries with a reduced energy life which influences the network lifetime. A possible solution involves only using the subset of nodes that have the most useful data with the aim to avoid that redundant information can be shared. The other nodes can use low power mode to save battery life. Finding the optimal subset of nodes and thus reducing the computational burden of the algorithms is a challenge and lately heuristic methods are used (genetic or memetic algorithms) [30].

- *Synchronization*. Note that, in real-time applications, some topologies may introduce delays that could seriously affect the system performance, introducing communication delays [2] and requiring the use of synchronization mechanisms [31, 32]. Therefore, the use of synchronization mechanisms among nodes is required in practical scenarios.

Note that many of these implementation aspects are related each other since dealing to improve one may help improve the others.

On the other hand, two types of ASNs are usually differentiated, *homogeneous* and *heterogeneous* sensor networks. In homogeneous networks, all nodes are identical in terms of battery life and hardware complexity. In addition, a single and fixed network topology is usually used. On the contrary, in heterogeneous networks, different topologies and different types of nodes with different functionality and battery energy may be used. In the context of this thesis, we consider that acoustic signal processing applications work over homogeneous ASN. To this end, we redefined the term of homogeneous ASN as a single-topology sensor network composed of acoustic single-channel nodes, i.e., nodes are equipped with a single sensor and a single actuator, have the same communications and computation capacities and execute the same acoustic signal processing task.

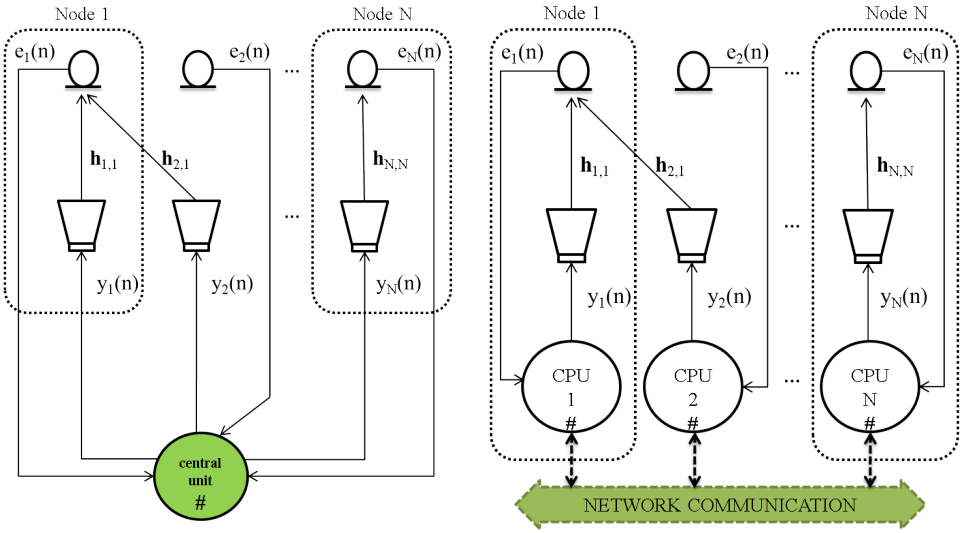


Figure 2.2: Scheme of (a) centralized audio system with a single central unit (controller) and (b) distributed audio system with single-channel nodes. Symbol # represents processing and communication capability and the posibility of communication among nodes is represented by dashed lines.

2.2 Distributed signal processing

As previously commented, the selection of the network topology affect how data is processed by each node. Note that, for example, in the case of a star topology (see Figure 2.1.a), a single controller or central unit receives, process and transmits all the information collected by a set of nodes. This strategy is referred to as *centralized processing* and is considered as the best solution in terms of the estimation performance since it has access to all the available information of the network. However, the addition of new nodes may considerably increase the computational load and data transfer rate (amount of data that is moved from the central unit to the nodes) of the central unit. Moreover, note that a failure of the single controller means that no information is processed. Another option is to use a ring topology (see Figure 2.1.d) where *distributed processing* is considered. In this case, each node locally processes its own information and the information received from its neighbors with the aim to share the results with its neighboring nodes. In this way, the communication and computational burden of the centralized processing is distributed among several nodes of the network.

In the field of audio signal processing, different applications have traditionally used centralized signal processing to modify or control the acoustic environment

through the use of multiple actuators and multiple sensors. Typically, these *multichannel audio systems* use a single centralized processor (central unit) managed by a *control algorithm* that has access to all the signals generated by the actuators and captured by the sensors. However, large multichannel systems distributed over a wide area require larger number of communication channels among transducers and controller. This results in an increase in the amount of cabling and, consequently, in a more costly infrastructure. Moreover, adding multiple transducers may increase drastically the computational cost required to capture, manage, and generate multiple signals.

As stated previously, an independent processing and control can be achieved by using a distributed approach which is often preferred, especially in terms of *flexibility*, *versatility* and *scalability*. Flexibility allows the system to select the suitable strategy depending on the objective application. In addition, they have the versatility to adapt quickly and easily to different situations. Moreover, these distributed systems can increase the number of controllers (or nodes) without redesigning the system. A *distributed audio system* consists of autonomous acoustic nodes which control a subset of loudspeakers from the signals picked up by a subset of microphones to reach a common target. However, note that a multichannel centralized system can be divided into a distributed system composed of several single-channel nodes (see Figure 2.2). In the context of this thesis, the term CPU depicted in Figure 2.2 refers to a Communication and Processing Unit and symbol # represents processing and communication capability. In both systems, $e_k[n]$ are the signals captured by the N microphones, $y_j[n]$ are the output signals reproduced by the N loudspeakers and the acoustic channels $h_{j,k}$ are the impulse responses between the j-th loudspeaker and the k-th microphone (where $k{=}1, 2, \ldots, N$ and $j{=}1, 2, \ldots, N$). Every node process signals independently and, when there exists communication among them, it is capable to generate the proper output signal as a result of processing the signal captured by itself as well as the information received from other nodes. Thus, all the nodes are relevant for the proper performance of the global system. These distributed single-channel systems are known to be computationally efficient with respect to its multichannel centralized version, since the computation burden as well as the acquisition and signal generation are distributed among several nodes of the network. One of the main problems of the distributed systems is how to share the information between the nodes in a controlled and efficient way.

Depending on the capacity of cooperation between nodes, the network may be ruled by a *collaborative* or *non-collaborative* learning. A collaborative learning may be understood as a data exchange among nodes following a cooperative strategy [5, 33]. Nodes are devoted to collaborate in order to achieve the same global network solution but distributing the computational burden as well as the acquisition and signal generation. Since some collaborative strategies usually aim to achieve an equivalent performance with respect to the centralized fashion, we consider in-

differently any kind of collaboration, regardless of being centralized or distributed cooperation. However, one of the main problems of the collaborative strategies is how to share information among nodes in a controlled and efficient way. On the other hand, in a non-cooperative or *decentralized network*, each node process data independently of the other nodes but sharing the computational burden among them. In this way, a decentralized system can be viewed as a set of single-channel centralized subsystems working in an autonomous way. Some advantages of this strategy compared to the collaborative method are scalability and low computational cost. Note that, in contrast to the collaborative fashion, in a decentralized scheme the addition or subtraction of nodes in the network does not require the readjustment of the learning rules. Moreover, in the case of addition new nodes, both the computational complexity and wiring effort of the system would be lower [34].

In recent years, distributed signal processing has been extensively developed due in part to the increase of many applications emerged from ASNs. This may lead to the use of novel distributed signal processing techniques with the aim to be applied over heterogeneous and multitask networks [35]. Every node of the network estimates a set of unknowk parameters in most of the distributed estimation methods. This scenario may be seen as a particular case of a more general problem where each node calculates a specific estimation but related to the estimations of the rest of the nodes [36]. *Node-specific* techniques are interested in solving different but overlapping estimation problems over distributed networks [20, 37, 38]. The term node-specific means that each node of the network estimates a signal or a parameter different from the rest and specific to each node. In this regard, two types of distributed node-specific estimation problems can be distinguished: node-specific *signal estimation* and node-specific *parameter estimation* [13]. Distributed node-specific signal estimation aim to estimate a signal in real time while the number of estimation variables grows linearly over time. This means that a new sample of the node-specific desired signals needs to be estimated at each sample time. In this case, nodes exchange fused or compresed versions of the samples captured by the sensors by using some compress-and-fuse techniques [39,40]. Therefore, in the case of *adaptive processing* [41], the algorithm can iterate and update these fusion rules instead of the estimates themselves, improving the overall estimation performance and reducing the communication bandwidth [21]. However, in node-specific parameter estimation (NSPE), the number of estimation variables is fixed, i.e. it does not grow over time. In this case, the node-specific parameters exchanged between nodes are derived from the samples captured by the sensors. Since these parameters vary slowly over time as compared to the sampling rate, adaptive algorithms may directly iterate and update over the estimated node-specific parameters. Therefore, the communication bandwidth is reduced in comparison to the node-specific signal estimation, meaning that network may consume less energy and may be composed of low-power, low-cost, and small-size nodes.

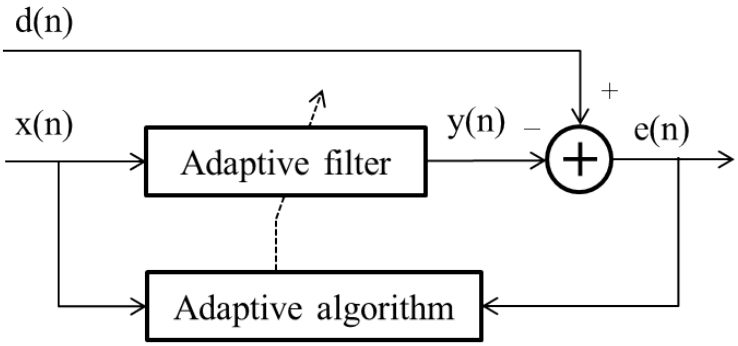


Figure 2.3: Block diagram of a typical adaptive algorithm.

2.3 Adaptive filtering algorithms

This thesis is focused on the control of the sound field by means of active methods applying adaptive strategies in signal processing. *Optimal signal processing* is concerned with the design, analysis, and implementation of systems which require a previous knowledge of the statistics of the data to be processed [42]. However, in unknown environments where it is required to deal with nonstationary signals with the aim to estimate time-varying parameters, this processing may no longer be optimum and it may require excessively elaborate and costly hardware. In those cases, *adaptive filtering processing* are usually considered since provides new signal processing techniques to solve the problem improving the system performance over the use of a fixed filter designed by conventional methods [43].

These techniques are based on *adaptive filters* [43, 44], i.e, time-variant filters whose parameters are adjusted to satisfy some predetermined optimization criterion. More specifically, a set of filter coefficients is periodically updated and controlled by a *recursive algorithm* with the aim to minimize a *cost function* that depends on the acoustic field to be controlled. Thus, these systems can automatically track changes in unknown environments or even changes in the system requirements. Adaptive filters have been successfully applied in such diverse control and signal processing applications such as system identification, noise and echo cancellation, equalization or beamforming among ohters.

As it can be shown in Figure 2.3, two basic stages, which interact with each other, are involved on the operation of an adaptive filtering algorithm: (1) a filtering stage where given an input signal $x(n)$, a determined output signal $y(n)$ is produced and (2) an adaptation stage where an adjustable set of coefficients are designed to be used in the filtering process. The difference between some desired response $d(n)$ and the

actual filter output $y(n)$ is defined as error signal $e(n)$. The adaptive strategy aims to match $y(n)$ with $d(n)$ while the time instant n increases. Saying it another way, adaptive systems are devoted to minimize some parameter of $e(n)$, usually related to its power. Thus, in steady-state, the adaptive algorithm converges to the optimum solution in some statistical sense while, in nonstationary environments, the adaptive strategy can track slow time variations in input data statistics.

On the design of the adaptive filters, some factors must be taken into account. The selection of the type of structure of the adaptive filter will determine to a significant extent the algorithm performance. Given its inherent stability, Finite Impulse Response (FIR) filters are the most widely used option for the design of linear adaptive filters. Therefore, for simplicity, we assume that the adaptive filters used throughout this thesis are causal FIR type implemented in direct form. The choice of the cost function depends on the approach used and the application of interest. The most commonly used cost function is the *mean square error (MSE)* criterion whichs minimizes $J(n)=E\{e^2(n)\}$, where $E\{.\}$ denotes expectation operation. The selection of the adaptive algorithm depends on the cost function used. However, in practical scenarios, several parameters influence the choice of the adaptive algorithm:

- *Rate of convergence*. This is defined as the time required for the adaptive algorithm to converge to the optimum solution.
- *Misadjustment*. This parameter provides a measure of how close the algorithm is to the optimum solution in the mean-square-error sense.
- *Tracking capability*. This refers to the ability of the algorithm to track statistical variations in a nonstationary environment.
- *Computational complexity*. This quantity describes the total number of operations (multiplications, divisions, and additions or subtractions) required to execute the algorithm on a computer.
- *Robustness*. The robustness of the algorithm is related to the ability of an algorithm to respond adequately to an unsuitable configuration, such as measurement mismatches, noise addition or data loss, among others.

However, since some of these aspects are incompatible in mostly cases, a trade-off between them is usually made depending on the application of interest. Within adaptive filtering algorithms, among the many adaptive strategies that can be considered to calculate the adaptive filter coefficients, two of the most common are: the *least-mean-square (LMS)* and the *recursive least-squares (RLS)* algorithms. The LMS algorithm [41], based on the method of stochastic gradient descent, aims to find the minimum value of the instantaneous squared value of the error signal. The LMS algorithm is simple in terms of computational complexity, it does not require knowledge

of statistical characteristics of the input signal and it is robust in unknown enviroments. However, it present slow convergence behaviour, especially for colored input signals. On the other hand, the RLS algorithm [45, 46] is based on the method of least squares which aims to minimize the sum of weighted error squares. It has the advantage that its convergence rate is faster than the LMS strategy at the expense of higher computational complexity. Due to its robustness and low computational cost, the LMS algorithm is one of the most popular adaptive filtering algorithms in use. The basis for understanding the LMS algorithm is provided below.

2.3.1 The LMS algorithm

As previously stated and following the nomenclature depicted in Figure 2.3, the objective is to estimate the filter coefficients that minimizes a cost function $J(n)$ that depends on the error signals $e(n)$. To do this, gradient-descent methods estimate the coefficients in an iterative manner as follows

$$\mathbf{w}(n) = \mathbf{w}(n-1) \ - \mu \nabla_{\mathbf{w}} \, J(n), \tag{2.1}$$

where vector $\mathbf{w}(n) = [w_1(n) \; w_2(n) \; \cdots \; w_L(n) \,]^T$ contains the L coefficients of the adaptive filter, μ is the step-size parameter and $\nabla_{\mathbf{w}}$ is the gradient operator defined as the partial derivatives with respect to the coefficients vector $\mathbf{w}(n)$, i.e.,

$$\nabla_{\mathbf{w}} \, J(n) = \frac{\partial E\{e^2(n)\}}{\partial \mathbf{w}(n)} \tag{2.2}$$

with $J(n)$ defined by the MSE criterion previously introduced. As it can be seen in Figure 2.3, the error signal can be written as

$$e(n) = d(n) - y(n), \tag{2.3}$$

where, the filter output, $y(n)$, can be expresssed, using the matricial notation, as

$$y(n) = \mathbf{w}^T(n) \, \mathbf{x}_L(n) = \mathbf{x}_L^T(n) \, \mathbf{w}(n) \tag{2.4}$$

with $\mathbf{x}_L(n)$=$[x(n) \; x(n-1) \; \cdots \; x(n-L+1)]^T$. When the signals $x(n)$ and $d(n)$ are both stationary, it is possible to find the optimal values of $\mathbf{w}(n)$ that minimize $J(n)$, denominated as $\mathbf{w}_o$. Thus, the optimum filter is obtained by calculating the coefficients that fulfills that $\nabla_{\mathbf{w}} \, J(n)$=0, obtaining

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}, \tag{2.5}$$

where $\mathbf{R}$=$E\{\mathbf{x}_L(n)\mathbf{x}_L^T(n))\}$ is the autocorrelation matrix of the input signal $x(n)$,

$$\mathbf{R} = E \left\{ \begin{array}{cccc} x(n)^2 & x(n)x(n-1) & \cdots & x(n)x(n-L+1) \\ x(n-1)x(n) & x(n-1)^2 & \cdots & x(n-1)x(n-L+1) \\ \vdots & \vdots & \ddots & \vdots \\ x(n-L+1)x(n) & x(n-L+1)x(n-1) & \cdots & x(n-L+1)^2 \end{array} \right\}, \tag{2.6}$$

and $\mathbf{p}{=}E\{d(n)\mathbf{x}_L(n)\}$ is the cross correlation vector between the desired signal $d(n)$ and $x(n)$,

$$\mathbf{p}{=}E\{d(n)x(n),\ d(n)x(n{-}1),\ d(n)x(n{-}2),\ \ldots\ ,\ d(n)x(n{-}L{+}1)\}^T \tag{2.7}$$

Since the LMS method is a stochastic implementation of the steepest-descent algorithm, $J(n)$ can be approximated by its instantaneous value ($J(n) = E\{e^2(n)\} \approx J(n) = e^2(n)$). Therefore, considering 2.3 and taking into account that $d(n)$ is independent of $\mathbf{w}$, we obtain

$$\nabla_{\mathbf{w}}\, J(n) = \frac{\partial e^2(n)}{\partial \mathbf{w}(n)} = 2\frac{\partial e(n)}{\partial \mathbf{w}(n)} e(n) = -2\, e(n)\mathbf{x}_L^T(n) \tag{2.8}$$

Therefore, substituting it in 2.1, the update equation of the filter coefficients considering the LMS algorithm can be expressed as

$$\mathbf{w}(n) = \mathbf{w}(n-1)\ + 2\mu e(n)\mathbf{x}_L(n), \tag{2.9}$$

Note that the selection of L, the value of μ and certain statistical parameters of $x(n)$ will affect to the stability, convergence and steady-state properties of the LMS algorithm. There are many versions and variants of this algorithm that try to improve some of its features. The *affine projection* algorithm [47, 48] based on the *normalized LMS* algorithm (NLMS) [41, 43], improves the convergence, stability and steady-state properties of the LMS algorithm. For improving the processing efficiency in practical scenarios where data acquisition systems work by blocks of samples, the LMS algorithm was also presented in frequency domain with block processing in [49]. The details of these strategies are described below.

2.3.2 The NLMS algorithm

One of the variants of the LMS algorithm that improves its convergence, stability and steady state performance is the Normalized LMS (NLMS). This variant takes into account fluctuations in the power level of the signal to be controlled. To this end, the NLMS algorithm uses a step-size parameter which is normalized by an estimate of the power of the filter input signal providing stability as well as faster convergence than the LMS algorithm.

Considering (2.9), the filter updating equation of the NLMS algorithm can be defined as

$$\mathbf{w}(n) = \mathbf{w}(n-1)\ + \bar{\mu}(n)e(n)\mathbf{x}_L(n), \tag{2.10}$$

where $\bar{\mu}(n)$ is expressed as

$$\bar{\mu}(n) = \frac{2\mu}{\mathbf{x}_L^T(n)\mathbf{x}_L(n)} = \frac{2\mu}{\|\mathbf{x}_L(n)\|^2} \tag{2.11}$$

where the operator $\|.\|^2$ is the ℓ_2 norm.

2.3.3 The Affine projection algorithm

The affine projection (AP) algorithm is a minimization problem with constraints based on the use of orthogonal projections in affine subspaces of adaptive filters [47]. This strategy could be understood as an extension of the LMS algorithm which improves the convergence properties of the LMS at the cost of introducing higher computational complexity.

The update equation of the filter coefficients considering the AP algorithm is given by

$$\mathbf{w}(n) = \mathbf{w}(n-1) \ + 2\mu \mathbf{U}_Q(n)[\mathbf{U}_Q^T(n)\mathbf{U}_Q(n) + \delta \mathbf{I}_Q]^{-1} \ \mathbf{e}_Q(n)\cdot \tag{2.12}$$

where $\mathbf{U}_Q(n)$ is a $[L\times Q]$ matrix defined as $\mathbf{U}_Q^T(n)$=$[\mathbf{x}_L(n) \ \mathbf{x}_L(n-1)\ldots \ \mathbf{x}_L(n-Q+1)]$. $\mathbf{I}_Q$ is a identity matrix of size $[Q \times Q]$ and vector $\mathbf{e}_Q(n)$ contains the last Q samples of the error signal, i.e., $\mathbf{e}_Q(n) = [e(n) \ e(n-1)\ldots \ e(n-Q+1)]^T$. Note that the NLMS is a particular case of the AP algorithm when Q=1. A more detail description of the affine projection algorithm can be found in [50].

2.3.4 The Block LMS algorithm

As the basis of block processing, the block LMS (BLMS) algorithm gather a block of samples of the involved signals to be efficiently processed in a parallel way. By means of this method, the adaptive filter coefficients are updated once every block of B data samples are collected. In this regard, the index n between brackets denotes block iteration.

In this way, the filter coefficients update of the BLMS algorithm is obtained as

$$\mathbf{w}[n] = \mathbf{w}[n-1] \ + 2\mu \mathbf{X}_B^T[n]\mathbf{e}_B[n], \tag{2.13}$$

where the matrix $\mathbf{X}_B[n]$=$[\mathbf{x}_L(Bn) \ \mathbf{x}_L(Bn-1) \ \ldots \ \mathbf{x}_L(Bn-B+1)]^T$ contains the last B block of samples of size L of $x(n)$ and $\mathbf{e}_B[n]$ is the vector with the last block of size B of the error signal $e(n)$, that is, $\mathbf{e}_B[n]$=$[e(Bn) \ e(Bn-1)\ldots \ e(Bn-B+1)]^T$. The output and the error vectors are also calculated as $\mathbf{y}_B[n]$=$\mathbf{X}_B[n]\mathbf{w}[n]$ and $\mathbf{e}_B[n]$=$\mathbf{d}_B[n]$−$\mathbf{y}_B[n]$ respectively, where $\mathbf{d}_B[n]$=$[d(Bn) \ d(Bn-1)\ldots \ d(Bn-B+1)]^T$ is the vector with the last block of size B of the desired signal $d(n)$.

2.3.5 The Fast BLMS algorithm

The fast BLMS (FBLMS) algorithm is a computationally efficient implementation of the BLMS algorithm in the frequency domain. The block diagram of the FBLMS algorithm is depicted in Figure 2.4 and its formulation is outlined below.

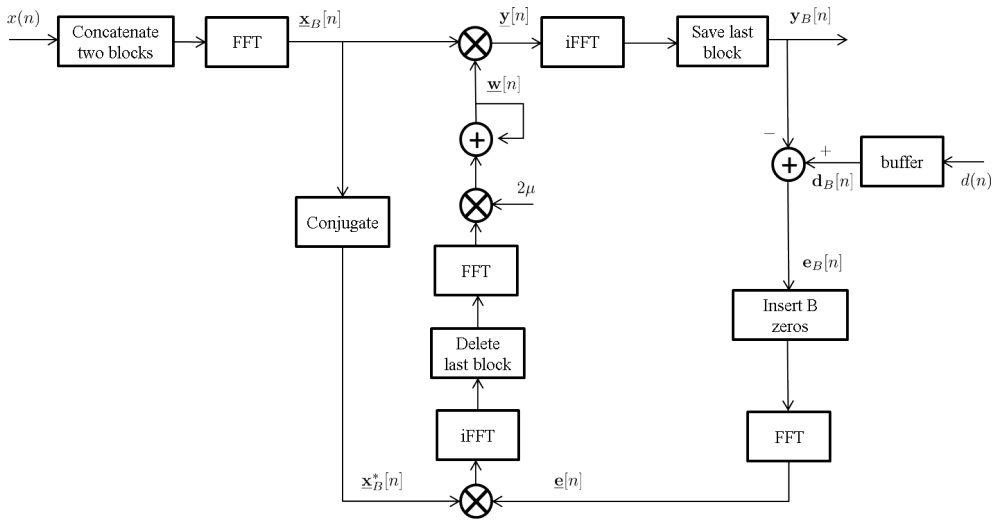


Figure 2.4: Block diagram of the FBLMS algorithm.

We define $\mathbf{x}_B[n]=[x(Bn)\ x(Bn-1)\dots\ x(Bn-B+1)]^T$ as the vector that contains the last B samples of $x(n)$ and $\underline{\mathbf{x}}_B[n]$ as the vector that contains the Fast Fourier Transform (FFT) of size $2B$ of the vector $\mathbf{x}_B[n]$ in the actual block iteration and the same vector in the previous block iteration, $\underline{\mathbf{x}}_B[n]=\mathrm{FFT}[\mathbf{x}_B[n-1]\ \ \mathbf{x}_B[n]]$.

The 2B-FFT of the filtered output vector at the n-th block iteration, $\underline{\mathbf{y}}[n]$, is calculated as ast

$$\underline{\mathbf{y}}[n] = \underline{\mathbf{x}}_B[n] \circ \underline{\mathbf{w}}[n] \tag{2.14}$$

where $\underline{\mathbf{w}}[n]$ is the FFT of size $2B$ of the coefficients of the adaptive filter at the n-th block iteration and $\circ$ denotes the element-wise product of two vectors. Note that we only consider the last B samples of the $2B$-Inverse Fast Fourier Transform (IFFT) operation, $\mathrm{IFFT}\{\underline{\mathbf{y}}[n]\}$, as the valid samples of the adaptive filter output $\mathbf{y}_B[n]=[y(Bn)\ y(Bn-1)\dots\ y(Bn-B+1)]^T$ since the first B samples suffer the effects of circular convolution due both to data management and the FFT sizes.

The vector $\underline{\mathbf{e}}[n]$ is the FFT of size $2B$ of the error vector $\mathbf{e}_B[n]=\mathbf{d}_B[n]-\mathbf{y}_B[n]$ preceded by a vector of zeros of size B, $\mathbf{0}_{B\times 1}$, that is, $\underline{\mathbf{e}}[n] = \mathrm{FFT}\{[\ \mathbf{0}_{B\times 1}\quad \mathbf{e}_B[n]\ ]\}$. Once the error vector is calculated, the filter coefficients of the FBLMS algorithm are updated as

$$\underline{\mathbf{w}}[n] = \underline{\mathbf{w}}[n-1]\ + 2\mu\ \mathrm{FFT}\{[\ [\ \mathrm{IFFT}\{\underline{\mathbf{e}}[n] \circ \underline{\mathbf{x}}_B^*[n]\}\ ]_{[1:B]}\quad \mathbf{0}_{B\times 1}\ ]\}. \tag{2.15}$$

where $*$ denotes complex conjugation and $[.]_{[1:B]}$ refers to the first B elements of the vector between brackets.

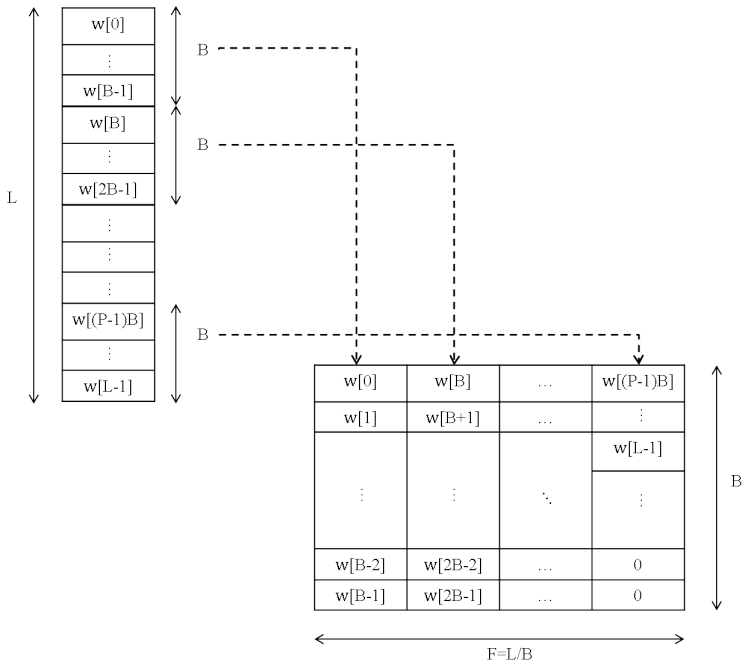


Figure 2.5: Scheme of the partition of an adaptive filter of size L into F partitions of size B.

2.3.6 The Partitioned FBLMS algorithm

If the length of the adaptive filters, L, is higher than the block size, B, an efficient implementation of the FBLMS algorithm can be derived by splitting up (partitioning) the adaptive filters into F partitions. The resulting implementation is called the partitioned FBLMS (PFBLMS) algorithm.

Assuming that $L{=}F{\times}B$, the partition of the filters may be carried out as depicted in Figure 2.5. Equation (2.14) now can be redefined as

$$\underline{\mathbf{y}}[n] = \sum_{f=1}^{F} \underline{\mathbf{x}}_B[n-f+1] \circ \underline{\mathbf{w}}^f[n] \tag{2.16}$$

where superscript f denotes the partition number and $\underline{\mathbf{w}}^f[n]$ contains the $2B$-FFT of the fth partition of the adaptive filter. As an example, the arrangement of $\underline{\mathbf{x}}_B[n]$ when $F{=}3$ is depicted in Figure 2.6. As explained in subsection 2.3.5, only the last B samples of $\underline{\mathbf{y}}[n]$ are valid. Finally, the update equation of the filter coefficients

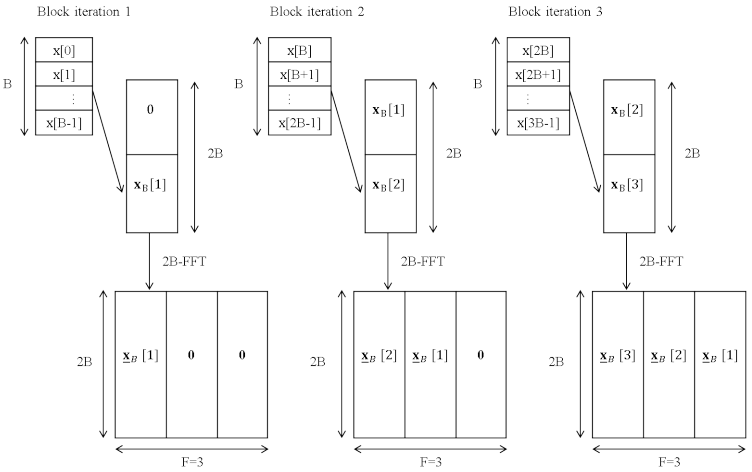


Figure 2.6: Arrangement of $\underline{\mathbf{x}}_B[n]$ when F=3.

considering the PFBLMS algorithm at the f-th partition is calculated as

$$\underline{\mathbf{w}}^f[n] = \underline{\mathbf{w}}^f[n-1] \; + 2\mu \, \mathrm{FFT}\{[\,[\,\mathrm{IFFT}\{\underline{\mathbf{e}}[n] \circ \underline{\mathbf{x}}_B^*[n-f+1]\}\,]_{[1:B]} \quad \mathbf{0}_{B\times 1}\,]\}. \tag{2.17}$$

Therefore, the presented LMS algorithm and some of its variants have been considered throughout this dissertation to implement adaptive signal processing applications related to the control of the sound field.

2.4 Sound Field Control applications

Sound field control (SFC) involves the use of electroacoustic transducers to capture, process and generate audio signals over an acoustical environment. Major technology development in recent years allows to implement more complex audio signal processing algorithms for specific applications such as personal sound zones [51], room acoustics control [52, 53], active noise control [54] or spatial audio [55, 56] among others. These applications can be used in the cabin of a public or private transport, consumer entertainment systems, mobile devices, concert halls, museums, and other public venues.

In this thesis, we apply the researched distributed processing to develope two (SFC) applications: *active noise control (ANC)* and *active noise equalization (ANE)*. The objective of these applications is to ensure the acoustic comfort of multiple users located in different positions inside an enclosure. A brief introduction to both applications is given below.

2.4.1 Active Noise Control (ANC)

Noise reduction techniques have been gaining an increasing importance due to the health problems related to the exposure to high levels of noise from industrial enviroment or traffic noise, among others. In addition to the adverse and direct effects of noise in society (often valued in euros), hearing loss, stress, sleep disturbance or aggressive behaviors indirectly involve a damaging impact on human health [57]. In particular, environmental noise, caused by traffic, industrial and recreational activities is considered to be a significant local environmental problem in Europe [58]. Noise complaints have increased in Europe since 1992 and it is estimated that roughly 20% of the Union population or close to 80 million people suffer from noise levels which scientists and health experts consider being unacceptable [59].

During the last decades, the design of *passive systems* composed of sound absorbing materials has been usually considered in order to try to attenuate an acoustic noise field. However, these passive techniques generally do not work well at low frequencies, where they tend to be expensive and bulky. This is due to the thickness of a sound absorber is low when compared to the large wavelengths at low frequencies. Note that most disturbance noises generated from our environment are composed of low frequency spectral components. On the contrary, *active noise reduction* techniques have been presented as an effective solution to attenuate the noise field at low frequencies.

The first active control technique was described in [60, 61] where Lueg used a loudspeaker to try to generate anti-noise sound of equal magnitude but out of phase with the acoustic wave produced by a noise source in an acoustic duct. However and although Lueg is considered as a pioneer in the field of active control, his experiment did not work due to technological limitations. Almost 20 years later, Olson and May proposed in [62] an active system positioned on the backrest of a seat with the aim to cancel the noise captured by a microphone located close to a loudspeaker. At the same time, Conover [63] was working on the active reduction of acoustic noise from large mains transformers. However, analog technology did not allow a very accurate adaptation to enviroment changes and consequently, the results obtained in both works were not entirely satisfactory. The digital revolution at the late 1970s facilitated the implementation of adaptive techniques to active noise control [64]. Since then, due to the digital signal processing techniques and recent advances in computing field, active control of noise has gained intensive development in the last decades.

Based on the superposition principle of sound waves, the objective of an *active noise control* (ANC) system is to create a zone of *destructive interference* by generating the appropriate acoustic waves in order to cancel an undesired noise [65]. To this end, the system makes use of loudspeakers devoted to emit the anti-noise signals to try to reduce the disturbance signal at specific spatial points monitored by micro-

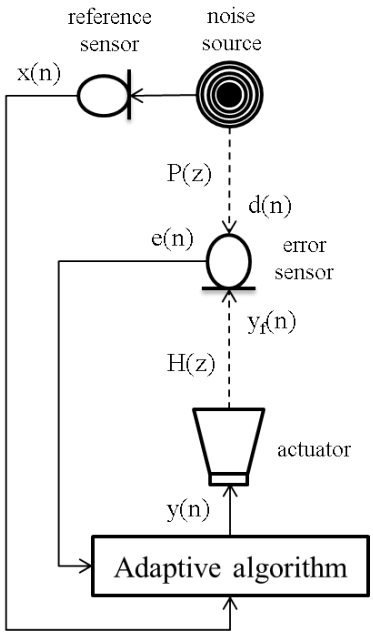


Figure 2.7: Single-channel feedforward active noise control system.

phones. The greater the wavelength of the signal to be canceled, the larger the zones of quiet. It has been shown [66] that a considerable attenuation can be achieved in an area around the control point with an approximate size of one tenth of the wavelength of the signal to be canceled. Out of these quiet zones, noise level may even increase [67].

The *global noise cancelling* in an entire enclosure is practically unfeasible [54]. Due to the room reflections, *acoustic modes* appear, i.e, standing waves of the room at speficic resonance frequencies. At low frequencies, an acoustic mode can dominate the room response and therefore it is possible to achieve high noise attenuation. The problem is that as frequency increases, more modes contribute to room response. Therefore, it is not possible to attenuate each of them without amplifying others. A possible solution would be to increase the number of actuators in order to increase the number of acoustic modes to be controlled. Unfortunately, the number of modes increases at high frequencies in approximate proportion to the cube of the excitation frequency [65]. The solution involves to control the noise field within a certain area in order to create *local zones of quiet* [68]. In addition, it is required to avoid that the actuators and sensors be located at certain points in the enclosure where the acoustic pressure is zero.

Since the anti-noise signal is usually unknown as well as the characteristics

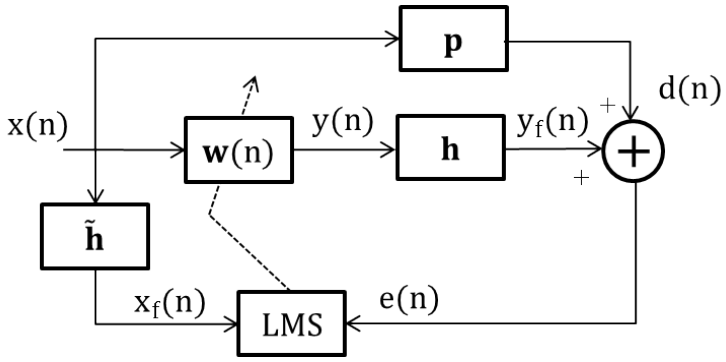


Figure 2.8: Block diagram of a single-channel feedforward active noise control system using filtered-x LMS algorithm.

the environment are time varying, ANC control system commonly use adaptive filters [43] in order to deal with these problems [69]. The active noise controller adjusts the adaptive filters coefficients to minimize the noise signal picked up by an *error sensor*. The most common form of adaptive filter for ANC systems is a finite impulse response (FIR) filter using the LMS algorithm. The filter output signal, $y(n)$, emitted by the actuator (called *secondary source*) and filtered through the acoustic system $H(z)$ are designed to minimize the signal recorded at the error sensor, called error signal and denoted by $e(n)$. $H(z)$ (*secondary path*) and $P(z)$ (*primary path*) are the acoustic paths which link the actuator and the noise source with the error sensor, respectively. Note that, in order to achieve a properly adaptation of the filter coefficients, a previous estimation of $H(z)$ may be required in most cases. Since the acoustic paths act as acoustic filters which change the amplitude and phase characteristics of the sound signals, to recreate its effects on the adaptive algorithm is desirable. For example, the secondary path introduces a delay in error signal which involves a deviation in the synchronization between $e(n)$ and $x(n)$. A possible solution is to compensate the secondary path effects by introducing an estimate of $H(z)$ on the LMS algorithm. This structure, known as filtered-reference or *filtered-x LMS* (FxLMS) algorithm [70, 71], produce compensating delay and attenuation to reference signal before using at the updating of the adaptive filter (see Figure 2.8).

In this case, the filter updating equation of the FxLMS algorithm can be expressed as

$$\mathbf{w}(n) = \mathbf{w}(n-1) \ + 2\mu e(n)\mathbf{x}_f(n), \tag{2.18}$$

where $\mathbf{x}_f(n)$ is a L-length vector that contains the last L samples of reference signal $x(n)$ filtered through the estimation of the secondary path ($\mathbf{h}$) modelled as a FIR filter and denoted as $\tilde{\mathbf{h}}$. Thus, $x_f(n)$ and $y_f(n)$ (calculated as $y(n)$ filtered through $\mathbf{h}$) are related by the filter coefficient vector, $\mathbf{w}(n)$.

It is well-known that introducing an additional filter will increase the number of algorithm operations. In addition, the inherent secondary path delay and the possible inaccuracies when this path is estimated, may lead to low convergence speed. Despite all this, the FxLMS solution is the most widely used in literature since provides a trade-off between convergence rate and computational cost.

Generally speaking, the use of a large number of microphones strategically located produces larger zones of quiet. Similarly, multiple transducers are commonly used to improve the system performance, resulting in a *multichannel ANC system*. In some cases, the use of a reference sensor to capture the noise signal before it reaches the control area is beneficial (*feedforward control*) since it provides robustness to the ANC system. However, in order to avoid stability problems, the reference sensor should be located as far to the cancelling actuators as possible. In cases when this does not happen, the reference signal can be internally estimated from the error signal (*feedback control*). Although these systems are simpler than the feedforward approach, they may present unstability due to the possible positive feedback of the control system as well as they usually gives poor performance in case of broadband noise. In this regard, this thesis is focused on the feedforward control approach, as depicted in Figure 2.7). It can be shown that the *reference signal* $x(n)$ is correlated with the noise signal since the reference sensor is located closed to the acoustic noise source, usually called *primary source*. However in the cases where the noise signal are composed of tonal components, the reference signal is not required since it can be synthesized internally. This may lead to the use of systems that allows to control each frequencial component independently.

2.4.2 Active Noise equalization (ANE)

As previously mentioned, disturbance noises are often suppressed by using active noise cancelers with the aim to reduce annoyance [64]. However, in some ANC applications, instead of cancelling an undesired disturbance, it is desirable to create a desired acoustic sound field by retaining a residual noise with a specific *spectral shape*. This is motivated because some *auditory feedback* of the control system performance can be required for safety considerations or for creating a desired acoustic comfort [72]. Furthermore, different sound zones can be desired where, in some areas, noise may be adjusted to enhance the sound quality, while, in other areas, noise should be suppressed to reduce annoyance.

For these tasks, the use of *adaptive noise equalizer* (ANE) algorithms [73–75] is usually considered. Depending on the operating bandwidth, both *narrowband* and *broadband* noise sources can be controlled by an ANE algorithm.

Narrowband ANE algorithm

Many of current acoustic noise sources have their origin in rotating machinery, such as engines or fans, which generate narrowband noises composed of periodic components normally related to the rotational speed. With the aim to deal with such disturbances, narrowband ANE systems are usually considered. The narrowband ANE algorithm can control the amount of attenuation or amplification of any harmonic present in the noise at a given frequency by adjusting a *gain parameter* [73]. This algorithm is based on an *adaptive notch filter* which aims to eliminate a frequency component of a signal by introducing, in the adaptive filter, a sinusoid of the frequency to eliminate. As depicted in Figure 2.10, a two-coefficient adaptive filter which controls the in-phase and quadrature components of a reference signal, $x(n)$, is required to cancel the sinusoid at frequency ω_0. This reference signal does not have to be obtained since it can be generated internally provided that the controlled frequencies are previously known [75]. In this case, $x(n)$ is given by

$$x(n) = A\cos(\omega_0 \mathrm{n} + \phi_0), \tag{2.19}$$

where A is the amplitude, ω_0 is the frequency and ϕ_0 is the initial phase of sinusoid at the discrete time n. The quadrature reference signal is derived from $x(n)$ by using a 90° phase shifter,

$$\widehat{x}(n) = A\sin(\omega_0 \mathrm{n} + \phi_0). \tag{2.20}$$

Using the LMS strategy, the filter updating equation is stated as follows

$$\begin{aligned} c(n+1) &= c(n) + \mu e(n)x(n) \\ \widehat{c}(n+1) &= \widehat{c}(n) + \mu e(n)\widehat{x}(n). \end{aligned} \tag{2.21}$$

Note that the adaptive filter minimizes the error signal power by adjusting the amplitude and the phase of the output signal, $y(n)$, in order to cancel the component at the same frequency of the multitonal primary noise signal, $d(n)$. Thus, by varying the frequency of $x(n)$, it is possible to vary the frequency to be eliminated.

In addition, instead of cancelling a controlled frequency, it is possible to independently control a frequency component of $d(n)$ by using an ANE. This system minimizes a *pseudo-error signal*, $e'(n)$, instead of the error signal, $e(n)$, with the help of a LMS algorithm. Performing as an adaptive notch filter, the LMS algorithm equalizes (attenuates or amplifies) the reference frequency to be controlled. In this case, $y(n)$ is split into two branches, the canceling branch and the balancing branch (see Figure 2.9). The gain $1-\gamma$ is inserted in the canceling branch and the gain γ is inserted in the balancing branch to adjust the residual noise level, where γ is the gain parameter at the frequency ω_0. γ=0 indicates the cancellation mode, $0<\gamma<1$ indicates the attenuation mode and $\gamma \geq 1$ amplifies the signal by γ ($\gamma>1$) or keeps the signal intact (γ=1).

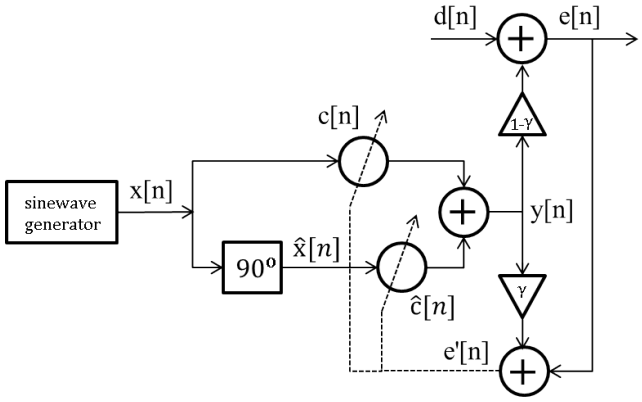


Figure 2.9: Block diagram of the single-channel single-frequency narrowband ANE algorithm.

The error signal $e(n)$ is now given by

$$e(n) = d(n) + (1 - \gamma)y(n), \tag{2.22}$$

and the pseudo-error signal $e'(n)$ is defined as

$$e'(n) = e(n) + \gamma y(n) = d(n) + (1 - \gamma)y(n) + y(n) = d(n) + y(n). \tag{2.23}$$

When convergence is achieved, $e'(n) \to 0$ and therefore

$$e(n) \to -\gamma y(n) \approx \gamma d(n), \tag{2.24}$$

where the output of the system, $e(n)$, is the input signal, $d(n)$, multiplied by a factor γ, i.e. equalization is achieved.

In practical active control applications, the previously stated FxLMS algorithm is used to adapt the weights of the adaptive filter as an alternative form of the LMS algorithm with the aim to minimize $e'(n)$. Using the approach developed in [76] and its application for ANC systems [3], the performance of the ANE algorithm can be evaluated by means of their *steady-state transfer functions* if the periodic reference signal is a synchronously sampled sinusoid. In this case, the transfer function between the desired signal and the error signal can be described by a linear time-invariant filter between these signals. The transfer function present a notch at the controlled frequency (as ANC algorithms) but allowing an independent gain adjustment. The narrowband ANE will perform different modes of operation by moving the zeros of the system into different regions inside the unit circle [74]. The zeros are moved by adjusting

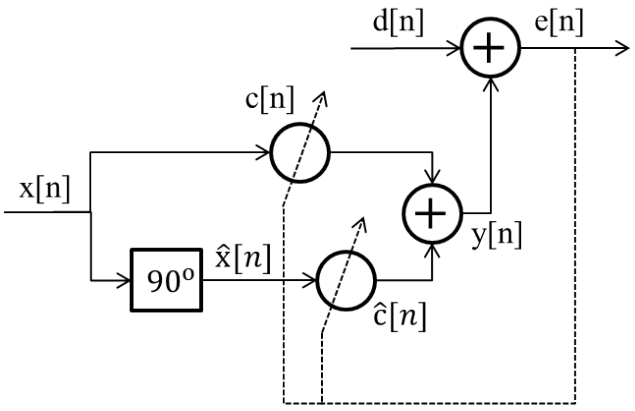


Figure 2.10: Block diagram of the adaptive notch filter.

the value of the gain parameter [75]. In addition, by locating the poles of the system transfer function, the ANE stability can be analyzed.

During the last decades, several studies have been presented in order to improve the performance of the single-channel narrowband ANE systems.

Multichannel ANE systems are used to produce larger equalization zones and to improve the control of the multifrequency noise by adding multiple loudspeakers and microphones [74]. In [77, 78], a multichannel adaptive equalizer that controls a noise signal composed of tonal components according to an equalization factor is presented. This factor may be different for each frequency component, but it is the same for each frequency at all sensors. *The multiple frequencies* of the disturbance noise are controlled by the superposition of several single-frequency ANE filters [75]. In addition, two adaptive strategies to solve the multichannel multi-frequency equalizer problem with a single equalization profile on all sensors are proposed in [78]. The *multiple-error* strategy uses a different pseudo-error signal for each frequency related to the same error sensor. On the contrary, only one pseudo-error signal for each error sensor is considered by the *common-error* strategy reducing the computational cost while providing a robust performance.

On the other hand, in practical scenarios, the secondary acoustic paths that links actuators and sensors may suffer variations over time due to changes in the acoustic environment [79]. This leads to the *imperfect secondary path estimation* [80] [81] and it may even cause the system divergence. The system stability is guaranteed if the phase error between the acoustic path and its estimation is less than 90° [70]. However, depending on the value of the system gain, the instability may appear due to the amplitude estimation error (denoted as *mis-equalization*) or caused by the phase estimation error [81]. Furthermore, the imperfect estimation also reduces the conver-

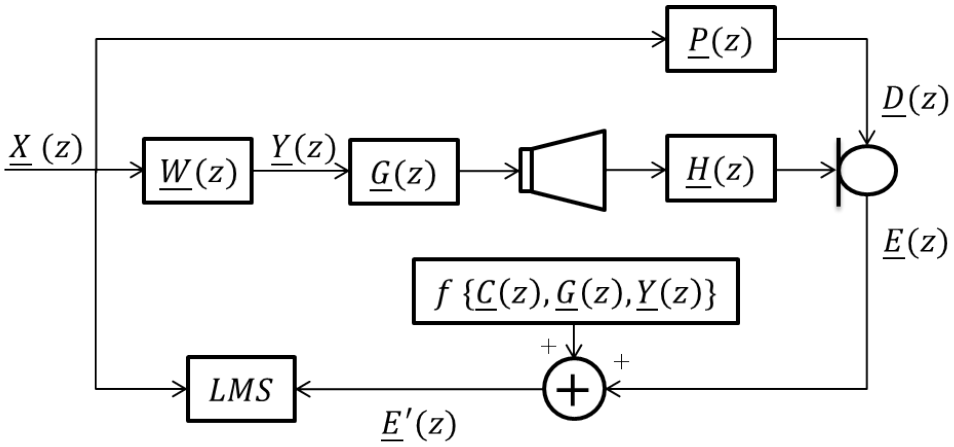


Figure 2.11: Block diagram of the adaptive algorithm for equalizing an acoustic signal.

gence speed of the ANE system [82]. Several strategies aim to deal with this instability problem by controlling the amplitude [80] [83] and the relative-phase [84] [85] of the narrowband components of the residual noise. Aside from the characterization of the acoustic environment, the interference problem involves an additional uncertainty with the aim to create *personal audio zones* [51]. In practical multichannel scenarios, achieving independent control over a specific personal zone without affecting other surrounding zones is challenging, especially at high frequencies [86]. As previously stated, note that at least 11 dB between the different personal zones is required to provide adequate separation [87]. These considerations must be taken into account in order to ensure an auditory comfort. This is the idea behind the *active sound quality control* (ASQC) techniques. An improvement of the control approach may be achieved by adapting the residual noise spectrum to the listener ear's response [88] [89]. Therefore, *psychoacoustic metrics*, such as loudness and roughness [90], must be taken into account in order to make the undesired noise sound comfortable [72]. The perceived acoustic comfort is often associated with low sound preassure levels [72]. However, it was demostrated in [91] that the subjective improvement of an ANC system strongly depends on the power spectrum of the controlled sound, and not just the attenuation level achieved.

Broadband ANE algorithm

In addition, a broadband noise can also be controlled by an ANE technique [88, 92, 93]. The broadband algorithm can shape the spectrum of the residual noise with a predetermined filter, denoted as *shaping filter*, which is usually designed on the basis

of ASQC considerations [94].

Consider a single-channel ANE system composed of one reference signal, a single actuator and a single error sensor (1:1:1 configuration), as shown in Figure 2.11. Considering the transform domain, we aim to equalize $\underline{D}(z)$ by generating $\underline{Y}(z)$ with the aim to obtain the following residual signal $\underline{E}(z)\approx\underline{C}(z)\underline{D}(z)$, being $\underline{D}(z)$ the z-transform of the noise signal, $\underline{Y}(z)$ the z-transform of the control signal and $\underline{C}(z)$ the shaping filter, i.e., the transfer function of the desired equalization. Note that the noise signal at the error sensor may be expressed as $\underline{D}(z)=\underline{Y}(z)\underline{P}(z)$ being $\underline{P}(z)$ the acoustic transfer function that links the noise source with the error sensor. The control signal $\underline{Y}(z)$ is calculated as $\underline{Y}(z)$ filtered through an adaptive filter $\underline{W}(z)$, i.e., $\underline{Y}(z)=\underline{Y}(z)\underline{W}(z)$ with $\underline{Y}(z)$ and $\underline{W}(z)$ as the z-transforms of the reference signal and the adaptive filter, respectively. In a general case, the control signal may be affected by a generic filter $\underline{G}(z)$. In order to reach the objective, a pseudoerror signal, $\underline{E'}(z)$ which depends on the system signals and the configuration parameters ($\underline{E'}(z)=f\{\underline{E}(z),\underline{Y}(z),\underline{C}(z),\underline{G}(z)\}$) must be minimized. Note that, when the proposed objective is reached, $\underline{E'}(z)\approx 0$, the expression $\underline{E}(z)\approx\underline{C}(z)\underline{D}(z)$ must be fulfilled.

In order to obtain the pseudoerror signal, we consider that the information captured by the error sensor is calculated as

$$\underline{E}(z) = \underline{D}(z) + \underline{Y}(z)\underline{G}(z)\underline{H}(z), \tag{2.25}$$

being $\underline{H}(z)$ the acoustic transfer function that links the actuator with the error sensor. We proposed that the pseudoerror signal can be calculated as a lineal combination between error signal and a function dependent on $f\{\underline{Y}(z),\underline{C}(z),\underline{G}(z)\}$, i.e.,

$$\underline{E'}(z) = \underline{E}(z) + f\{\underline{Y}(z)\underline{C}(z)\underline{G}(z)\}. \tag{2.26}$$

Therefore, when the pseudoerror signal is minimized, $\underline{E'}(z)\rightarrow 0$, the function must fulfilled that $f\{\underline{Y}(z)\underline{G}(z)\underline{H}(z)\}=-\underline{E}(z)$. Considering $\underline{E}(z)\approx\underline{C}(z)\underline{D}(z)$, from (2.25) we obtain that

$$\begin{aligned} \underline{D}(z)(\underline{C}(z)-1) &= \underline{Y}(z)\underline{G}(z)\underline{H}(z), \\ \underline{D}(z) &= \frac{\underline{G}(z)}{\underline{C}(z)-1}\underline{H}(z)\underline{Y}(z). \end{aligned} \tag{2.27}$$

Multiplying both sides of (2.27) by $\underline{C}(z)$, we get

$$\underline{C}(z)\underline{D}(z) = \frac{\underline{C}(z)\underline{G}(z)}{\underline{C}(z)-1}\underline{H}(z)\underline{Y}(z). \tag{2.28}$$

obtaining the value of $\underline{E}(z)$ from which it is possible to derive the value of the desired function as the negative value of the right term of (2.28). Therefore, when $\underline{E'}(z)\rightarrow 0$,

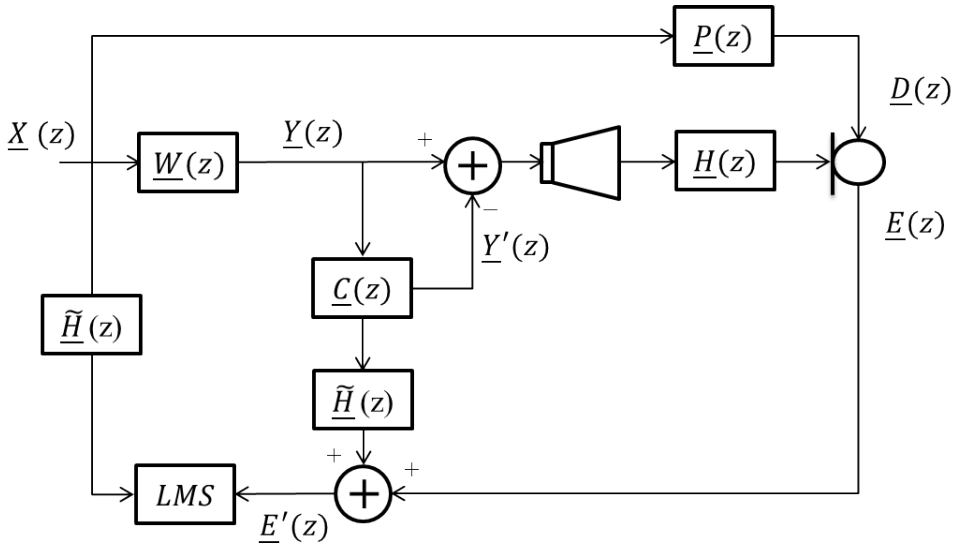


Figure 2.12: Block diagram of the adaptive algorithm for broadband ANE when $\underline{G}(z)=1-\underline{C}(z)$.

we get

$$f\{\underline{Y}(z),\underline{H}(z),\underline{G}(z)\} = -\frac{\underline{C}(z)\underline{G}(z)}{\underline{C}(z)-1}\underline{H}(z)\underline{Y}(z). \tag{2.29}$$

and therefore, the pseudoerror signal to be minimized can be obtained as

$$\underline{E}'(z) = \underline{E}(z) + \frac{\underline{C}(z)\underline{G}(z)}{1-\underline{C}(z)}\underline{H}(z)\underline{Y}(z). \tag{2.30}$$

Note that the computation of $\underline{E}'(z)$ involves the real acoustic path $\underline{H}(z)$ which, in practical environments, can be estimated as $\widetilde{\underline{H}}(z)$. In this way, we consider perfect estimation, i.e., $\widetilde{\underline{H}}(z)=\underline{H}(z)$. In addition, if $\underline{G}(z)=1-\underline{C}(z)$, we obtain from (2.30) that $\underline{E}'(z) = \underline{E}(z) + \underline{C}(z)\underline{H}(z)\underline{Y}(z)$ obtaining the same case as proposed in [92].

With the aim to minimize the pseudoerror signal $\underline{E}'(z)$, we use the LMS strategy where an adaptive filter must be estimated in order to equalize an acoustic noise by generating the appropriate control signal. In this way, the filter coefficients vector $\mathbf{w}(n)$ are updated over time by following the iterative solution,

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \mu\nabla_{\mathbf{w}}e'^2(n) \tag{2.31}$$

where $e'(n)$ is the time domain version of $\underline{E}'(z)$. Substituting (2.25) in (2.30), we obtain that

$$\underline{E}'(z) = \underline{D}(z) + \frac{\underline{G}(z)}{1-\underline{C}(z)}\underline{H}(z)\underline{Y}(z) = \underline{D}(z) + \underline{S}(z)\underline{Y}(z). \tag{2.32}$$

Considering that $\underline{S}(z)$ can be estimated as a FIR filter whose impulse response is $s(n)$, we get that

$$e'(n) = d(n) + s(n) * x(n) * w(n) \tag{2.33}$$

where $d(n)$ and $x(n)$ are the time domain version of $\underline{D}(z)$ and $\underline{Y}(z)$ respectively, being $*$ denotes linear convolution operator. Defining $x_s(n) = x(n) * s(n)$, the right term of (2.31) can be calculated as

$$\nabla_{\mathbf{w}} e'^2(n) = 2e'(n)\mathbf{x}_s(n), \tag{2.34}$$

where $\mathbf{x}_s(n)$ is the vector that contain the last L samples of $x_s(n)$ being L the adaptive filter length. Therefore, (2.31) can be redefined as

$$\mathbf{w}(n) = \mathbf{w}(n-1) - 2\mu e'(n)\mathbf{x}_s(n). \tag{2.35}$$

It should be noted that in the particular case where $\underline{G}(z)=1-\underline{C}(z)$ and then $\underline{S}(z)=\underline{H}(z)$, the equation of the adaptive filter coefficients update is the classical one of the filtered-x structure (since the infinite impulse response (IIR) filter $\underline{S}(z)$ can be now modelled as a FIR filter). However and due to this, the output signal $y(n)$ will suffer a delay which may affect the convergence speed of the algorithm. This problem can be solved using $\underline{G}(z)=1$, although it may occurr that the resultant filters cannot be estimated as transversal FIR filters. Furthermore, note that depending on $\underline{C}(z)$, filter singularities in both $\underline{S}(z)$ and $(1-\underline{C}(z))^{-1}$ may appear.

For these and other reasons, the narrowband technique is often preferred for equalizing tonal noise signals with multiple components [95]. In the narrowband ANE, only two-coefficient adaptive filter for each frequency to be controlled is required. However, in the case of broadband ANE, the size of the adaptive filter needs to be much longer in order to model $\underline{P}(z)/\underline{H}(z)$. In the narrowband approach, the reference signal is not required since it can be generated internally. On the other hand, in broadband ANE, a reference sensor is needed to monitor the noise signal. In the case of the reference sensor detects the signal coming from the cancelling actuator, acoustic feedback can be produced, which will degrade the system behavior, even leading to unstability [69]. In the narrowband ANE, the causality condition is preserved due to periodicity. However, in the broaband approach, causality problems may appear provoking the system performance degradation. However, the narrowband strategy can also present some issuess in certain scenarios. In real applications, the reference signal generated internally may suffer a deviation in its frequencies in relation to the components of the original tonal noise. This problem is refered as frequency missmatch (FM) and it has been proved that a FM of 1% is enough to degrade the ANE system performance [96]. In those cases, the use of the broadband method may solve the FM problem provided that the shaping filter be properly designed.

In summary, ANE strategies provide an immense range of possibilities for dealing with tonal noises. They not only can reduce existing noise levels but they can also modify its spectrum seeking for psychoacoustic or subjective criteria.

2.5 Conclusions

In this chapter, a brief overview of the necessary concepts for the understanding of this dissertation has been presented. As summary and for the sake of clarity, in the context of this thesis some considerations and terms must be remembered and taken into account:

- Our target is to take advantage of the benefits of acoustic sensor networks (*ASN*) to implement sound field control (SFC) systems in real environments with the help of adaptive signal processing. Concretely, we focus on the *active noise control (ANC)* and *active noise equalization* (*ANE*) applications.
- Regarding the adaptive signal algorithm, we have considered the *LMS strategy* and its variants with the *filtered-x structure* (*FxLMS*) to implement the previously commented SFC applications. As an exception, the *filtered-error* structure (*FeLMS*) is also used but only in a single ANE application.
- All SFC applications implemented in this dissertation work over *homogeneous ASN* composed of acoustic single-channel nodes, i.e., nodes are composed of a single sensor, a single actuator and a single processor with the same computational and communication capabilities.
- We define *acoustic node* as a device capable of obtaining information from one or more microphones and capable of generating signals via one or more loudspeakers. Moreover, every acoustic node has the ability to individually process signals as well as to interchange the necessary information with the other nodes using a suitable communication network.
- The term *centralized network* refers to a star-topology ASN where a central unit gathers data from all nodes and processes it in a centralized mannner. Similarly, we consider a *distributed network* as a ring-topology ASN which uses a set of nodes, placed strategically to reach a common objective. An output signal at each node is generated as a result of processing the signal captured by the node as well as the information received from other nodes, when there exists communication among the nodes. Every node processes signals independently and all the nodes are relevant for the proper performance of the global system.
- We define *centralized SFC system* (both ANC and ANE) as an ASN composed of passive nodes where the data processing is carried out by an unit central which controls multiple loudspeakers and multiple microphones to modify the soundfield. On the other hand, a *distributed SFC system* (both ANC and ANE)

is a distributed ASN which executes a SFC application by using a set of acoustic nodes placed strategically to reach the common objective of controlling an undesirable noise in some areas of interest. In the case of *non-collaborative distributed* or *decentralized SFC systems*, nodes do not exchange information. On the other hand, in *collaborative distributed SFC systems* communication among nodes is considered.

- For all cases, we assume that a single disturbance noise is devoted to be controlled. In addition we consider that all nodes haves access to the information of the acoustic noise signal either by using a reference sensor used to detect the single disturbance noise or a sinewave generator, as previously commented.

Note that, since the use of multiple actuators and sensors involves the presence of the acoustic interferences, collaboration among nodes may be required to reach the best possible performance i.e. the same as the centralized method. These acoustic coupling among nodes may degrade the performance of the sound control system since the signals captured by each node also depend on the signals generated by the other nodes of the network. However, depending on the nodes location, decentralized or non-collaborative strategies may be desirable in order to reduce the computational cost of the system and to avoid the inherent problems in communication among nodes [2]. For all these reasons, the use of distributed solutions for both ANC and ANE over ASN may be used to be a flexible and efficient solution for the creation of local control zones in enclosures.

Once the review of the state of the art from a multi-disciplinary point of view has been presented, the following chapters deal with the implementation of sound field control applications over distributed networks.

Chapter 3

On the implementation of ANC systems over distributed networks

Before starting the development of multichannel ANC systems over networks of distributed acoustic nodes, there are several aspects to keep in mind that can help us better understand the system behavior in order to achieve a successful implementation. In this regard, two relevant issues are covered by this chapter: 1) how the ANC system stability is affected by the acoustic enviroment and 2) if it is possible to gather centralized and distributed algorithms in an unique and generalized nomenclature. To this end, first a brief explanation regarding the implementation of ANC systems over ASNs is introduced. Then, we will analyze the importance of the acoustic system in the behavior of ANC applications. In addition, a generic formulation of the ANC problem over distributed networks based on the LMS algorithm is presented. Thus, both centralized and distributed (non-collaborative and collaborative) strategies can be derived from a generic point of view of the problem. Finally, with the aim to evaluate the ANC system behavior, some performance measures are introduced.

3.1 Analysis of acoustical interaction over distributed networks

In this section, we analyze the influence of the acoustic enviroment on the ANC system performance. Prior knowledge of the acoustic conditions where our control system is located is crucial and it can play a significant role in the system behaviour in terms of convergence rate and stability. To this end, we will introduce the concept of acoustic coupling trying to identify the conditions where it happens and trying to answer questions such as how it affects the coupling degree on collaboration between nodes or in which cases it is convenient for specific nodes to collaborate. These and

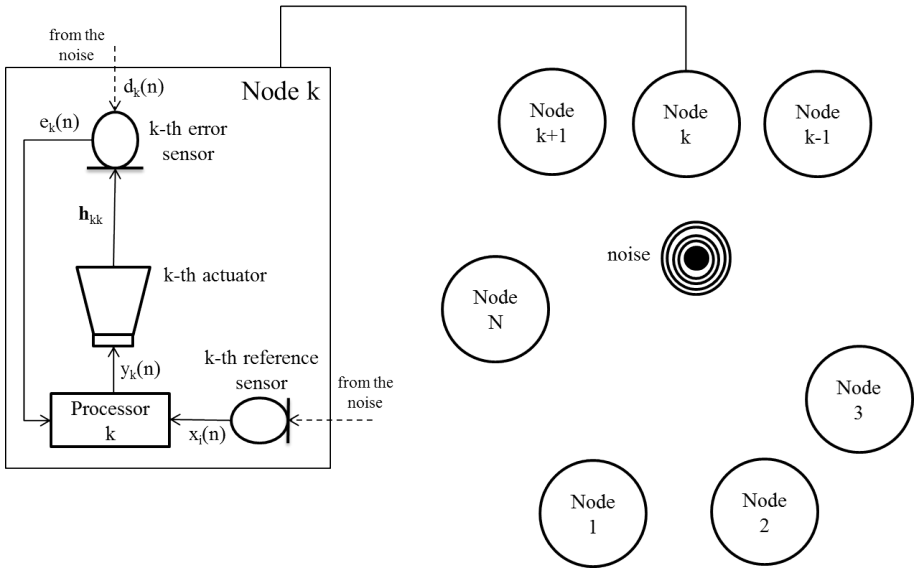


Figure 3.1: ANC system working over an ASN of N single-channel nodes.

other questions will try to be answered throughout this section. As a consequence, several experimental methods for trying to estimate the acoustic system influence on the ANC system stability are also presented in Appendix A.

However, let us first introduce a brief explanation regarding the implementation of ANC systems over distributed networks with the aim to justify the acoustic coupling problem.

3.1.1 Generic ANC problem statement

Let us consider a ANC system working over an homogeneous ASN of N single-channel nodes spatially distributed in some area, as shown is Figure 3.1. We assume that all the nodes have access to the same reference signal $x(n)$ at the discrete time instant n which is correlated with the unwanted noise. Our objetive is to estimate filter coefficients $\mathbf{w}_k(n)$ at every node to control the acoustic noise signal at the sensor locations, $d_k(n)$ (where $k{=}1, 2, \ldots, N$). As discussed in Section 2.4.1, adaptive techniques are usually considered on the estimation of $\mathbf{w}_k(n)$ [43]. To this end, the control signals $y_j(n)$ (where $j{=}1, 2, \ldots, N$), emitted by the actuators and propagated through the acoustic system, are designed to minimize the signals recorded at the sensors, called error signals and denoted by $e_k(n)$. As previously commented in the conclusions of Chapter 2, we consider the filtered-x structure in the adaptive algorithm. Therefore, a previous estimation of the acoustic channels is required. The

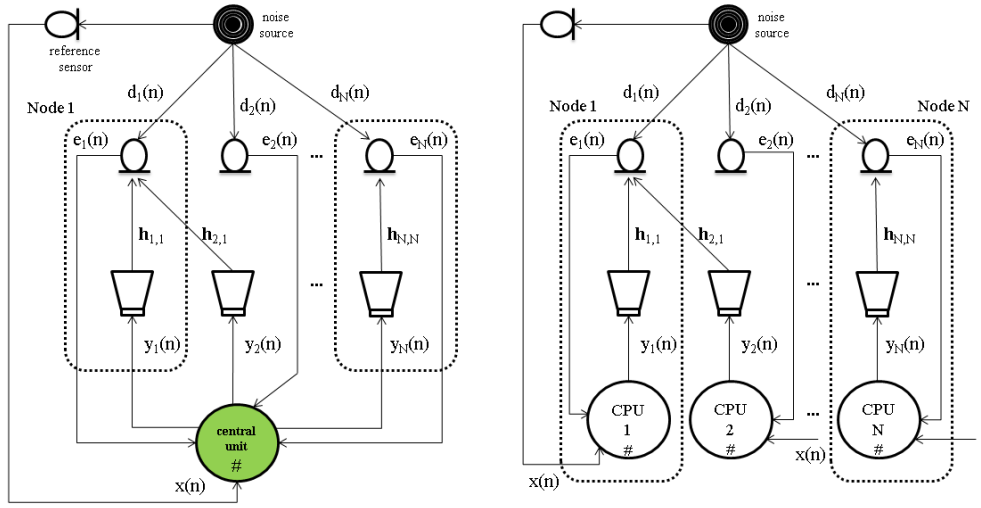


Figure 3.2: Scheme of (a) centralized and (b) decentralized ANC systems working over an ASN of N single-channel nodes.

acoustic channel that links the actuator of the node j and the sensor of the node k ($\mathbf{h}_{jk}$) is usually estimated by means of a FIR filter of M coefficients denoted as $\tilde{\mathbf{h}}_{jk}$. With regard to this, from now we assume ideal path estimation, i.e. $\tilde{\mathbf{h}}_{jk}$=$\mathbf{h}_{jk}$. The acoustic noise signal at the sensor of node k, $d_k(n)$, is calculated as the noise signal filtered through the acoustic channel between the noise source and the sensor of node k, denoted as $\mathbf{p}_k$. Without loss of generality, the reference signal used in the algorithm is considered to be the same as the noise signal. There exist several ways to achieve this objetive depending on the selected strategy. The ANC system aims to minimize a cost function $J(n)$ that depends on the acoustic field to be controlled. However, depending on the type of network, the specific objective, and consequently, this cost function may vary.

3.1.2 Centralized ANC system

For example, in a centralized network, all the error signals are necessary to calculate the coefficients of each filter $\mathbf{w}_k(n)$. Figure 3.2.(a) shows an centralized ANC system composed of N passive nodes and a single unit central which control the N actuators and the N sensors. Gathering the signals involved in the ANC system, the information captured by all the error sensors of the network is defined as

$$\mathbf{e}(n) = \mathbf{d}(n) \; + \; \mathbf{U}^T(n)\mathbf{w}(n). \tag{3.1}$$

where $\mathbf{e}(n)=[e_1(n)\ e_2(n)\dots\ e_N(n)]^T$ and $\mathbf{d}(n)=[d_1(n)\ d_2(n)\dots\ d_N(n)]^T$. The $[LN\times N]$ matrix $\mathbf{U}(n)$ is defined as

$$\mathbf{U}(n)=[\mathbf{u}_1(n)\ \mathbf{u}_2(n)\dots\ \mathbf{u}_N(n)]\ =\ \begin{bmatrix}\mathbf{u}_{11}(n) & \mathbf{u}_{12}(n) & \cdots & \mathbf{u}_{1N}(n)\\ \mathbf{u}_{21}(n) & \mathbf{u}_{22}(n) & \cdots & \mathbf{u}_{2N}(n)\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{u}_{N1}(n) & \mathbf{u}_{N2}(n) & \cdots & \mathbf{u}_{NN}(n)\end{bmatrix}, \tag{3.2}$$

where the vector $\mathbf{u}_{jk}(n)$ contains the last L samples of reference signal $x(n)$ filtered through $\mathbf{h}_{jk}$, $\mathbf{u}_{jk}(n)=\mathbf{X}(n)\mathbf{h}_{jk}$ with $\mathbf{X}(n)$ being a properly arranged matrix of size $L\times M$ defined as

$$\mathbf{X}(n)=\begin{bmatrix}x(n) & x(n-1) & \cdots & x(n-M+1)\\ x(n-1) & x(n-2) & \cdots & x(n-M)\\ \vdots & \vdots & \ddots & \vdots\\ x(n-L+1) & x(n-L) & \cdots & x(n-(L+M)+2)\end{bmatrix}, \tag{3.3}$$

and with $\mathbf{h}_{jk}=[h_{jk,1}\ h_{jk,2}\dots\ h_{jk,M}\ ]^T$. Vector $\mathbf{w}(n)=[\mathbf{w}_1^T(n)\ \mathbf{w}_2^T(n)\ \cdots\ \mathbf{w}_N^T(n)\ ]^T$ is defined as the global state of the network of size $[LN\times 1]$ with $\mathbf{w}_k(n)$ defined as $\mathbf{w}_k(n)=[w_{k,1}(n)\ w_{k,2}(n)\ \dots\ \ w_{k,L}(n)\ ]^T$, i.e., the vector that contains the L filter coefficients of the kth node.

These centralized systems are usually based on the well known multiple-error least-mean-square algorithm with the filtered-x structure (MEFxLMS) [3] which is devoted to minimize the sum of the squares of the measured signals at the N error sensors,

$$J(n)=E\Big\{\sum_{k=1}^{N}e_k^2(n)\Big\}=E\Big\{\sum_{k=1}^{N}|d_k(n)\ +\ \mathbf{u}_k^T(n)\mathbf{w}(n)|^2\Big\}=E\{\mathbf{e}^T(n)\mathbf{e}(n)\}, \tag{3.4}$$

being

$$e_k(n)=d_k(n)\ +\ \mathbf{u}_k^T(n)\mathbf{w}(n), \tag{3.5}$$

where the $[LN\times 1]$ vector $\mathbf{u}_k(n)$ is arranged as $\mathbf{u}_k(n)=[\mathbf{u}_{1k}^T(n)\ \mathbf{u}_{2k}^T(n)\ \dots\ \mathbf{u}_{Nk}^T(n)]^T$. The noise signal at the sensor of node k is defined as $d_k(n)=\mathbf{X}(n)_{(1,:)}\mathbf{p}_k$ where $\mathbf{X}(n)_{(1,:)}$ is the $[L\times 1]$ vector formed by the first column of $\mathbf{X}(n)$ (see (3.3)) and $\mathbf{p}_k=[p_{k,1}\ \ p_{k,2}\ \dots\ \ p_{k,M}\ ]^T$. It can be shown that the minimum MSE solution of (3.4), $\mathbf{w}_o$, can be obtained by solving

$$-\mathbf{R}_{\mathbf{U}}\mathbf{w}_o=\mathbf{r}_{\mathbf{U},\mathbf{d}}, \tag{3.6}$$

where $\mathbf{R}_{\mathbf{U}}=E\{\mathbf{U}(n)\mathbf{U}^T(n)\}$ and $\mathbf{r}_{\mathbf{U},\mathbf{d}}=E\{\mathbf{U}(n)\mathbf{d}(n)\}$. A practical solution is to use the traditional steepest-descent method to search the coefficients vector $\mathbf{w}(n)$ that minimizes $J(n)$

$$\mathbf{w}(n+1)=\mathbf{w}(n)\ -\mu\nabla_{\mathbf{w}}J(n), \tag{3.7}$$

where μ is the step-size parameter of the algorithm and $\nabla_{\mathbf{w}}=[\nabla^T_{\mathbf{w}_1}\nabla^T_{\mathbf{w}_2}\dots\ \nabla^T_{\mathbf{w}_N}]^T$ with $\nabla_{\mathbf{w}_k}$ as the gradient operator defined as the partial derivatives with respect to the coefficients vector $\mathbf{w}_k(n)$.

Assuming time invariance of $\mathbf{w}(n)$ and considering that $J(n)$ is approximated by its instantaneous value by using the LMS method, the global filter updating equation of the centralized (MEFxLMS) algorithm is stated as follows

$$\mathbf{w}(n+1) = \mathbf{w}(n)\ - 2\mu\mathbf{U}(n)\mathbf{e}(n) = \mathbf{w}(n)\ - 2\mu\sum_{k=1}^{N}\mathbf{u}_k(n)e_k(n), \tag{3.8}$$

where the solution for the j-th adaptive filter is obtained as

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n)\ - 2\mu\sum_{k=1}^{N}\mathbf{u}_{jk}(n)e_k(n). \tag{3.9}$$

Once (3.9) is calculated, the k-th node will generate the following output signal,

$$y_k(n) = \mathbf{w}_k^T(n)\ \mathbf{X}(n)_{(:,1)}. \tag{3.10}$$

However, the use of a centralized ANC system may entail major drawbacks, as discussed in Section 2.2. For these reasons, a distributed implementation of the ANC system may be required. Note that, as also commented in Section 2.2, a $(1{:}N{:}N)$ multichannel centralized ANC system can be distributed into N single-channel acoustic nodes (see Figure 3.2.(b)). In this way, while centralized systems work with all the signals generated by the loudspeakers and captured by the microphones, decentralized systems employ several nodes composed of independent processors which control a subset of loudspeakers from the signals picked up by a subset of microphones. The analysis of decentralized networks is outlined below.

3.1.3 Decentralized ANC system

In a decentralized distributed network, each node is usually devoted to minimize the mean power of the error signal, $e_k(n)$, picked up at its sensor k, adjusting the output signal $y_k(n)$ generated by its actuator k., i.e., only by using its available local data. Thus, a local control is carried out at each node independent of the rest of the nodes. Let us then consider that all single-channel nodes of the network converge to an optimal solution similar as reached by the nodes working in an isolated manner [34]. Then, asumming that convergence is achieved, the information captured by the error sensor at the node k in steady-state, may be approximated as

$$e_k(n) = d_k(n)\ +\ \mathbf{u}_{kk}^T(n)\mathbf{w}_k(n), \tag{3.11}$$

Thus, the cost function of the node k is given by

$$J_k(n) = E\{e_k^2(n)\} = E\{|d_k(n) \ + \ \mathbf{u}_{kk}^T(n)\mathbf{w}_k(n)|^2\}, \tag{3.12}$$

with its local minimum solution, $\mathbf{w}_{k,o}$, obtained from

$$-\mathbf{R}_{kk}\mathbf{w}_{k,o} = \mathbf{r}_{kk}, \tag{3.13}$$

where $\mathbf{R}_{kk}{=}E\{\mathbf{u}_{kk}(n)\mathbf{u}_{kk}^T(n)\}$ and $\mathbf{r}_{kk}{=}E\{\mathbf{u}_{kk}(n)d_k(n)\}$. Using a gradient-descent method to estimate the coefficients in an iterative manner, we obtain that

$$\mathbf{w}_k(n+1) = \mathbf{w}_k(n) \ - \mu\nabla_{\mathbf{w}_k} J_k(n), \tag{3.14}$$

As previously stated, the mean value in the cost function is approximated by its instantaneous value by using the LMS method. From (3.12) and applying the gradient operator in 3.11, we obtain the filter updating equation of the non-collaborative distributed FxLMS (NC-DFxLMS) algorithm for the kth node as follows

$$\mathbf{w}_k(n+1) = \mathbf{w}_k(n) \ - 2\mu\mathbf{u}_{kk}(n)e_k(n). \tag{3.15}$$

However, note that, in real scenarios, the ASN is usually acoustically coupled and consequently, the error signal captured by each node, $e_k(n)$, depends on, not only the noise signal to be cancelled and the control signal generated by that node, but also the acoustical interferences produced by the rest of the nodes. Thus, the information captured by the error sensor at the k-th node must take into account these acoustic interference signals, as defined in (3.5).

Therefore, the global update solution of the decentralized network $\mathbf{w}(n)$ is given by

$$\mathbf{w}(n+1) = \mathbf{w}(n) \ - 2\mu\tilde{\mathbf{U}}(n)\mathbf{e}(n), \tag{3.16}$$

where $\tilde{\mathbf{U}}(n)$ is a matrix of size $[LN{\times}N]$ defined as follows

$$\tilde{\mathbf{U}}(n) = \begin{bmatrix} \mathbf{u}_{11}(n) & \mathbf{0}_{L\times 1} & \cdots & \mathbf{0}_{L\times 1} \\ \mathbf{0}_{L\times 1} & \mathbf{u}_{22}(n) & \cdots & \mathbf{0}_{L\times 1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{L\times 1} & \mathbf{0}_{L\times 1} & \cdots & \mathbf{u}_{NN}(n) \end{bmatrix}. \tag{3.17}$$

where $\mathbf{0}_{L\times 1}$ is a null vector of size $L{\times}1$. Thus, the output signal that feeds the actuator of the k-th node is obtained as

$$y_k(n) = \mathbf{w}_k^T(n) \ \mathbf{X}(n)_{(:,1)} \tag{3.18}$$

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

Therefore, note that, in (3.16), the update of the $\mathbf{w}(n)$ does not consider all the information of the acoustic system. This lack of information in the NC-DFxLMS algorithm may cause that the ANC system becomes unstable in acoustically coupled networks (when the level of acoustic coupling is significant).

Due to the importance of this acoustic interference on the decentralized ANC system behavior, an introduction to the acoustic coupling problem is presented in the next section.

3.1.4 Introduction to the acoustic coupling

As previously commented, the acoustic coupling may degrade the performance of the decentralized ANC system since the signals captured by each node also depend on the signals generated by the other nodes of the network. For the sake of clarity, we define the concept of acoustic coupling in the scope of this thesis as the degree of cross acoustic interference among acoustic channels. Depending on the nodes location, acoustic coupling among their transducers may provoke degradation on the decentralized ANC system behaviour even causing instability. Furthermore, with the aim to cover larger sound zones, multiple acoustic nodes must be introduced in the network. Consequently, the behaviour of each node will be more affected by the acoustic coupling produced by the rest of the nodes. In the last decades, several approaches have been proposed for solving the acoustic coupling problem with the aim to stabilize the control system. They can be summarized in the following strategies:

- *Optimum nodes location*: the level of acoustic coupling depends mainly on how the actuators and sensors of each node are located within an enclosure. Therefore, the optimization of the sensor/actuator placement may be an effective solution to reach the system convergence [97]. But knowing the separation between nodes for which the acoustic coupling is relevant to the system stability would help to better understand the problem. Considering a two-nodes network in free space, a first approach might be to think that the system stability is guaranteed if the distance between the actuator-sensor pair of node j is smaller than the distance between the actuator of node k and sensor of node j, as demonstrated in [34] (for $j{=}1, 2$, $k{=}1, 2$ and $j{\neq}k$). The problem is that nodes are usually located inside enclosed spaces where room reflections may provoke that the previous assumption may not be fulfilled. In addition, in some enclosures, such as a cabin of a public transport (train, plane, bus, etc.), it is not possible to freely locate the actuators.

- *Control effort strategy*: Assume an optimum nodes location where the acoustic coupling is not relevant to the system stability. Depending on the power

level of the signals emitted by the actuators of the nodes, the acoustic coupling among nodes may increase provoking stability problems. In these cases, the use of effort constraints to limit the output signal generated by the actuators may ensure the system stability by introducing a control effort weighting in the cost function of the adaptive algorithm at each node [34]. However, larger constraint values may introduce strong degradation in the performance of the control system [43, 98].

- *Shaping eigenvalues matrix*: Recently, a frequency domain optimization method based on shaping the eigenvalues of a transfer matrix related to the acoustic channels was presented in [99]. The effectiveness of the method was validated but it was only analyzed for a two-channel decentralized system and, therefore, adapting the method for use in multichannel environments does not seem trivial. In [100] an optimized decentralized filtered-x least mean square (ODcFxLMS) algorithm has been proposed. In order to stabilize a decentralized system, the frequency responses of a set of filters designed by a weighted Least-Square method are optimized according to a frequency domain stability condition. It has been proved theoretically that a decentralized system could always be stabilized. However, although this method seems promising, the feasibility of its practical implementation in multichannel systems, in terms of computational complexity, is not discussed and it must be studied.

- *Collaborative networks*: Allowing information exchange between nodes, it is possible to stabilize the distributed ANC system achieving the same solution as centralized systems [22], even taking into account practical constraints.

- *Interference control*: A novel strategy based on controlling the signals emitted by the loudspeakers at each node of the network may be an interesting alternative with the aim of minimizing the effects of the acoustical interferences among nodes. This method will be presented in Chapter 4 since it is beyond the scope of this section.

In this thesis, we mainly focus on collaborative strategies as well as some non-collaborative techniques considering effort and interference control. However, before applying any of these strategies, a brief analysis attempting to relate the stability of ANC systems with the collaboration between nodes is presented below.

As proved in [101], the stability of a $(1{:}N{:}N)$ centralized ANC system based on the MEFxLMS algorithm analyzed in the frequency domain could be determined by examining the magnitude of the eigenvalues of a $[N \times N]$ transfer matrix related to

the acoustic channels, $\underline{\mathbf{H}}(\omega_0)\underline{\mathbf{H}}^H(\omega_0)$ evaluated at frequency ω_0, which is defined as

$$\underline{\mathbf{H}}(\omega_0)\underline{\mathbf{H}}^H(\omega_0)= \begin{bmatrix} \underline{H}_{11}(\omega_0) & \underline{H}_{12}(\omega_0) & \cdots & \underline{H}_{1N}(\omega_0) \\ \underline{H}_{21}(\omega_0) & \underline{H}_{22}(\omega_0) & \cdots & \underline{H}_{2N}(\omega_0) \\ \vdots & \vdots & \ddots & \vdots \\ \underline{H}_{N1}(\omega_0) & \underline{H}_{N2}(\omega_0) & \cdots & \underline{H}_{NN}(\omega_0) \end{bmatrix} \begin{bmatrix} \underline{H}^*_{11}(\omega_0) & \underline{H}^*_{21}(\omega_0) & \cdots & \underline{H}^*_{N1}(\omega_0) \\ \underline{H}^*_{12}(\omega_0) & \underline{H}^*_{22}(\omega_0) & \cdots & \underline{H}^*_{N2}(\omega_0) \\ \vdots & \vdots & \ddots & \vdots \\ \underline{H}^*_{1N}(\omega_0) & \underline{H}^*_{2N}(\omega_0) & \cdots & \underline{H}^*_{NN}(\omega_0) \end{bmatrix} \tag{3.19}$$

where $\underline{H}_{jk}(\omega_0)$ is the complex component of $\mathbf{h}_{jk}$ at frequency ω_0, the superscript $*$ denotes the conjugate operator and the superscript ^{H} is the Hermitian or conjugate-transpose operator, For simplicity, the frequency index will be omitted throughout the section.

Note that all eigenvalues of (3.19) are real and positives because $\underline{\mathbf{H}}\underline{\mathbf{H}}^H$ is Hermitian [102]. Thus, by selecting a proper value of the step-size parameter [101], the centralized ANC system will be stable. Therefore, the stability of the ANC system can be determined in advance by knowing the acoustic paths between each actuator-sensor pair of each node. This is viable since, in practical implementations, the real acoustic responses are usually identified off-line in a previous stage.

Bearing in mind how is arranged the matrix related to the reference signal filtered by the acoustic channels in both the centralized (3.2) and decentralized (3.16) cases, note that a more generic version of (3.19) can be considered as

$$(\mathbf{A}\circ\underline{\mathbf{H}})\underline{\mathbf{H}}^H= \left(\begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1N} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{N1} & \alpha_{N2} & \cdots & \alpha_{NN} \end{bmatrix} \circ \begin{bmatrix} \underline{H}_{11} & \underline{H}_{12} & \cdots & \underline{H}_{1N} \\ \underline{H}_{21} & \underline{H}_{22} & \cdots & \underline{H}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{H}_{N1} & \underline{H}_{N2} & \cdots & \underline{H}_{NN} \end{bmatrix} \right) \begin{bmatrix} \underline{H}^*_{11} & \underline{H}^*_{21} & \cdots & \underline{H}^*_{N1} \\ \underline{H}^*_{12} & \underline{H}^*_{22} & \cdots & \underline{H}^*_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{H}^*_{1N} & \underline{H}^*_{2N} & \cdots & \underline{H}^*_{NN} \end{bmatrix} \tag{3.20}$$

where matrix $\mathbf{A}$ is defined as a $[N\times N]$ collaboration matrix being α_{jk} a value which represents the cooperation between the node j and the node k and it may be chosen to be either 0 or 1. More specifically, the design of matrix $\mathbf{A}$ may be related to how nodes interchange information into the network. Note that, if $\mathbf{A}$ is a matrix with all entries equal to 1, then $(\mathbf{A}\circ\underline{\mathbf{H}})\underline{\mathbf{H}}^H$, i.e., nodes are ruled by a the centralized ANC system. Since $\underline{\mathbf{H}}\underline{\mathbf{H}}^H$ is a semi-positive definite matrix, all its eigenvalues are real and nonnegative, as previously stated. On the other hand, in a decentralized system, $\mathbf{A}$ should be an identity matrix of size $[N\times N]$. In this case or in a more general case

where any of the entries of $\mathbf{A}$ is 0, we cannot assert that $(\mathbf{A} \circ \underline{\mathbf{H}})\underline{\mathbf{H}}^H$ is semi-positive definite (not even symmetric). In that case, its eigenvalues may not be real or positive, causing instability. Therefore, the main problem here is to determine what matrix $\mathbf{A}$ must fulfill in order to avoid the system to be unstable. This can be approached by analyzing how changes in the entries of $\mathbf{A}$ affect on the system convergence. This may be viewed as an optimization problem described as follows: for any $[N \times N]$ complex matrix $\underline{\mathbf{H}}$, find a matrix $\mathbf{A}$ so that the real part of all the eigenvalues of $(\mathbf{A} \circ \underline{\mathbf{H}})\underline{\mathbf{H}}^H$ are equal to or greater than 0. However this problem it is not trivial since the extraction of the contribution of the entries of $\mathbf{A}$ from the eigendecomposition of $(\mathbf{A} \circ \underline{\mathbf{H}})\underline{\mathbf{H}}^H$ is challenging. Another possibility it is to design the matrix $\mathbf{A}$ from analysing certain characteristics of the acoustic paths matrix $\underline{\mathbf{H}}$. A discussion about this approach is carried out in Appendix A.

In summary, depending on the degree of acoustic coupling between nodes, their collaboration and the resulting design of matrix $\mathbf{A}$ is determined. A proper design of $\mathbf{A}$ will allow the acoustic system not to negatively affect the ANC system stability, which will finally depend on the configuration parameters of the LMS algorithm (mainly the step-size parameter) once the ANC is running.

3.2 Generic ANC formulation based on the LMS algorithm

The aim of this section is to present a generalized nomenclature to solve the problem of ANC over distributed networks. In this way, it will be possible to derive from this generic formulation, all the ANC algorithms used throughout this thesis, both centralized and distributed, collaborative and non-collaborative. As discussed in [22], from a network centralized approach it is possible to derive the contribution of every node in a ring-topology distributed network using an incremental strategy of the data exchange. Therefore, centralized approaches will be first obtained and then, the collaborative distributed algorithms will be derived from them in the following chapters. Thus, by simply introducing certain matrices and constants within a certain cost function, it is possible to rely on a generic formulation of the problem to derive certain adaptive solutions for the ANC problem.

3.2.1 Generic formulation for the ANC problem

Since both centralized and decentralized strategies are based on different objectives, different cost functions are derived. However, note that both (3.16) and (3.8) updating equations are very similar differing only the way the matrix $\mathbf{U}(n)$ is arranged. This may lead to the development of a generic notation with the aim to derive from it most of the algorithmic proposals carried out throughout this thesis. With this objective in mind, let us redefine some formulation and introduce some parameters with the aim

to provide a generic cost function which works in both centralized and distributed (non-collaborative and collaborative) networks.

Note that the cost functions defined (3.4) in and (3.12) are related to the acoustic field in the controlled zone. However, in the context of this thesis, the proposed cost function $J(n)$ is defined as a properly designed global function from which it is possible to derive most of the proposed algorithms presented in this thesis. Therefore, the relation between $J(n)$ and the acoustic field to be controlled may be implicit or not.

A brief summary of some different cases which are aimed to be included within this generic formulation is presented as follows:

- Non-collaborative and collaborative distributed ANC systems can be selected through the design of the following matrices $\boldsymbol{\Upsilon}_i$, $\boldsymbol{\Theta}_{i'i}$, $\boldsymbol{\Psi}_{i'i}$ and $\boldsymbol{\Phi}_{i'}$. These matrices may determine how $\mathbf{U}(n)$ is arranged and consequently, which strategies are considered to be used. They are based on the linear sum form of the matrix vectorization operation.
- The collaboration matrix $\mathbf{A}$ that determines which nodes collaborate in a distributed network is also included.
- In order to improve the convergence behavior of the control system, some efficient solution based on the affine projection algorithm will be included through the matrix $\mathbf{S}$.
- In practical ANC systems, certain constraints are usually considered within the adaptive algorithm to avoid system unstability. In this regard, control effort strategies [103, 104] are intended to be included in this formulation by means of the matrix $\boldsymbol{\beta}$.
- Finally, a non-collaborative strategy based on minimizing the acoustical interferences among nodes is desired to be derived from this generic formulation. To this end, the matrix $\boldsymbol{\Gamma}_i$ and the vector $\boldsymbol{\rho}_i$ are also included in $J(n)$.

As introduced in Section 3.1.2, a practical solution to minimize $J(n)$ is to use the traditional steepest-descent method to search the coefficients vector $\mathbf{w}(n)$,

$$\mathbf{w}(n+1) = \mathbf{w}(n) \; - \mu \nabla_{\mathbf{w}} J(n), \tag{3.21}$$

In this regard, we assume from now that the proposed cost function $J(n)$ is approximated by its instantaneous value by using the LMS method ($\nabla_{\mathbf{w}} J(n) \approx \nabla_{\mathbf{w}} \tilde{J}(n)$). Considering all these aspects, the *generic cost function of the network*, $J(n)$, is presented as

$$J(n) = \hat{\mathbf{e}}^T(n)\, \hat{\mathbf{e}}(n) + \mathbf{w}^T(n)\, \boldsymbol{\beta}\, \mathbf{w}(n) + \epsilon\, \mathbf{w}^T(n)\, \tilde{\mathbf{w}}(n)\, \bar{\mathbf{u}}(n), \tag{3.22}$$

where

$$\hat{\mathbf{e}}(n) = \hat{\mathbf{d}}(n) \; + \; [\hat{\mathbf{U}}^T]_c(n)\mathbf{w}(n), \tag{3.23}$$

with $\hat{\mathbf{e}}(n)$=$[\hat{\mathbf{e}}_1^T(n) \; \hat{\mathbf{e}}_2^T(n) \ldots \; \hat{\mathbf{e}}_N^T(n)]^T$ and $\hat{\mathbf{d}}(n)$=$[\hat{\mathbf{d}}_1^T(n) \; \hat{\mathbf{d}}_2^T(n) \ldots \; \hat{\mathbf{d}}_N^T(n)]^T$ as vectors of size $[QN\times1]$. Vectors $\hat{\mathbf{e}}_k(n)$ and $\hat{\mathbf{d}}_k(n)$ contain the last Q samples of the error and noise signals at the k-th node, respectively

$$\begin{aligned} \hat{\mathbf{e}}_k(n) &= [e_k(n) \; e_k(n-1) \ldots \; e_k(n-Q+1)]^T, \\ \hat{\mathbf{d}}_k(n) &= [d_k(n) \; d_k(n-1) \ldots \; d_k(n-Q+1)]^T. \end{aligned} \tag{3.24}$$

The matrix $[\hat{\mathbf{U}}]_c(n)$ of size $[LN\times QN]$ is calculated as

$$[\hat{\mathbf{U}}]_c(n) = \sum_{i'=1}^{I'} \sum_{i=1}^{I} \boldsymbol{\Upsilon}_i \mathbf{A} \boldsymbol{\Theta}_{i'i} \hat{\mathbf{U}}(n) \boldsymbol{\Psi}_{i'i} \mathbf{S}(n) \boldsymbol{\Phi}_{i'} \tag{3.25}$$

where I and I' are constants related with the number of nodes and $\boldsymbol{\Upsilon}_i$ is a weighting matrix of size $[LN\times LN]$ defined as

$$\boldsymbol{\Upsilon}_i = \begin{bmatrix} \boldsymbol{\upsilon}_{11} & \boldsymbol{\upsilon}_{12} & \cdots & \boldsymbol{\upsilon}_{1N} \\ \boldsymbol{\upsilon}_{21} & \boldsymbol{\upsilon}_{22} & \cdots & \boldsymbol{\upsilon}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\upsilon}_{N1} & \boldsymbol{\upsilon}_{N2} & \cdots & \boldsymbol{\upsilon}_{NN} \end{bmatrix}, \tag{3.26}$$

with $\boldsymbol{\upsilon}_{jk}$ as a $[L\times L]$ matrix given by

$$\boldsymbol{\upsilon}_{jk} = \begin{bmatrix} \upsilon_{jk}^{11} & \upsilon_{jk}^{12} & \cdots & \upsilon_{jk}^{1L} \\ \upsilon_{jk}^{21} & \upsilon_{jk}^{22} & \cdots & \upsilon_{jk}^{2L} \\ \vdots & \vdots & \ddots & \vdots \\ \upsilon_{jk}^{L1} & \upsilon_{jk}^{L2} & \cdots & \upsilon_{jk}^{LL} \end{bmatrix}. \tag{3.27}$$

$\mathbf{A}$ is a matrix of size $[LN\times LN]$ whose elements are diagonal matrices of size $[L\times L]$ with the same weighting parameter replicated L times,

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{\alpha}_{11} & \boldsymbol{\alpha}_{12} & \cdots & \boldsymbol{\alpha}_{1N} \\ \boldsymbol{\alpha}_{21} & \boldsymbol{\alpha}_{22} & \cdots & \boldsymbol{\alpha}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\alpha}_{N1} & \boldsymbol{\alpha}_{N2} & \cdots & \boldsymbol{\alpha}_{NN} \end{bmatrix}, \qquad \boldsymbol{\alpha}_{jk} = \begin{bmatrix} \alpha_{jk} & 0 & \cdots & 0 \\ 0 & \alpha_{jk} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_{jk} \end{bmatrix}. \tag{3.28}$$

where α_{jk} is the weighting parameter ($0\leqslant\alpha_{jk}\leqslant1$). Similarly as (3.26), the $[LN\times LN]$ matrix $\boldsymbol{\Theta}_{i'i}$ is defined as

$$\boldsymbol{\Theta}_{i'i} = \begin{bmatrix} \boldsymbol{\theta}_{11} & \boldsymbol{\theta}_{12} & \cdots & \boldsymbol{\theta}_{1N} \\ \boldsymbol{\theta}_{21} & \boldsymbol{\theta}_{22} & \cdots & \boldsymbol{\theta}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\theta}_{N1} & \boldsymbol{\theta}_{N2} & \cdots & \boldsymbol{\theta}_{NN} \end{bmatrix}, \tag{3.29}$$

with $\boldsymbol{\theta}_{jk}$ as a $[L\times L]$ matrix given by

$$\boldsymbol{\theta}_{jk} = \begin{bmatrix} \theta_{jk}^{11} & \theta_{jk}^{12} & \cdots & \theta_{jk}^{1L} \\ \theta_{jk}^{21} & \theta_{jk}^{22} & \cdots & \theta_{jk}^{2L} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{jk}^{L1} & \theta_{jk}^{L2} & \cdots & \theta_{jk}^{LL} \end{bmatrix}. \tag{3.30}$$

The matrix $\hat{\mathbf{U}}(n)$ of size $[LN\times QN]$ is defined as $\hat{\mathbf{U}}(n)=[\hat{\mathbf{U}}_1(n)\hat{\mathbf{U}}_2(n)\ldots\ \hat{\mathbf{U}}_N(n)]$, with the $LN\times Q$ matrix $\hat{\mathbf{U}}_k(n)$ composed by N matrices $\hat{\mathbf{U}}_{jk}(n)$ of size $L\times Q$ such as

$$\begin{aligned} \hat{\mathbf{U}}_k(n) &= [\hat{\mathbf{U}}_{1k}^T(n)\ \hat{\mathbf{U}}_{2k}^T(n)\ \ldots\ \hat{\mathbf{U}}_{Nk}^T(n)]^T, \\ \hat{\mathbf{U}}_{jk}(n) &= [\mathbf{u}_{jk}(n)\ \mathbf{u}_{jk}(n-1)\ \ldots\ \mathbf{u}_{jk}(n-Q+1)]. \end{aligned} \tag{3.31}$$

Moreover, we define the $[QN\times QN]$ weighting matrix $\boldsymbol{\Psi}_{i'i}$ as

$$\boldsymbol{\Psi}_{i'i}= \begin{bmatrix} \boldsymbol{\psi}_{11} & \boldsymbol{\psi}_{12} & \cdots & \boldsymbol{\psi}_{1N} \\ \boldsymbol{\psi}_{21} & \boldsymbol{\psi}_{22} & \cdots & \boldsymbol{\psi}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\psi}_{N1} & \boldsymbol{\psi}_{N2} & \cdots & \boldsymbol{\psi}_{NN} \end{bmatrix}, \tag{3.32}$$

where $\boldsymbol{\psi}_{jk}$ is a matrix of size $[Q\times Q]$ defined as

$$\boldsymbol{\psi}_{jk} = \begin{bmatrix} \psi_{jk}^{11} & \psi_{jk}^{12} & \cdots & \psi_{jk}^{1Q} \\ \psi_{jk}^{21} & \psi_{jk}^{22} & \cdots & \psi_{jk}^{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{jk}^{Q1} & \psi_{jk}^{Q2} & \cdots & \psi_{jk}^{QQ} \end{bmatrix}. \tag{3.33}$$

$\mathbf{S}(n)$ is a weighting matrix of size $[QN\times QN]$ defined as

$$\mathbf{S}(n)= \begin{bmatrix} \mathbf{S}_{11}(n) & \mathbf{S}_{12}(n) & \cdots & \mathbf{S}_{1N}(n) \\ \mathbf{S}_{21}(n) & \mathbf{S}_{22}(n) & \cdots & \mathbf{S}_{2N}(n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{N1}(n) & \mathbf{S}_{N2}(n) & \cdots & \mathbf{S}_{NN}(n) \end{bmatrix}, \tag{3.34}$$

with $\mathbf{S}_{jk}(n)$ as a $[Q\times Q]$ matrix given by

$$\mathbf{S}_{jk}(n) = \begin{bmatrix} s_{jk}^{11}(n) & s_{jk}^{12}(n) & \cdots & s_{jk}^{1Q}(n) \\ s_{jk}^{21}(n) & s_{jk}^{22}(n) & \cdots & s_{jk}^{2Q}(n) \\ \vdots & \vdots & \ddots & \vdots \\ s_{jk}^{Q1}(n) & s_{jk}^{Q2}(n) & \cdots & s_{jk}^{QQ}(n) \end{bmatrix}. \tag{3.35}$$

Similarly, we define the $[QN \times QN]$ weighting matrix $\mathbf{\Phi}_{i'}$ as

$$\mathbf{\Phi}_{i'} = \begin{bmatrix} \boldsymbol{\phi}_{11} & \boldsymbol{\phi}_{12} & \cdots & \boldsymbol{\phi}_{1N} \\ \boldsymbol{\phi}_{21} & \boldsymbol{\phi}_{22} & \cdots & \boldsymbol{\phi}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\phi}_{N1} & \boldsymbol{\phi}_{N2} & \cdots & \boldsymbol{\phi}_{NN} \end{bmatrix}, \tag{3.36}$$

where $\boldsymbol{\phi}_{jk}$ is a matrix of size $[Q \times Q]$ defined as

$$\boldsymbol{\phi}_{jk} = \begin{bmatrix} \phi_{jk}^{11} & \phi_{jk}^{12} & \cdots & \phi_{jk}^{1Q} \\ \phi_{jk}^{21} & \phi_{jk}^{22} & \cdots & \phi_{jk}^{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{jk}^{Q1} & \phi_{jk}^{Q2} & \cdots & \phi_{jk}^{QQ} \end{bmatrix}. \tag{3.37}$$

Coming back to equation (3.22), $\boldsymbol{\beta}$ is a diagonal matrix of size $LN \times LN$ whose diagonal elements are the N values of β_k replicated L times,

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1 & \mathbf{0}_L & \cdots & \mathbf{0}_L \\ \mathbf{0}_L & \boldsymbol{\beta}_2 & \cdots & \mathbf{0}_L \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_L & \mathbf{0}_L & \cdots & \boldsymbol{\beta}_N \end{bmatrix}, \qquad \boldsymbol{\beta}_k = \begin{bmatrix} \beta_k & 0 & \cdots & 0 \\ 0 & \beta_k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \beta_k \end{bmatrix}. \tag{3.38}$$

where $\mathbf{0}_L$ is a null matrix of size $L \times L$. The constant ϵ get values between 0 and 1, i.e., $\epsilon = [0, 1]$ and the $LN \times N$ matrix $\tilde{\mathbf{w}}(n)$ is calculated as

$$\tilde{\mathbf{w}}(n) = \sum_{i=1}^{I} \mathbf{\Gamma}_i \, \mathbf{w}(n) \boldsymbol{\rho}_i \tag{3.39}$$

where the $LN \times LN$ matrix $\mathbf{\Gamma}_i$ is defined as a diagonal matrix whose diagonal entries are the N values of γ_k replicated L times,

$$\mathbf{\Gamma} = \begin{bmatrix} \boldsymbol{\gamma}_{11} & \mathbf{0}_L & \cdots & \mathbf{0}_L \\ \mathbf{0}_L & \boldsymbol{\gamma}_{22} & \cdots & \mathbf{0}_L \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_L & \mathbf{0}_L & \cdots & \boldsymbol{\gamma}_{NN} \end{bmatrix}, \qquad \boldsymbol{\gamma}_{kk} = \begin{bmatrix} \gamma_k & 0 & \cdots & 0 \\ 0 & \gamma_k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_k \end{bmatrix}. \tag{3.40}$$

and $\boldsymbol{\rho}_i$ is a row vector of size $1 \times N$ composed of N values of ρ_k, i.e., $\boldsymbol{\rho}_i = [\rho_1 \; \rho_2 \; \ldots \; \rho_N]$. The $[N \times 1]$ vector $\bar{\mathbf{u}}(n)$ is arranged as

$$\bar{\mathbf{u}}(n) = \begin{bmatrix} \bar{\mathbf{u}}_1^T(n) \; \bar{\mathbf{u}}_1(n) \\ \bar{\mathbf{u}}_2^T(n) \; \bar{\mathbf{u}}_2(n) \\ \vdots \\ \bar{\mathbf{u}}_N^T(n) \; \bar{\mathbf{u}}_N(n) \end{bmatrix}, \tag{3.41}$$

where the vector $\bar{\mathbf{u}}_k(n)$ of size $[L\bar{N}\times 1]$ is defined as

$$\bar{\mathbf{u}}_k(n) = [\mathbf{u}_{k1}^T(n)\ \mathbf{u}_{k2}^T(n)\ \ldots\ \mathbf{u}_{kN}^T(n)]^T, \tag{3.42}$$

where the vector $\mathbf{u}_{kk}(n)$ is eliminated from (3.42) and, consequently, $\bar{N}=N-1$. In other words,

$$\bar{\mathbf{u}}_k^T(n)\ \bar{\mathbf{u}}_k(n) = \sum_{\substack{j=1\\ j\neq k}}^{N} \mathbf{u}_{kj}^T(n)\mathbf{u}_{kj}(n). \tag{3.43}$$

For attempting to clarify again the purpose of using these weighting matrices and constants, remerber that, for example, in (3.25), the design of $\boldsymbol{\Upsilon}_i$, $\boldsymbol{\Theta}_{i'i}$, $\boldsymbol{\Psi}_{i'i}$ and $\boldsymbol{\Phi}_{i'}$ may determine how $\mathbf{U}(n)$ is arranged and consequently, which strategy is considered to be used. $\mathbf{A}$ is related with collaboration matrix while $\mathbf{S}(n)$ will be designed to derive affine projection strategies. On the other hand, $Q=1$ means that affine projection strategies are not considered. Otherwise, in the case of $Q>1$, the matrix $\mathbf{S}(n)$ should be properly defined. β_k is a constant which would be non-zero in the case of control effort strategies are required. The constant ϵ controls whether or not the interference signals emitted from one node to the rest of nodes are considered to be also minimized. Therefore, it only makes sense to use it at the same time as non collaborative strategies.

Finally, the *generic filter updating equation of the network* based on the MEFxLMS algorithm is stated as follows,

$$\begin{aligned}\nabla_{\mathbf{w}}\tilde{J}(n) &= 2\mathbf{d}^T(n)[\hat{\mathbf{U}}^T]_c(n) + 2\mathbf{w}^T(n)[\hat{\mathbf{U}}]_c(n)[\hat{\mathbf{U}}^T]_c(n)\\ &\quad + 2\mathbf{w}^T(n)\,\boldsymbol{\beta} + \epsilon\sum_{i=1}^{I}\boldsymbol{\Gamma}_i\,\mathbf{w}(n)\boldsymbol{\rho}_i\,\bar{\mathbf{u}}(n)\\ &= 2\left(\boldsymbol{\beta}\mathbf{w}(n) + [\hat{\mathbf{U}}]_c(n)\,\hat{\mathbf{e}}(n) + \epsilon\sum_{i=1}^{I}\boldsymbol{\Gamma}_i\,\mathbf{w}(n)\boldsymbol{\rho}_i\,\bar{\mathbf{u}}(n)\right)\end{aligned} \tag{3.44}$$

and therefore

$$\mathbf{w}(n+1) = \mathbf{w}(n)\ - 2\mu\left(\boldsymbol{\beta}\mathbf{w}(n) + [\hat{\mathbf{U}}]_c(n)\,\hat{\mathbf{e}}(n) + \epsilon\sum_{i=1}^{I}\boldsymbol{\Gamma}_i\,\mathbf{w}(n)\boldsymbol{\rho}_i\,\bar{\mathbf{u}}(n)\right). \tag{3.45}$$

In the following chapters, we present the specific solution achieved at each distributed strategy derived from the generic cost function depicted in (3.22).

3.3 Performance measures

In order to evaluate the performance of the different distributed algorithms for both broadband and tonal noises, in this section we present different parameters which

can be used. We define the Noise Reduction at node k, $\mathrm{NR}_k(n)$, as the ratio in dB between the estimated error power with and without the application of the active noise controller,

$$\mathrm{NR}_k(n) = 10 \cdot \log_{10}\left[\frac{P_{e_k}(n)}{P_{d_k}(n)}\right], \tag{3.46}$$

where $P_{d_k}(n)$ is the signal power of $d_k(n)$ picked up at the kth microphone when the ANC system is inactive and $P_{e_k}(n)$ is the signal power of $e_k(n)$ measured at the kth microphone when the ANC system is on. Moreover, these signal powers have been estimated using an exponential windowing from the instantaneous signals. Moreover, we define the Noise Reduction in the whole network, $NR(n)$, as

$$\mathrm{NR}(n) = 10 \cdot \log_{10}\left[\frac{1}{N}\sum_{k=1}^{N}\frac{P_{e_k}(n)}{P_{d_k}(n)}\right]. \tag{3.47}$$

In addition and with the aim to evaluate the performance of the algorithms for tonal noises (single or multi-frequency), we define the steady-state Noise Reduction of the network at excitation frequency f_m, $\mathrm{NR}_{f_m}(n)$, expressed in dB as:

$$\lim_{n\to\infty}\mathrm{NR}_{f_m}(n) = \lim_{n\to\infty} 10 \cdot \log_{10}\left[\frac{1}{N}\sum_{k=1}^{N}\frac{P_{e_k}(n)}{P_{d_k}(n)}\right], \tag{3.48}$$

where $m = 0, 1, 2, .., M/2$ being M the size of the Discrete Fourier Transform (DFT) used. With the aim to describe the variation of the error signal power versus frequency for tonal noises, we use the power spectral density (PSD) estimate of the error signal using Welch's overlapped segment averaging estimator [105]. The error signal is divided into the longest possible segments with 50% overlap. Each segment is windowed with a Hamming window. The modified periodograms are averaged to obtain the PSD estimate.

3.3.1 Simulation setting

In regard to the simulations configuration, throughout this thesis we have used real acoustic channels measured in a listening room of 9.36 meters long, 4.78 meters wide and 2.63 meters high, located at the Audio Processing Laboratory of the Polytechnic University of Valencia, as depicted in Figure 3.4. Some examples of the impulses responses of this listening room are available at [106]. These acoustic channels have been modeled as FIR filters of M=256 coefficients and at a sampling rate of 2 kHz. The election of this number of coefficients is due to the impulse responses have hardly any energy in the following samples.

With respect to networks design, all the algorithms have been tested in homogeneous ASNs composed of N single-channel nodes, as shown in Figure 3.3. The most

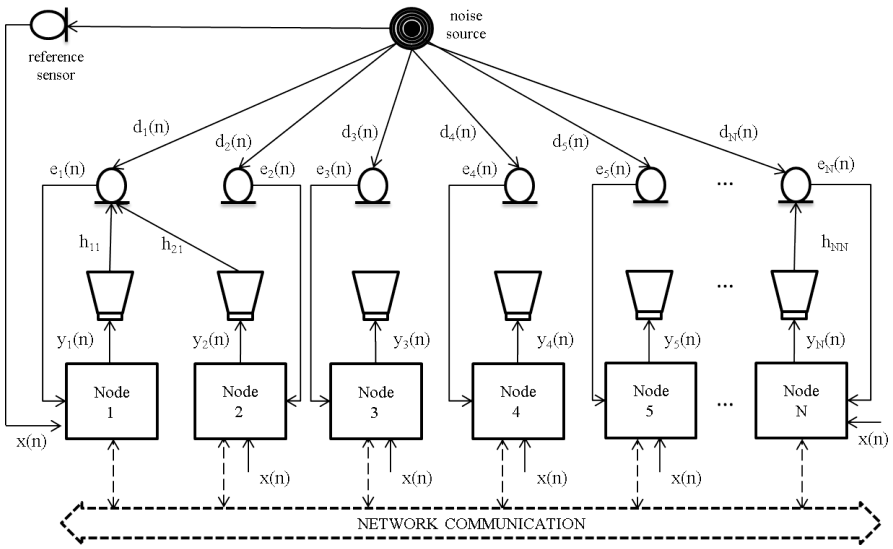


Figure 3.3: ASN of N acoustic nodes that suports a ANC system. The posibility of communication among nodes is represented by dashed lines.

commonly used ASN configuration is depicted as follows: N nodes composed of one loudspeaker and one microphone were considered. An equal separation of 20 cm between adjacent loudspeakers was selected. The microphones were placed opposite to the loudspeakers and separated 74 cm away from them. The separation between the microphones was 20 cm. All microphones and loudspeakers involved were located at a height of 147 cm. The tested distribution emulates a real ANC application where we would seek to create local quiet zones in enclosures (such as a cabin of a public transport) using ASNs of acoustic nodes with similar separation as detailed above. We have usually considered a wideband zero-mean Gaussian white noise with unit variance as disturbance signal and a step size parameter as the highest value that ensures the stability of the algorithms at each case. For simplicity, we assume that each node has access to the reference signal through an alternative broadcast channel (wideband noises) or by using a sinewave generator (tonal noises), as commented in Chapter 2. Although in some cases a lower number of coefficients may be sufficient to acheive the adaptation, we have usually selected the adaptive filter length to $L = 150$ coefficients in order to ensure accurate results. The noise signal has been generated by a primary loudspeaker located 320 cm away from both the secondary loudspeakers and error microphones. Therefore, all simulations carried out throughout the thesis usually consider this configuration, unless otherwise indicated.

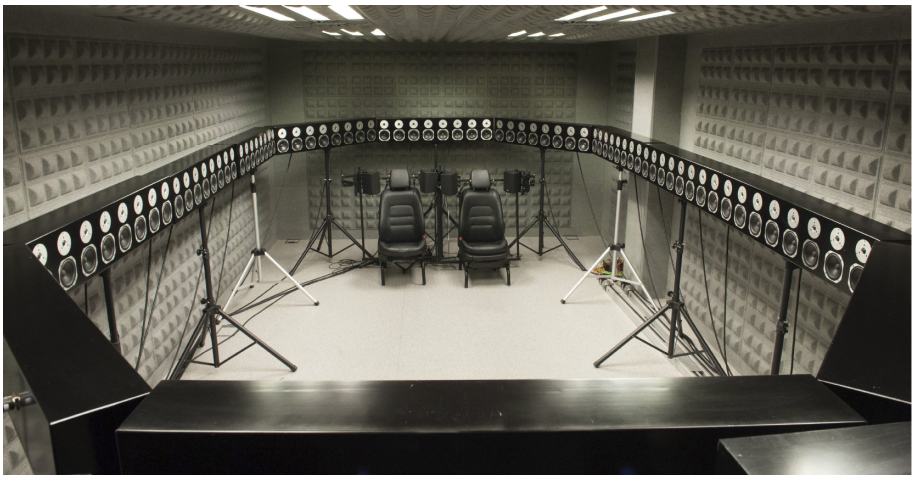


Figure 3.4: A photo of the GTAC listening room.

Chapter 4

ANC applications over non-collaborative networks

In this chapter, a performance analysis of a distributed ANC system over an acoustically coupled decentralized network is presented. To this end, the decentralized algorithm for distributed networks is presented as well as an analysis of its mean behavior with the aim to study its stability conditions. In addition, the acoustical interferences which may degrade the performance of the decentralized ANC system are modelled. This chapter also includes the description of some practical strategies based on the previous analysis with the aim to minimize the acoustical coupling among the nodes of a decentralized network provided that the system stability is maintained. Firstly, some existing methods based on limiting the power of the signals emitted by the loudspeakers are presented. Then, an intuitive strategy to reach the objective considering an interference control in the cost function of the distributed algorithm at each node is proposed. The simulation results carried out to compare the performance of these decentralized methods are also presented. Finally, the main conclusions of the chapter are outlined.

4.1 Introduction

Decentralized systems are more sensitive to the acoustic enviroment where nodes are located. In the case that nodes are located at such a distance that there is no acoustic interaction among the nodes (uncoupled nodes), the system stability may be achieved by using a decentralized ANC system [34] [107]. [34] concludes that an analysis of the eigenvalues distribution of the autocorrelation matrix of the filtered reference signals provides a necessary and sufficient condition for convergence. However, in many ANC systems, microphones are located very close to each other in order to

increase the size of the quiet zone by overlapping several control areas. As a result, the different acoustic channels involved will be acoustically coupled. In those cases, the decentralized distributed algorithm should therefore be modified to achieve good results even in coupled systems.

Several strategies have been proposed in order to reach the convergence in decentralized systems, as discussed in Section 3.1. Since the level of acoustic interaction will depend on the location of both actuators and sensors, a possible efficient solution may be the optimization of the sensor/actuator placement [97, 108]. However, in some enclosures, such as a cabin of a public transport (train, plane, bus, etc.), it is not possible to freely locate the actuators [109]. On the other hand, it is known that it is possible to stabilize the decentralized ANC system controlling the value of a regularization parameter [34]. The stabilization of such an arrangement can be assured by limiting the output signal at each node. However, the steady-state error increases with the regularization parameter [98, 110]. A simple solution to improve the performance of this leaky LMS algorithm is to use constraints strategies [104] to saturate the control signal, when the output is too large, or also scale the filter weights in order to avoid large oscillations in the coefficients' update. In [111], a decentralized algorithm for implementing multichannel active sound profiling applications that do not interfere with each other channel is presented. However, since the control units exchange data to compensate for interferences caused by the others controllers, it can not be implemented in a decentralized network. For this reason, in this chapter we propose a simple but effective decentralized strategy based on minimizing the effects of the acoustical intereferences among nodes over a distributed network.

Moreover, the stability of a decentralized distributed ANC system must be analyzed in order to know if collaboration is required to achieve the system stability. With regard to this, in this chapter we aim to identify the conditions in which single-channel nodes of an acoustically coupled network achieve the ANC system convergence working in an independent way or, on the contrary, nodes need to be clustered in a multichannel centralized system or using a collaborative learning. In other words, we want to know when it is necessary to execute a multichannel algorithm instead of several single-channel algorithms working in a parallel and independent way. To this end, the performance of the decentralized ANC (NC-DFxLMS) algorithm presented in [22] and previously discussed in 3.1.3, has been analyzed for a distributed network composed of acoustically coupled nodes and with no communication constraints. Since the acoustic interaction among nodes is usually present, an interference model that give us an approximation about the degradation of an ANC system when a node is working collaboratively with respect to work isolated has also been proposed. In addition, some decentralized strategies are analyzed in order to stabilize an ANC system over a distributed network where acoustic interaction among nodes lead to the system divergence. Regarding to this, as mentioned previously, a novel decentralized algorithm is proposed based on controlling the interference signals emitted by each

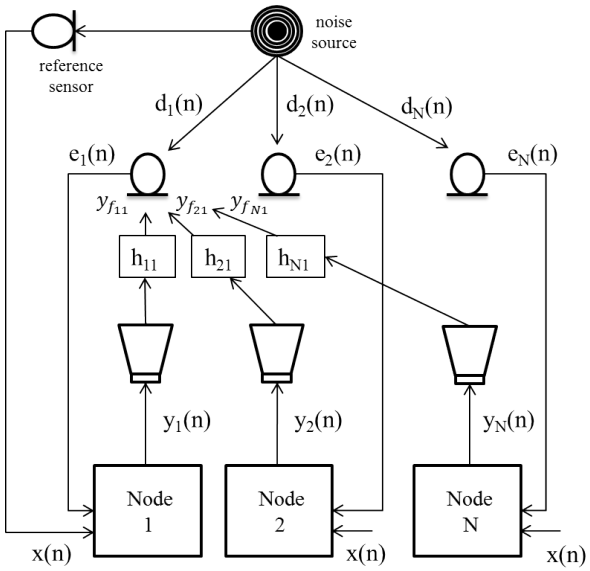


Figure 4.1: Decentralized ANC system working over an ASN of N single-channel nodes.

node of the network.

4.2 Decentralized problem statement

Therefore, let us consider a decentralized distributed network composed of N single-channel nodes randomly distributed in some area as shown in Figure 4.1. The objective is to control a sound field cancelling an acoustic noise at specific sensor locations using this network. More specifically, each node aims at cancelling the acoustic noise signal in its sensor location only by using its available local data. To this end, we present below the specific solution achieves at each decentralized strategy derived from the generic cost function depicted in (3.22). For all decentralized cases presented below, $\boldsymbol{\Upsilon}_i$ is denoted as a diagonal matrix of size $[LN \times LN]$ with only a non-zero matrix at the $(1+L(i-1):iL, 1+L(i-1):iL)$ position, $\boldsymbol{v}_{ii}$, which is an identity matrix of size $[L \times L]$,

$$\boldsymbol{\Upsilon}_i = \begin{bmatrix} \mathbf{0}_L & \cdots & \mathbf{0}_L & \cdots & \mathbf{0}_L \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_L & \cdots & \boldsymbol{v}_{ii} & \cdots & \mathbf{0}_L \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_L & \cdots & \mathbf{0}_L & \cdots & \mathbf{0}_L \end{bmatrix}, \qquad \boldsymbol{v}_{ii} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \tag{4.1}$$

From (3.25), we assume that $I'=1$ and $i'=i$. Therefore, $\boldsymbol{\Theta}_{ii}$ is denoted as a diagonal matrix of size $[LN\times LN]$ with only a non-zero matrix at the $(1+L(i-1){:}iL, 1+L(i-1){:}iL)$ position, $\boldsymbol{\theta}_{ii}$, which is an identity matrix of size $[L\times L]$,

$$\boldsymbol{\Theta}_{ii}=\begin{bmatrix} \mathbf{0}_L & \cdots & \mathbf{0}_L & \cdots & \mathbf{0}_L \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_L & \cdots & \boldsymbol{\theta}_{ii} & \cdots & \mathbf{0}_L \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_L & \cdots & \mathbf{0}_L & \cdots & \mathbf{0}_L \end{bmatrix}, \qquad \boldsymbol{\theta}_{ii}=\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \tag{4.2}$$

Moreover, the $[QN\times QN]$ matrix $\boldsymbol{\Psi}_{ii}$ is designed as a diagonal matrix with only a non-zero matrix at the $(1+L(i-1){:}iL, 1+L(i-1){:}iL)$ position, $\boldsymbol{\psi}_{ii}$, which is defined as an identity matrix of size $[Q\times Q]$,

$$\boldsymbol{\Psi}_{ii}=\begin{bmatrix} \mathbf{0}_Q & \cdots & \mathbf{0}_Q & \cdots & \mathbf{0}_Q \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_Q & \cdots & \boldsymbol{\psi}_{ii} & \cdots & \mathbf{0}_Q \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_Q & \cdots & \mathbf{0}_Q & \cdots & \mathbf{0}_Q \end{bmatrix}, \qquad \boldsymbol{\psi}_{ii}=\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \tag{4.3}$$

where $\mathbf{0}_Q$ is a null matrix of size $Q\times Q$. The $[QN\times QN]$ matrix $\boldsymbol{\phi}_i$ is similarly defined as matrix $\boldsymbol{\Psi}_{ii}$, that is, a diagonal matrix of size $[QN\times QN]$ with only an $[Q\times Q]$ identity matrix at the $(1+L(i-1){:}iL, 1+L(i-1){:}iL)$ position, $\boldsymbol{\psi}_{ii}$,

$$\boldsymbol{\Phi}_i=\begin{bmatrix} \mathbf{0}_Q & \cdots & \mathbf{0}_Q & \cdots & \mathbf{0}_Q \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_Q & \cdots & \boldsymbol{\phi}_{ii} & \cdots & \mathbf{0}_Q \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_Q & \cdots & \mathbf{0}_Q & \cdots & \mathbf{0}_Q \end{bmatrix}, \qquad \boldsymbol{\phi}_{ii}=\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}. \tag{4.4}$$

Therefore, considering $I=N$, $[\hat{\mathbf{U}}]_c(n)$ can be calculated as

$$[\hat{\mathbf{U}}]_c(n)=\sum_{i=1}^{N}\boldsymbol{\Upsilon}_i\mathbf{A}\boldsymbol{\Theta}_{ii}\hat{\mathbf{U}}(n)\boldsymbol{\Psi}_{ii}\mathbf{S}(n)\boldsymbol{\Phi}_i$$

$$=\begin{bmatrix} \alpha_{11}\hat{\mathbf{U}}_{11}(n)\mathbf{S}_{11}(n) & \mathbf{0}_{L\times Q} & \cdots & \mathbf{0}_{L\times Q} \\ \mathbf{0}_{L\times Q} & \alpha_{22}\hat{\mathbf{U}}_{22}(n)\mathbf{S}_{22}(n) & \cdots & \mathbf{0}_{L\times Q} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{L\times Q} & \mathbf{0}_{L\times Q} & \cdots & \alpha_{NN}\hat{\mathbf{U}}_{NN}(n)\mathbf{S}_{NN}(n) \end{bmatrix}, \tag{4.5}$$

where $\mathbf{0}_{L\times Q}$ is a null matrix of size $L\times Q$.

4.3 Decentralized FxLMS algorithm

Note that, for the fully decentralized case, we must consider Q=1 and $\mathbf{S}(n)$ defined as an identity matrix of size $N \times N$ such as

$$\mathbf{S}(n)= \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}. \tag{4.6}$$

Thus and also considering α_{kk}=1, (4.5) can be simplified as

$$[\hat{\mathbf{U}}]_c(n) = \begin{bmatrix} \mathbf{u}_{11}(n) & \mathbf{0}_{L\times 1} & \cdots & \mathbf{0}_{L\times 1} \\ \mathbf{0}_{L\times 1} & \mathbf{u}_{22}(n) & \cdots & \mathbf{0}_{L\times 1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{L\times 1} & \mathbf{0}_{L\times 1} & \cdots & \mathbf{u}_{NN}(n) \end{bmatrix}, \tag{4.7}$$

which is the same matrix as $\tilde{\mathbf{U}}(n)$ defined in (3.17) for the decentralized approach. In the particular case where $\boldsymbol{\beta}$ is a null matrix and ϵ=0, the cost function (3.22) can be expressed as

$$J(n) = (\mathbf{d}(n) + \tilde{\mathbf{U}}^T(n)\mathbf{w}(n))^T(\mathbf{d}(n) + \tilde{\mathbf{U}}^T(n)\mathbf{w}(n)), \tag{4.8}$$

from which we can derive the global update solution for a decentralized ANC system as

$$\mathbf{w}(n+1) = \mathbf{w}(n) \ - 2\mu\tilde{\mathbf{U}}(n)\mathbf{e}(n), \tag{4.9}$$

and consequently, the filter updating equation of the *decentralized* or *non-collaborative distributed FxLMS* (*NC-DFxLMS*) algorithm for the k-th node, calculated as

$$\mathbf{w}_k(n+1) = \mathbf{w}_k(n) \ - 2\mu\mathbf{u}_{kk}(n)e_k(n). \tag{4.10}$$

It is easy to note that both solutions (4.9) and (4.10) are the same as obtained in Section 3.1.3. Complementary, note that (4.8) can be expressed for the k-th node of the network as

$$J_k(n) = |d_k(n) \ + \ \mathbf{u}_{kk}^T(n)\mathbf{w}_k(n)|^2, \tag{4.11}$$

which is the cost function for the fully decentralized approach depicted in (3.12). A description of the NC-DFxLMS algorithm is given in ***Algorithm 1***.

In the next section, we present the conditions under which all the nodes of the decentralized network converge when they work simultaneously.

Algorithm 1: *NC-DFxLMS algorithm for N-nodes ASN.*

1: **for all** *node* $1 \leq k \leq N$ **do**
2: $y_k(n) = \mathbf{w}_k^T(n-1)\,[\mathbf{X}(n)]_{(:,1)}$ % Generate output signal
3: $\mathbf{u}_{kk}(n) = \mathbf{X}(n)\mathbf{h}_{kk}$ % Vector that contains reference signal filtered by estimated acoustic channels
4: $\mathbf{w}_k(n) = \mathbf{w}_k(n-1) - 2\mu\mathbf{u}_{kk}(n)e_k(n)$ % Update local state
5: **end for**

4.3.1 Convergence analysis

Then, the performance of the NC-DFxLMS algorithm is analyzed in terms of the mean of the decentralized network filter coefficients. Let us consider a network of N single-channel nodes where each node converges to its optimal solution when working in isolation. The objective of this section is to determine what conditions must be satisfied for all nodes to achieve convergence when working simultaneously in a non-collaborative distributed network.

Since the ANC system is usually acoustically coupled, the error signal captured by each node depends on, not only the noise signal to be cancelled and the control signal generated by that node, but also the acoustical interferences produced by the rest of the nodes. Therefore, the information captured by all the error sensors of the network is defined as

$$\mathbf{e}(n) = \mathbf{d}(n) \; + \; \mathbf{U}^T(n)\mathbf{w}(n). \tag{4.12}$$

Substituting (4.12) in (4.9), we obtain

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) - 2\mu\tilde{\mathbf{U}}(n)(\mathbf{d}(n) + \mathbf{U}^T(n)\mathbf{w}(n)) \\ &= \mathbf{w}(n) - 2\mu\tilde{\mathbf{U}}(n)\mathbf{d}(n) - 2\mu\tilde{\mathbf{U}}(n)\mathbf{U}^T(n)\mathbf{w}(n). \end{aligned} \tag{4.13}$$

As we need to deal with the expectation operator, we shall rely on several assumptions in the following analysis. Therefore, to study the mean behavior of the filter coefficients $\mathbf{w}(n)$, some statistical assumptions between different vectors and matrices must be considered:

- *A.1)* $\tilde{\mathbf{U}}_k(n)$ *and* $\mathbf{d}(n)$ *are wide-sense stationary.*
- *A.2) Elements of* $\tilde{\mathbf{U}}_k(n)$ *are statistically uncorrelated of elements of* $\mathbf{w}(n)$.
- *A.3) Time invariance of* $\mathbf{w}(n)$ *when convergence is achieved.*

(4.14)

Assuming that the network achieves the convergence, i.e., $\mathbf{w}(n)$ converges to its optimal solution $\mathbf{w}_o$ as $n \to \infty$, we can obtain from (4.13) the next expression

$$\begin{aligned} \mathbf{w}_o &= \mathbf{w}_o \; - 2\mu\tilde{\mathbf{U}}(n)\mathbf{d}(n) \; - 2\mu\tilde{\mathbf{U}}(n)\mathbf{U}^T(n)\mathbf{w}_o, \\ \tilde{\mathbf{U}}(n)\mathbf{d}(n) &= \; -\tilde{\mathbf{U}}(n)\mathbf{U}^T(n)\mathbf{w}_o. \end{aligned} \tag{4.15}$$

Substituting (4.15) in (4.13) and substracting $\mathbf{w}_o$ from both sides of the equation, we obtain

$$\begin{aligned}\mathbf{w}(n+1)-\mathbf{w}_o &= \mathbf{w}(n) \ +2\mu\tilde{\mathbf{U}}(n)\mathbf{U}^T(n)\mathbf{w}_o \\ &\quad -2\mu\tilde{\mathbf{U}}(n)\mathbf{U}^T(n)\mathbf{w}(n)-\mathbf{w}_o \\ &= (I_{LN}-2\mu\tilde{\mathbf{U}}(n)\mathbf{U}^T(n))(\mathbf{w}(n)-\mathbf{w}_o),\end{aligned} \tag{4.16}$$

where I_{LN} is a identity matriz of size $[LN\times LN]$. To see how $\mathbf{w}(n)$ converges to $\mathbf{w}_o$, we analyze its mean behaviour. Taking expectations of both sides and considering the coefficient error vector as $\mathbf{v}(n)=E\{\mathbf{w}(n)-\mathbf{w}_o\}$, we can rewrite (4.16) as

$$\mathbf{v}(n+1) = (I_{LN}-2\mu\mathbf{R})\mathbf{v}(n). \tag{4.17}$$

where $\mathbf{R}$ is a matrix of size $[LN\times LN]$ defined as

$$\mathbf{R} = E\{\tilde{\mathbf{U}}(n)\mathbf{U}^T(n)\} = \begin{bmatrix}\mathbf{R}_{11} & \mathbf{R}_{12} & \cdots & \mathbf{R}_{1N}\\ \mathbf{R}_{21} & \mathbf{R}_{22} & \cdots & \mathbf{R}_{2N}\\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{N1} & \mathbf{R}_{N2} & \cdots & \mathbf{R}_{NN}\end{bmatrix}, \tag{4.18}$$

with

$$\mathbf{R}_{jk} = \begin{bmatrix}R_{jk}(0) & R_{jk}(1) & \cdots & R_{jk}(L-1)\\ R_{jk}(1) & R_{jk}(0) & \cdots & R_{jk}(L-2)\\ \vdots & \vdots & \ddots & \vdots \\ R_{jk}(L-1) & R_{jk}(L-2) & \cdots & R_{jk}(0)\end{bmatrix}, \tag{4.19}$$

being $R_{jk}(l)=R_{xx}(l)*\mathbf{h}_{jj}(l)*\hat{\mathbf{h}}_{kj}(l)$ with $R_{xx}(l)=E\{x(n+l)\,x(n)\}$ and $\hat{\mathbf{h}}_{kj}(l)=\mathbf{h}_{kj}(-l)$. Both expressions are calculated as follows.

Note that the matrix $\mathbf{R}$ defined in (4.18) can be expressed as

$$\begin{aligned}
\mathbf{R} &= E\{\tilde{\mathbf{U}}(n)\mathbf{U}^T(n)\} \\
&= E\left\{ \begin{bmatrix} \mathbf{u}_{11}(n) & 0 & \cdots & 0 \\ 0 & \mathbf{u}_{22}(n) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{u}_{NN}(n) \end{bmatrix} \right. \\
&\qquad\qquad \left. \begin{bmatrix} \mathbf{u}_{11}^T(n) & \mathbf{u}_{21}^T(n) & \cdots & \mathbf{u}_{N1}^T(n) \\ \mathbf{u}_{12}^T(n) & \mathbf{u}_{22}^T(n) & \cdots & \mathbf{u}_{N2}^T(n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_{1N}^T(n) & \mathbf{u}_{2N}^T(n)0 & \cdots & \mathbf{u}_{NN}^T(n) \end{bmatrix} \right\} \\
&= E\left\{ \begin{bmatrix} \mathbf{u}_{11}(n)\mathbf{u}_{11}^T(n) & \mathbf{u}_{11}(n)\mathbf{u}_{21}^T(n) & \cdots & \mathbf{u}_{11}(n)\mathbf{u}_{N1}^T(n) \\ \mathbf{u}_{22}(n)\mathbf{u}_{12}^T(n) & \mathbf{u}_{22}(n)\mathbf{u}_{22}^T(n) & \cdots & \mathbf{u}_{22}(n)\mathbf{u}_{N2}^T(n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_{NN}(n)\mathbf{u}_{1N}^T(n) & \mathbf{u}_{NN}(n)\mathbf{u}_{2N}^T(n) & \cdots & \mathbf{u}_{NN}(n)\mathbf{u}_{NN}^T(n) \end{bmatrix} \right\}
\end{aligned} \tag{4.20}$$

Therefore, $\mathbf{R}$ is composed of $[N\times N]$ submatrices $\mathbf{R}_{jk}$ of size $[L\times L]$ defined as

$$\mathbf{R}_{jk} = E\{\mathbf{u}_{jj}(n)\mathbf{u}_{kj}^T(n)\}. \tag{4.21}$$

If we rewrite $\mathbf{u}_{jk}(n) = [u_{jk}(n)\ u_{jk}(n-1)\ \ldots\ u_{jk}(n-L+1)\ ]^T$, being $u_{jk}(n)=\sum_{m=0}^{M-1} h_{jk}(m)\ x(n-m)$, note that the $R_{jk}(l)$ element used in the matrix defined in (4.19), can be expressed as

$$\begin{aligned}
R_{jk}(l) &= E\{u_{jj}(n+l)u_{kj}(n)\} \\
&= E\{u_{jj}(n+l)\sum_{m=0}^{M-1} h_{kj}(m)\ x(n-m)\} \\
&= \sum_{m=0}^{M-1} h_{kj}(m)E\{u_{jj}(n+l)\ x(n-m)\} \\
&= \sum_{m=0}^{M-1} h_{kj}(m)R_{u_{jj}x}(l+m) = R_{u_{jj}x}(l) * \hat{\mathbf{h}}_{kj}(l)
\end{aligned} \tag{4.22}$$

where $\hat{\mathbf{h}}_{kj}(l)$=$\mathbf{h}_{kj}(-l)$ and

$$\begin{aligned} R_{u_{jj}x}(l) &= E\{u_{jj}(n+l)\ x(n)\} \\ &= E\{\sum_{m=0}^{M-1} h_{jj}(m)\ x(n+l-m)x(n)\} \\ &= \sum_{m=0}^{M-1} h_{jj}(m)E\{x(n+l-m)x(n)\} \\ &= \sum_{m=0}^{M-1} h_{jj}(m)R_{xx}(l-m) = R_{xx}(l) * \mathbf{h}_{jj}(l), \end{aligned} \tag{4.23}$$

we obtain that,

$$R_{jk}(l) = R_{xx}(l) * \mathbf{h}_{jj}(l) * \hat{\mathbf{h}}_{kj}(l) \tag{4.24}$$

Note that (4.17) describes the evolutionary behavior of the mean values of the error in the filter coefficients $\mathbf{w}(n)$. On the other hand, $\mathbf{R}$ can be described by its eigenvalue decomposition as

$$\mathbf{RQ} = \mathbf{Q\Lambda}, \tag{4.25}$$

being $\mathbf{Q}$ the $[LN\times LN]$ eigenvectors matrix of $\mathbf{R}$ and $\mathbf{\Lambda}$ the diagonal matrix containing the LN eigenvalues (λ_p) of $\mathbf{R}$. Substituting (4.25) in (4.17) and noting that $I_{LN} = \mathbf{QQ}^{-1}$, we obtain

$$\mathbf{v}(n+1) = \mathbf{Q}(I_{LN} - 2\mu\mathbf{\Lambda})\mathbf{Q}^{-1}\mathbf{v}(n). \tag{4.26}$$

Defining the rotated error vector as $\mathbf{v}^{'}(n)$=$\mathbf{Q}^{-1}\mathbf{v}(n)$ and multiplying both sides of (4.26) by $\mathbf{Q}^{-1}$, we get

$$\mathbf{v}^{'}(n+1) = (I_{LN} - 2\mu\mathbf{\Lambda})\mathbf{v}^{'}(n). \tag{4.27}$$

This expression shows that the algorithm converges in a series of independent modes and, therefore, (4.27) can be expressed as $[LN\times LN]$ recursive equations

$$v_p^{'}(n+1) = (1-2\mu\lambda_p)v_p^{'}(n) = (1-2\mu\lambda_p)^{n+1}\ v_p^{'}(0), \tag{4.28}$$

for $p = 1, 2, \ldots, LN$. This implies that $\mathbf{w}(n)$ converges to $\mathbf{w}_o$ if every step-size μ is selected so that

$$|1 - 2\mu\lambda_p| < 1, \tag{4.29}$$

that it may be expanded as

$$(1 - 2\mu\,\mathbb{R}\mathrm{e}\{\lambda_p\})^2 + (2\mu\,\mathbb{I}\mathrm{m}\{\lambda_p\})^2 < 1, \tag{4.30}$$

Note that, if any of the real parts of the eigenvalues of $\mathbf{R}$ are negative, (4.30) is not fulfilled and therefore, the system will not converge. On the contrary, the system will converge if $\mathbb{R}\mathrm{e}\{\lambda_p\}>0$ and the step-size fulfills that

$$\mu < \frac{1}{\mathbb{R}\mathrm{e}\{\lambda_p\}},\ \mu < \frac{1}{2|\mathbb{I}\mathrm{m}\{\lambda_p\}|},\ \mu < \frac{\mathbb{R}\mathrm{e}\{\lambda_p\}}{|\lambda_p|^2} \tag{4.31}$$

It should be noted that, these conditions depends, on the one hand, on the setting parameter of the algorithm (μ) and, on the other hand, on the statistics of the reference signals and the acoustic paths, represented both by the eigenvalues of the matrix $\mathbf{R}$. Note that, assuming a random noise as reference signal, $R_{xx}(l)=\delta(l)$ and thus, $R_{jk}(l)=\mathbf{h}_{jj}*\hat{\mathbf{h}}_{kj}$. In these cases, the use of statistically characterized reference signals implies that $\mathbf{R}$ is only dependent on the features of the acoustic system. However, the condition $\mathbb{R}\mathrm{e}\{\lambda_p\}<0$ implies that the system will not converge, independently of (4.31).

4.3.2 Interference model

As previously commented, certain levels of acoustical interaction among nodes may worsen the convergence performance of the decentralized network, even causing the unstability of the system. Therefore, an interference model based on the ratio between the degradation suffered by a node with and without the influence of other nodes of a decentralized network is required. However, it is important to note that, in order to obtain a realistic interference model, a considerable number of performances would be required due to certain uncertainties inherent in sound field control [51]. For this reason, a theoretical model of the interference level by analysing its statistical behaviour (on mean) is proposed below.

To this end, as described in (3.13), $\mathbf{r}_{jk}$ is a vector of size $[L\times 1]$ defined as follows

$$\mathbf{r}_{jk} = E\{\mathbf{u}_{jk}(n)d_k(n)\} = \begin{bmatrix} r_{jk}(0) \\ r_{jk}(1) \\ \vdots \\ r_{jk}(L-1) \end{bmatrix}, \tag{4.32}$$

where

$$\begin{aligned} r_{jk}(l) &= E\{u_{jk}(n+l)d_k(n)\} \\ &= E\{u_{jk}(n+l)\sum_{m=0}^{M-1} p_k(m)\, x(n-m)\} \\ &= \sum_{m=0}^{M-1} p_k(m)E\{u_{jk}(n+l)\, x(n-m)\} \\ &= \sum_{m=0}^{M-1} p_k(m)R_{u_{jk}x}(l+m) = R_{u_{jk}x}(l) * \hat{\mathbf{p}}_k(l) \end{aligned} \tag{4.33}$$

where $\hat{\mathbf{p}}_k(l)$=$\mathbf{p}_k(-l)$ being $\hat{\mathbf{p}}_k$=$[p_{k,M}\ \ldots\ p_{k,2}\ p_{k,1}\ ]^T$ and

$$\begin{aligned} R_{u_{jk}x}(l) &= E\{u_{jk}(n+l)\, x(n)\} \\ &= E\{\sum_{m=0}^{M-1} h_{jk}(m)\, x(n+l-m)x(n)\} \\ &= \sum_{m=0}^{M-1} h_{jk}(m)E\{x(n+l-m)x(n)\} \\ &= \sum_{m=0}^{M-1} h_{jk}(m)R_{xx}(l-m) = R_{xx}(l) * \mathbf{h}_{jk}(l) \end{aligned} \tag{4.34}$$

Therefore,

$$r_{jk}(l) = R_{xx}(l) * \mathbf{h}_{jk}(l) * \hat{\mathbf{p}}_k(l) \tag{4.35}$$

When a node is working isolated, the cost function to be minimized, $J_k(n)_{node}$, is defined in (3.12) with its optimal solution obtained in (3.13). On the contrary, if we consider a N-single-channel node network where all nodes work at the same time, now the MSE criterion of the node k is given by

$$J_k(n)_{network} = E\{|d_k(n)\ +\ \sum_{j=1}^{N} \mathbf{u}_{jk}^T(n)\mathbf{w}_j(n)|^2\}. \tag{4.36}$$

Both (3.12) and (4.36) can be written in terms of the statistics. In the first case,

$J_k(n)_{node}$ is given as

$$\begin{aligned} J_k(n)_{node} &= E\{|d_k(n) + \mathbf{u}_{kk}^T(n)\mathbf{w}_k(n)|^2\} \\ &= E\{d_k^2(n)\} + E\{\mathbf{w}_k^T(n)\mathbf{u}_{kk}(n)\mathbf{u}_{kk}^T(n)\mathbf{w}_k(n)\} \\ &\quad + 2E\{\mathbf{w}_k^T(n)\mathbf{u}_{kk}(n)\}d_k(n)\} \\ &= E\{d_k^2(n)\} + \mathbf{w}_k^T(n)E\{\mathbf{u}_{kk}(n)\mathbf{u}_{kk}^T(n)\}\mathbf{w}_k(n) \\ &\quad + 2\mathbf{w}_k^T(n)E\{\mathbf{u}_{kk}(n)\}d_k(n)\} \\ &= E\{d_k^2(n)\} + \mathbf{w}_k^T(n)\mathbf{R}_{kk}\mathbf{w}_k(n) + 2\mathbf{w}_k^T(n)\mathbf{r}_{kk} \end{aligned} \tag{4.37}$$

Note that the first term of the previous equation can be expressed as

$$\begin{aligned} E\{d_k^2(n)\} &= E\{\mathbf{p}_k^T\mathbf{X}^T(n)_{(1,:)}\mathbf{X}(n)_{(1,:)}\mathbf{p}_k\} \\ &= \mathbf{p}_k^T E\{\mathbf{X}^T(n)_{(1,:)}\mathbf{X}(n)_{(1,:)}\}\mathbf{p}_k \\ &= \mathbf{p}_k^T\mathbf{R}_{xx}\mathbf{p}_k \end{aligned} \tag{4.38}$$

where

$$\begin{aligned} \mathbf{R}_{xx} &= E\Bigg\{ \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-M+1) \end{bmatrix} \\ &\qquad \begin{bmatrix} x(n) & x(n-1) & \cdots & x(n-M+1) \end{bmatrix} \Bigg\} \\ &= \begin{bmatrix} R_{xx}(0) & R_{xx}(1) & \cdots & R_{xx}(M-1) \\ R_{xx}(1) & R_{xx}(0) & \cdots & R_{xx}(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_{xx}(M-1) & R_{xx}(M-2) & \cdots & R_{xx}(0) \end{bmatrix} \end{aligned} \tag{4.39}$$

Then, the cost function, $J_k(n)_{node}$, can be defined as

$$\begin{aligned} J_k(n)_{node} &= \mathbf{p}_k^T\mathbf{R}_{xx}\mathbf{p}_k + \mathbf{w}_k^T(n)\mathbf{R}_{kk}\mathbf{w}_k(n) \\ &\quad + 2\mathbf{w}_k^T(n)\mathbf{r}_{kk} \\ &= \mathbf{p}_k^T\mathbf{R}_{xx}\mathbf{p}_k + \mathbf{w}_k^T(n)(\mathbf{R}_{kk}\mathbf{w}_k(n) + 2\mathbf{r}_{kk}) \end{aligned} \tag{4.40}$$

On the other hand, (4.36) can be expressed as

$$\begin{aligned} J_k(n)_{network} &= E\{|d_k(n) + \sum_{j=1}^{N} \mathbf{u}_{jk}^T(n)\mathbf{w}_j(n)|^2\} \\ &= E\{d_k^2(n)\} \\ &\quad + \sum_{j=1}^{N} \mathbf{w}_j^T(n)E\{\mathbf{u}_{jk}(n)\mathbf{u}_{jk}^T(n)\}\mathbf{w}_j(n) \\ &\quad + 2\sum_{j=1}^{N} \mathbf{w}_j^T(n)E\{\mathbf{u}_{jk}(n)\}d_k(n)\} \\ &= \mathbf{p}_k^T\mathbf{R}_{xx}\mathbf{p}_k + \sum_{j=1}^{N} \mathbf{w}_j^T(n)\mathbf{R}_{\mathbf{u}_{jk}\mathbf{u}_{jk}}\mathbf{w}_j(n) \\ &\quad + 2\sum_{j=1}^{N} \mathbf{w}_j^T(n)\mathbf{r}_{jk} \\ &= \mathbf{p}_k^T\mathbf{R}_{xx}\mathbf{p}_k \\ &\quad + \sum_{j=1}^{N} \mathbf{w}_j^T(n)(\mathbf{R}_{\mathbf{u}_{jk}\mathbf{u}_{jk}}\mathbf{w}_j(n) + 2\mathbf{r}_{jk}) \end{aligned} \tag{4.41}$$

where $\mathbf{R}_{\mathbf{u}_{jk}\mathbf{u}_{jk}}$ is a matrix of size $[L\times L]$ constructed in a similar manner as (4.19) and defined as

$$\mathbf{R}_{\mathbf{u}_{jk}\mathbf{u}_{jk}} = E\{\mathbf{u}_{jk}(n)\mathbf{u}_{jk}^T(n)\} \tag{4.42}$$

being its l-th element, $R_{u_{jk}u_{jk}}(l)=R_{xx}(l)*\mathbf{h}_{jk}(l)*\hat{\mathbf{h}}_{jk}(l)$ (obtained in a similar manner as described in (4.24) and (4.35)). In summary, $J_k(n)_{node}$ and $J_k(n)_{network}$ can be rewritten as

$$J_k(n)_{node} = \mathbf{p}_k^T\mathbf{R}_{xx}\mathbf{p}_k + \mathbf{w}_k^T(n)(\mathbf{R}_{kk}\mathbf{w}_k(n) + 2\mathbf{r}_{kk}), \tag{4.43}$$

$$\begin{aligned} J_k(n)_{network} &= \mathbf{p}_k^T\mathbf{R}_{xx}\mathbf{p}_k \\ &\quad + \sum_{j=1}^{N} \mathbf{w}_j^T(n)(\mathbf{R}_{\mathbf{u}_{jk}\mathbf{u}_{jk}}\mathbf{w}_j(n) + 2\mathbf{r}_{jk}). \end{aligned} \tag{4.44}$$

When convergence is achieved, $\mathbf{w}_k(n)=\mathbf{w}_k^o$ and considering (3.13), the minimum MSE of (4.43) is given by

$$\begin{aligned} (J_k)_{node}^{min} &= \mathbf{p}_k^T\mathbf{R}_{xx}\mathbf{p}_k \\ &\quad + \mathbf{r}_{kk}^T(\mathbf{R}_{kk}^T)^{-1}(\mathbf{R}_{kk}(\mathbf{R}_{kk})^{-1}\mathbf{r}_{kk} - 2\mathbf{r}_{kk}) \\ &= \mathbf{p}_k^T\mathbf{R}_{xx}\mathbf{p}_k - \mathbf{r}_{kk}^T(\mathbf{R}_{kk}^T)^{-1}\mathbf{r}_{kk} \end{aligned} \tag{4.45}$$

Similarly, the local minimum solution of (4.44) is given by

$$\begin{aligned}(J_k)_{network}^{min} =& \mathbf{p}_k^T \mathbf{R}_{xx} \mathbf{p}_k \\ &+ \sum_{j=1}^{N} \mathbf{r}_{jj}^T (\mathbf{R}_{jj}^T)^{-1} (\mathbf{R}_{\mathbf{u}_{jk}\mathbf{u}_{jk}} (\mathbf{R}_{jj})^{-1} \mathbf{r}_{jj} - 2\mathbf{r}_{jk}) \\ =& \mathbf{p}_k^T \mathbf{R}_{xx} \mathbf{p}_k \\ &+ \mathbf{r}_{kk}^T (\mathbf{R}_{kk}^T)^{-1} (\mathbf{R}_{kk} (\mathbf{R}_{kk})^{-1} \mathbf{r}_{kk} - 2\mathbf{r}_{kk}) \\ &+ \sum_{\substack{j=1 \\ j \neq k}}^{N} \mathbf{r}_{jj}^T (\mathbf{R}_{jj}^T)^{-1} (\mathbf{R}_{\mathbf{u}_{jk}\mathbf{u}_{jk}} (\mathbf{R}_{jj})^{-1} \mathbf{r}_{jj} - 2\mathbf{r}_{jk}) \\ =& \mathbf{p}_k^T \mathbf{R}_{xx} \mathbf{p}_k - \mathbf{r}_{kk}^T (\mathbf{R}_{kk}^T)^{-1} \mathbf{r}_{kk} \\ &+ \sum_{\substack{j=1 \\ j \neq k}}^{N} \mathbf{r}_{jj}^T (\mathbf{R}_{jj}^T)^{-1} (\mathbf{R}_{\mathbf{u}_{jk}\mathbf{u}_{jk}} (\mathbf{R}_{jj})^{-1} \mathbf{r}_{jj} - 2\mathbf{r}_{jk}) \end{aligned} \tag{4.46}$$

Considering a N-nodes non-collaborative network where convergence conditions are assumed to be fulfilled, note that, (4.46) is fulfilled if every node reaches the same least squares optimal solution as if it were working in isolation. The interference ratio (IR_k) or how much the system has been degraded when a node k is working simultaneously with respect to work isolated, may be calculated as

$$IR_k = 10 \cdot \log_{10} \left[\frac{(J_k)_{network}^{min}}{(J_k)_{node}^{min}} \right] \tag{4.47}$$

Substituting (4.45) and (4.46) in (4.47), we obtain that

$$IR_k = 10 \cdot \log_{10} \left[1 + \frac{\sum_{\substack{j=1 \\ j \neq k}}^{N} \mathbf{r}_{jj}^T (\mathbf{R}_{jj}^T)^{-1} (\mathbf{R}_{\mathbf{u}_{jk}\mathbf{u}_{jk}} (\mathbf{R}_{jj})^{-1} \mathbf{r}_{jj} - 2\mathbf{r}_{jk})}{\mathbf{p}_k^T \mathbf{R}_{xx} \mathbf{p}_k - \mathbf{r}_{kk}^T (\mathbf{R}_{kk}^T)^{-1} \mathbf{r}_{kk}} \right] \tag{4.48}$$

If we define

$$\begin{aligned} \mathbf{A}_j &= \mathbf{r}_{jj}^T (\mathbf{R}_{jj}^T)^{-1} \mathbf{R}_{\mathbf{u}_{jk}\mathbf{u}_{jk}} (\mathbf{R}_{jj})^{-1} \mathbf{r}_{jj} \\ \mathbf{B}_j &= \mathbf{r}_{jj}^T (\mathbf{R}_{jj}^T)^{-1} \mathbf{r}_{jk} \\ \mathbf{C}_k &= \mathbf{r}_{kk}^T (\mathbf{R}_{kk}^T)^{-1} \mathbf{r}_{kk} \end{aligned} \tag{4.49}$$

(4.48) can be rewritten as

$$IR_k = 10 \cdot \log_{10} \left[1 + \frac{\sum_{\substack{j=1 \\ j \neq k}}^{N} (\mathbf{A}_j - 2\mathbf{B}_j)}{\mathbf{p}_k^T \mathbf{R}_{xx} \mathbf{p}_k - \mathbf{C}_k} \right]. \tag{4.50}$$

4.4 Decentralized FxLMS algorithm using control effort

Since acoustic nodes are equipped with power constrained actuators, these constraints need to be considered in the cost function of the adaptive algorithm at each node. Otherwise, control signals may increase unlimitedly, causing system instability [112]. In the proposed generic cost function (3.22) presented in 3.2.1, this control effort weighting is represented by the entries of matrix $\boldsymbol{\beta}$ where β_k is the leakage coefficient used to adjust the amplitude of the adaptive filter coefficients of the node k.

In this case, the matrix $\boldsymbol{\beta}$ is similar as defined in (3.38). Therefore, the cost function (3.22) is redefined as

$$J(n) = (\mathbf{d}(n) + \tilde{\mathbf{U}}^T(n)\mathbf{w}(n))^T(\mathbf{d}(n) + \tilde{\mathbf{U}}^T(n)\mathbf{w}(n)) \; + \; \mathbf{w}^T(n)\,\boldsymbol{\beta}\,\mathbf{w}(n). \quad (4.51)$$

and consequently, the global update solution for a decentralized ANC system using control effort as

$$\mathbf{w}(n+1) = (I_{LN} - 2\boldsymbol{\beta}\mu)\mathbf{w}(n) \; - 2\mu\tilde{\mathbf{U}}(n)\mathbf{e}(n) \quad (4.52)$$

Therefore, the filter updating equation of the NC-DFxLMS algorithm using control effort (*leaky or l-NC-DFxLMS*) at node k is given by

$$\mathbf{w}_k(n+1) = (1 - 2\beta_k\mu)\mathbf{w}_k(n) \; - 2\mu\,\mathbf{u}_{kk}(n)\,e_k(n)\cdot \quad (4.53)$$

Note that, for β_k=0, the updating filter equation of the leaky NC-DFxLMS algorithm (4.53) is equal to (4.10).

Different methods to apply constraints by limiting the output power at each node for a single-channel ANC system are described in [104]. However, the analysis of the performance of these constrained methods over a decentralized network has not been already reported. In addition, the effect of these methods on the behavior of a distributed ANC system over a network with distributed and collaborative nodes will be analyzed in Section 5.3.

Clipping method just describes a real situation in practical scenarios where output amplifier or loudspeaker saturation appears. If the output signal power is greater than an upper threshold, a simple solution is to limit the output signal power to the threshold value. Defining $y_{k_{max}}$ as the maximum allowed value of the output signal at each node k, the clipping leaky NC-DFxLMS (*clipping l-NC-DFxLMS*) algorithm is given by

$$\begin{aligned} &\text{if } |y_k(n)| > y_{k_{max}}, \\ &\qquad y_k(n) = y_k(n)\xi_k\tfrac{y_{k_{max}}}{|y_k(n)|}, \quad (0 < \xi_k \leq 1)\cdot \end{aligned} \quad (4.54)$$

Depending on the value of the parameter ξ_k, a saturation (ξ_k=1) or compression ($0 < \xi_k$<1) effect on the dynamic range of the ouput may be applied at a certain threshold $y_{k_{max}}$. To avoid large oscillations in the coefficients' update, the re-scaling algorithm can be used. To this end, it is only necessary to add to (4.54) the re-scaling of the adaptive filters as follows

$$\begin{aligned} &\text{if } |y_k(n)| > y_{k_{max}}, \\ &\quad y_k(n) = y_k(n)\xi_k \tfrac{y_{k_{max}}}{|y_k(n)|}, \\ &\quad \mathbf{w}_k(n) = \mathbf{w}_k(n)\ \xi_k \tfrac{y_{k_{max}}}{|y_k(n)|}. \end{aligned} \qquad (0 < \xi_k < 1) \tag{4.55}$$

Note that, while the clipping l-NC-DFxLMS algorithm just re-scales the output, the re-scaling leaky NC-DFxLMS (*re-scaling l-NC-DFxLMS*) algorithm, described in (4.55), re-scales both the output and the filter coefficients. This allows to avoid potential stability problems due to the coefficients update is uncorrelated with the filter output when the clipping strategy is working [104] [113]. As previously commented, applying the suitable constraints over the output signal, the stability of the system will be ensured but, to be too restrictive in saturation, can result in low performance attenuation.

4.4.1 Convergence analysis

As previously commented, a control effort weighting may be introduced in the cost function of the distributed algorithm at each node to stabilize the decentralized ANC system. Similarly as the procedure described in 4.3.1, we aim to know the necessary condition to achieve the system stability. For this analysis, we assume the same assumptions as described in (4.14). Considering that all nodes use the same leakage parameter (denoted as β) and substituting (4.12) in (4.52), now the global update solution of the decentralized network using control effort can be rewritten as

$$\mathbf{w}(n+1) = (1 - 2\beta\mu)\mathbf{w}(n)\ - 2\mu\tilde{\mathbf{U}}(n)\mathbf{d}(n) - 2\mu\tilde{\mathbf{U}}(n)\mathbf{U}^T(n)\mathbf{w}(n) \tag{4.56}$$

Assuming that the network achieves the convergence, we can obtain

$$\mathbf{w}_o = -(\tilde{\mathbf{U}}(n)\mathbf{U}^T(n) + \beta I)^{-1}\tilde{\mathbf{U}}(n)\mathbf{d}(n) \tag{4.57}$$

and, hence, (4.17) is modified as

$$\begin{aligned} \mathbf{v}(n+1) &= (I_{LN} - 2\mu(\mathbf{R} + \beta I_{LN}))\mathbf{v}(n) \\ &= (I_{LN}(1 - 2\mu\beta) - 2\mu\mathbf{R}))\mathbf{v}(n). \end{aligned} \tag{4.58}$$

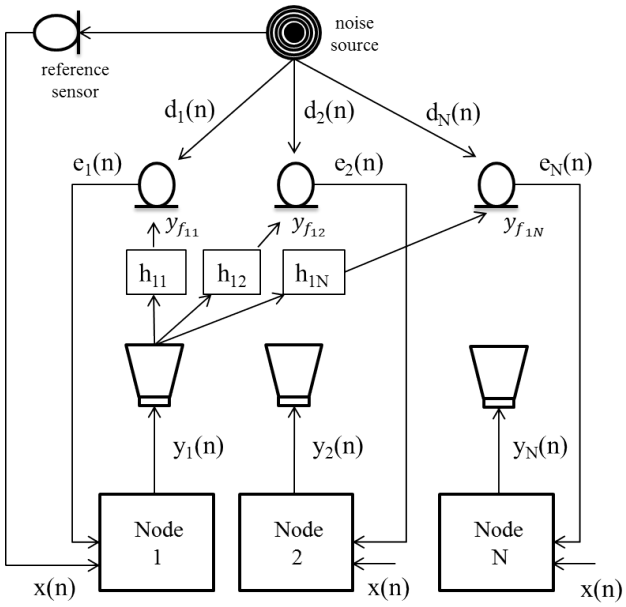


Figure 4.2: Decentralized ANC system working over an ASN of N single-channel nodes using an intereference control strategy.

This implies that $\mathbf{w}(n)$ converges to $\mathbf{w}_o$ if every step-size μ is selected so that

$$|1 - 2\mu(\beta + \lambda_p)| < 1, \tag{4.59}$$

i.e.

$$\mu < \frac{1}{\beta + \mathbb{R}\mathrm{e}\{\lambda_p\}},\ \mu < \frac{1}{2|\mathbb{I}\mathrm{m}\{\lambda_p\}|},\ \mu < \frac{\beta + \mathbb{R}\mathrm{e}\{\lambda_p\}}{|\lambda_p|^2 + \beta^2 + 2\beta\,\mathbb{R}\mathrm{e}\{\lambda_p\}}. \tag{4.60}$$

Therefore, if the real part of any eigenvalue of matrix $\mathbf{R}$ is negative, i.e. $\mathbb{R}\mathrm{e}\{\lambda_p\}<0$, the unstability of the decentralized ANC system can be avoided by using a proper value of β (as well as a proper step-size parameter μ). However, values of $\beta>0$ imply the performance degradation of the ANC system in order to guarantee its convergence [43].

4.5 Decentralized FxLMS algorithm using interference control

As commented in 3.1, the level of acoustical interaction is usually related to the distance between nodes. The lower distance, the higher level of coupling. In Figure

4.1, the signals $y_{f_{jk}}(n)$ (for $j{\neq}k$) represent the acoustic interferences that suffer the sensor of the node k and which may degrade its performance. The straightforward strategy may be to eliminate these interference signals from the information captured by each sensor. For example, the cost function to minimize at node k could be expressed as

$$J_k(n) = e_k^2(n) + \sum_{\substack{j=1 \\ j\neq k}}^{N} y_{f_{jk}}^2(n), \tag{4.61}$$

It can easily be realized that, since the information of the rest of the nodes is not available at each node, the performing of this method is not possible in a non-collaborative network. However, a possible solution to avoid the acoustical interaction of the decentralized networks, it is to minimize the effect produced by each node in the rest of the nodes by introducing the interferences among them in the cost function as

$$J_k(n) = e_k^2(n) + \sum_{\substack{j=1 \\ j\neq k}}^{N} y_{f_{kj}}^2(n), \tag{4.62}$$

being $y_{f_{kj}}(n)$ the interference signal that the node k-th estimates that it is arriving at the sensor of node j-th due to the control signal generated by its actuator, $y_k(n)$. Note that $y_{f_{kj}}(n)$ is calculated as,

$$y_{f_{kj}}(n) = \mathbf{y}_{Mk}^T(n)\mathbf{h}_{kj} = \mathbf{w}_k^T(n)\mathbf{x}(n)\mathbf{h}_{kj} = \mathbf{w}_k^T(n)\mathbf{u}_{kj}(n), \tag{4.63}$$

where $\mathbf{y}_{Mk}^T(n)$ is a M-lenght vector that contains the last M samples of output signal $y_k(n)$. The scheme of this strategy is summarized in Figure 4.2.

It should be noted that (4.62) can be derived from (3.22) by selecting the following parameters configuration. Consider $Q{=}1$, $\alpha_{kk}{=}1$, $\mathbf{S}(n)$ as an identity matrix as defined in (4.6) and $\boldsymbol{\beta}$ as a null matrix. Since interference signals must be now considered, then $\epsilon{=}1$. Therefore, in (3.39) consider that $I{=}N$, $\boldsymbol{\Gamma}_i$ is only composed of $\boldsymbol{\gamma}_{ii}$ at the $(1{+}L(i{-}1){:}iL, 1{+}L(i{-}1){:}iL)$ position, defined as an identity matrix, such as

$$\boldsymbol{\Gamma}_i{=}\begin{bmatrix} \mathbf{0}_L & \cdots & \mathbf{0}_L & \cdots & \mathbf{0}_L \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_L & \cdots & \boldsymbol{\gamma}_{ii} & \cdots & \mathbf{0}_L \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_L & \cdots & \mathbf{0}_L & \cdots & \mathbf{0}_L \end{bmatrix}, \qquad \boldsymbol{\gamma}_{ii}{=}\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}. \tag{4.64}$$

and $\boldsymbol{\rho}_i$ is defined as $\boldsymbol{\rho}_i = [0 \ \ldots \ \rho_i \ \ldots \ 0]$ where ρ_i is equal to 1 at position i of the vector. This causes that the cost function (3.22) is now given by

$$J(n) = (\mathbf{d}(n) + \tilde{\mathbf{U}}^T(n)\mathbf{w}(n))^T(\mathbf{d}(n) + \tilde{\mathbf{U}}^T(n)\mathbf{w}(n)) \ + \ \mathbf{w}^T(n)\,\tilde{\mathbf{w}}(n)\,\bar{\mathbf{u}}(n), \tag{4.65}$$

where the $LN \times N$ matrix $\tilde{\mathbf{w}}(n)$ is given by

$$\tilde{\mathbf{w}}(n)= \begin{bmatrix} \mathbf{w}_1(n) & \mathbf{0}_{L\times 1} & \cdots & \mathbf{0}_{L\times 1} \\ \mathbf{0}_{L\times 1} & \mathbf{w}_2(n) & \cdots & \mathbf{0}_{L\times 1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{L\times 1} & \mathbf{0}_{L\times 1} & \cdots & \mathbf{w}_N(n) \end{bmatrix}, \tag{4.66}$$

From (4.65) it is possible to obtain the global update solution for a decentralized ANC system considering interference control as

$$\mathbf{w}(n+1) = \mathbf{w}(n) \ - 2\mu(\tilde{\mathbf{U}}(n)\mathbf{e}(n) + \tilde{\mathbf{w}}(n)\ \bar{\mathbf{u}}(n)). \tag{4.67}$$

Splitting up from (4.67) the contribution of each node, the filter updating equation of the NC-DFxLMS algorithm using interference control (*icNC-DFxLMS*) at node k is calculated as follows:

$$\mathbf{w}_k(n+1) = \mathbf{w}_k(n) \ - 2\mu(\mathbf{u}_{kk}(n)\ e_k(n) + \sum_{\substack{j=1 \\ j\neq k}}^{N} \mathbf{u}_{kj}(n) y_{f_{kj}}(n))\cdot \tag{4.68}$$

Note that previously it is necessary that each node has information about the estimated acoustic channels which link its actuator with each sensor of the rest of the nodes in order to carry out a proper filter updating. However and as commented in 2.4.1, in practical ANC implementations, the real acoustic responses between all the loudspeakers and all microphones are usually identified off-line in a previous stage.

4.5.1 Convergence analysis

Similarly as described in (4.3.1) and (4.4.1), the aim of this section is to analyse the mean weight behaviour of the adaptive coefficients at each node of the proposed icNC-DFxLMS algorithm. As previous cases, the same assumptions as depicted in (4.14) are assumed.

Substituting (4.12) in (4.67), the global update solution of the decentralized network using interference control can be expressed as

$$\mathbf{w}(n+1) = \mathbf{w}(n) \ - 2\mu(\tilde{\mathbf{U}}(n)\mathbf{d}(n) + \tilde{\mathbf{U}}(n)\mathbf{U}^T(n)\mathbf{w}(n) + \tilde{\mathbf{w}}(n)\ \bar{\mathbf{u}}(n)). \tag{4.69}$$

Note that, if $\mathbf{w}(n)$ converges to its optimal filters $\mathbf{w}_o$, it is fulfilled that

$$\tilde{\mathbf{U}}(n)\mathbf{d}(n) = \ -(\tilde{\mathbf{U}}(n)\mathbf{U}^T(n) + \bar{\mathbf{U}}(n))\mathbf{w}_o \tag{4.70}$$

where $\bar{\mathbf{U}}(n))=\sum_{i=1}^{I}\boldsymbol{\rho}_i\ \bar{\mathbf{u}}(n)\ \boldsymbol{\Gamma}_i$, i.e, a diagonal matrix of size $LN\times LN$ whose diagonal elements are the N values of $\bar{u}_k(n)$ replicated L times,

$$\bar{\mathbf{U}}(n)=\begin{bmatrix}\bar{\mathbf{U}}_1(n) & \mathbf{0}_L & \cdots & \mathbf{0}_L\\ \mathbf{0}_L & \bar{\mathbf{U}}_2(n) & \cdots & \mathbf{0}_L\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0}_L & \mathbf{0}_L & \cdots & \bar{\mathbf{U}}_N(n)\end{bmatrix},\bar{\mathbf{U}}_k(n)=\begin{bmatrix}\bar{u}_k(n) & 0 & \cdots & 0\\ 0 & \bar{u}_k(n) & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \bar{u}_k(n)\end{bmatrix}. \tag{4.71}$$

where $\bar{u}_k(n)=\bar{\mathbf{u}}_k^T(n)\bar{\mathbf{u}}_k(n)$. Under this steady-state condition, substracting $\mathbf{w}_o$ from both sides of (4.69), taking expectations of both sides and using (4.70), we obtain the evolutionary behavior of the mean values of the errors in the global adaptive filter as

$$\mathbf{v}(n+1)=(I_{LN}-2\mu(\mathbf{R}+\mathbf{R}_{\bar{\mathbf{U}}})\mathbf{v}(n). \tag{4.72}$$

where $\mathbf{R}_{\bar{\mathbf{U}}}=E\{\bar{\mathbf{U}}(n)\}$. Therefore, $\mathbf{w}(n)$ will converge to $\mathbf{w}_o$ if the following condition is fulfillfed

$$|1-2\mu\lambda_{p,U}|<1, \tag{4.73}$$

$$\mu<\frac{1}{\mathbb{R}\mathrm{e}\{\lambda_{pU}\}},\ \mu<\frac{1}{2|\mathbb{I}\mathrm{m}\{\lambda_{pU}\}|},\ \mu<\frac{\mathbb{R}\mathrm{e}\{\lambda_{pU}\}}{|\lambda_{pU}|^2} \tag{4.74}$$

where λ_{pU} is the p-th eigenvalue of the matrix $\mathbf{RU}=\mathbf{R}+\mathbf{R}_{\bar{\mathbf{U}}}$. Since the sum of eigenvalues of a matrix is equal to its trace, therefore, the following can be stated

$$\lambda_{p,U}\ \leq\sum_{p=1}^{LN}\lambda_{p,U}=\mathrm{Tr}\{\mathbf{R}+\mathbf{R}_{\bar{\mathbf{U}}}\}=E\Big\{\sum_{k=1}^{N}(\mathbf{u}_{kk}^T(n)\mathbf{u}_{kk}(n)+\bar{u}_k(n))\Big\}. \tag{4.75}$$

Substituting (4.75) in (4.74),

$$\begin{aligned}\mu&<\frac{1}{\mathrm{Re}\{E\Big\{\sum_{k=1}^{N}(\mathbf{u}_{kk}^T(n)\mathbf{u}_{kk}(n)+\bar{u}_k(n))\Big\}\}},\\ \mu&<\frac{1}{2|\mathbb{I}\mathrm{m}\{E\Big\{\sum_{k=1}^{N}(\mathbf{u}_{kk}^T(n)\mathbf{u}_{kk}(n)+\bar{u}_k(n))\Big\}\}|},\\ \mu&<\frac{\mathbb{R}\mathrm{e}\{E\Big\{\sum_{k=1}^{N}(\mathbf{u}_{kk}^T(n)\mathbf{u}_{kk}(n)+\bar{u}_k(n))\Big\}\}}{|E\Big\{\sum_{k=1}^{N}(\mathbf{u}_{kk}^T(n)\mathbf{u}_{kk}(n)+\bar{u}_k(n))\Big\}|^2}\end{aligned} \tag{4.76}$$

Note that the convergence condition presented in (4.74) can be viewed as to introduce a regularization factor to avoid the decentralized system unstability (similar conclusion as discussed in (4.60)) but now this factor depends on the sum of interefference signals which node k-th causes on the rest of the nodes. However, unlike in the control effort strategy, the value of λ_{pU} is determined by the energy of the interference signals generated by each node. The more the node interferes with the rest, the larger the value of this parameter. And therefore, although this may also cause higher performance degradation, the ANC system stability may be guaranteed.

4.6 Other decentralized strategies

Note that due to the design of the cost function presented in equation (3.22), other decentralized strategies can be derived from it. Although the behavior of these strategies has not been analyzed in this thesis, a brief summary is presented below.

Decentralized strategy using quality of service

Consider a similar paramater configuration as defined in Section 4.3, that is, Q=1, $\mathbf{S}(n)$ defined as (4.6) with $\boldsymbol{\beta}$ as a null matrix and ϵ=0. However, note that a simple way to obtain different performances at each node of the decentralized network may be achieved by considering $0<\alpha_{kk}<1$ in (4.5). Thus, matrix $[\hat{\mathbf{U}}]_c(n)$ can be defined as

$$[\hat{\mathbf{U}}]_c(n) = \begin{bmatrix} \alpha_{11}\mathbf{u}_{11}(n) & \mathbf{0}_{L\times 1} & \cdots & \mathbf{0}_{L\times 1} \\ \mathbf{0}_{L\times 1} & \alpha_{22}\mathbf{u}_{22}(n) & \cdots & \mathbf{0}_{L\times 1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{L\times 1} & \mathbf{0}_{L\times 1} & \cdots & \alpha_{NN}\mathbf{u}_{NN}(n) \end{bmatrix} \tag{4.77}$$

Note that, the greater the value of α_{kk}, the better the performance or quality of service (QoS) of node i (for i=1, 2, . . ., N). Therefore, in this case the global solution is similar to 4.9 but weighting the error signal as follows

$$\mathbf{w}(n+1) = \mathbf{w}(n) \ - 2\mu[\hat{\mathbf{U}}]_c(n)\mathbf{e}(n), \tag{4.78}$$

and consequently, the filter updating equation of the NC-DFxLMS strategy using quality of service (*QoS NC-DFxLMS*) for the k-th node is calculated as

$$\mathbf{w}_k(n+1) = \mathbf{w}_k(n) \ - 2\mu\alpha_{kk}\mathbf{u}_{kk}(n)e_k(n). \tag{4.79}$$

Decentralized strategy based on affine projection approach

The affine projection algorithms appear due to the need to improve the convergence behaviour in transient state of the LMS algorithms. Speed convergence improvement

depends on the increase of the projection order (Q). Therefore, in this case we assume that $Q>1$. Starting now from (4.5), consider that $\alpha_{kk}=1$. Since the affine projection approach is considered, we define $\mathbf{S}_{kk}=[\hat{\mathbf{U}}_k^T(n)\hat{\mathbf{U}}_k(n)+\delta\boldsymbol{I}_\phi]^{-1}$ being δ a regularization factor to avoid instability in matrix inversion and $\boldsymbol{I}_\phi$ an identity matrix of size $[Q\times Q]$. Therefore, consdering that $\epsilon=0$ and $\boldsymbol{\beta}$ is a null matrix, the cost function (3.22) is redefined as

$$J(n)=(\mathbf{d}(n)+[\hat{\mathbf{U}}^T]_c(n)\mathbf{w}(n))^T(\mathbf{d}(n)+[\hat{\mathbf{U}}^T]_c(n)\mathbf{w}(n)) \tag{4.80}$$

where $[\hat{\mathbf{U}}]_c(n)$ is arranged as

$$[\hat{\mathbf{U}}]_c(n)=\begin{bmatrix}\hat{\mathbf{U}}_{11}(n)\mathbf{S}_{11}(n) & \mathbf{0}_{L\times Q} & \cdots & \mathbf{0}_{L\times Q}\\ \mathbf{0}_{L\times Q} & \hat{\mathbf{U}}_{22}(n)\mathbf{S}_{22}(n) & \cdots & \mathbf{0}_{L\times Q}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0}_{L\times Q} & \mathbf{0}_{L\times Q} & \cdots & \hat{\mathbf{U}}_{NN}(n)\mathbf{S}_{NN}(n)\end{bmatrix} \tag{4.81}$$

Therefore, the global update solution is now calculated as

$$\mathbf{w}(n+1)=\mathbf{w}(n)\ -2\mu[\hat{\mathbf{U}}]_c(n)\hat{\mathbf{e}}(n), \tag{4.82}$$

and therefore, the filter updating equation of the decentralized filtered-x affine projection (*NC-DFxAP*) algorithm is calculated at the k-th node as follows:

$$\mathbf{w}_k(n+1)=\mathbf{w}_k(n)\ -2\mu\hat{\mathbf{U}}_k(n)[\hat{\mathbf{U}}_k^T(n)\hat{\mathbf{U}}_k(n)+\delta\boldsymbol{I}_\phi]^{-1}\ \hat{\mathbf{e}}_k(n)\cdot \tag{4.83}$$

4.7 Simulation results

In this section, we present the simulations carried out to evaluate the performance of the presented non-collaborative algorithms over several ideal ASNs composed of two, four and six single-channel nodes respectively, as depicted in Figure 4.3. To this end, the icNC-DFxLMS algorithm has been compared to the NC-DFxLMS, the l-NC-DFxLMS, the clipping l-NC-DFxLMS and the rescaling l-NC-DFxLMS algorithms in terms of noise reduction convergence speed and computational demands. Initially we have compared the performance of the algorithms in both uncoupled and coupled two-node ASNs with the aim to justify the use of constrained non-collaborative methods in networks composed of acoustically coupled single-channel nodes. Furthermore, in order to validate the behaviour of the decentralized algorithms in larger ASN, four-node and six-node networks composed of acoustically coupled single-channel nodes that suports a distributed ANC system are considered. For the designed ASNs, we use the real acoustic channels and the parameters configuration depicted in Section 3.3. The nodes selected for each presented ASN are: nodes 1 and 2 for

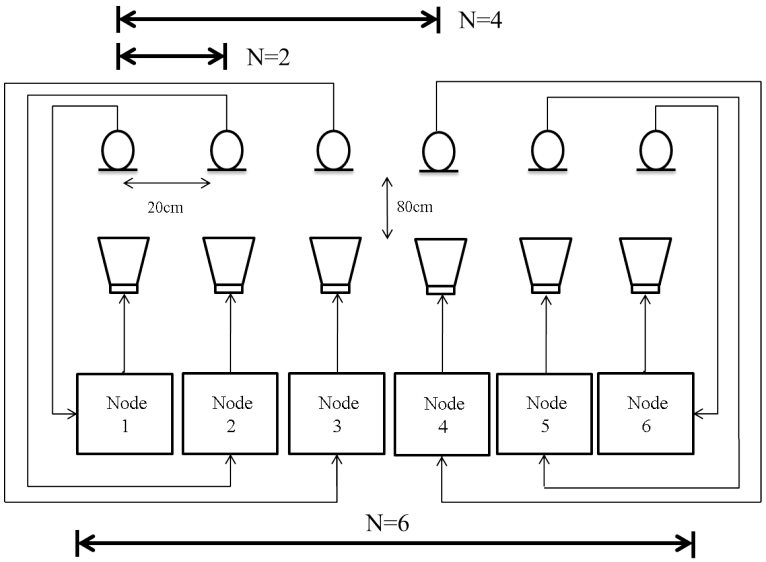
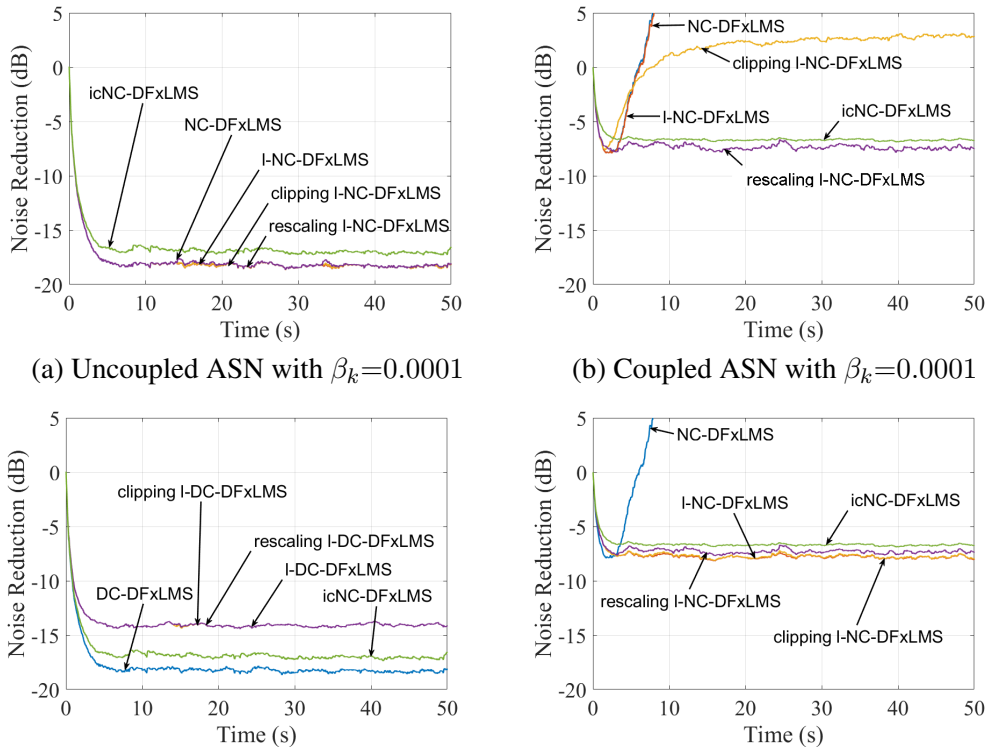


Figure 4.3: Distributed ASN two nodes, four nodes and six nodes for a coupled ANC system. Nodes selected for each ASN are indicated.

$N = 2$, nodes 1 to 4 for $N = 4$, and nodes 1 to 1 for $N = 6$. For all scenarios, fixed values of $y_{k_{max}}$=1.0 and ξ_k=1 are considered as the maximum allowed value of the output signals and the attenuation parameter, respectively. Initially, we use a leakage parameter of β_k=0.0001 for all the nodes. The reason of these values is explained in Section 5.3 where a more detailed analysis of the performance of the constrained techniques as well as the motivation for using them in distributed networks will be addressed.

In the first stage, we have evaluated the performance of NC-DFxLMS, the l-NC-DFxLMS, the clipping l-NC-DFxLMS, the rescaling l-NC-DFxLMS and the icNC-DFxLMS algorithms in both uncoupled and coupled networks composed of two single-channel nodes. Figure 4.4) illustrates the time evolution of the $NR(n)$ with a step-size parameter fixed at μ=0.05 for all the algorithms in both type of networks. As Figure 4.4.(a) shows, in the uncoupled network case, all the algorithms provide a stable behaviour achieving a noise reduction of 17.7 dB for the icNC-DFxLMS algorithm and 18.2 dB for all other algorithms. In this case, for the leaky algorithms, a very low beta value has been selected (β_k=0.0001) since a regularization parameter to stabilize the ANC system is not required. On the other hand, the NC-DFxLMS algorithm and its leaky version diverge in the coupled network case, as show in Figure 4.4.(a). In the same way, the clipping l-NC-DFxLMS algorithm starts to diverge and although over time, it stabilizes its behavior, it is not able to reduce the noise (even amplifies it). However, the rescaling l-NC-DFxLMS and the icNC-DFxLMS algorithms are stable providing a noise reduction around 7 dB. Therefore,



(a) Uncoupled ASN with β_k=0.0001

(b) Coupled ASN with β_k=0.0001

(c) Uncoupled ASN with β_k=0.01

(d) Coupled ASN with β_k=0.01

Figure 4.4: Noise reduction of the uncoupled and coupled two-node networks using the decentralized algorithms with different values of β_k for the leaky methods.

a higher value of β_k is required for the leaky methods in order to ensure the system stability. For that reason, in Figures 4.4.(c) and 4.4.(d), the same ASNs are considered but using β_k=0.01. Now, although the noise reduction of the leaky algorithms in Figure 4.4.(c) is lower in comparison to Figure 4.4.(a) (almost 5 dB less), they presents a robust behavior in terms of stability in the case of coupled network providing an attenuation close to 8 dB (see Figure 4.4.(d)). The NC-DFxLMS and the icNC-DFxLMS algortihms present the same behaviour as the previous simulation.

However, it should be noted that, as it can be seen in Figure 4.5.(a), the l-NC-DFxLMS algorithm exceeds the output threshold which can lead to instability. On the contrary, the icNC-DFxLMS algorithm along with clipping and rescaling l-NC-DFxLMS algorithms satisfy the constraint since their output signals are under $y_{k_{max}}$, as shown in Figures 4.5(b), 4.5(c) and 4.5(d).

In the second stage, Figure 4.6.(a) illustrates the time evolution of the $NR(n)$

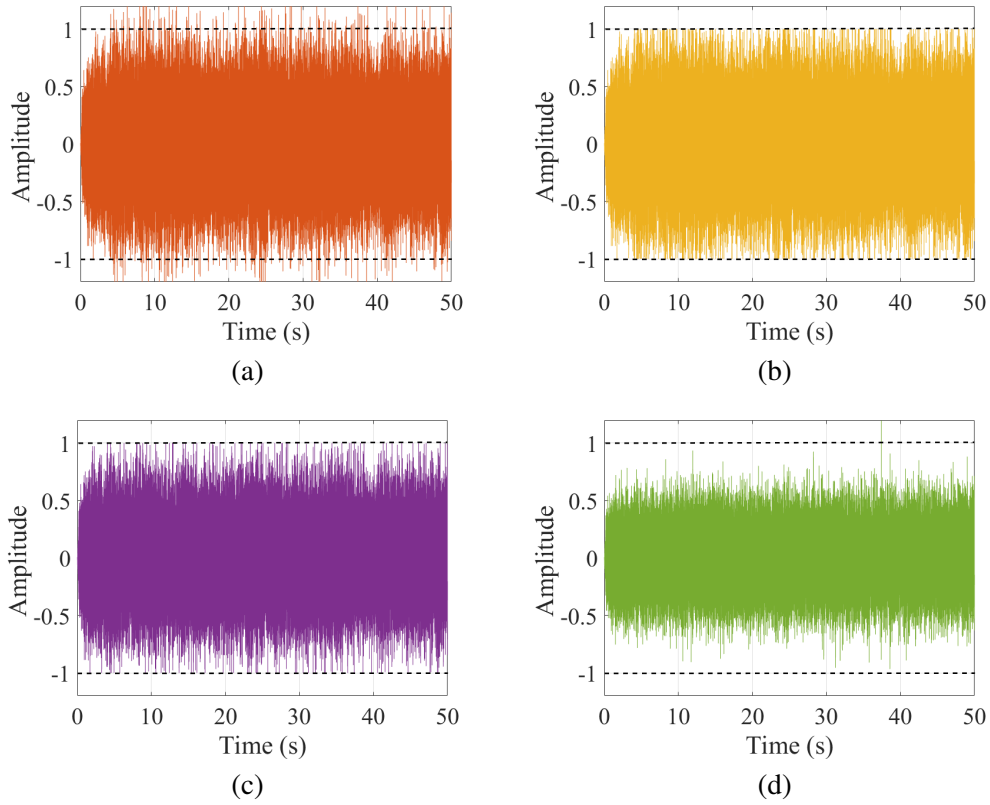


Figure 4.5: Time evolution of the control signals for the (a) l-NC-DFxLMS algorithm, (b) clipping l-NC-DFxLMS algorithm, (c) rescaling l-NC-DFxLMS algorithm and (d) icNC-DFxLMS algorithm at the best node of a coupled ASN with $\beta_k = 0.01$. The threshold is represented by dashed lines.

of a distributed system with four single-channel nodes using the fully decentralized approach, its leaky variants and the proposed decentralized version using the interference control. The step-size parameter was set to μ=0.025 for all the algorithms and the leakage parameter β_k must be increased to β_k=0.05 in order to ensure the stability of the leaky algorithms. The attenuation achieved for all the constrained distributed algorithms in this case is lower than the previous one depicted in Figure 4.4.(d) (4 dB less) but their performance is similar to the previous scenario. Similarly, the NC-DFxLMS algorithm starts cancelling the noise but at a certain point it turns unstable and does not converge.

Finally, we have evaluated the performance of the decentralized algorithms in a six-node ASNs. The $NR(n)$ curves obtained for all the algorithms are shown in

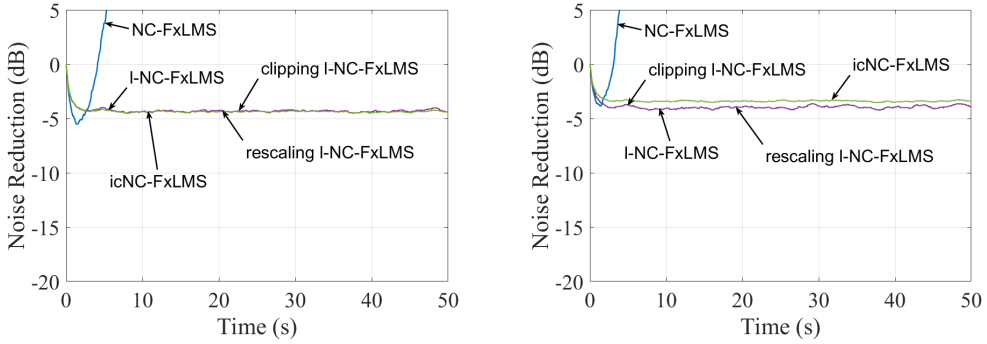


Figure 4.6: Noise reduction of the (a) four-node and (b) six-node coupled networks using the decentralized algorithms.

Table 4.1: Total number of multiplications (MUX) per iteration regarding the computational complexity of the algorithms. L: length of the adaptive filters; M: length of the estimated acoustic paths; N: number of nodes. As example, some typical cases considering L=150, M=256 and N=2, 4 and 6 nodes, have been evaluated.

	Algorithms	**Generic**	***N* = 2**	***N* = 4**	***N* = 6**
Computational complexity (MUX)	NC-DFxLMS	$2L+M+1$	557	557	557
	l-NC-DFxLMS	$3L+M+1$	707	707	707
	clipping l-NC-DFxLMS	$3L+M+2$	708	708	708
	re-scaling l-NC-DFxLMS	$4L+M+2$	858	858	858
	icNC-DFxLMS	$LN+M(2N-1)$	1068	2392	3716

Figure 4.6.(b). The step-size parameter was set to μ=0.016 for all the algorithms and β_k=0.05 for the leaky methods. As expected and similarly as the previous cases, the behavior of NC-DFxLMS algorithm makes the system not stable. However, the proposed decentralized strategies shows a stable performance despite providing an attenuation slightly less than 4 dB.

It is important to note that there exists remarkable aspects regarding the computational burden to be taken into account. Table 4.1 compares the computational complexity (in terms of multiplications per iteration) of the presented non-collaborative distributed ANC algorithms. Note that the computational complexity of all algorithms depends on L and M except for the icNC-DFxLMS algorithm which also depends on N. Results show that there are no significant differences between the leaky versions of the NC-DFxLMS algorithm indenpendently of the number of nodes. However, the icNC-DFxLMS algorithm requires initially higher computational re-

quirements which increases significantly with the number of nodes. Finally, note that all the results accomplished in this section depend on particular settings but their behavior can be easily extrapolated to other configurations.

4.8 Conclusions

In this chapter, several decentralized ANC strategies have been analyzed in order to be implemented in distributed networks. More specifically, the fully distributed algorithm (NC-DFxLMS) has been presented as a simple and efficient method to solve the ANC problem in ASN composed of uncoupled nodes. However, its performance in networks composed of acoustically coupled nodes may not be stable which may lead to system divergence. For that reason, new decentralized strategies, as the proposed constrained methods, may be considered with the aim to stabilize the distributed ANC system. The control effort version of the NC-DFxLMS (leaky NC-DFxLMS) algorithm arises from the idea of considering the effect of practical constraints in the ANC problem. In addition two approaches have been derived from this method in order to keep the output signal controlled denoted as clipping l-NC-DFxLMS and rescaling l-NC-DFxLMS algorithms. Moreover, in order to minimize the effects of the acoustic coupling, a decentralized strategy based on to control the interference signals produced among nodes has been introduced and denoted as icNC-DFxLMS algorithm.

For each decentralized strategy, the stability conditions of a distributed ANC system composed of non-collaborative acoustic nodes has been analyzed. These strategies have been derived from a proposed generic cost function by selecting a proper parameter configuration. Moreover and with the aim to estimate how much the performance of one node in a decentralized ANC system is degraded by the effect of the acoustical interference signals from the rest of nodes of the network, an acoustical interferences model has been presented.

Several simulations have been carried out over different ASNs in order to evaluate the performance of the decentralized algoritms. Results show that the icNC-DFxLMS algorithm achieves the system convergence in those cases where the fully decentralized algorithm diverges achieving a similar performance as the classic control effort strategies for ANC systems over coupled decentralized networks. In addition, by using the icNC-DFxLMS algorithm, it is possible to stabilize the decentralized system automatically independently of the location of the nodes and without the need to adjust any regularization parameter, in contrast to the control effort algorithms. As disadvantage, it requires higher computational demands than the other decentralized constrained techniques. It is clear from the obtained results that, if the number of nodes increases, the attenuation of the constrained algorithms gets worse.

Likewise, the proposed algorithms avoid the divergence of the distributed ANC system in the case of networks with acoustically coupled nodes.

In summary, in this chapter we have demonstrated that in acoustically coupled environments where nodes cannot work independently, there are methods to ensure stability without requiring collaboration between nodes (among other techniques). And although these methods may also cause higher performance degradation, the ANC system stability may be guaranteed. In the next chapter, we address the same ANC problem over acoustically coupled networks but considering collaborative distributed algorithms.

Chapter 5

ANC applications over collaborative networks

In this chapter, we present the implementation of ANC systems over distributed networks composed of collaborative acoustic nodes considering the LMS method and some of its variants. In cases when decentralized methods are not capable of properly solving the ANC problem, collaborative distributed algorithms may be considered with the aim to reach a solution as similar as possible to the centralized one. To this end and with the aim to improve the implementation of these algorithms in real scenarios, several collaborative distributed approaches have been proposed taking into account implementation aspects such as hardware constraints, convergenge rate, efficient performance, sensor locations or computational and communication burden, among others. Simulation results have been carried out at each case to validate the theoretical assumptions.

5.1 Introduction

As discussed in the previous chapter, in the case that there was no acoustic interaction among the nodes, it is possible to achieve the centralized cancelation solution (and consequently, the system stability) by using a decentralized ANC system. On the contrary, when the acoustic interaction among loudspeakers and microphones is present, the use of decentralized ANC strategies may offer an efficient solution for solving the acoustic coupling problem in acoustically coupled networks, as demonstrated in Chapter 4. However, these methods may suffer performance degradation in large ASNs. A possible solution lies in the use of distributed ANC system over a network of collaborative nodes. In this case, allowing cooperation among nodes, it is possible to stabilize the control system as well as to reach results equivalent to those

of the centralized method. Previous works [5, 6] showed that the implementation of the LMS algorithm over distributed networks using collaborative strategies achieves good results. However, most of the adaptive distributed networks approaches use collaborative nodes to try to estimate the same network-wide filter at each node by using the LMS algorithm. This is because they tackle a distributed system identification problem over an ASN composed of passive nodes (equipped with sensors) that do not interfere or modify the behavior of the rest of the nodes. However, if the ASN has to support an ANC application, each node estimates a local version of the network-wide filter, different but inter-related with the rest of the nodes. In this case, nodes are also equipped with actuators capable of generating signals that may affect not only its sensor but also the sensors of the rest of the nodes. Therefore, the solution estimated at each node depend on the rest of the nodes since they share information related to the acoustic system. This leads to the use of the FxLMS approach. A distributed version of the MEFxLMS algorithm using incremental communication strategies with sample-by-sample data acquisition was presented in [22]. This strategy, denoted as DMEFxLMS algorithm, has been presented as an efficient solution to solve the problem of creating quiet zones in acoustically coupled networks achieving the same result as the centralized fashion. For this reason, the DMEFxLMS algorithm is the basis for all the collaborative distributed algorithms presented throughout this chapter. However, it should be noted that, on the practical implementation of ANC over ASN, practical constraints need to be considered to ensure the distributed ANC system stability.

For example, acoustic nodes are usually equipped with power constrained actuators. If these constraints are not considered within the adaptive algorithm, the control signals may increase and saturate the hardware devices, causing system instability. To avoid this drawback, a control effort weighting can be considered in the cost function of the distributed algorithm at each node to keep the output signal under a given threshold. In addition note that, in other cases, it is necessary to improve the performance of the LMS algorithms in order to improve the convergence behavior in transient state. The Affine Projection (AP) algorithms [47] has been developed with that objective at the cost of higher computational demands. With regard to this, in this chapter we present a scalable and low-computational distributed approach of the affine-projection-like algorithm [114] using an incremental communication among the nodes. Moreover, and due to the data acquisition performed in real-time applications, the frequency domain implementation of the DMEFxLMS algorithm working with blocks of samples may be required. Therefore, the implementation of an ANC system on a distributed network is then proposed using both an incremental [5] and a diffusion strategies [33] between neighboring nodes. This is a previous step to the real-time implementation of a distributed ANC prototype. On the other hand, since it is widely assumed the time invariance of the ANC filters [3], a distributed ANC application could be seen as a NSPE problem. Therefore, a clustered-version

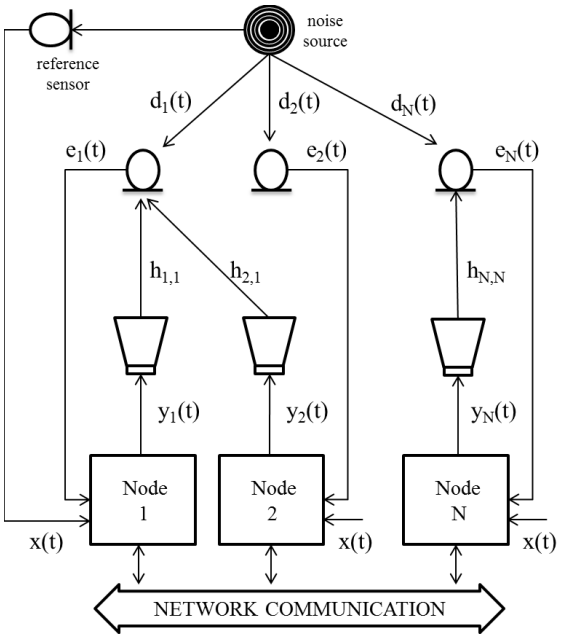


Figure 5.1: ASN of N nodes for an ANC system.

of the DMEFxLMS algorithm based on the node-specific technique proposed in [36] is presented with the aim to reduce the computational and communication demands of ASNs. In this way, only using a subset of nodes that have the most useful data, it is possible to reduce the amount of data shared among nodes. Finally, another limitation arises related to the physical location of the error sensors. The addition of the remote microphone (RM) technique to the ANC systems has recently allowed to create local zones of quiet avoiding an uncomfortable location of error sensors. This technique was developed to shift the zone of quiet to a desired location that is remote from the physical sensor. However, the implementation of this combined ANC system over distributed networks has not been already aborded. At the end of this chapter, a distributed version of the MEFxLMS algorithm considering the remote microphone method is proposed.

As a result, in the following sections, several collaborative approaches based on the DMEFxLMS algorithm are proposed considering all these practical aspects with the aim to implement them over real ASNs. To this end, the solutions of these strategies can be derived from (3.22) as demonstrated in the following sections. Note that, as previously commented, firstly the centralized solution is calculated and then, the contribution of every node in a ring-topology distributed network will be derived.

5.2 Collaborative distributed algorithm

Consider a network of N nodes that supports an ANC system composed by N sensors and N actuators, as shown in Figure 5.1. In this case, a communication network is available to allow data exchange among the nodes. Unless otherwise stated, we assume ideal networks with no communications constraints. In order to deal with the ANC problem, each node should be able to locally process its own information and the information received from its neighbors to cancel a disturbance noise at specific sensor locations. As previously commented, in [22] a new formulation to introduce the distributed incremental version of the centralized MEFxLMS algorithm has been presented. In this section we aim to present this distributed formulation as the basis for the distributed strategies proposed below. To this end, the centralized MEFxLMS algorithm must be first derived from (3.22).

In this case, we consider $I'{=}I{=}1$. Consequently, $\mathbf{A}$, $\boldsymbol{\Upsilon}_i$ and $\boldsymbol{\Theta}_{i'i}$ are defined as $[LN{\times}LN]$ identity matrices with $\boldsymbol{\Psi}_{i'i}$ and $\boldsymbol{\phi}_{i'}$ being both also identity matrices of size $[QN{\times}QN]$,

$$\mathbf{A}{=}\boldsymbol{\Upsilon}_i{=}\boldsymbol{\Theta}_{i'i}{=}\begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & 1 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 1\end{bmatrix}, \qquad \boldsymbol{\Psi}_{i'i}{=}\boldsymbol{\phi}_{i'}{=}\begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & 1 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 1\end{bmatrix}. \tag{5.1}$$

Therefore, $[\hat{\mathbf{U}}]_c(n)$ can be calculated as

$$[\hat{\mathbf{U}}]_c(n) = \hat{\mathbf{U}}(n)\mathbf{S}(n) = \begin{bmatrix}[1.5]\sum_{k=1}^{N}\hat{\mathbf{U}}_{1k}(n)\mathbf{S}_{k1}(n) & \sum_{k=1}^{N}\hat{\mathbf{U}}_{1k}(n)\mathbf{S}_{k2}(n) & \cdots & \sum_{k=1}^{N}\hat{\mathbf{U}}_{1k}(n)\mathbf{S}_{kN}(n)\\ \sum_{k=1}^{N}\hat{\mathbf{U}}_{2k}(n)\mathbf{S}_{k1}(n) & \sum_{k=1}^{N}\hat{\mathbf{U}}_{2k}(n)\mathbf{S}_{k2}(n) & \cdots & \sum_{k=1}^{N}\hat{\mathbf{U}}_{2k}(n)\mathbf{S}_{kN}(n)\\ \vdots & \vdots & \ddots & \vdots\\ \sum_{k=1}^{N}\hat{\mathbf{U}}_{Nk}(n)\mathbf{S}_{k1}(n) & \sum_{k=1}^{N}\hat{\mathbf{U}}_{Nk}(n)\mathbf{S}_{k2}(n) & \cdots & \sum_{k=1}^{N}\hat{\mathbf{U}}_{Nk}(n)\mathbf{S}_{kN}(n)\end{bmatrix} \tag{5.2}$$

Note that, in the case of $Q{=}1$ and $\mathbf{S}(n)$ as an identity matrix of size $[LN{\times}LN]$, (5.2) can be simplified as

$$[\hat{\mathbf{U}}]_c(n) = \begin{bmatrix}\mathbf{u}_{11}(n) & \mathbf{u}_{12}(n) & \cdots & \mathbf{u}_{1N}(n)\\ \mathbf{u}_{21}(n) & \mathbf{u}_{22}(n) & \cdots & \mathbf{u}_{2N}(n)\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{u}_{N1}(n) & \mathbf{u}_{N2}(n) & \cdots & \mathbf{u}_{NN}(n)\end{bmatrix} \tag{5.3}$$

which is the same matrix as $\mathbf{U}(n)$ defined in (3.2) for the centralized case. Considering both $\boldsymbol{\beta}$ and $\boldsymbol{\Gamma}$ as null matrices and ϵ_k=0, the cost function of the centralized network is given by

$$J(n) = \mathbf{e}^T(n)\mathbf{e}(n)\} = (\mathbf{d}(n) + \mathbf{U}^T(n)\mathbf{w}(n))^T(\mathbf{d}(n) + \mathbf{U}^T(n)\mathbf{w}(n)) \tag{5.4}$$

which is the same as defined in (3.4) and therefore, the global filter updating equation of the centralized strategy (MEFxLMS) is obtained as

$$\mathbf{w}(n+1) = \mathbf{w}(n) \ - 2\mu\mathbf{U}(n)\mathbf{e}(n), \tag{5.5}$$

which is equal to (3.8).

5.2.1 DMEFxLMS algorithm

It can be seen in (5.5), that all the error signals are necessary to calculate the coefficients of each filter, so that a central unit that receives and transmits all the information through the network is required (see Figure 5.2.a). The problem is that, if the number of nodes increases or if multichannel nodes are used, it is not straightforward to transmit information between the central unit and each node due to an increase in the bandwidth required for communication. Moreover, any failure in the central unit will cause no information to be processed. Therefore, a distributed network (see Figure 5.2.b) with the computational burden shared among the nodes becomes necessary to solve these problems. Since the number of signals processed at every node is low, this type of processing provides a more efficient computational performance. In addition, depending on the strategy used to exchange information among the nodes, the bandwidth in data transmission can be reduced. It should be noted that a distributed ASN in the context of this paper means that, not only are the nodes physically distributed in the area of interest, but also the processing (or computation) is divided among the nodes. Previous works [5,6] showed that the implementation of the LMS algorithm over distributed networks using collaborative strategies achieves good results. However, those works do not consider acoustic nodes that acoustically interact with the environment both controlling and modifying it, as it was introduced in [3]. The distributed version of the MEFxLMS algorithm presented in [22] aims to distribute the calculation of $\mathbf{w}(n)$ among the N nodes of the ASN described in Figure 5.1 but considering a ring topology with an incremental strategy when there are no communication constraints in the network (see Figure 5.2.b). The formulation of this distributed approach is developed below.

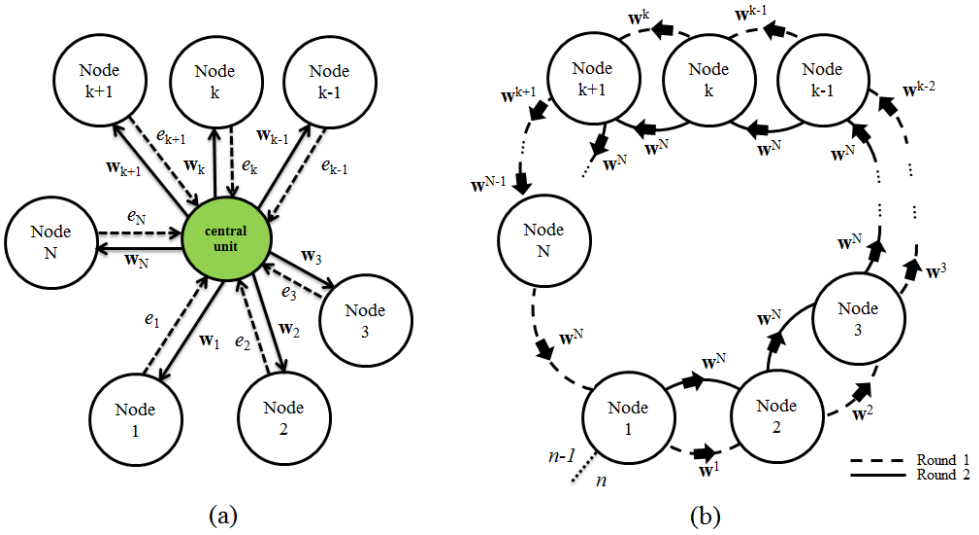


Figure 5.2: (a) A centralized ASN and (b) a ring topology distributed ASN with incremental communication. In (b), data transfer rounds are represented with different type of lines.

Note that, in distributed networks, only local data are available at each node. Thus, the objective is to split up the sum of the global updating equation into the contributions of each node, that is, the kth term in the sum of (5.5) can be only calculated by the kth node. In the case of a ring network based on an incremental strategy, the coefficients of the adaptive filters are calculated by distributing the calculation among different nodes by transmitting information to an adjacent node in a consecutive order. Thus, every node can calculate a portion of the sum of the filter updating equation and supplies to the next node the partial result to update the coefficients with its respective information. If this step is performed with an incremental strategy, the last node will have the complete updated coefficients. Moreover, each node should calculate the adaptive filters of all the nodes of the network so, in a network of N nodes, the network state would be defined by N adaptive filters one of each node. To this end, we define the global state of the network $\mathbf{w}(n)$ as the adaptive filter coefficients of each node of the network at the time instant n and $\mathbf{w}^k(n)$ as the local version of $\mathbf{w}(n)$ at the k-th node, $\mathbf{w}^k(n)$=[$[\mathbf{w}_1^k(n)]^T$ $[\mathbf{w}_2^k(n)]^T$ ... $[\mathbf{w}_N^k(n)]$]^T. Thus, by splitting up the contribution of each node from k=1 to k=N, equation (5.5) may be

Algorithm 2: *DMEFxLMS algorithm for N-nodes ASN.*

1: $\mathbf{w}^0(n) = \mathbf{w}^N(n-1)$ % Needed at node k=1
2: **for all** *node* $1 \leq k \leq N$ **do**
3: $\mathbf{w}_k(n) = \mathbf{w}^0(n)_{(1+L(k-1):Lk)}$ % Obtain local coefficients to generate the output signal
4: $y_k(n) = \mathbf{w}_k^T(n)\,[\mathbf{X}(n)]_{(:,1)}$ % Generate output signal
5: **for all** $1 \leq j \leq N$ **do**
6: $\mathbf{u}_{jk}(n) = \mathbf{X}(n)\mathbf{h}_{jk}$ % Vector that contains reference signal filtered by estimated acoustic channels
7: **end for**
8: $\mathbf{w}^k(n) = \mathbf{w}^{k-1}(n) - 2\mu\mathbf{u}_k(n)e_k(n)$ % Update local state
15: **end for**
16: $\mathbf{w}(n) = \mathbf{w}^N(n)$ % Updated global state of the network
17: **for all** *node* $1 \leq k \leq (N-1)$ **do**
18: $\mathbf{w}^k(n) = \mathbf{w}(n)$ % Disseminate global state of the network
19: **end for**

written as

$$\begin{aligned}
\mathbf{w}^1(n) &= \mathbf{w}^0(n) - 2\mu\mathbf{u}_1(n)e_1(n),\\
\mathbf{w}^2(n) &= \mathbf{w}^1(n) - 2\mu\mathbf{u}_2(n)e_2(n),\\
&\vdots\\
\mathbf{w}^N(n) &= \mathbf{w}^{N-1}(n) - 2\mu\mathbf{u}_N(n)e_N(n),
\end{aligned} \tag{5.6}$$

which is equal to (5.5) , i.e, $\mathbf{w}^N(n)$=$\mathbf{w}(n)$. Therefore, for each node of the network, we obtain

$$\mathbf{w}^k(n) = \mathbf{w}^{k-1}(n) \; - 2\mu\mathbf{u}_k(n)\; e_k(n). \tag{5.7}$$

which represents the updating rule of the state of the k-th node by using the DME-FxLMS algorithm. Finally, these coefficients are disseminated to the rest of the nodes to allow the system to generate the appropriate cancelation signals before the next iteration begins. This means $2(N-1)$ interchanges of the filter coefficients among the nodes (see Figure 5.2.(b)). To clarify the communications cycle, a differentation between consecutive time indexes, $n-1$ and n, as represented in Figure 5.2.(b). Note that, at each sample time n, all the necessary data transfer rounds in which nodes interchange their information, must be completed. Moreover, note that since $\mathbf{w}(n-1)$ is unavailable in (5.7), each node uses the local state of the last node, which contains the global state of the network at the previous iteration, that is,

$$\mathbf{w}^0(n)=\mathbf{w}^N(n-1)=\mathbf{w}(n-1). \tag{5.8}$$

It should be noted that only the local information, $\mathbf{w}_k(n)$=$\mathbf{w}^k(n-1)_{(1+L(k-1):Lk)}$, is needed to generate the kth node output signal $y_k(n)$. A summary of the DME-FxLMS algorithm pseudocodes executed per sample time at each node is given in ***Algorithm 2***.

Since most of the distributed versions proposed in this chapter are based on incremental learning, the performance analysis of the DMEFxLMS algorithm are presented. As previously commented, acoustical coupling is usually very common in multichannel ANC systems. Therefore, the adaptive filter estimated at each node is related with the rest of adaptive filters. Due to this, the following performance analysis is quite different as the previous ones for traditional distributed networks. Since we aim to prove that the distributed algorithm converges in mean to the centralized solution, first we will analyze the strategy for the centralized fashion while the analysis of the mean performance of the distributed solution will be presented below.

5.2.2 Convergence analysis

The performance of the DMEFxLMS algorithm is analyzed in terms of the mean of the network filter coefficients. As we need to deal with the expectation operator, we shall rely on several assumptions in the following analysis. Therefore, to study the mean behavior of the filter coefficients $\mathbf{w}(n)$, in addition to taking into account the list defined in (4.14), we must consider the following assumption:

- *A.4) In steady-state, the local state at each node tends to the global optimal solution.* (5.9)

As it is well known, the convergence characteristics of the MEFxLMS algorithm analyzed in the time domain are affected by the eigenvalue distribution of the autocorrelation matrix of the filtered reference signals [101]. However, in this case, the analysis of the convergence properties of the algorithm has been considered in the frequency domain due to provides a better understanding of the physical problem [115]. We assume that a primary disturbance at a single known frequency (ω_0) is being controlled and consequently, a tonal reference signal at the same frequency can be internally generated. Consequently, the formulation can be simplified by using a frequency domain analysis. This allows the modulus and phase of the signals in the steady state can be described by a single complex number. Therefore, the use of a statistically characterized reference signal implies that the autocorrelation matrix (3.6) is only dependent on the features of the acoustic system evaluated at ω_0. Based on the convergence analysis described in [3], we study the necessary conditions to guarantee the stability of the DMEFxLMS algorithm, starting from the centralized MEFxLMS approach to derive into the distributed strategy.

5.3.1.1. Mean analysis of the centralized solution

Consider that the signals involved in the ANC system depicted in Figure 5.1 can

be defined in steady-state by a single complex number. To this end, we define

$$\begin{aligned}
&\underline{\mathbf{E}}(\omega_0) = [\underline{E}_1(\omega_0)\ \underline{E}_2(\omega_0)\cdots\ \underline{E}_N(\omega_0)]^T,\\
&\underline{\mathbf{D}}(\omega_0) = [\underline{D}_1(\omega_0)\ \underline{D}_2(\omega_0)\cdots\ \underline{D}_N(\omega_0)]^T,\\
&\underline{\mathbf{W}}(\omega_0) = [\underline{W}_1(\omega_0)\ \underline{W}_2(\omega_0)\cdots\ \underline{W}_N(\omega_0)]^T,\\
&\underline{\mathbf{H}}(\omega_0) = \begin{bmatrix} \underline{H}_{11}(\omega_0) & \underline{H}_{12}(\omega_0) & \cdots & \underline{H}_{1N}(\omega_0)\\ \underline{H}_{21}(\omega_0) & \underline{H}_{22}(\omega_0) & \cdots & \underline{H}_{2N}(\omega_0)\\ \vdots & \vdots & \ddots & \vdots\\ \underline{H}_{N1}(\omega_0) & \underline{H}_{N2}(\omega_0) & \cdots & \underline{H}_{NN}(\omega_0) \end{bmatrix},
\end{aligned} \tag{5.10}$$

where $\underline{E}_j(\omega_0)$, $\underline{D}_j(\omega_0)$, $\underline{W}_j(\omega_0)$ and $\underline{H}_{jk}(\omega_0)$, are the complex component of $e_k(n)$, $d_k(n)$, $w_j(n)$ and h_{jk}, respectively, evaluated at frequency ω_0. For simplicity, the frequency index will be omitted. We can rewrite (5.4) as the instantaneous MSE solution in the frequency domain given by

$$\underline{\mathbf{J}}(n) = \underline{\mathbf{E}}^H(n)\underline{\mathbf{E}}(n) \tag{5.11}$$

where

$$\underline{\mathbf{E}}(n) = \underline{\mathbf{D}}(n) \ + \ \underline{\mathbf{H}}\underline{\mathbf{W}}(n) \tag{5.12}$$

It can be shown that the minimum MSE solution, $\underline{\mathbf{W}}_o$, can be obtained by solving the Wiener equation

$$\underline{\mathbf{W}}_o = -[\underline{\mathbf{H}}^H\underline{\mathbf{H}}]^{-1}\underline{\mathbf{H}}^H\underline{\mathbf{D}}(n). \tag{5.13}$$

By using the traditional LMS method to search the coefficients vector $\underline{\mathbf{W}}(n)$ that minimizes (5.11), we obtain

$$\begin{aligned}
\underline{\mathbf{W}}(n+1) &= \underline{\mathbf{W}}(n) - 2\mu\underline{\mathbf{H}}^H\mathbf{E}(n),\\
&= \underline{\mathbf{W}}(n) - 2\mu\underline{\mathbf{H}}^H\underline{\mathbf{D}}(n) - 2\mu\underline{\mathbf{H}}^H\underline{\mathbf{H}}\underline{\mathbf{W}}(n),
\end{aligned} \tag{5.14}$$

Note that, if $\underline{\mathbf{W}}$ converges to its optimal filters $\underline{\mathbf{W}}_o$, it is fulfilled that

$$\begin{aligned}
&\underline{\mathbf{W}}_o = \underline{\mathbf{W}}_o - 2\mu\underline{\mathbf{H}}^H\underline{\mathbf{D}}(n) - 2\mu\underline{\mathbf{H}}^H\underline{\mathbf{H}}\underline{\mathbf{W}}_o,\\
&\underline{\mathbf{H}}^H\underline{\mathbf{D}}(n) = -\underline{\mathbf{H}}^H\underline{\mathbf{H}}\underline{\mathbf{W}}_o.
\end{aligned} \tag{5.15}$$

This expression is only valid when convergence is achieved. Under these conditions, substracting $\underline{\mathbf{W}}_o$ from both sides of (5.14) and using (5.15), we obtain

$$\begin{aligned}
\underline{\mathbf{W}}(n+1) - \underline{\mathbf{W}}_o =& \underline{\mathbf{W}}(n) - 2\mu\underline{\mathbf{H}}^H\underline{\mathbf{D}}(n)\\
&- 2\mu\underline{\mathbf{H}}^H\underline{\mathbf{H}}\underline{\mathbf{W}}(n) - \underline{\mathbf{W}}_o,\\
=& \underline{\mathbf{W}}(n) + 2\mu\underline{\mathbf{H}}^H\underline{\mathbf{H}}\underline{\mathbf{W}}_o\\
&- 2\mu\underline{\mathbf{H}}^H\underline{\mathbf{H}}\underline{\mathbf{W}}(n) - \underline{\mathbf{W}}_o,\\
=& (I_N - 2\mu\underline{\mathbf{H}}^H\underline{\mathbf{H}})(\underline{\mathbf{W}}(n) - \underline{\mathbf{W}}_o),
\end{aligned} \tag{5.16}$$

where I_N is a identity matriz of size $N \times N$. To see how $\underline{\mathbf{W}}(n)$ converges to $\underline{\mathbf{W}}_o$, we analyze its mean behavior. Considering the coefficient error vector as $\underline{\mathbf{V}}(n) = E\{\underline{\mathbf{W}}(n) - \underline{\mathbf{W}}_o\}$, we can rewrite (5.16) as

$$\underline{\mathbf{V}}(n+1) = (I_N - 2\mu \underline{\mathbf{H}}^H \underline{\mathbf{H}})\underline{\mathbf{V}}(n). \tag{5.17}$$

This equation describes the evolutionary behavior of the mean values of the error vector $\underline{\mathbf{E}}$ in the filter coefficients $\underline{\mathbf{W}}$. If the matrix $\underline{\mathbf{H}}^H \underline{\mathbf{H}}$ is described by its eigenvalue decomposition, then we can write

$$\underline{\mathbf{H}}^H \underline{\mathbf{H}} = \mathbf{Q} \underline{\mathbf{\Lambda}} \mathbf{Q}^H, \tag{5.18}$$

being $\mathbf{Q}$ the $N \times N$ eigenvectors matrix of $\underline{\mathbf{H}}^H \underline{\mathbf{H}}$ and $\underline{\mathbf{\Lambda}}$ the diagonal matrix containing the N eigenvalues (λ_p) of $\underline{\mathbf{H}}^H \underline{\mathbf{H}}$. Substituting (5.18) in (5.17) and noting that $I_N = \mathbf{Q}\mathbf{Q}^H$, we obtain

$$\underline{\mathbf{V}}(n+1) = \mathbf{Q}(I_N - 2\mu \underline{\mathbf{\Lambda}})\mathbf{Q}^H \underline{\mathbf{V}}(n). \tag{5.19}$$

Defining the rotated error vector as $\underline{\mathbf{V}}'(n) = \mathbf{Q}^H \underline{\mathbf{V}}(n)$ and multiplying both sides of (5.19) by $\mathbf{Q}^H$, we get

$$\underline{\mathbf{V}}'(n+1) = (I_N - 2\mu \underline{\mathbf{\Lambda}})\underline{\mathbf{V}}'(n). \tag{5.20}$$

This expression shows that the algorithm converges in a series of independent modes and, therefore, (5.20) can be expressed as $N \times N$ recursive equations

$$V_p'(n+1) = (1 - 2\mu\lambda_p)V_p'(n) = (1 - 2\mu\lambda_p)^{f+1}\, V_p'(0), \tag{5.21}$$

for $p = 1, 2, \ldots, N$. This implies that $\underline{\mathbf{W}}(n)$ converges to $\underline{\mathbf{W}}_o$ if the step-size μ is selected so that

$$|1 - 2\mu\lambda_p| < 1, \tag{5.22}$$

that it may be expanded as

$$0 < \mu_{max} < \frac{1}{\lambda_{max}(\underline{\mathbf{H}}^H \underline{\mathbf{H}})}, \tag{5.23}$$

where $\lambda_{max}(\underline{\mathbf{H}}^H \underline{\mathbf{H}})$ is the maximum of the eigenvalues of $\underline{\mathbf{H}}^H \underline{\mathbf{H}}$. Therefore, (5.23) provides the necessary condition to guarantee the centralized system stability.

5.3.1.2. Mean analysis of the distributed solution

Now, we examinate the mean behaviour of the distributed algorithm defined by analyzing the mean behaviour of the adaptive weights at each node of the network evaluated at the frequency ω_0.

Equation (5.7) can be expressed in the frequency domain as

$$\underline{\mathbf{W}}^{k}(n) = \underline{\mathbf{W}}^{k-1}(n) - 2\mu\underline{\mathbf{H}}_k\underline{E}_k(n), \tag{5.24}$$

where $\underline{\mathbf{H}}_k{=}[H_{1k}\ H_{2k}\cdots\ H_{Nk}]^H$ and the error signal at the k-th node is redefined as

$$\underline{E}_k(n) = \underline{D}_k(n) + \underline{\mathbf{H}}_k^H\underline{\mathbf{W}}(n). \tag{5.25}$$

Substituting (5.25) in (5.24), we obtain

$$\underline{\mathbf{W}}^{k}(n){=}\underline{\mathbf{W}}^{k-1}(n){-}2\mu\underline{\mathbf{H}}_k\underline{E}_k(n){-}2\mu\underline{\mathbf{H}}_k\underline{\mathbf{H}}_k^H\underline{\mathbf{W}}(f{-}1). \tag{5.26}$$

Considering the assumption (5.9), when $f{\to}\infty$, the local state of the nodes tends to the global optimal solution, i.e. $\underline{\mathbf{W}}_o^k{=}\underline{\mathbf{W}}_o^{k-1}$ and therefore, $\underline{\mathbf{W}}_o^k{\to}\underline{\mathbf{W}}_o$ for $k{=}1, 2, \ldots, N$, then

$$\begin{aligned}\underline{\mathbf{W}}_o^k &= \underline{\mathbf{W}}_o^{k-1} - 2\mu\underline{\mathbf{H}}_k\underline{D}_k(n) - 2\mu\underline{\mathbf{H}}_k\underline{\mathbf{H}}_k^H\underline{\mathbf{W}}_o,\\ \underline{\mathbf{H}}_k\underline{D}_k(n) &= -\underline{\mathbf{H}}_k\underline{\mathbf{H}}_k^H\underline{\mathbf{W}}_o,\end{aligned} \tag{5.27}$$

Note that this expression is only valid when convergence is achieved. Under these conditions, subtracting $\underline{\mathbf{W}}_o$ from both sides of (5.26) and using (5.27), (5.26) can be written as

$$\begin{aligned}\underline{\mathbf{W}}^{k}(n) - \underline{\mathbf{W}}_o &= \underline{\mathbf{W}}^{k-1}(n) - 2\mu\underline{\mathbf{H}}_k\underline{E}_k(n)\\ &\quad - 2\mu\underline{\mathbf{H}}_k\underline{\mathbf{H}}_k^H\underline{\mathbf{W}}(n-1) - \underline{\mathbf{W}}_o\\ &= \underline{\mathbf{W}}^{k-1}(n) + 2\mu\underline{\mathbf{H}}_k\underline{\mathbf{H}}_k^H\underline{\mathbf{W}}_o\\ &\quad - 2\mu\underline{\mathbf{H}}_k\underline{\mathbf{H}}_k^H\underline{\mathbf{W}}(n-1) - \underline{\mathbf{W}}_o\\ &= \underline{\mathbf{W}}^{k-1}(n) - \underline{\mathbf{W}}_o\\ &\quad - 2\mu\underline{\mathbf{H}}_k\underline{\mathbf{H}}_k^H(\underline{\mathbf{W}}(n-1) - \underline{\mathbf{W}}_o).\end{aligned} \tag{5.28}$$

Note that, in an incremental method, the expression $\underline{\mathbf{W}}(f{-}1){=}\underline{\mathbf{W}}^0(n)$ can be considered. Defining $\underline{\mathbf{V}}^k(n){=}E\{\underline{\mathbf{W}}^k(n){-}\underline{\mathbf{W}}_o\}$, we obtain

$$\underline{\mathbf{V}}^{k}(n) = \underline{\mathbf{V}}^{k-1}(n) - 2\mu\underline{\mathbf{H}}_k\underline{\mathbf{H}}_k^H\underline{\mathbf{V}}^0(n). \tag{5.29}$$

Iterating (5.29), the solution evolves to

$$\begin{aligned}
\underline{\mathbf{V}}^1(n) &= \underline{\mathbf{V}}^0(n) - 2\mu\underline{\mathbf{H}}_1\underline{\mathbf{H}}_1^H\underline{\mathbf{V}}^0(n) \\
&= (I_N - 2\mu\underline{\mathbf{H}}_1\underline{\mathbf{H}}_1^H)\underline{\mathbf{V}}^0(n), \\
\underline{\mathbf{V}}^2(n) &= \underline{\mathbf{V}}^1(n) - 2\mu\underline{\mathbf{H}}_2\underline{\mathbf{H}}_2^H\underline{\mathbf{V}}^0(n) \\
&= \underline{\mathbf{V}}^0(n) - 2\mu\underline{\mathbf{H}}_1\underline{\mathbf{H}}_1^H\underline{\mathbf{V}}^0(n) - 2\mu\underline{\mathbf{H}}_2\underline{\mathbf{H}}_2^H\underline{\mathbf{V}}^0(n) \\
&= (I_N - 2\mu\underline{\mathbf{H}}_1\underline{\mathbf{H}}_1^H - 2\mu\underline{\mathbf{H}}_2\underline{\mathbf{H}}_2^H)\underline{\mathbf{V}}^0(n), \\
&\vdots \\
\underline{\mathbf{V}}^N(n) &= \underline{\mathbf{V}}^{N-1}(n) - 2\mu\underline{\mathbf{H}}_N\underline{\mathbf{H}}_N^H\underline{\mathbf{V}}^0(n) \\
&= [I_N - (2\mu\underline{\mathbf{H}}_1\underline{\mathbf{H}}_1^H + 2\mu\underline{\mathbf{H}}_2\underline{\mathbf{H}}_2^H + \dots \\
&\quad + 2\mu\underline{\mathbf{H}}_N\underline{\mathbf{H}}_N^H)]\underline{\mathbf{V}}^0(n).
\end{aligned} \tag{5.30}$$

Therefore, we obtain that the k-th term of (5.30) can be expressed as

$$\underline{\mathbf{V}}^k(n) = \left[I_N - 2\mu\sum_{l=1}^{k}\underline{\mathbf{H}}_l\underline{\mathbf{H}}_l^H\right]\underline{\mathbf{V}}^0(n). \tag{5.31}$$

Setting $k{=}N$, we obtain

$$\underline{\mathbf{V}}^N(n) = \left[I_N - 2\mu\sum_{k=1}^{N}\underline{\mathbf{H}}_k\underline{\mathbf{H}}_k^H\right]\underline{\mathbf{V}}^0(n). \tag{5.32}$$

Noting that, in an incremental strategy, $\underline{\mathbf{V}}(n){=}\underline{\mathbf{V}}^N(n)$ and $\underline{\mathbf{V}}^0(n){=}\underline{\mathbf{V}}(f{-}1)$, we obtain

$$\underline{\mathbf{V}}(n) = \left[I_N - 2\mu\sum_{k=1}^{N}\underline{\mathbf{H}}_k\underline{\mathbf{H}}_k^H\right]\underline{\mathbf{V}}(n-1). \tag{5.33}$$

However, note that $\displaystyle\sum_{k=1}^{N}\underline{\mathbf{H}}_k\underline{\mathbf{H}}_k^H{=}\underline{\mathbf{H}}\underline{\mathbf{H}}^H$. In order to prove it, we consider

$$\begin{aligned}
\underline{\mathbf{H}}_k\underline{\mathbf{H}}_k^H &= \begin{bmatrix}\underline{H}_{1k}\\ \underline{H}_{2k}\\ \vdots \\ \underline{H}_{Nk}\end{bmatrix}\left[\underline{H}_{1k}^H\underline{H}_{2k}^H\cdots\underline{H}_{Nk}^H\right] \\
&= \begin{bmatrix}\underline{H}_{1k}\underline{H}_{1k}^H & \underline{H}_{1k}\underline{H}_{2k}^H & \cdots & \underline{H}_{1k}\underline{H}_{Nk}^H \\ \underline{H}_{2k}\underline{H}_{1k}^H & \underline{H}_{2k}\underline{H}_{2k}^H & \cdots & \underline{H}_{2k}\underline{H}_{Nk}^H \\ \vdots & \vdots & \ddots & \vdots \\ \underline{H}_{Nk}\underline{H}_{1k}^H & \underline{H}_{Nk}\underline{H}_{2k}^H & \cdots & \underline{H}_{Nk}\underline{H}_{Nk}^H\end{bmatrix}.
\end{aligned} \tag{5.34}$$

On the other hand, we have

$$\underline{\mathbf{H}} = \left[\underline{\mathbf{H}}_1 \underline{\mathbf{H}}_2 \cdots \underline{\mathbf{H}}_N\right] = \begin{bmatrix} \underline{H}_{11} & \underline{H}_{12} & \cdots & \underline{H}_{1N} \\ \underline{H}_{21} & \underline{H}_{22} & \cdots & \underline{H}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{H}_{N1} & \underline{H}_{N2} & \cdots & \underline{H}_{NN} \end{bmatrix}, \tag{5.35}$$

$$\underline{\underline{\mathbf{H}}}^H = \left[\underline{\mathbf{H}}_1^H \underline{\mathbf{H}}_2^H \cdots \underline{\mathbf{H}}_N^H\right] = \begin{bmatrix} \underline{H}_{11}^H & \underline{H}_{21}^H & \cdots & \underline{H}_{N1}^H \\ \underline{H}_{12}^H & \underline{H}_{22}^H & \cdots & \underline{H}_{N2}^H \\ \vdots & \vdots & \ddots & \vdots \\ \underline{H}_{1N}^H & \underline{H}_{2N}^H & \cdots & \underline{H}_{NN}^H \end{bmatrix}. \tag{5.36}$$

And therefore, we obtain

$$\underline{\mathbf{H}}\underline{\mathbf{H}}^H = \begin{bmatrix} \underline{H}_{11}\underline{H}_{11}^H + \underline{H}_{12}\underline{H}_{12}^H + \underline{H}_{1N}\underline{H}_{1N}^H & \underline{H}_{11}\underline{H}_{21}^H + \underline{H}_{12}\underline{H}_{22}^H + \underline{H}_{1N}\underline{H}_{2N}^H \\ \underline{H}_{21}\underline{H}_{11}^H + \underline{H}_{22}\underline{H}_{12}^H + \underline{H}_{2N}\underline{H}_{1N}^H & \underline{H}_{21}\underline{H}_{21}^H + \underline{H}_{22}\underline{H}_{22}^H + \underline{H}_{2N}\underline{H}_{2N}^H \\ \vdots & \vdots \\ \underline{H}_{N1}\underline{H}_{11}^H + \underline{H}_{N2}\underline{H}_{12}^H + \underline{H}_{NN}\underline{H}_{1N}^H & \underline{H}_{N1}\underline{H}_{21}^H + \underline{H}_{N2}\underline{H}_{22}^H + \underline{H}_{NN}\underline{H}_{2N}^H \end{bmatrix}$$
$$\begin{matrix} \cdots & \underline{H}_{11}\underline{H}_{N1}^H + \underline{H}_{12}\underline{H}_{N2}^H + \underline{H}_{1N}\underline{H}_{NN}^H \\ \cdots & \underline{H}_{21}\underline{H}_{N1}^H + \underline{H}_{22}\underline{H}_{N2}^H + \underline{H}_{2N}\underline{H}_{NN}^H \\ \ddots & \vdots \\ \cdots & \underline{H}_{N1}\underline{H}_{N1}^H + \underline{H}_{N2}\underline{H}_{N2}^H + \underline{H}_{NN}\underline{H}_{NN}^H \end{matrix} \Bigg] = \sum_{k=1}^{N} \underline{\mathbf{H}}_k \underline{\mathbf{H}}_k^H. \tag{5.37}$$

Proved that $\sum_{k=1}^{N} \underline{\mathbf{H}}_k \underline{\mathbf{H}}_k^H = \underline{\mathbf{H}}\underline{\mathbf{H}}^H$, (5.33) can be expressed as

$$\underline{\mathbf{V}}(n) = (I_N - 2\mu \underline{\mathbf{H}}^H \underline{\mathbf{H}}) \underline{\mathbf{V}}(n-1), \tag{5.38}$$

which is the same expression as (5.17) in the centralized approach. Following the same eigendecomposition procedure, the necessary condition for mean convergence of the distributed method is calculated as

$$0 < \mu_{max} < \frac{1}{\lambda_{max}(\underline{\mathbf{H}}^H \underline{\mathbf{H}})}. \tag{5.39}$$

Since (5.39) is equal to (5.23), for small step-sizes, it has been proved that the distributed algorithm converges in mean to the centralized solution.

5.2.3 Simulation results

In this section the performance of the DMEFxLMS compared to the centralized MEFxLMS have been evaluated in terms of noise reduction convergence speed and

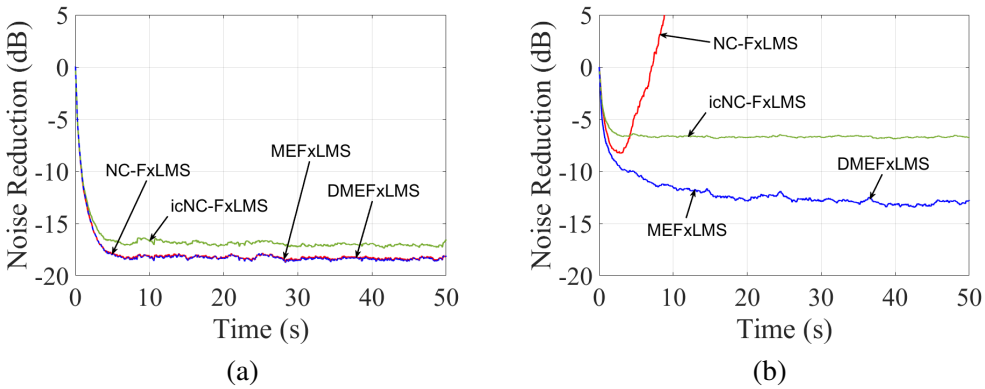


Figure 5.3: Noise reduction of the (a) uncoupled and (b) coupled two-node coupled networks comparing non-collaborative and collaborative distributed algorithms.

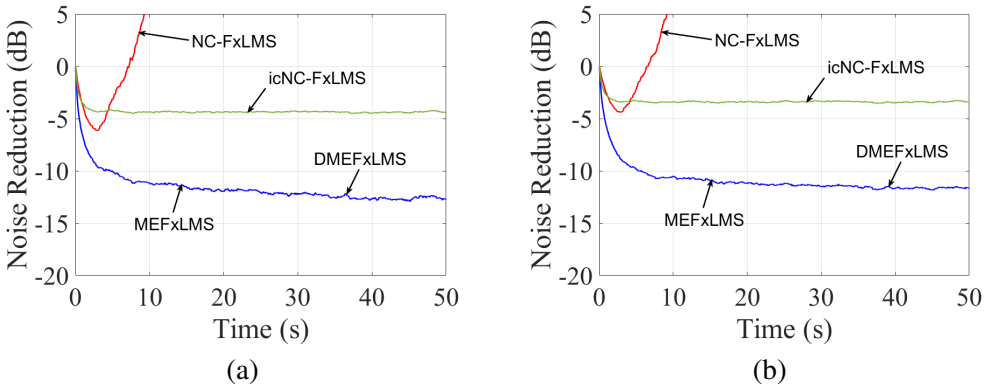


Figure 5.4: Noise reduction of the (a) four-node and (b) six-node coupled networks comparing non-collaborative and collaborative distributed algorithms.

computational demands. In addition and with the aim to justify the use of collaborative statregies, it can be useful for the sake of comparison to include the results for the NC-DFxLMS and ic-NC-DFxLMS algorithms, as representative methods of decentralized strategies. Similarly as the simulations carried out in 4.7, initially we have compared the performance of the algorithms in both uncoupled and coupled two-node ASNs and then, four-node and six-node coupled networks are considered. The designed ASNs are depicted in Figure 4.3 with nodes 1 and 2 for N=2, nodes 1 to 4 for N=4, and nodes 1 to 1 for N=6. In addition, we use the real acoustic channels and the parameters configuration depicted in Section 3.3.

Figure 5.3.(a) presents the $NR(n)$ obtained by an uncoupled two-nodes ASN for the four algorithms. The NC-DFxLMS, the MEFxLMS and the DMEFxLMS algorithms exhibit the same stable performance achieving an attenuation of 18.4 dB. The ic-NC-DFxLMS algorithm also achieves a robust performance but its $NR(n)$ is slighty lower (17 dB) than the other methods. On the contrary, in a coupled ASN, the NC-DMEFxLMS turns unstable while the ic-NC-DFxLMS, DMEFxLMS and MEFxLMS algorithms mantain stable, providing 6.7 dB of noise reduction for the non-collaborative method and 13 dB of noise reduction for both the collaborative and the centralized strategies, as shown in Figure 5.3.(b). Note that in both uncoupled and coupled cases, DMEFxLMS and MEFxLMS algorithms exhibit equal performance.

Results for larger coupled ASNs are quite similar as the results depicted in Figure 5.3.(b). In both four-node and six-node ASNs, the NC-DMEFxLMS algorithm does not converge and the behaviour of the ic-NC-DFxLMS algorithm is stable but the $NR(n)$ achieved is very poor (below 5 dB). On the other hand, the $NR(n)$ curves obtained by the DMEFxLMS and MEFxLMS algorithms show a similar behaviour in both the four-node and six-node ASNs, where the $NR(n)$ in both cases fluctuates around 12 dB.

Therefore, from Figures 5.3 and 5.4, it can be concluded that the DMEFxLMS algorithm has the same performance as the MEFxLMS algorithm showing a robust and stable performance in terms of stability and outperforming the results obtained by the non-collaborative strategies. Note that, although the $NR(n)$ obtained by the DMEFxLMS algorithm will depend of the number and physical location of nodes, the non-collaborative strategies seems to be more sensitive to the ASNs settings, as it can be seen from Figures 5.4.(a) and 5.4.(b).

Regarding implementation aspects, the DMEFxLMS algorithm has exactly the same computational complexity as the MEFxLMS algorithm. A total number of $L(N{+}1){+}MN{+}1$ multiplications per iteration and per node are computed in both strategies. Note that this amount is slightly lower than the $LN{+}M(2N{-}1)$ multiplications required for the icNC-DFxLMS method. However, decentralized strategies do not need to transfer information through the communications network. The communication demands of the DMEFxLMS algorithm requires that a data stream of $2LN(N{-}1)$ samples should be propagated through the nodes in an incremental manner since every node must transfer LN coefficients to the following node $2(N{-}1)$ times in each sample iteration (see Figure 5.2.(b)). Let us present a practical example of a real-time ANC application over an incremental ASN composed of 4 nodes ($N{=}4$). Consider an audio card working with blocks of 2048 samples ($B{=}2048$) and with the common sampling frequency (f_s) of 44100 Hz. The buffering time, defined as B/f_s, is then 0.0464 s. Using the single-precision floating-point format (that occupies 4 bytes) and a general filter length of 4096 taps ($L{=}4096$), we obtain the amount of transferred data through four nodes at the buffering time as $(L{\times}N){\times}2(N{-}1){\times}4$ bytes $=384$ kB (kilobytes). Therefore, the data that we must transfer through four

nodes in one second is 384 kB / 0.0464 s $=8268.75$ kBps (kB per second). In summary, a transfer rate at least of 8.1 megabytes per second (MBps) would be necessary implement the DMEFxLMS algorithm over an four-node incremental network. Therefore, using a standard Ethernet network of 1 Gbps (125 MBps), we would have enough rate capacity to perform the required data transfer among the nodes. In addition, the standard network capacity allows to increase the number of nodes up to ten nodes. Anyway, if we did not have enough transmission speed, the proposed system would not achieve the performance of the centralized system but it would work with a slower convergence.

5.2.4 Conclusions

In ths section, the scalable and versatile distributed implementation of the MEFxLMS algorithm for an ANC system using an incremental strategy in the network proposed in [22] has been introduced here as the basis of the collaborative strategies which will be presented in the following sections. It has been proven that the DMEFxLMS algorithm has the same performance than the centralized version when there are no communication constraints in the network. 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.

5.3 Collaborative distributed algorithm using control effort

In order to provide a more realistic solution for practical implementations, this section aims to analyze the effect of control effort weighting on the behavior of a distributed collaborative ANC system over an ASN. It should be noted that, in practical ANC systems, the hardware used to generate output signals at each node has power limitations. In case of saturation of loudspeakers or amplifiers outputs, the control signals generated by the adaptive filters may increase making the system unstable. Note that, in those cases, nonlinearities may cause the system divergence. A possible strategy is based on limiting the control signal power by minimizing the energy of the adaptive filters to avoid that the signals emitted by the loudspeakers may increase unlimitedly. In this case, the objective is to control the signals generated by the adaptive filters at each node in order to limit the amplitude of the signals reproduced by the loudspeakers. Constraint techniques have been widely used in practical control systems [103] [104]. Some of them may be intended for use in real scenarios to improve the processing efficiency [116] [113] [109] or even to reduce nonlinearity effects of

the system [112]. A common way is to use a leakage during the updating of the control filter coefficients in the LMS algorithm [41]. Because of the addition of bias to the coefficients' update, this leaky LMS algorithm suffers from a degradation in the steady-state error attenuation [43]. On the contrary, it is possible to stabilize the system by controlling the value of the leakage coefficient [34]. A possible solution to improve the performance of the leaky algorithm is to use the clipping and the re-scaling strategies [104]. While clipping method just saturates the output signal, the re-scaling method also scales the filter weights when the output is too large in order to avoid large oscillations in the coefficients' update. The behavior of these methods is analyzed in [104] for a single-channel ANC system. Regarding the proper value of the leakage coefficient, although it is possible to calculate a range to assure the system linearity [112] or introduce an uneven weight at each node [117], it is usually choosen by trial and error depending on the signal power supported by the system loudspeakers. For all these reasons, in this section we considers the effect of effort constraints on the behavior of an ANC system over a distributed network composed of single-channel acoustic nodes. To this end, we analyze the performance of several constrained methods (described in [104] and previously introduced in 4.4) when applied over a network with distributed nodes and incremental learning without communication constraints. A study of implementation aspects such as computational complexity and communication capabilities among the nodes in the network for the different control effort strategies is also presented. To our knowledge, no other analysis of this type has been already reported. In addition, we propose an intuitive strategy based on limiting the control signal power to avoid overdriving the loudspeakers that ensures distributed ANC system stability while reducing the communication demands of the applied constrained strategies.

5.3.1 Leaky DMEFxLMS algorithms

Following the same procedures as previous algorithms, we start from a network centralized approach to derive into the contribution of every node in a distributed network using an incremental strategy of the data exchange and assuming practical constraints into the solution. To this end, the cost function (3.22) is now devoted to minimize both error and control signals by taking into account the effort penalty. Therefore, considering $[\hat{\mathbf{U}}]_c(n)$ as defined in (5.3), the matrix $\boldsymbol{\beta}$ as defined (3.38), Q=1, $\boldsymbol{\Gamma}$ as a null matrix and ϵ_k=0, the cost function is now given by

$$J(n) = (\mathbf{d}(n) + \mathbf{U}^T(n)\mathbf{w}(n))^T(\mathbf{d}(n) + \mathbf{U}^T(n)\mathbf{w}(n)) \; + \; \mathbf{w}^T(n)\,\boldsymbol{\beta}\,\mathbf{w}(n), \quad (5.40)$$

which can be also expressed as

$$\begin{aligned} J(n) &= \mathbf{e}(n)^T\mathbf{e}(n) \; + \mathbf{w}^T(n)\boldsymbol{\beta}\mathbf{w}(n)\}, \\ &= \sum_{k=1}^{N} e_k^2(n) \; + \sum_{k=1}^{N} \beta_k \mathbf{w}_k^T(n)\mathbf{w}_k(n). \end{aligned} \quad (5.41)$$

Therefore, the global filter updating equation of the centralized MEFxLMS algorithm using control effort (leaky or l-CMEFxLMS) is stated as follows

$$\begin{aligned}\mathbf{w}(n) &= (\boldsymbol{I}_{LN} - 2\boldsymbol{\beta}\mu)\mathbf{w}(n-1) - 2\mu\mathbf{U}(n)\mathbf{e}(n), \\ &= \mathbf{w}(n-1) - 2\mu(\boldsymbol{\beta}\mathbf{w}(n-1) + \sum_{k=1}^{N}\mathbf{u}_k(n)e_k(n)).\end{aligned} \tag{5.42}$$

Given the well-known drawbacks of the centralized system, such as large computational and communication demands, the use of a distributed network is required. Following the steps described in 5.2, the implementation of (5.42) over a network of distributed nodes will be presented below. Thus, we can split up the contribution of each node in (5.42) as,

$$\begin{aligned}\mathbf{w}^1(n) &= \mathbf{w}^0(n) - \mu(\frac{\boldsymbol{\beta}}{N}\mathbf{w}(n-1) + \mathbf{u}_1(n)e_1(n)), \\ \mathbf{w}^2(n) &= \mathbf{w}^1(n) - \mu(\frac{\boldsymbol{\beta}}{N}\mathbf{w}(n-1) + \mathbf{u}_2(n)e_2(n)), \\ &\vdots \\ \mathbf{w}^N(n) &= \mathbf{w}^{N-1}(n) - \mu(\frac{\boldsymbol{\beta}}{N}\mathbf{w}(n-1) + \mathbf{u}_N(n)e_N(n)),\end{aligned} \tag{5.43}$$

Therefore, the updating filter equation of the kth node by using the leaky DMEFxLMS algorithm (l-DMEFxLMS) can be expressed as

$$\mathbf{w}^k(n) = \mathbf{w}^{k-1}(n) - \mu(\frac{\boldsymbol{\beta}}{N}\mathbf{w}(n-1) + \mathbf{u}_k(n)e_k(n))\cdot \tag{5.44}$$

As commented previously, we have considered homogeneous nodes to achieve the updating equation presented in (5.42). However, considering an ASN composed by non-homogeneous nodes, (5.42) might be computed in a different manner. For instance, the first node which updates its coefficients in the incremental sequence, could assume the full computation of the control effort releasing the remaining nodes to perform them. More specifically, the first node could calculate its local state as

$$\mathbf{w}^1(n) = \mathbf{w}^0(n) - \mu(\boldsymbol{\beta}\mathbf{w}(n-1) + \mathbf{u}_1(n)e_1(n)), \tag{5.45}$$

while, the rest of the nodes could update their coefficients as

$$\mathbf{w}^p(n) = \mathbf{w}^{p-1}(n) - \mu\mathbf{u}_p(n)e_p(n), \tag{5.46}$$

where p=2, 3, ... , N. Thus, the dissemination of the updated coefficients can be eliminated reducing the communications demands at the expenses of an increase in the computational cost of the first node.

Algorithm 3: *re-scaling l-DMEFxLMS algorithm for N-nodes ASN.*

for all $node\ 1 \leq\ k \leq N$ **do**
  $\mathbf{w}^k(n) = \mathbf{w}^{k-1}(n)$ % Copy local state of previous node (at node k=1, $\mathbf{w}^0(n) = \mathbf{w}(n-1)$)
  $\mathbf{w}_k(n) = \mathbf{w}^k(n)_{(1+L(k-1):Lk)}$ % Obtain local coefficients to generate the output signal
  $\underline{y}_k(n) = \mathbf{w}_k^T(n)[\mathbf{X}(n)]_{(:,1)}$ % Provisional output signal
  for all $1 \leq\ j \leq N$ **do**
    $\mathbf{u}_{jk}(n) = \mathbf{X}(n)\mathbf{s}_{jk}$ % Vector that contains reference signal filtered by estimated acoustic channels
  end for
  $\mathbf{w}^k(n) = \mathbf{w}^{k-1}(n) - \mu(\frac{\boldsymbol{\beta}}{N}\mathbf{w}^k(n) + \mathbf{u}_k(n)e_k(n))$ % Update local state
  if $\left|\underline{y}_k(n)\right| > y_{k_{max}}$ % If provisional output signal is greater than threshold
    $\underline{y}_k(n) = \underline{y}_k(n)\xi_k \frac{y_{k_{max}}}{\left|\underline{y}_k(n)\right|}$ % Rescale provisional output signal
    $\underline{\mathbf{w}}_k(n) = \mathbf{w}^k(n)_{(L(k-1)+1:Lk)}\ \xi_k \frac{y_{k_{max}}}{\left|\underline{y}_k(n)\right|}$ % Rescale its portion within its local state
    $\mathbf{w}^k(n)_{(1+L(k-1):Lk)} = \underline{\mathbf{w}}_k(n)$ % Updated local state with rescaled coefficients
  end if
  $y_k(n) = \underline{y}_k(n)$ % Generate output signal
end for
$\mathbf{w}(n) = \mathbf{w}^N(n)$ % Updated global state of the network properly rescaled
for all $node\ 1 \leq\ k \leq (N-1)$ **do**
  $\mathbf{w}^k(n) = \mathbf{w}(n)$ % Disseminate global state of the network
end for

On the other hand, the clipping strategy used in 4.4 could be applied to the l-DMEFXLMS algorithm. That strategy addresses the problem of saturation in amplifiers or loudspeakers by limiting the output signal at each node k to a threshold value $y_{k_{max}}$. Thus, the clipping l-DMEFxLMS algorithm is given by

$$\begin{aligned}
&\text{if } \left|\underline{y}_k(n)\right| > y_{k_{max}}, \\
&\qquad \underline{y}_k(n) = \underline{y}_k(n)\xi_k \frac{y_{k_{max}}}{\left|\underline{y}_k(n)\right|}, \quad (0 < \xi_k < 1) \\
&\text{end if} \\
&y_k(n) = \underline{y}_k(n).
\end{aligned} \tag{5.47}$$

where $\underline{y}_k(n)$ is defined as the provisional output signal at kth node, calculated as

$$\underline{y}_k(n) = \mathbf{w}_k^T(n)[\mathbf{X}(n)]_{(:,1)}. \tag{5.48}$$

As an improvement, the re-scaling method could be applied to the clipping l-DMEFxLMS algorithm with the aim to avoid large oscillations of $\mathbf{w}_k(n)$ in (5.44).

Algorithm 4: *1r re-scaling l-DMEFxLMS algorithm for N-nodes ASN.*

for all *node* $1 \leq k \leq N$ **do**
  $\mathbf{w}_k(n) = \mathbf{w}^k(n)_{(1+L(k-1):Lk)}$ % Obtain local coefficients to generate the output signal
  $\underline{y}_k(n) = \mathbf{w}_k^T(n)[\mathbf{X}(n)]_{(:,1)}$ % Provisional output signal
  for all $1 \leq j \leq N$ **do**
    $\mathbf{u}_{jk}(n) = \mathbf{X}(n)\mathbf{s}_{jk}$ % Vector that contains reference signal filtered by estimated acoustic channels
  end for
  $\mathbf{w}^k(n) = \mathbf{w}^{k-1}(n) - \mu(\frac{\boldsymbol{\beta}}{N}\mathbf{w}^k(n-1) + \mathbf{u}_k(n)e_k(n))$ % Update local state
  if $\left|\underline{y}_k(n)\right| > y_{k_{max}}$ % If provisional output signal is greater than threshold
    $\underline{y}_k(n) = \underline{y}_k(n)\xi_k \frac{y_{k_{max}}}{\left|\underline{y}_k(n)\right|}$ % Rescale provisional output signal
    $\underline{\mathbf{w}}_k(n) = \mathbf{w}^k(n)_{(L(k-1)+1:Lk)}\ \xi_k \frac{y_{k_{max}}}{\left|\underline{y}_k(n)\right|}$ % Rescale its portion within its local state
    $\mathbf{w}^k(n)_{(1+L(k-1):Lk)} = \underline{\mathbf{w}}_k(n)$ % Updated local state with rescaled coefficients
  end if
  $y_k(n) = \underline{y}_k(n)$ % Generate output signal
end for

This leads to the re-scaling of the adaptive filters in (5.47) as follows

$$\begin{aligned}
&\text{if } \left|\underline{y}_k(n)\right| > y_{k_{max}},\\
&\quad \underline{y}_k(n) = \underline{y}_k(n)\xi_k \frac{y_{k_{max}}}{\left|\underline{y}_k(n)\right|},\\
&\quad \underline{\mathbf{w}}_k(n) = \mathbf{w}^k(n)_{(L(k-1)+1:Lk)}\ \xi_k \frac{y_{k_{max}}}{\left|\underline{y}_k(n)\right|},\\
&\quad \mathbf{w}^k(n)_{(1+L(k-1):Lk)} = \underline{\mathbf{w}}_k(n),\\
&\text{end if}\\
&y_k(n) = \underline{y}_k(n).
\end{aligned} \tag{5.49}$$

Algorithm 3 illustrates a summary of the re-scaling l-DMEFxLMS algorithm pseudocode, which are executed per sample time at each node. Comparing (5.47) and (5.49), it can be seen that, while the clipping l-DMEFxLMS algorithm only re-scales the output, the re-scaling l-DMEFxLMS algorithm re-scales both the output and the filter coefficients. The dual re-scaling prevents stability problems since the coefficients update is uncorrelated with the filter output when the clipping strategy is working [104] [113]. It is important to take into account that the system stability is ensured by applying the suitable constraints over the output signal, although too restrictive saturation levels could result in performance impairments. Furthermore, the re-scaling l-DMEFxLMS algorithm requires a higher data transfer speed compared to the l-DMEFxLMS algorithm. This result from the fact that $3(N-1)$ coefficients are exchanged among the nodes (see Figure 5.5.a): a first round where each node passes the global state of the network at the previous iteration, $\mathbf{w}(n-1)$, to the following node; a second round where each node receives the information of the previous node, $\mathbf{w}^{k-1}(n)$, calculates its local version $\mathbf{w}^k(n)$ with the help of $\mathbf{w}(n-1)$, and supplies to the following node its partial result; and finally, a third round where each node

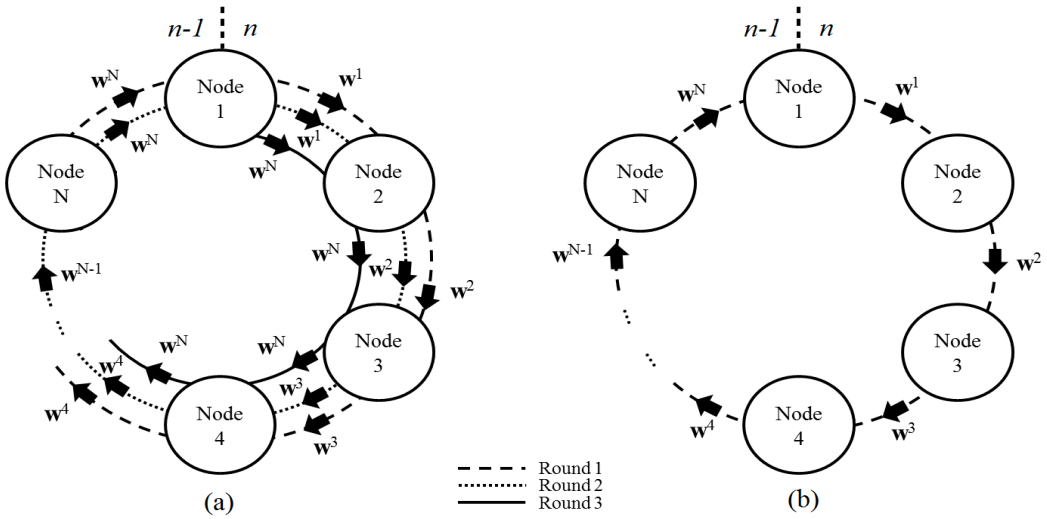


Figure 5.5: Two ring topology distributed ASNs with incremental communication using (a) the re-scaling l-DMEFxLMS algorithm and (b) the 1r re-scaling l-DMEFxLMS algorithm. Data transfer rounds are represented with different type of lines.

rescales its portion within $\mathbf{w}^k(n)$, passes its local state to the following node and generates its output signal with the rescaled coefficients. Thus, the last node will obtain the updated global state of the network properly rescaled which will be necessary at the time of dissemination to the rest of the nodes in the first round of the following iteration. Note that, in the last round, each node may supplies only its rescaled portion directly to the last node. However, it would be necessary a different commnunication strategy to fulfill this solution.

With the aim to reduce the communication demands of this algorithm, a one-round strategy is proposed (1r re-scaling l-DMEFxLMS). Using this method, in one round each node receives the information from its precedent node, $\mathbf{w}^{k-1}(n)$, calculates its local version $\mathbf{w}^k(n)$ with the help of its information at the previous iteration, $\mathbf{w}^k(n-1)$, and supplies to the following node its partial result with its portion previously rescaled. At the same time, each node generates its rescaled output signal. Note that, in this case, instead of using the global state of the network at the previous iteration in the updating equation, each node uses its local state at the previous iteration, $\mathbf{w}^k(n-1)$. Thus, the dissemination of the updated coefficients is avoided and it is possible to reduce the data transfer among the nodes making $(N-1)$ interchanges of the filter coefficients, as it can be seen in Figure 5.5.b. ***Algorithm 4*** illustrates a summary of the 1r re-scaling l-DMEFxLMS algorithm pseudocode, which are executed per sample time at each node. Note that, since the updated coefficients are

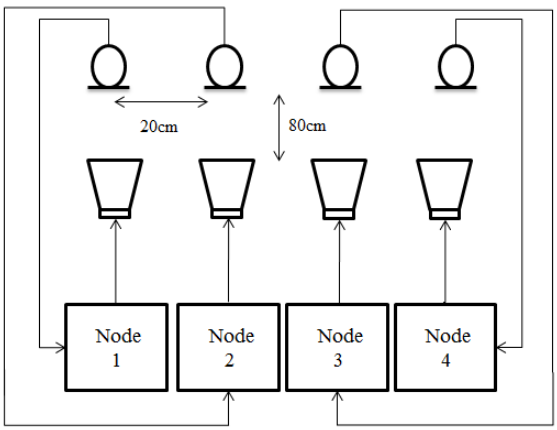


Figure 5.6: ASN of four nodes for an ANC system.

not fully shared among the nodes, the behavior of the 1r re-scaling l-DMEFxLMS algorithm is not exactly the same as the re-scaling approach. Due to the coefficients exchange is carried out in a consecutive order, those differences may result more relevant in larger ASN. This is a consequence of the incremental learning where each node computes a part of the global filter, aggregates it to the given filter and passes it to the following node. In this way, nodes closer to the last node have a more accurate estimation of the global filter. This could deteriorate the convergence speed of the 1r re-scaling l-DMEFxLMS algorithm in comparison to the re-scaling l-DMEFxLMS strategy. However, in the case of ASN composed of few nodes, boths methods will obtain a similar performance in terms of convergence speed and noise reduction. A performance comparison between all the proposed algorithms is presented in the following section.

5.3.2 Simulation results

In this section, we present the simulations carried out to evaluate the performance of the presented distributed algorithms over networks with no communication constraints. In a first stage, we have compared the performance of both the l-CMEFxLMS and l-DMEFxLMS algorithms in order to validate the theoretical solution outlined in (5.44). In a second stage, we have justified the use of the constrained techniques comparing the performance of the DMEFxLMS algorithm and its leaky version (l-DMEFxLMS). Subsequently, we have validated the need of using the clipping and re-scaling methods in order to fulfill the loudspeakers output constraint as well as the use of the proposed one-round strategy (1r re-scaling l-DMEFxLMS) to reduce the communication demands of the network. To this end, we have compared the l-

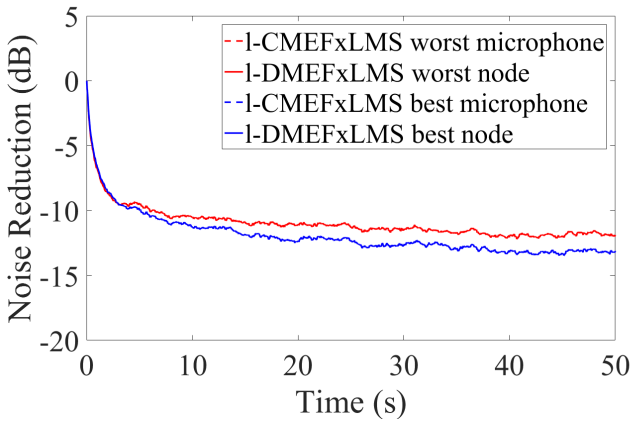


Figure 5.7: Noise reduction of the distributed system (solid line) with four single-channel nodes and the centralized system (dashed line) with a *1:4:4* configuration with β_k=0.01 represented for the best and worst microphone.

DMEFxLMS, the clipping l-DMEFxLMS, the re-scaling l-DMEFxLMS and the 1r re-scaling l-DMEFxLMS algorithms in terms of final noise reduction, convergence behavior, computational complexity and communication requirements. All the algorithms have been tested in a homogeneous ASN composed of four single-channel nodes, as shown in Figure 5.6. Only the nodes with the best and the worst performance are shown in the simulations in order to assess the behavior of the ASN. The performance of the other nodes remains within this range. For the designed ASN, we have considered the simulation parameters setting explained in Section 3.3. An initial step size parameter of μ=0.0125 as the highest value that ensures the stability of the algorithms, has been used in the first simulations. Furthermore, we have considered a maximum allowed value of the output signals of $y_{k_{max}}$=1.0 and an attenuation parameter of ξ_k=1 for all the nodes and all the simulations. Initially, we use a leakage parameter of β_k=0.01 for all the nodes. The reason of this leakage value is explained below.

5.4.1.1. Comparison between centralized and distributed leaky approaches

In the first simulation, we compare the noise reduction of a leaky centralized ANC system with one reference signal, four actuators and four sensors (*1:4:4* configuration) and a leaky distributed ANC system with 4 single-channel nodes. Figure 5.7 shows the time evolution of the $NR_k(n)$ for both the l-CMEFxLMS and the l-DMEFxLMS algorithms for the microphone with best and worst performance in the centralized case and the node with the best and worst performance in the distributed case. As expected, the distributed implementation exhibits exactly the same results

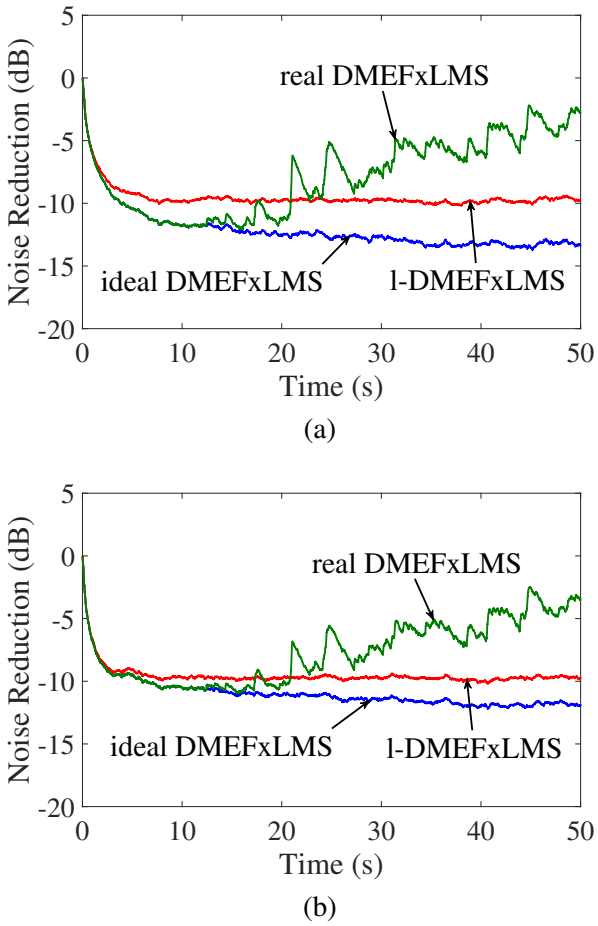
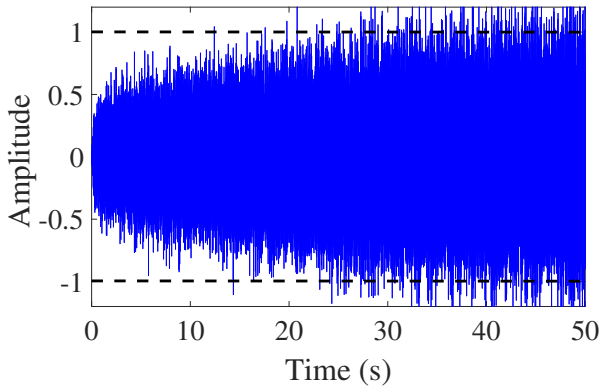


Figure 5.8: Noise reduction obtained for the ideal DMEFxLMS, the real DMEFxLMS and the l-DMEFxLMS algorithms using a four-node ASN with β_k=0.01 at the nodes (a) with the best performance and (b) with the worst performance.

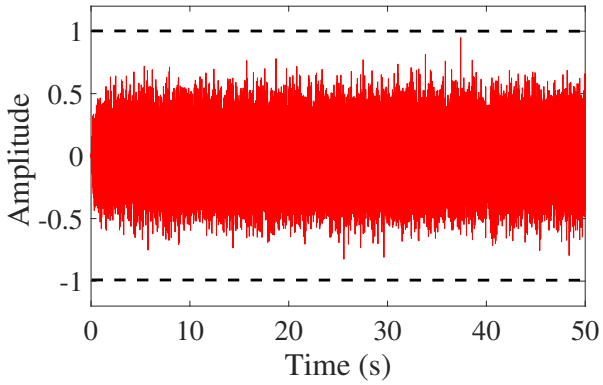
as the centralized version in terms of convergence speed and final residual noise, providing an attenuation up to 12 dB for the worst node and almost 14 dB for the best node.

5.4.1.2. Improving the performance of the DMEFxLMs in practical scenarios

In the second simulation, we compare the performance of the DMEFxLMS algorithm with the l-DMEFxLMS strategy in order to justify its use in practical scenarios. As the previous simulation, an ANC system over a 4-single-channel-node ASN has been considered. Figure 5.8 shows the $NR_k(n)$ for both algorithms at the nodes with



(a)



(b)

Figure 5.9: Time evolution of the control signals for both the (a) DMEFxLMS and (b) l-DMEFxLMS algorithms at the worst node. The threshold is represented by dashed lines.

the best and the worst performance. Two behaviors of the DMEFxLMS algorithm have been differentiated: its performance in a real scenario (denoted as real DMEFxLMS) and its performance in an ideal scenario (denoted as ideal DMEFxLMS). The real version emulates how the loudspeaker saturation influences the behaviour of the distributed algorithm. On the other hand, the effect of the loudspeaker saturation is not considered in the ideal version. The time variation of the control signal for both algorithms at the worst node are shown in Figure 5.9.(a) and Figure 5.9.(b), respectively. The amplitude of the control signals is normalized being $+1$ and -1 the maximum and the minimum signal amplitudes that can be produced by the loud-

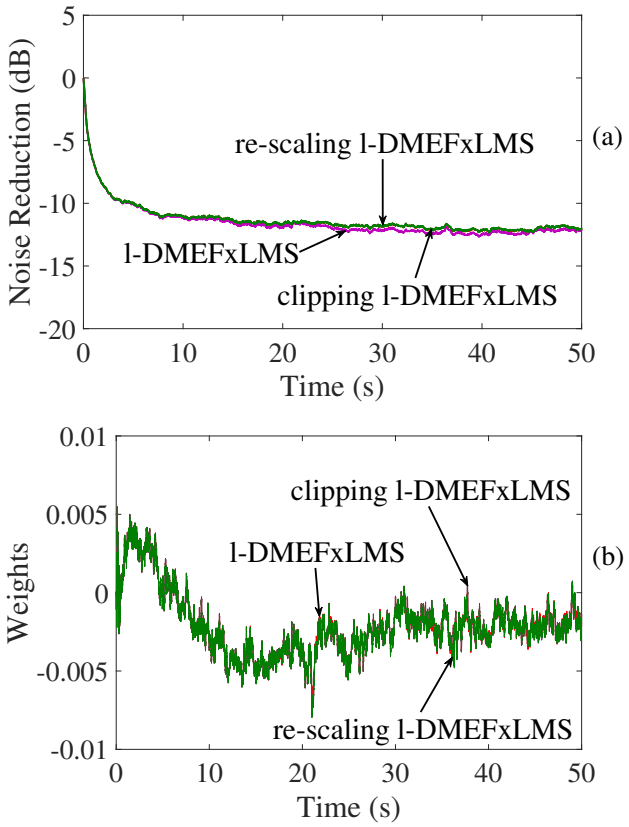


Figure 5.10: Behavior of the l-DMEFxLMS, the clipping l-DMEFxLMS and the re-scaling l-DMEFxLMS algorithms in a four-node ASN with β_k=0.001 at the best node: (a) time evolution of the noise reduction obtained and (b) time evolution of the first filter coefficient.

speakers. It can be shown that both the ideal and the real DMEFxLMS behaviors initially outperform the l-DMEFxLMS algorithm in terms of convergence speed and noise reduction but when they reach certain point, the real version turns unstable and does not converge. This is because the DMEFxLMS method fails to satisfy the control output constraint as shown in Figure 5.9.(a). Consequently, the appearance of non-linearities due to the loudspeakers saturation leads to the divergence of the DME-FxLMS algorithm. To avoid this, we use the l-DMEFxLMS algorithm with β_k=0.01 for all the nodes selected by trial and error. Thus, the l-DMEFxLMS algorithm fulfills the constraint by limiting the maximum output signal (see Figure 5.9.(b)), achieving a $NR_k(n)$ around 10 dB for the best and the worst node, as it can be seen in Figure

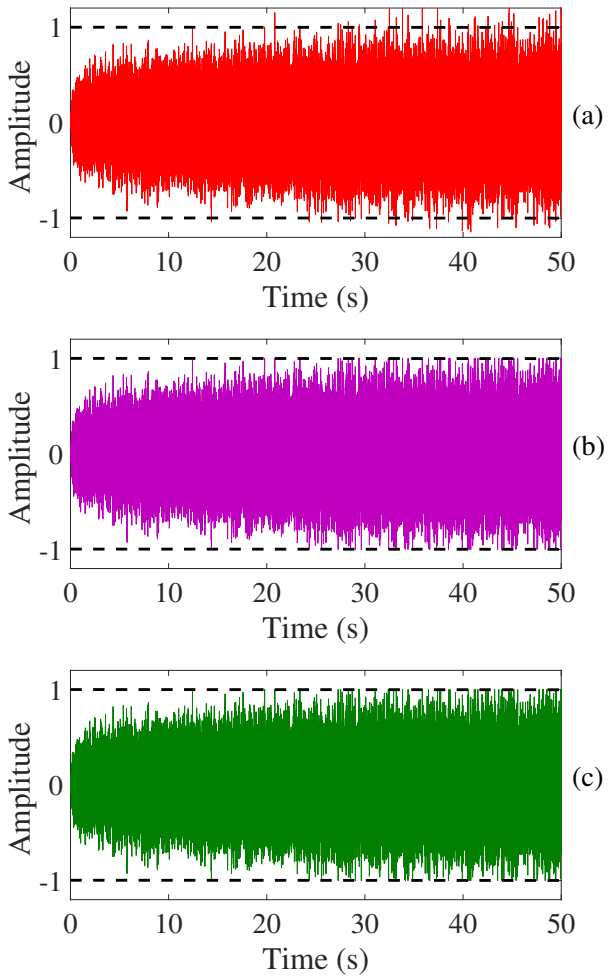


Figure 5.11: Time evolution of the control signals for the (a) l-DMEFxLMS algorithm, (b) clipping l-DMEFxLMS algorithm and (c) re-scaling l-DMEFxLMS algorithm at the best node. The threshold is represented by dashed lines.

5.8. For simplicity, and since the use of the leaky method in real scenarios has been justified, the effect of loudspeakers saturation on the performance of the leaky algorithms has not been considered in the next simulations.

5.4.1.3. Comparison between distributed leaky approaches

Furthermore, it is possible to improve the performance of the l-DMEFxLMS algorithm, in terms of final attenuation, fulfilling the output signal constraint addressed

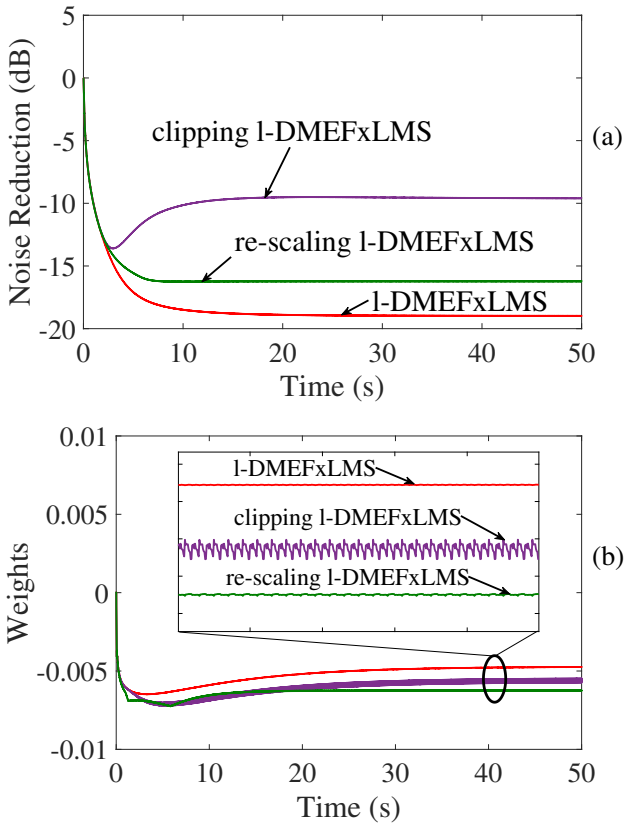


Figure 5.12: Behavior of the l-DMEFxLMS, the clipping l-DMEFxLMS and the re-scaling l-DMEFxLMS algorithms in a four-node ASN with β_k=0.01 at the best node: (a) time evolution of the noise reduction obtained and (b) time evolution of the first filter coefficient.

by the clipping and the re-scaling l-DMEFxLMS algorithms. This can be seen in Figure 5.10 and Figure 5.11 where the leakage parameter β_k was selected as 0.001 for all nodes of a 4-node ASN, in order to increase the noise reduction of the leaky methods. Note that for β_k=0.01, we will obtain the same results as the previous simulation since the leaky algorithms keep their output signal powers below the allowed threshold, $y_{k_{max}}$, as it was described in (5.47) and (5.49). As the behavior of the four nodes of the network is similar, only the results obtained for the node with the best performance have been shown. As shown in Figure 5.10.(a), all algorithms present similar noise attenuation levels providing 12 dB of noise reduction in the three cases. Regarding the convergence behavior of the first adaptive filter coefficient, plotted in

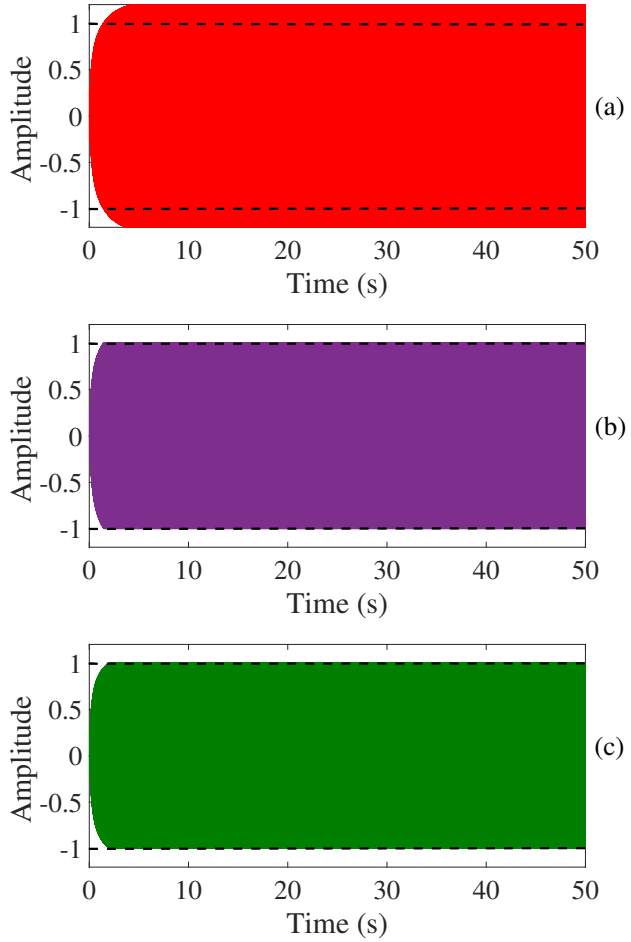


Figure 5.13: Time evolution of the control signals for the (a) l-DMEFxLMS algorithm, (b) clipping l-DMEFxLMS algorithm and (c) re-scaling l-DMEFxLMS algorithm at the best node. The threshold is represented by dashed lines.

Figure 5.10.(b), it can be observed that is almost identical for the three methods. However, the l-DMEFxLMS algorithm, as seen from Figure 5.11.(a), fails to satisfy the output constraint requirement at certain time instants which may lead to system unstability, as justified in the previous simulation. On the contrary, the output signal level of both clipping and re-scaling l-DMEFxLMS algorithms is under the threshold fixed by the constraint, as it is illustrated by Figures 5.11.(b) and 5.11.(c).

5.4.1.4. Experiments with periodic noise as input signal

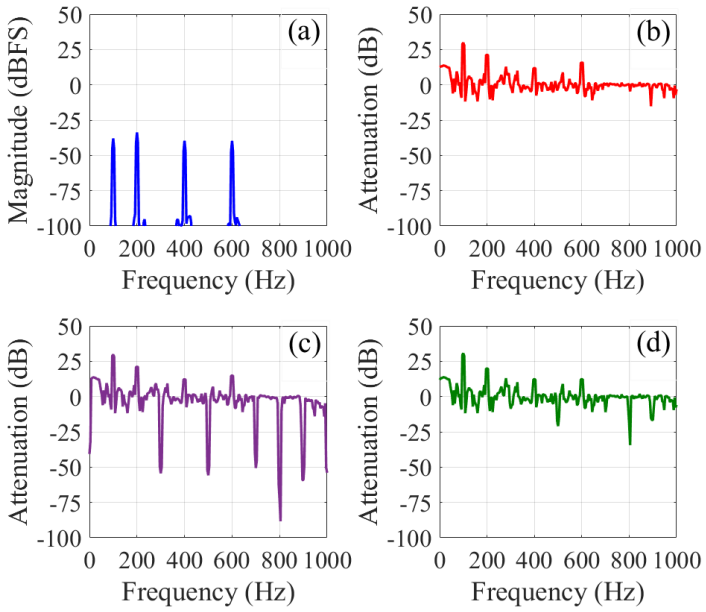


Figure 5.14: (a) Power spectrum of the error signal for the ANC off and attenuation reached by the (b) l-DMEFxLMS algorithm, (c) clipping l-DMEFxLMS algorithm and (d) re-scaling l-DMEFxLMS algorithm compared to (a) at the best node of the four-node ASN.

On the other hand, the clipping algorithm can lead to system unstability as a result of the unpredictable behavior of its adaptive filter coefficients [104]. This can be observed in the next simulation where a periodic noise with 100 Hz, 200 Hz, 400 Hz and 600 Hz components has been used as disturbance signal. As the previous case, Figures 5.12-5.13-5.14 show only the results obtained for the best node of a four-node ASN. In this case, the highest value of the step size parameter that ensures the stability of the algorithms was set to μ=0.005. Regarding the leakage parameter, a value of β_k=0.01 for all the nodes was selected. Note that a higher value of β_k would maintain the output signal power of the l-DMEFxLMS algorithm below the threshold (see Figure 5.13.(a)) obtaining a similar performance for the three algorithms. However, and as it can be seen in the previous simulations, the smaller leakage parameter, the larger noise attenuation.

Under these conditions, as shown in Figure 5.12.(a), both the re-scaling l-DMEFxLMS and the l-DMEFxLMS algorithms show a stable behavior providing an attenuation up to 16 dB for the first one and almost 20 dB for the second one. However, as the previous simulation, note that the control output constraint of the l-DMEFxLMS algo-

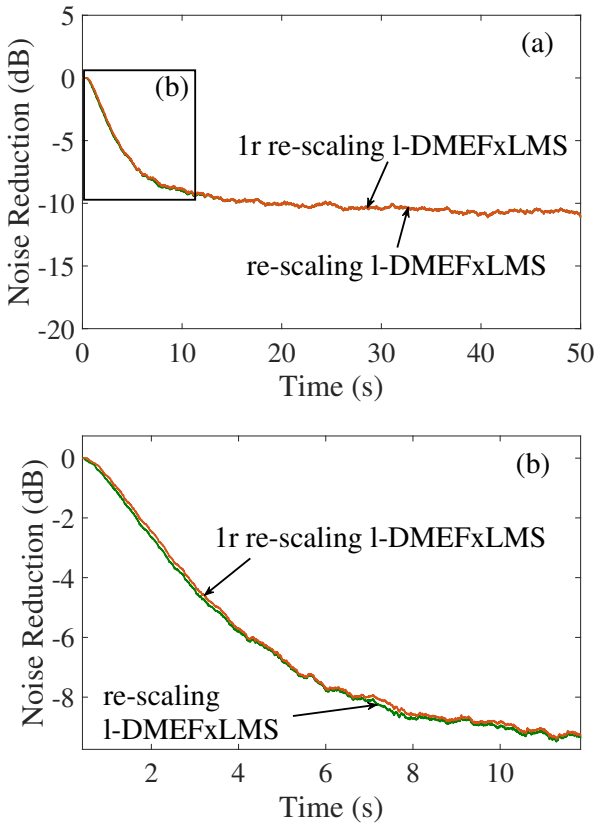


Figure 5.15: Noise reduction obtained for both the re-scaling and the 1r re-scaling l-DMEFxLMS algorithms using a four-node ASN at the node with the best performance with β_k=0.001. To observe (a) both transient and steady state or (b) only the transient state.

rithm was exceeded, as shown in Figure 5.13.(a), while the re-scaling l-DMEFxLMS strategy satisfies the output constraint (see Figure 5.13.(c)). Although the clipping l-DMEFxLMS method fulfills the maximum output signal constraint (see Figure 5.13.(b)), it achieves the worst performance in terms of final residual noise, obtaining a $NR_k(n)$ of 10 dB for the best node. As it can be seen in Figure 5.12.(b) and regarding the time evolution of the first filter coefficient, while both, the l-DMEFxLMS and the re-scaling l-DMEFxLMS algorithms, exhibit a stable convergence, the coefficient of the clipping l-DMEFxLMS algorithm presents a significant oscillation. Although this behavior does not lead to system unstability, it can result in the appearance of undesired frequency components at the spectrum of the error signal. The magnitude of

Table 5.1: Total number of multiplications (MUX) and data transfer per iteration regarding (1) the computational complexity and (2) the communication requirements of the algorithms, respectively. L: length of the adaptive filters; M: length of the estimated acoustic paths; N: number of nodes. As example, some typical cases considering L=150, M=256 and N=2, 4 and 8 nodes, have been evaluated.

	Algorithms	Generic	N = 2	N = 4	N = 8
(1) **Computational complexity (MUX)**	l-DMEFxLMS	$LN(1+LN)+MN+L+1$	90963	361775	1443399
	clipping l-DMEFxLMS	$LN(1+LN)+MN+L+2$	90964	361776	1443400
	re-scaling l-DMEFxLMS	$LN(2+LN)+MN+L+1$	91263	362376	1444600
	1r re-scaling l-DMEFxLMS	$LN(2+LN)+MN+L+1$	91263	362376	1444600
(2) **Communication requirement (data transfer)**	l-DMEFxLMS	$2LN(N-1)$	600	3600	16800
	clipping l-DMEFxLMS	$2LN(N-1)$	600	3600	16800
	re-scaling l-DMEFxLMS	$3LN(N-1)$	900	5400	25200
	1r re-scaling l-DMEFxLMS	$LN(N-1)$	300	1800	8400

the power spectral density of the error signal when the ANC system is off compared to the attenuation obtained by the three leaky strategies are represented in Figure 5.14. It can be seen from Figure 5.14.(b), 5.14.(c) and 5.14.(d), that, all the methods, the l-DMEFxLMS algorithm, the clipping l-DMEFxLMS algorithm and the re-scaling l-DMEFxLMS algorithm, reduce the noise at the frequencies of interest, 100 Hz, 200 Hz, 400 Hz and 600 Hz, obtaining an attenuation of almost 30 dB, 20 dB, 12 dB and 14 dB, respectively. It can be observed that the re-scaling l-DMEFxLMS algorithm introduces undesired noise at 500 Hz and 800 Hz, as can be seen in Figure 5.14.(d). However, the clipping l-DMEFxLMS algorithm introduces much more additional harmonics, significantly at high-frequencies, as shown in Figure 5.14.(c). At least, five new harmonics appear at the frequencies 300 Hz, 500 Hz, 700 Hz, 800 Hz and 900 Hz, achieving more than 50 dB in all of them. This may lead to the unstability of the ANC system in real scenarios, probably caused by the strong saturation applied to the control signal (see Figure 5.13.(b)). Since the l-DMEFxLMS algorithm does not ensure fulfillment of the constraint and the clipping l-DMEFxLMS algorithm may present some potential problems of stability, the re-scaling l-DMEFxLMS algorithm exhibits the best overall performance, providing both the higher noise reduction and the system stability and fulfilling the output signal constraint.

5.4.1.5. Computational complexity and communication requirements

Table 5.1 compares the computational complexity (in terms of multiplications per iteration) and the communication requirements (data transfer) of the distributed leaky ANC algorithms. The transmitted filter coefficients will be proportionally related to the transmitted bits depending on the used coding. The l-CMEFxLMS algorithm has not been included in the table because it has exactly the same computational complexity as the l-DMEFxLMS algorithm. To this end, we consider a

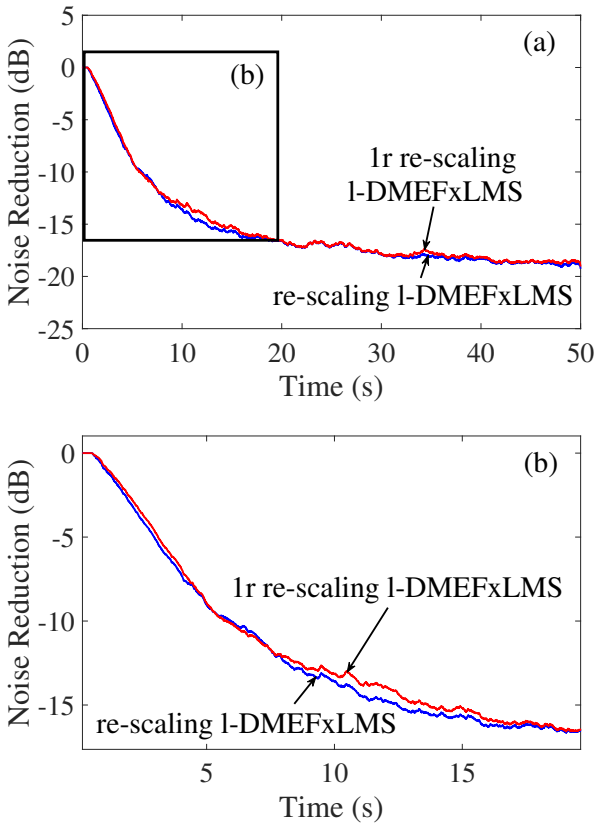


Figure 5.16: Noise reduction obtained for both the re-scaling and the 1r re-scaling l-DMEFxLMS algorithms using a sixteen-node ASN at the node with the best performance with β_k=0.001. To observe (a) both transient and steady state or (b) only the transient state.

network of N single-channel nodes. For simplicity, we assume that each node has access to $x(n)$ through an alternative broadcast channel. Therefore, the reference signal has not been considered in the calculation of the data transfer. Note that, for all the algorithms, the computational complexity depends on L, M and N while the communication requirements only depends on L and N. Then, both implementation aspects are particularized for N=2, N=4 and N=8. Results show that the computational cost of the re-scaling versions is slightly higher than the l-DMEFxLMS and clipping l-DMEFxLMS algorithms. Although there are no significant differences between them, it is important to take into account that the computational complexity of all l-DMEFxLMS strategies increases significatively with the number of nodes. Re-

garding the communication needs, the re-scaling l-DMEFxLMS strategy has higher requirements as shown in Table 5.1. For an incremental-learning N-node network, the re-scaling l-DMEFxLMS method needs that every node transfers $LN\times1$ coefficients to the following node $3(N-1)$ times in each iteration. However, using the 1r re-scaling l-DMEFxLMS algorithm, a data stream of $LN(N-1)$ samples is propagated through the nodes, reducing the data transfer of the network. Note that, as expected, the same relation is maintained as the number of nodes increases. Following a practical example similar as defined in 5.2.3, note that the clipping and re-scaling l-DMEFxLMS algorithms would need a transfer rate of 16.1 and 24.2 megabytes per second (MBps) respectively, on an incremental four-node ASN. However, the 1r re-scaling l-DMEFxLMS algorithm would need a transfer rate of at least 8.1 MBps. Therefore, in the last case, we could use a network of 10 MBps to perform the required data transfer among the nodes.

5.4.1.6. Comparison between re-scaling approaches

Finally, to compare the behavior of the two re-scaling strategies, Figure 5.15 shows the time evolution of the $NR_k(n)$ for the re-scaling l-DMEFxLMS algorithm and the 1r re-scaling l-DMEFxLMS algorithm for the node with the best performance of a four-node ASN. We have considered, as disturbance signal, the wideband noise used at the first simulations as well as a step size parameter of μ=0.001 and a leakage parameter of β_k=0.001 for all the nodes. As it can be seen in Figure 5.15.(a), the 1r re-scaling l-DMEFxLMS algorithm presents similar results as the re-scaling l-DMEFxLMS method in terms of final residual noise providing an attenuation up to 11 dB. As commented in Section 2, to avoid the dissemination of the update coefficients would affect the convergence speed of the 1r re-scaling l-DMEFxLMS algorithm. However, differences between both strategies are negligible, as it can be seen from Figure 5.15.(b). While in this case, the 1r re-scaling l-DMEFxLMS strategy does not introduce strong degradation in the performance of the re-scaling l-DMEFxLMS algorithm, it should be noted that in other cases, such as larger ASN, the differences could be more relevant. To this end, in Figure 5.16, it can be seen the time evolution of the $NR_k(n)$ for the re-scaling l-DMEFxLMS algorithm and the 1r re-scaling l-DMEFxLMS algorithm for the node with the best performance of a sixteen-node ASN. In this case, differences between boths algorithms, in terms of convergence speed, are slightly larger than the previous simulation, as it is shown in Figure 5.16.(b). Differences in the result of the algorithms may become more evident if we use an increased numbers of nodes (maybe as many nodes as filter coefficients) or we work by block of samples. Since the 1r re-scaling l-DMEFxLMS algorithm exhibits similar behavior to the re-scaling l-DMEFxLMS algorithm, their results have not been included in the previous simulations. Note that all the results accomplished in this work depend on particular settings and parameter configuration.

5.3.3 Conclusions

In this section, several control effort strategies have been implemented on a distributed ANC system over an ideal ASN using a collaborative incremental strategy. For this purpose, four new approaches have been derived from the DMEFxLMS to ensure ANC system stability in practical scenarios. Results show that the distributed version of the leaky MEFxLMS (l-DMEFxLMS) algorithm exhibits the same performance as its leaky centralized version (l-CMEFxLMS) when there are no communication constraints in the network. We have carried out simulations to show the performance of the distributed leaky approaches in different scenarios. It has been justified the use of the l-DMEFxLMS algorithm in practical cases at the expense of worsening the system performance. In this regard, an appropiate leakage parameter value is required to reduce the performance degradation: high enough to fulfill the constraint but also low enough to obtain a good noise reduction. It has been demonstrated that both the clippling and the rescaling l-DMEFxLMS algorithms allow us to use low leakage values keeping the output constraint controlled. In addition, attenuation noise levels close to those obtained by the DMEFxLMS algorithm in an ideal scenario have been achieved for both methods. Due to the stability problems of the clipping l-DMEFxLMS strategy, the use of the re-scaling l-DMEFxLMS algorithm is more promising in most cases. A simplified version of the re-scaling l-DMEFxLMS algorithm is proposed, called 1r re-scaling l-DMEFxLMS algorithm, which aims to reduce the communications demands of the re-scaling l-DMEFxLMS method in ASN. Moreover, the computational complexity of the distributed leaky algorithms has been reported, bearing in mind that the computational capability of the nodes should be considered in practical scenarios. Regarding the communication requirements, the proposed 1r re-scaling l-DMEFxLMS strategy achieves a noticeable data transfer saving maintainig similar performance to the re-scaling l-DMEFxLMS algorithm. Therefore, it has been shown that the proposed 1r re-scaling l-DMEFxLMS algorithm achieves a good control effort with a computational cost similar to the other strategies but reducing significantly the communication requirements of the network.

5.4 Collaborative distributed algorithm based on affine projection approach

The Affine Projection (AP) algorithms [43, 47] have been presented as an efficient alternative to the conventional LMS algorithms since they improve the LMS convergence speed maintaining robustness and stability. Speed convergence improvement depends on the increase of the projection order (Q), which produces a negative impact on the computational cost. Several variants have been introduced with the aim of improving the computational performance of the AP algorithm [118]. Recently,

a distributed approach of the multichannel AP algorithm has been proposed in [119] with the aim to provides a significant computational saving in comparison with its centralized version as well as to outperform the DMEFxLMS algorithm in terms of convergence speed and noise attenuation. In addition, two affine-projection-like (APL) algorithms that avoid matrix inversion have been introduced in [114]. We focus on the development of a distributed version of the affine-projection-like-I (APL-I) algorithm since it requires a lower processing time to converge.

5.4.1 DFxAPL-I algorithm

As in the previous distributed algorithms, the multichannel version of the AP algorithm based on the filtered-x strategy (FxAP) will be obtained from (3.22) and then a distributed approach will be proposed.

We assume now the affine projection approach considering ϵ_k=0 and both $\boldsymbol{\beta}$ and $\boldsymbol{\Gamma}$ are null matrices. Bearing in mind (5.2) and the weighting matrix $\mathbf{S}(n)$ presented in (3.34), we define $\mathbf{S}_{jk}(n)$=$[\hat{\mathbf{U}}_k^T(n)\hat{\mathbf{U}}_j(n) + \delta \boldsymbol{I}_Q]^{-1}$ where $\boldsymbol{I}_Q$ is an identity matrix of size $[Q\times Q]$. Therefore, the cost function (3.22) is redefined as

$$J(n) = (\mathbf{d}(n) + \hat{\mathbf{U}}(n)\mathbf{S}(n)\mathbf{w}(n))^T(\mathbf{d}(n) + \hat{\mathbf{U}}(n)\mathbf{S}(n)\mathbf{w}(n)) \tag{5.50}$$

where $\mathbf{S}(n)$=$[\hat{\mathbf{U}}^T(n)\hat{\mathbf{U}}(n)+\delta\mathbf{I}_{QN}]^{-1}$ with $\boldsymbol{I}_{QN}$ as an identity matrix of size $[QN\times QN]$. Therefore, the filter updating equation of the FxAP algorithm based on its multichannel centralized version is then calculated as

$$\mathbf{w}(n+1) = \mathbf{w}(n) \ - 2\mu\hat{\mathbf{U}}(n)[\hat{\mathbf{U}}^T(n)\hat{\mathbf{U}}(n) + \delta\mathbf{I}_{QN}]^{-1}\ \mathbf{e}(n)\cdot \tag{5.51}$$

In order to reduce the computational burden of the AP algorithm, the affine projection like-I strategy can be considered. The global solution of the filtered-x affine projection like-I (FxAPL-I) algorithm can be derived from (5.50) but considering $\mathbf{S}(n)$ as a weighting matrix of size $[N\times N]$ defined as

$$\mathbf{S}(n)=\begin{bmatrix} S_{11}(n) & S_{12}(n) & \cdots & S_{1N}(n) \\ S_{21}(n) & S_{22}(n) & \cdots & S_{2N}(n) \\ \vdots & \vdots & \ddots & \vdots \\ S_{N1}(n) & S_{N2}(n) & \cdots & S_{NN}(n) \end{bmatrix}, \tag{5.52}$$

with $S_{jk}(n)$ as a scalar value given by

$$S_{jk}(n)=\frac{\|\hat{\mathbf{U}}_{jk}(n)\hat{\mathbf{e}}_k(n)\|^2}{\|\hat{\mathbf{U}}_{jk}(n)^T\hat{\mathbf{U}}_{jk}(n)\hat{\mathbf{e}}_k(n)\|^2}\cdot \tag{5.53}$$

Therefore, the cost function (5.50) is redefined as

$$J(n) = (\mathbf{d}(n) + [\hat{\mathbf{U}}]_c(n)\mathbf{w}(n))^T(\mathbf{d}(n) + [\hat{\mathbf{U}}]_c(n)\mathbf{w}(n)) \tag{5.54}$$

where $[\hat{\mathbf{U}}]_c(n)$ is now arranged as

$$[\hat{\mathbf{U}}]_c(n) = \begin{bmatrix} \hat{\mathbf{U}}_{11}(n)S_{11}(n) & \hat{\mathbf{U}}_{21}(n)S_{21}(n) & \cdots & \hat{\mathbf{U}}_{1N}(n)S_{1N}(n) \\ \hat{\mathbf{U}}_{21}(n)S_{21}(n) & \hat{\mathbf{U}}_{22}(n)S_{22}(n) & \cdots & \hat{\mathbf{U}}_{2N}(n)S_{2N}(n) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathbf{U}}_{N1}(n)S_{N1}(n) & \hat{\mathbf{U}}_{N2}(n)S_{N2}(n) & \cdots & \hat{\mathbf{U}}_{NN}(n)S_{NN}(n) \end{bmatrix} \tag{5.55}$$

Finally, the global updating equation of the FxAPL-I algorithm is obtained as

$$\mathbf{w}(n+1) = \mathbf{w}(n) \; - \sum_{k=1}^{N} \hat{\mathbf{U}}_{k,S}(n)\hat{\mathbf{e}}_k(n), \tag{5.56}$$

where $\hat{\mathbf{U}}_{k,S}(n)$=$[\hat{\mathbf{U}}_{1k,S}(n)\ \hat{\mathbf{U}}_{2k,S}(n)\ldots\hat{\mathbf{U}}_{Nk,S}(n)]$ of size $[LN\times Q]$ is composed by N matrices $\hat{\mathbf{U}}_{jk,S}(n)$=$\hat{\mathbf{U}}_{jk}(n)S_{jk}(n)$. Note that, to obtain $[\hat{\mathbf{U}}]_c(n)$ as arranged in (5.55), a specific configuration of the elements which composed (3.25) must be required. In this regard, I'=I=N with $\mathbf{A}$ as an identity matrix as depicted in (5.1). The matrix $\boldsymbol{\Theta}_{i'i}$ must be redefined as $\boldsymbol{\Theta}_i$, i.e., a $[LN\times LN]$ diagonal matrix with only a non-zero matrix at the $(1{+}L(i{-}1){:}iL, 1{+}L(i{-}1){:}iL)$ position, $\boldsymbol{\upsilon}_{ii}$, which is defined as an identity matrix of size $[L\times L]$,

$$\boldsymbol{\Theta}_i{=}\begin{bmatrix} \mathbf{0}_L & \cdots & \mathbf{0}_L & \cdots & \mathbf{0}_L \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_L & \cdots & \boldsymbol{\theta}_{ii} & \cdots & \mathbf{0}_L \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_L & \cdots & \mathbf{0}_L & \cdots & \mathbf{0}_L \end{bmatrix}, \qquad \boldsymbol{\theta}_{ii}{=}\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \tag{5.57}$$

In addition, matrices $\boldsymbol{\Upsilon}_i$, $\boldsymbol{\Psi}_{i'i}$ and $\boldsymbol{\phi}_{i'}$ must be defined as follows

$$\boldsymbol{\Upsilon}_i{=}\begin{bmatrix} \mathbf{0}_L & \cdots & \mathbf{0}_L & \cdots & \mathbf{0}_L \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_L & \cdots & \boldsymbol{\upsilon}_{ii} & \cdots & \mathbf{0}_L \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_L & \cdots & \mathbf{0}_L & \cdots & \mathbf{0}_L \end{bmatrix}, \qquad \boldsymbol{\upsilon}_{ii}{=}\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \tag{5.58}$$

$$\boldsymbol{\Psi}_{i'i}{=}\begin{bmatrix} \mathbf{0}_Q & \cdots & \mathbf{0}_Q & \cdots & \mathbf{0}_Q \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_Q & \cdots & \boldsymbol{\psi}_{i'i} & \cdots & \mathbf{0}_Q \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_Q & \cdots & \mathbf{0}_Q & \cdots & \mathbf{0}_Q \end{bmatrix}, \qquad \boldsymbol{\psi}_{i'i}{=}\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \tag{5.59}$$

$$\boldsymbol{\phi}_{i'}=\begin{bmatrix}\mathbf{0}_Q & \cdots & \mathbf{0}_Q & \cdots & \mathbf{0}_Q\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ \mathbf{0}_Q & \cdots & \boldsymbol{\phi}_{i'i'} & \cdots & \mathbf{0}_Q\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ \mathbf{0}_Q & \cdots & \mathbf{0}_Q & \cdots & \mathbf{0}_Q\end{bmatrix}, \qquad \boldsymbol{\phi}_{i'i'}=\begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & 1 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 1\end{bmatrix}. \tag{5.60}$$

where $\boldsymbol{\Upsilon}_i$ is similarly defined as (4.1), the $[QN\times QN]$ matrix $\boldsymbol{\Psi}_{i'i}$ is designed as a diagonal matrix with only a non-zero matrix at the $(1+L(i'-1):i'L, 1+L(i-1):iL)$ position, $\boldsymbol{\psi}_{i'i}$, which is defined as an identity matrix of size $[Q\times Q]$ and $\boldsymbol{\phi}_{i'}$ is a diagonal matrix of size $[QN\times QN]$ with only an $[Q\times Q]$ identity matrix at the $(1+L(i'-1):i'L, 1+L(i'-1):i'L)$ position, $\boldsymbol{\phi}_{i'i'}$.

Note that the coefficient update equation (5.56) can be expressed for a single-channel node ASN composed by one loudspeaker and one microphone ($N=1$) as follows

$$\mathbf{w}_1(n+1) = \mathbf{w}_1(n) - \frac{\|\hat{\mathbf{U}}_1(n)\mathbf{e}_1(n)\|^2}{\|\hat{\mathbf{U}}_1^T(n)\hat{\mathbf{U}}_1(n)\mathbf{e}_1(n)\|^2}\hat{\mathbf{U}}_1(n)\mathbf{e}_1(n), \tag{5.61}$$

where $\hat{\mathbf{U}}_1(n)=[\mathbf{u}_{1k}(n)\ \mathbf{u}_{1k}(n-1)\cdots\ \mathbf{u}_{1k}(n-Q+1)]$ is a matrix of size $[L\times Q]$ that contains the last Q vectors of $\mathbf{U}_{1k}(n)$ (where $k=1$). Equation 5.61 is an approximate version of the FxAP algorithm, whose single-channel node version of (5.51) is given by

$$\mathbf{w}_1(n+1) = \mathbf{w}_1(n) - 2\mu\hat{\mathbf{U}}_1(n)[\hat{\mathbf{U}}_1^T(n)\hat{\mathbf{U}}_1(n)+\delta\mathbf{I}_Q]^{-1}\,\mathbf{e}_1(n) \tag{5.62}$$

Figure 5.17 illustrates the convergence behavior of both, the FxAP (solid lines) and its approximate version, the FxAPL-I (dashed lines) algorithms. Different values of the projection order and the maximum value of the step-size parameter that assures algorithm stability have been considered ($\mu=0.1$ for the FxAP algorithm and $\mu=3\cdot10^{-2}$ for the FxAPL-I algorithm). As expected, in both cases, as Q increases, the convergence speed of both algorithms increases although the FxAP achieves a convergence behavior slightly better than the FxAPL-I. However, both of them exhibit a similar final residual error. Moreover, we can compare the computational cost in terms of multiplications per iteration for both the FxAP and FxAPL-I. The required computational cost of the updating equation is $O(Q^3)+LQ+1$ multiplications and $(3+3LQ)+LQ+1$ multiplications for both FxAP and the FxAPL-I algorithms. Note that the computational cost of the FxAP is much higher than the one of the FxAPL-I, mainly for high values of Q or L. Therefore, although the speed of convergence of the FxAPL-I algorithm is lower than the FxAP algorithm, FxAPL-I algorithm requires much less computational cost because it avoids matrix inversion. In addition, the problem of the FxAP algorithm is that, in a distributed network composed for several nodes, each node has to calculate the whole matrix inversion $[\hat{\mathbf{U}}^T(n)\hat{\mathbf{U}}(n)+\delta\mathbf{I}_Q]^{-1}$, although only a portion of the inverse matrix is needed.

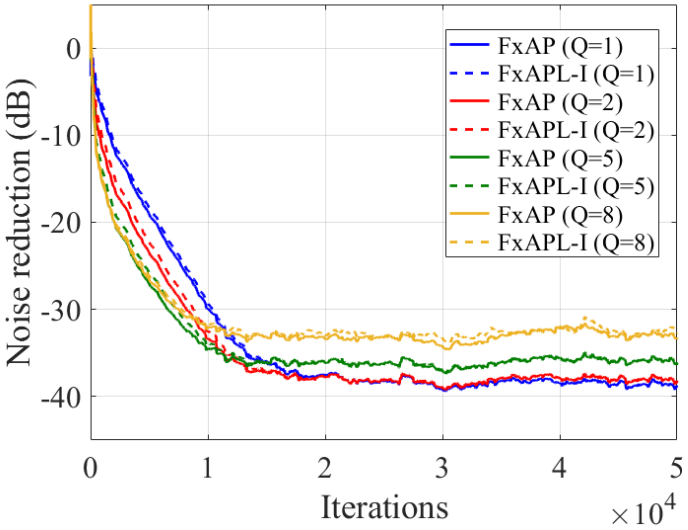


Figure 5.17: Noise reduction of an ASN composed by one single-channel node using the FxAP (solid lines) and the FxAPL-I (dashed lines) algorithms for different values of the projection order.

Therefore, in order to implement efficiently an affine-projection algorithm in a distributed network, we must find a solution which allows to split up the sum of the updating equation (5.56) into the contributions of each node. Following the same procedure used in the previous algorithms, we develop the formulation for this strategy by splitting up the contribution of each node in (5.56) as

$$\begin{aligned}\mathbf{w}(n) = \mathbf{w}(n-1) \; &- \hat{\mathbf{U}}_{1,S}(n)\mathbf{e}_1(n) - \hat{\mathbf{U}}_{2,S}(n)\mathbf{e}_2(n) - \\ &\cdots - \hat{\mathbf{U}}_{N,S}(n)\mathbf{e}_N(n)\cdot\end{aligned} \tag{5.63}$$

Considering $\mathbf{w}^k(n)$ a local version of $\mathbf{w}(n)$ at the k-th node and assuming (5.8), from node k=1 to node k=N, we obtain

$$\begin{aligned}\mathbf{w}^1(n) &= \mathbf{w}(n-1) \; - \hat{\mathbf{U}}_{1,S}(n)\, \mathbf{e}_1(n), \\ \mathbf{w}^2(n) &= \mathbf{w}^1(n) \; - \hat{\mathbf{U}}_{2,S}(n)\, \mathbf{e}_2(n), \\ &\vdots \\ \mathbf{w}^N(n) &= \mathbf{w}^{N-1}(n-1) \; - \hat{\mathbf{U}}_{N,S}(n)\, \mathbf{e}_N(n)\cdot\end{aligned} \tag{5.64}$$

Therefore, equation (5.56) may be rewritten as

$$\mathbf{w}^k(n) = \mathbf{w}^{k-1}(n) \; - \hat{\mathbf{U}}_{k,S}(n)\, \mathbf{e}_k(n)\cdot \tag{5.65}$$

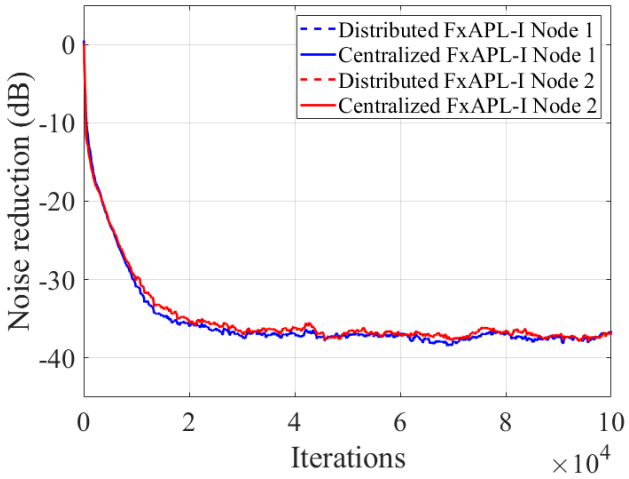


Figure 5.18: Noise reduction of the distributed system with 2 single-channel nodes (dashed line) and the centralized system with a 1:2:2 configuration (solid line) using the APL-I algorithm in both cases.

Equation (5.65) is the updating rule of the state of the k-th node by using the FxAPL-I algorithm in a distributed network (DFxAPL-I). Once all the nodes have finished the filter coefficient updates, the global vector $\mathbf{w}^N(n)$ is disseminated to the rest of the nodes for the $(n+1)$-th iteration ($\mathbf{w}^k(n)=\mathbf{w}^N(n)$).

5.4.2 Simulation results

In this section, we present the simulations carried out to evaluate the performance of the DFxAPL-I algorithm in an unconstrained ASN composed of two single-channel nodes. To do this, in a first stage, we have compared the convergence behavior and the noise reduction of both, the centralized FxAPL-I (see equation 5.56) and the distributed FxAPL-I (see equation 5.65). In a second stage, the performance of the DFxAPL-I algorithm compared to the DMEFxLMS algorithm ((5.7)) for differents values of the projection order has been evaluated. Regarding the simulation parameters setting, we consider the configuration explained in Section 3.3.

In the first stage, we have evaluated the performance of both the distributed and the centralized FxAPL-I algorithms in a network composed of two single-channel nodes with no communication constraints. The step-size parameter and the projection order was set to $\mu=3\cdot10^{-2}$ and $Q=5$ for both algorithms respectively. Figure 5.18 illustrates the time evolution of the $NR_k(n)$ for both algorithms. As expected, the distributed implementation exhibits the same performance than the centralized

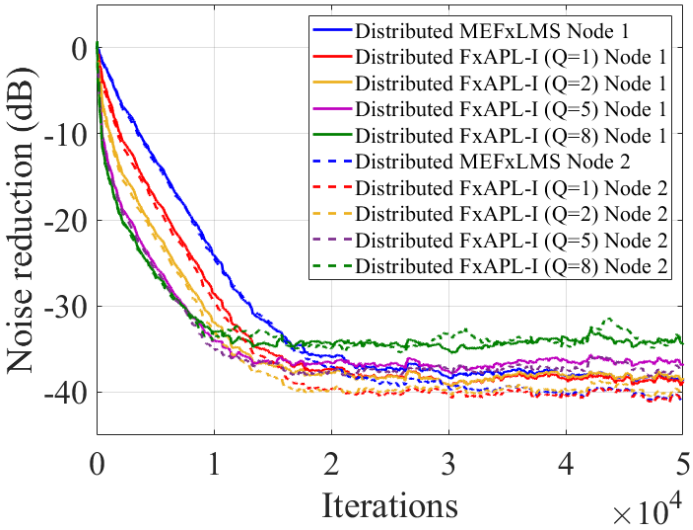


Figure 5.19: Noise reduction of the distributed system with 2 single-channel nodes using the MEFxLMS and the FxAPL-I algorithms for different values of the projection order for the node 1 (solid line) and for the node 2 (dashed line).

version for the two nodes. In the second stage, Figure 5.19 illustrates the time evolution of the $NR_k(n)$ of a distributed system with 2 single-channel nodes using the distributed versions of the MEFxLMS and the FxAPL-I for the node 1 and for the node 2. The step-size parameter was set to μ_1=$5{\cdot}10^{-3}$ for the MEFxLMS algorithm and μ_3=$3{\cdot}10^{-2}$ for the FxAPL-I algorithm. As figure shows, the proposed algorithm presents a robust behavior in terms of stability and exhibits faster convergenge speed than the MEFxLMS algorithm as Q increases for both nodes, maintaining similar attenuation for Q=1 and Q=2. For the cases of Q=5 and Q=8, the speed of convergence improves but the attenuation decreases almost 5 dB for the last case. However, the computational cost of the updating equation for the MEFxLMS algorithm is LN multiplications while the FxAPL-I algorithm requires $(3{+}3LQ){+}2LQN$ multiplications in order to update the coefficients in each iteration, which means a difference between the computational burden to be taken into account. Finally, note that all the results accomplished in this work depend on particular settings but their behavior can be easily extrapolated to other configurations.

5.4.3 Conclusions

In this section, an ANC system has been implemented over a network of distributed acoustic nodes using a collaborative incremental strategy. For this purpose, a distributed version of the FxAPL-I algorithm has been introduced. The simulations show that the distributed implementation of the APL-I algorithm has the same performance as its centralized version when there are no communication constraints in the network. It has been demonstrated that the proposed algorithm exhibits faster speed of convergenge than the MEFxLMS algorithm for several values of the projection order.

5.5 Blockwise collaborative distributed algorithm

With the aim to implement the proposed distributed ANC systems in real-time applications, the obtained equations in 5.2 have to be considered using block processing in the frequency domain in order to ensure a more efficient computational performance. The reason is because most common audio cards work with block data buffers. Moreover, hardware platforms that work with blocks of samples such as Digital Signal Processors (DSPs) or Graphics Processing Units (GPUs), use libraries of frequency-domain operations for a more efficient processing [120]. Furthermore, if the adaptive filters and the estimated secondary paths are longer than the sample block, they have to be split up into partitions [121]. For all these reasons, we consider the Frequency-domain Partitioned Block technique for the adaptive filtering operation [44, 122] based on the conventional filtered-x scheme (FPBFxLMS).

It is important to note that, the derivation of this strategy from the generic formulation presented in 3.2 is not going to be approached since the application of this technique over distributed ANC systems requires the use of a different structure.

5.5.1 FPBFxLMS algorithm

This section focuses on illustrating the FPBFxLMS algorithm applied to a centralized multichannel ANC system. Derived from the single-channel FPBFxLMS algorithm introduced in 2.3.6 and widely explained in [122], we can rewrite (3.9) in the frequency domain with blocks of B samples. Notation in Table 5.2 will be used to describe the following equations. Samples are processed by blocks of size B, L is the length of the adaptive filters and M is the length of the FIR filters that model the estimated secondary path. If L and M are higher than B, we have to split up both the adaptive filters and the estimated secondary paths into F and P partitions, respectively. In line with the notation of the thesis, boldface underlined letters denote vectors and matrices that contain information of signals in the frequency-domain

(e.g., $\underline{\mathbf{W}}$). Furthermore, the index n between brackets denotes block iteration and the super-indexes f and p denotes the number of the partition.

Considering the formulation outlined in Section 2.3.6, the control signal of the j-th node is calculated as follows

$$\underline{\mathbf{y}}_j[n] = \sum_{f=1}^{F} \underline{\mathbf{w}}_j^f[n-1] \circ \underline{\mathbf{x}}_B[n-f+1] \tag{5.66}$$

where $\underline{\mathbf{w}}_j^f[n]$ contains the $2B$-FFT of the fth partition of the adaptive filter of the j-th node. The valid samples of the adaptive filter output $\underline{\mathbf{y}}_j[n]$ are the last B samples of $\text{IFFT}\{\underline{\mathbf{y}}_j[n]\}$, as previously commented. Now we define $\mathbf{e}_{k,B}[n]$ as the error vector that contains the last block of size B of the error signal in the sensor of node k, $e_k(n)$, that is, $\mathbf{e}_{k,B}[n]=[e_k(Bn) \;\; e_k(Bn-1)\ldots \; e_k(Bn-B+1)]^T$. We define $\underline{\mathbf{e}}_k[n]$ as the FFT of size $2B$ of $\mathbf{e}_{k,B}[n]$ preceded by $\mathbf{0}_{B\times 1}$, i.e., $\underline{\mathbf{e}}_k[n] = \text{FFT}\{[\,\mathbf{0}_B \quad \mathbf{e}_{k,B}[n]\,]\}$. The filter coefficients are updated in the frequency domain by calculating the correlations between the reference signal that are filtered through the estimated secondary paths, and the error signals. To this end, the update of the coefficients of each partition of the adaptive filter at the j-th node is calculated as

$$\underline{\mathbf{w}}_j^f[n]=\underline{\mathbf{w}}_j^f[n-1] -2\mu \sum_{k=1}^{N} \text{FFT}\{[\,[\,\text{IFFT}\{\underline{\mathbf{e}}_k[n]\circ\underline{\mathbf{V}}_{jk}[n-f+1]^*\}\,]_{[1:B,:]} \quad \mathbf{0}_{[B\times FN]}\,]\}, \tag{5.67}$$

with $\underline{\mathbf{V}}_{jk}[n]=[\underline{\mathbf{v}}_{jk}^1[n], \underline{\mathbf{v}}_{jk}^2[n], \ldots, \underline{\mathbf{v}}_{jk}^F[n]]$ as a matrix of size $[2B\times F]$ where vector $\underline{\mathbf{v}}_{jk}^f[n]$ contains the $2B$-FFT of the reference signal filtered by the estimated acoustic channel $\mathbf{h}_{jk}$ to update the f-th partition of the adaptive filter,

$$\underline{\mathbf{V}}_{jk}[n] = \sum_{p=1}^{P} \underline{\mathbf{H}}_{jk}^p \circ \underline{\mathbf{x}}_B[n-p+1], \tag{5.68}$$

being $\underline{\mathbf{H}}_{jk}^p$ the FFT of size $2B$ of the pth partition of the acoustic path h_{jk}. Note that we only consider the first B samples of the $2B$-IFFT operation. $\mathbf{0}_{[B\times FN]}$ is a matrix of zeros of size $[B\times FN]$.

In the next sextion, the distributed version of the FPBFxLMS algorithm is derived from (5.67). Regarding the communication among the nodes, two types of collaboration strategy are considered: incremental learning and diffusion learning. Both strategies are discussed below.

Table 5.2: Additional notation of the description of the FPBFxLMS algorithm

B	Block size
F	L/B, number of partitions of the adaptive filters
P	M/B, Number of partitions of the estimated secondary paths
n	index that denotes block iteration
f, p	super-indexes that denote partition number.
$\underline{\mathbf{H}}_{jk}^{p}$	Fast Fourier Transform (FFT) of size $2B$ of the p-th partition of the acoustic path h_{jk}.

5.5.2 Incremental DFPBFxLMS algorithm

In this case, consider a ring-topology ASN with incremental communication. Let us define the global state of the network, $\underline{\mathbf{W}}[n]$, as the following $[2B\times FN]$ matrix

$$\underline{\mathbf{W}}[n] = \left[\ \underline{\mathbf{W}}_1[n], \underline{\mathbf{W}}_2[n], \ldots, \underline{\mathbf{W}}_N[n]\ \right], \tag{5.69}$$

being $\underline{\mathbf{W}}_j[n]$=$[\underline{\mathbf{w}}_j^1[n], \underline{\mathbf{w}}_j^2[n], \ldots, \underline{\mathbf{w}}_j^F[n]]$ a matrix of size $[2B\times F]$. Moreover, we define

$$\underline{\mathbf{V}}_k[n] = \left[\ \underline{\mathbf{V}}_{1k}[n], \underline{\mathbf{V}}_{2k}[n],\ \ldots\ \underline{\mathbf{V}}_{Nk}[n],\ \right] \tag{5.70}$$

The N nodes collaborate with each other by updating their part of $\underline{\mathbf{W}}[n]$ and transferring $\underline{\mathbf{W}}[n]$ to the next node. Therefore, every node will use a local version of the global state of the network (denoted by $\underline{\hat{\mathbf{W}}}_k[n]$) at the kth node at the nth block iteration. Notice that only the F partitions of $2B$ coefficients of their adaptive filter are needed to generate the kth node output signal,

$$\underline{\mathbf{W}}_k[n-1] = \underline{\hat{\mathbf{W}}}_N[n-1]_{(:,1+F(k-1):Fk)}. \tag{5.71}$$

Each node estimates the coefficients of the rest of the nodes to achieve a global solution using the information of the previous node and the adaptation matrix calculated using the signals that each node own. Thus, the update equation of the adaptive filter coefficients of the kth node at the nth block iteration is given by

$$\underline{\hat{\mathbf{W}}}_k[n] = \underline{\hat{\mathbf{W}}}_{k-1}[n]\ - \mu\ \mathrm{FFT}\{[\ [\ \mathrm{IFFT}\{\underline{\mathbf{E}}_k[n] \circ \underline{\mathbf{V}}_k[n]^*\}\ ]_{[1:B,:]}\quad \mathbf{0}_{[B\times FN]}\ ]\}, \tag{5.72}$$

where $\underline{\mathbf{E}}_k[n]$ is the multiplication of vector $\underline{\mathbf{e}}_k[n]$ by $\mathbf{1}_{[1\times FN]}$, a row vector of ones of size FN. Once all the nodes have finished the filter coefficient updates, the global vector $\underline{\hat{\mathbf{W}}}_N[n]$ is disseminated to the rest of the nodes for the $[n+1]$-th iteration. Note that in (5.72), $\underline{\hat{\mathbf{W}}}_0[n]$=$\underline{\hat{\mathbf{W}}}_N[n-1]$, as we stated in (5.7). ***Algorithm 5*** illustrates the summary of the the collaborative distributed FPBFxLMS algorithm with incremental learning (incremental DFPBFxLMS) instructions, which are executed per block iteration at each node.

Algorithm 5: *Incremental DFPBFxLMS algorithm for N-nodes ASN*

for all *node* $1 \leq k \leq N$ **do**
  $\underline{\mathbf{W}}_k[n-1] = \underline{\hat{\mathbf{W}}}_k[n-1]_{(:,1+F(k-1):Fk)}$
  $\underline{\mathbf{Y}}_k[n] = \sum_{f=1}^{F} \underline{\mathbf{W}}_k[n-1] \circ \underline{\mathbf{x}}_B[n-f+1]$
  for all $1 \leq j \leq N$ **do**
    $\underline{\mathbf{V}}_{jk}[n] = \sum_{p=1}^{P} \underline{\mathbf{H}}_{jk}^{p} \circ \underline{\mathbf{x}}_B[n-p+1]$
  end for
  $\underline{\mathbf{V}}_k[n] = [\, \underline{\mathbf{V}}_{1k}[n], \underline{\mathbf{V}}_{2k}[n], \ldots \underline{\mathbf{V}}_{Nk}[n] \,]$
  $\underline{\mathbf{e}}_k[n] = \mathrm{FFT}\{[\, \mathbf{0}_{B\times 1} \quad \mathbf{e}_B[n] \,]\}$
  $\underline{\mathbf{E}}_k[n] = \underline{\mathbf{e}}_k[n] \cdot \mathbf{1}_{[1\times FN]}$
  $\underline{\hat{\mathbf{W}}}_k[n] = \underline{\hat{\mathbf{W}}}_{k-1}[n] \; - \mu\, \mathrm{FFT}\{[\, [\, \mathrm{IFFT}\{\underline{\mathbf{E}}_k[n] \circ \underline{\mathbf{V}}_k[n]^*\} \,]_{[1:B,:]} \quad \mathbf{0}_{[B\times FN]} \,]\}$
end for
for all *node* $0 \leq k \leq N$ **do**
  $\underline{\hat{\mathbf{W}}}_k[n] = \underline{\hat{\mathbf{W}}}_N[n]$
end for

5.5.2.1. Simulation result

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. In a first stage, the noise reduction and the convergence performance of the distributed ANC system is evaluated and compared with the centralized ANC system. In a second stage, we evaluate and compare the computational complexity of both ANC systems. The simulations have been carried out using the configuration parameters depicted in Section 3.3. 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.

First, we compare the noise reduction of a square centralized ANC system with a 1:4:4 configuration and a distributed ANC system with four single channel nodes. Figure 5.20 shows the $NR_k(n)$ of both the centralized and the distributed implementations of the FPBFxLMS algorithm 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 studied the convergence behavior of the block filtered-x LMS algorithm (BFxLMS). In [123], the maximum μ parameter that leads to the fastest

$$0 \leq \mu < \frac{1}{B\lambda_{max}}, \tag{5.73}$$

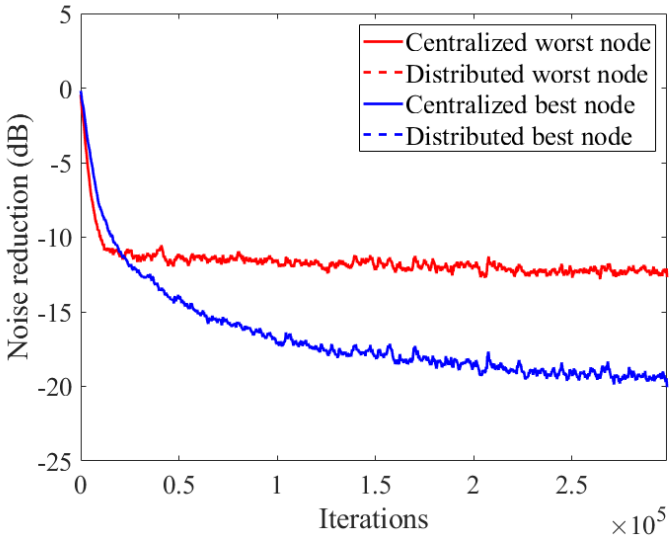


Figure 5.20: Noise reduction of the distributed system with 4 single-channel nodes and the centralized system with a 1:4:4 configuration represented for the best and worst microphone.

where λ_{max} is the maximum eigenvalue of $\mathbf{R}_{\mathbf{U}}$, defined in (3.6). 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, the size of B is also limited by the real-time condition. This implementation aspect will be addressed more broadly in 7. In summary, since in real time the processing delay must be less than the time spent to fill up the input-data buffers, defined as B/f_s, a block size large enough to handle processing and communications delays is required. Therefore, there is a minimum value of B for a given configuration that assures the real-time condition and maximum convergence speed. Figure 5.21 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 (5.73), where, for the same reference signal, the maximum μ is doubled by reducing the size of B by half.

Table 5.3 compares the computational complexity in terms of multiplications, additions, and FFTs per iteration of the FPBFxLMS algorithm, implemented for a

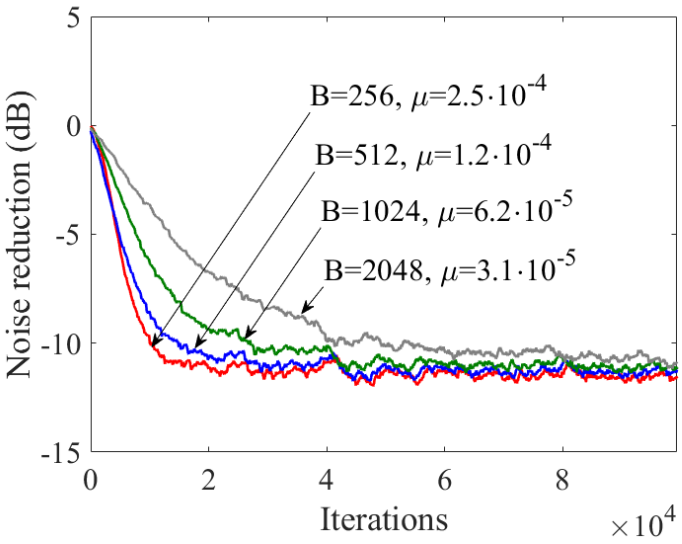


Figure 5.21: Noise reduction of the distributed algorithm for different size of B.

centralized and a distributed ANC systems. For the centralized implementation, we consider a multichannel ANC system with one disturbance noise and the same number of microphones and loudspeakers (1:N:N configuration). For the distributed implementation, we consider a network of N single-channel nodes. It is important to note that the complexity of the network as a whole is at least as high as the centralized algorithm. However, note that each node processes the algorithm simultaneously except the last addition of the global network state of the previous node (see line 11 in Algorithm 1). Therefore, in Table 5.3, we have only computed the operations of one single-channel 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 5.3 shows the computational complexity of both algorithms related to values of L and N. Then, this computational complexity is particularized for N=1, N=4 and N=8. As expected, when N=1, both implementations need the same number of operations. This is because both the centralized and the distributed ANC systems become a single-channel system. Moreover, for $N = 4$, we compare the operations of a 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. Finally, 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 represents a bottleneck in massive multichannel ANC systems. Otherwise, the increase of computational complexity in a distributed ANC system is not so significant.

Table 5.3: Total number of multiplications (MUX), additions (ADD) and FFTs per blockwise iteration of the FPBFxLMS algorithm in both: (1) centralized and (2) distributed ANC systems.

		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	$4 + 4N$	8	20	36

Algorithm 6: *Diffusion DFPBFxLMS algorithm for N-nodes ASN*

for all *node* $1 \leq k \leq N$ **do**
  $\underline{\mathbf{W}}_k[n-1] = \underline{\hat{\mathbf{W}}}_k[n-1]_{(:,1+F(k-1):Fk)}$
  $\underline{\mathbf{Y}}_k[n] = \sum_{f=1}^{F} \underline{\mathbf{W}}_k[n-1] \circ \underline{\mathbf{x}}_B[n-f+1]$
  for all $1 \leq j \leq N$ **do**
    $\underline{\mathbf{V}}_{jk}[n] = \sum_{p=1}^{P} \underline{\mathbf{H}}_{jk}^{p} \circ \underline{\mathbf{x}}_B[n-p+1]$
  end for
  $\underline{\mathbf{V}}_k[n] = [\, \underline{\mathbf{V}}_{1k}[n], \underline{\mathbf{V}}_{2k}[n], \ldots \underline{\mathbf{V}}_{Nk}[n] \,]$
  $\underline{\mathbf{e}}_k[n] = \mathrm{FFT}\{[\, \mathbf{0}_{B\times 1} \quad \mathbf{e}_B[n] \,]\}$
  $\underline{\mathbf{E}}_k[n] = \underline{\mathbf{e}}_k[n] \cdot \mathbf{1}_{[1\times FN]}$
  $\underline{\hat{\mathbf{W}}}_k[n] = \alpha \underline{\hat{\mathbf{W}}}_k[n-1] + (1-\alpha)\underline{\hat{\mathbf{W}}}_{k-1}[n]$
    $- \mu\, \mathrm{FFT}\{[\, [\, \mathrm{IFFT}\{\underline{\mathbf{E}}_k[n] \circ \underline{\mathbf{V}}_k[n]^*\} \,]_{[1:B,:]} \quad \mathbf{0}_{[B\times FN]} \,]\}$
end for

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 $(2N-1)$ times. This fact implies that the transmission speed of the network has to be considered.

5.5.3 Diffusion DFPBFxLMS algorithm

An alternative of the previous proposal is the case in which simultaneously all nodes can communicate with their neighbor. This is the case of a ASN which uses a ring topology with diffusion learning [6,124]. This collaboration method is not considered

as an incremental strategy, although with the passage of iterations we would arrive at a dissemination of information similar to incremental strategies. However, it is more robust if any node presents failures in its functioning. In addition, the diffusion strategy achieves improved use of the spatial diversity of available data and achieves better performance than the incremental learning for a similar convergence rate [33, 125]. By allowing parallel communication of the nodes and assuming that each pair of neighboring nodes do not share the communications channel, we would reduce the need for bandwidth for network communications since the total data flow for each iteration would be divided into transmissions between nodes of $L \times N$ samples. Based on this, we are going to describe the FPBFxLMS algorithm for a N-nodes ASN with diffusion-based communication. The notation in Table 5.2 is used to describe the algorithm.

In ***Algorithm 6*** a summary of the algorithm instructions executed by the diffusion DFPBFxLMS algorithm per block iteration n for each node is shown. As a simplified version of the proposed algorithm, we consider that each node collaborates only with a single node, which we will call adjoint node. So, the updated coefficients in each node depend on both its own coefficients and the coefficients calculated by the adjoint node in the previous iteration. In this case, the coefficients update equation are stated as follows,

$$\underline{\hat{\mathbf{W}}}_k[n] = \alpha\underline{\hat{\mathbf{W}}}_k[n] + (1-\alpha)\underline{\hat{\mathbf{W}}}_{k-1}[n] - \mu\,\mathrm{FFT}\{[\,[\,\mathrm{IFFT}\{\underline{\mathbf{E}}_k[n] \circ \underline{\mathbf{V}}_k[n]^*\}\,]_{[1:B,:]} \quad \mathbf{0}_{[B\times FN]}\,]\} \tag{5.74}$$

where α is a constant $(0 \leq \alpha \leq 1)$ that weights the local estimate in node k and the data received from its adjoint node $(k-1)$. In summary, the k-th node will use the state of the $(k-1)$th node in the previous iteration to calculate its own state in the current iteration. Therefore, the information of all the nodes spreads out over the network at each iteration. The value of $\alpha = 1$ assigned to all the nodes means that no network information is available, similarly to a distributed non-collaborative system using the FPBFxLMS algorithm (non-collaborative DFPBFxLMS). A summary of the non-collaborative algorithm instructions executed per block iteration n for a single node is given in ***Algorithm 7***.

5.5.3.1. Simulation result

In this section we present the simulation results carried out to evaluate the performance of the incremental DFPBFxLMS algorithm compared to the non-collaborative DFPBFxLMS algorithm for an ANC system over a six-nodes ASN. Regarding the acoustic system, we consider the real acoustic channels presented in Section 3.3. Furthermore, a block-size of $B = 512$ and a filter length of $L = 1024$ have been considered.

We have evaluated the performance of both the non-collaborative FPBFxLMS and the incremental DFPBFxLMS algorithm in a network composed of six single-channel nodes. Three different scenarios have been considered by varying the acous-

Algorithm 7: *Non-collaborative DFPBFxLMS algorithm for a single-node ASN*

1: $\underline{\mathbf{Y}}[n] = \sum_{f=1}^{F} \underline{\mathbf{W}}^f[n-1] \circ \underline{\mathbf{X}}[n-f+1]$
2: $\underline{\mathbf{V}}[n] = \sum_{p=1}^{P} \underline{\mathbf{S}}^p \circ \underline{\mathbf{X}}[n-p+1]$
3: $\underline{\mathbf{e}}[n] = \mathrm{FFT}\{[\mathbf{0}_{B\times 1} \quad \mathbf{e}_B[n]]\}$
4: **for all** $1 \leq f \leq F$ **do**
5: $\underline{\mathbf{W}}^f[n] = \underline{\mathbf{W}}^f[n-1] - \mu\, \mathrm{FFT}\{[\,[\mathrm{IFFT}\{\underline{\mathbf{e}}[n] \circ \underline{\mathbf{V}}[n-f+1]^*\}\,]_{[1:B,:]} \quad \mathbf{0}_{B\times 1}]\}$
6: **end for**

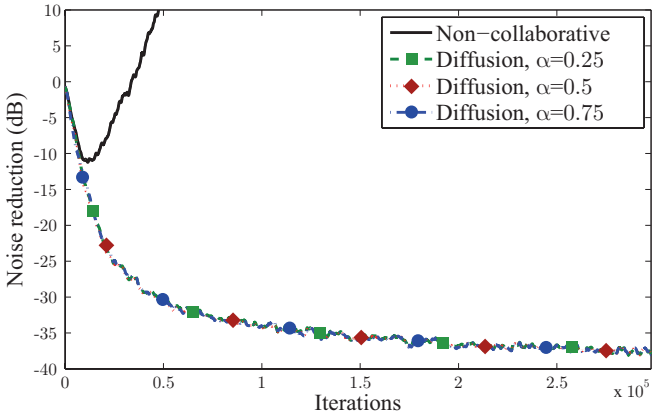
tical interaction among the nodes. Specifically a over-coupled, coupled and uncoupled system. The main difference among them is their respective separation among the nodes (minimum, medium and maximum separation respectively), which will affect the amount of acoustic coupling among them. The step-size parameter was set to μ_1=$4.1{\cdot}10^{-4}$, μ_2=$4.1{\cdot}10^{-5}$ and μ_3=$1.6{\cdot}10^{-4}$ for the three scenarios, respectively.

Figure 5.22, Figure 5.23, and Figure 5.24 illustrate the instantaneous Noise Reduction versus the number of iterations for the node with the best and the worst performance for different values of α. In the first two cases (over-coupled and coupled ASNs), the non-collaborative algorithm starts canceling the noise but, at a certain point, it turns unstable and does not converge. However, the diffusion algorithm is stable providing higher noise reduction for all values of α. So, the coupled systems need any kind of collaboration between the nodes to minimize the effects of the acoustic coupling and, therefore, to achieve the convergence of the algorithm. On the other hand, Figure 5.24 shows that the performance of the uncoupled system improves when there is no network information, that means that the node updates its filter coefficients using only its local information. Note that, in all the cases, the value of α does not lead to any significant change in the algorithm performance. From this fact, we can conclude that, in a coupled system, any collaboration between the nodes is better than the non-collaborative strategy. Moreover, the use of a specific α for each node would improve the performance of the system, both in terms of NR and the speed of convergence.

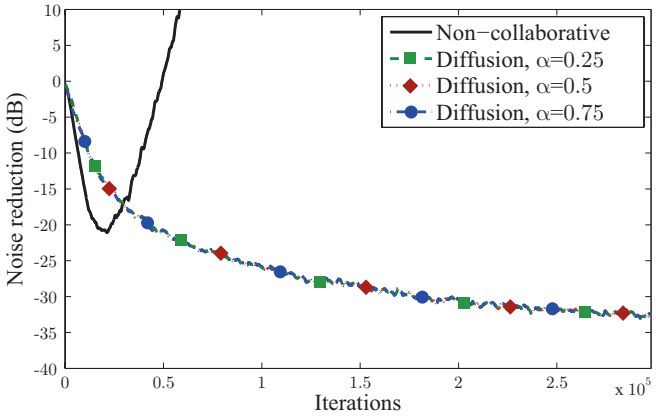
5.5.4 Conclusions of this section

In this section, two collaborative distributed implementations of the FPBFxLMS algorithm for an ANC system have been presented considering two types of cooperation among nodes: incremental and diffusion learning.

Initally, an scalable and versatile distributed implementation using an incremental strategy in the network has been presented. It has been demonstrated that the



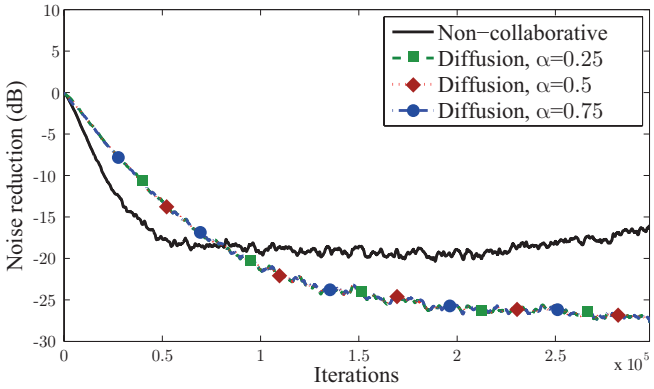
(a)



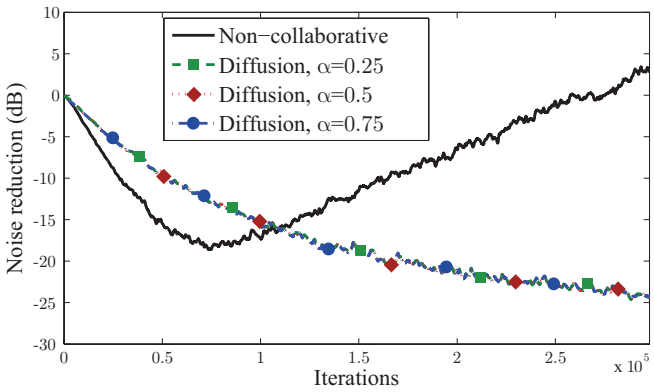
(b)

Figure 5.22: NR obtained for the non-collaborative DFPBFxLMS algorithm and the incremental DFPBFxLMS algorithm using a six-node over-coupled ASN at the node with the best performance (a) and with the worst performance (b).

proposed incremental 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. 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 converge 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,



(a)

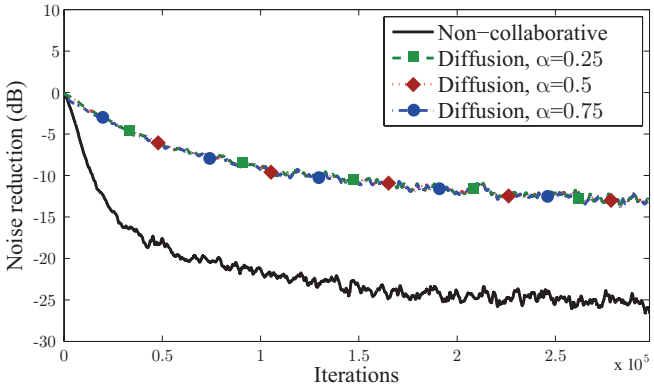


(b)

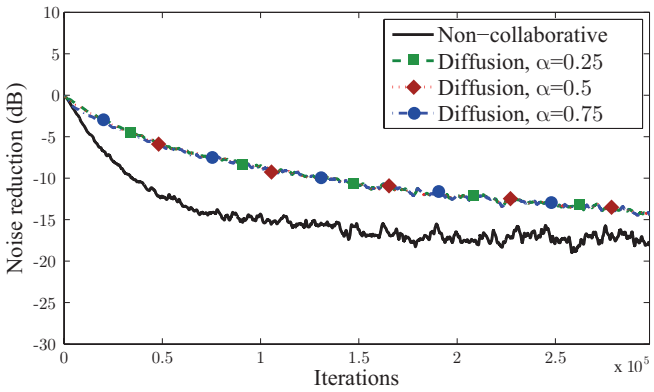
Figure 5.23: NR obtained for the non-collaborative DFPBFxLMS algorithm and the incremental DFPBFxLMS algorithm using a six-node coupled ASN at the node with the best performance (a) and with the worst performance (b).

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.

Finally, at the end of this section, a new approach based on the FPBFxLMS algorithm that introduces collaboration between nodes following a diffusion strategy has been presented. The proposed diffusion algorithms allow that every node updates the global state of the network by using local information and assuming some



(a)



(b)

Figure 5.24: NR obtained for the non-collaborative FPBFxLMS algorithm and the incremental DFPBFxLMS algorithm using a six-node uncoupled ASN at the node with the best performance (a) and with the worst performance (b).

weighted collaboration with its neighbor nodes, as well. The performance of this diffusion method has been evaluated in networks composed of both acoustically coupled and uncoupled nodes. In coupled networks, the proposed diffusion algorithm obtains a good performance in contrast to the non-collaborative strategy because of the information of the network state calculated at each node spread all over the nodes. On the contrary, in uncoupled systems, a non-collaborative strategy outperforms the collaborative method. In this way, it is better that the nodes do not interchange information between them because the performance of the system improves when the node updates its filter coefficients using only its local information.

5.6 Collaborative distributed algorithm using clusters

As commented in 2.2, one of the main problems of the collaborative strategies is how to effectively share information among nodes. The higher data transfer, the higher delay in transmitting the infomation between nodes of the network. This fact implies that the transmission speed of the network has to be also considered. In addition, network congestion or sensor failure are common problems that usually appears in real networks leading to data loss [126, 127]. All these aspects may lead to require new collaborative distributed techniques which aims to reduce data amount to be exchanged.

In order to reduce the computational and communication requirements of ASNs, a node-specific ANC based on the DMEFxLMS algorithm where each node collaborates in an incremental manner only with a subset of nodes acoustically coupled was proposed in [36]. This technique aims to avoid that redundant data can be shared. Every subset of nodes try to reach its own solution but if several subsets share the same nodes, theirs solutions will be affected. How much they are affected can be determined by the level of acoustic coupling among the actuators and microphones of these nodes. In the case that there exists acoustic interaction among the nodes, a distributed ANC system over subnetworks of collaborative nodes may be used to reach results equivalent to those of the centralized method. Unlike any other distributed algorithm for ANC, this node-specific method is scalable with the network size since the computational cost and the communication requirements are independent of the total number of nodes. However, the details of the proof by which the distributed algorithm for node-specific ANC converges to the centralized solution are omitted. Moreover, a criterion to define the subsets of nodes, i.e. which nodes should collaborate, is not specified.

Therefore, based on the node-specific technique proposed in [36], in this section we focus on a clustered-version of the DMEFxLMS algorithm over a network with distributed nodes and incremental learning without communication constraints. Thus, we analize the mean behavior of the proposed algorithm over the network and provide the conditions under which it converges in the mean to the centralized solution. In this regard, the analysis is carried out in the frequency domain. In order to validate the theoretical results, we present a set of simulations with real acoustic channel responses for ASNs of different sizes by using a collaborative method presented in Appendix A based on the acoustically coupling among the nodes to identify when collaboration is needed. A study of implementation aspects such as computational complexity and communication capabilities among the nodes in the network is also presented. It should be emphasised that, the strict meaning of the node-specific term means that nodes are interested in solving different estimation problems. However, in a distributed ANC system all nodes try to solve the same estimation problem. Due to this, in this section, we use the node-specific term to refer to that each node only

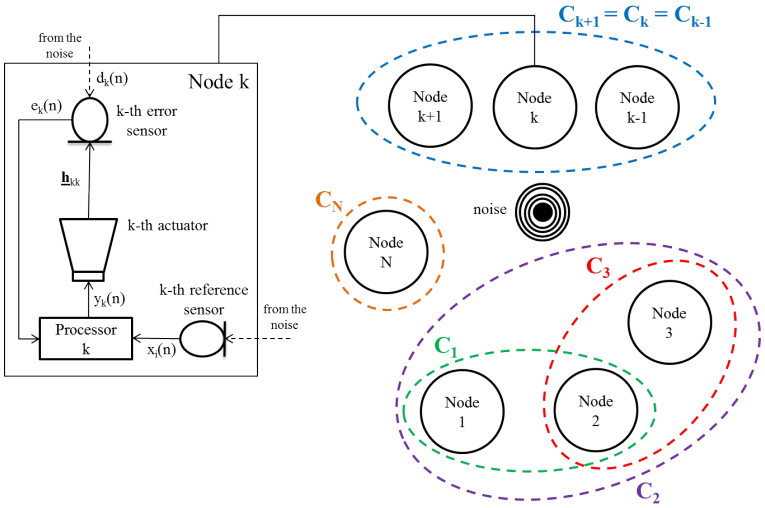


Figure 5.25: Example of a distributed clustered ASN with incremental communication and composed of N single-channel nodes supporting an ANC application.

communicates with a specific group of nodes in order to reach a solution similar to the one provided by the corresponding distributed alternatives derived from centralized systems.

Note that a collaborative strategy may imply that each node estimates the coefficients of the rest of the nodes to achieve a global solution. However, depending on the location of the nodes over the network, the solution reached at each node is mainly affected by the contribution of its closest nodes, while the behavior of distant nodes hardly affects its ANC filter estimation. Therefore, a possible strategy is to bring together certain nodes in order to obtain their group estimate as a part of the global network solution, as shown in Figure 5.25. In this way, each node may only transmit information to the nodes of its specific subset. We start from a network centralized solution applying a coupling mask and then, the implementation of this solution and its derivation to the clustering strategy over a network of distributed nodes will be presented below.

5.6.1 cl-DMEFxLMS algorithm

For this analysis, it should be recalled the coefficients update equation of the centralized MEFxLMS algorithm, described in (3.8) and expressed here as

$$\mathbf{w}(n+1) = \mathbf{w}(n) \; - 2\mu \sum_{k=1}^{N} \mathbf{u}_k(n) e_k(n). \tag{5.75}$$

Analyzing (5.75), note that, if the value of any component of $\mathbf{u}_k(n)$ is very low, they are not going to contribute to the sumatory and then, to the update of the adaptive filter. Therefore, at each component of $\mathbf{w}(n)$ (solution at each node), only some terms must be added. That case could happen if, for instance, the loudspeaker of node j would barely contribute to the error signal measured at node k with respect to its own loudspeaker k. By way of example, this may happen if the nodes are located at such a distance which does not allow the acoustic interaction among them. That is, generally speaking, node j and node k would be not acoustically coupled. This would be similar to use a coupling mask to define which nodes are acoustically coupled or not, as explained in Appendix A. To this end, we use the $[LN\times 1]$ vector $\tilde{\mathbf{u}}_k(n)$ defined in (A.2) being α_{jk} the value which represents the collaboration between the node j and the node k and defined as

$$\alpha_{jk} = \begin{cases} 1 & \text{, if } k \in C_j \\ 0 & \text{, if } k \notin C_j \end{cases} . \tag{5.76}$$

where C_j is defined as the set of nodes of size N_j devoted to estimate the adaptive filter of the j-th node, calculated as

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) \; - 2\mu \sum_{k=1}^{N} \alpha_{jk} \mathbf{u}_{jk}(n) e_k(n). \tag{5.77}$$

Note that the value of α_{jk} determines whether or not the error signal of the k-th node, $e_k(n)$, is needed for the calculation of the filter of the j-th node, $\mathbf{w}_j(n)$. In other words, if node k is acoustically coupled with node j, it will be within C_j and therefore, α_{jk}=1. The global filter updating equation of the MEFxLMS applying a coupling mask (cm-MEFxLMS) algorithm was introduced in (A.1).

On the other hand, the derivation of (A.1) from (3.22) can be obtained by considering the following assumptions. Consider I'=I=N with $\mathbf{A}$ as defined in (3.28). $\boldsymbol{\Upsilon}_i$ is similarly defined as (4.1) and the $[LN\times LN]$ matrix $\boldsymbol{\Theta}_{i'i}$ is designed as a diagonal matrix with only a non-zero matrix at the $(1{+}L(i'{-}1){:}i'L, 1{+}L(i{-}1){:}iL)$ position, $\boldsymbol{\theta}_{i'i}$, which is defined as an identity matrix of size $[L\times L]$,

$$\boldsymbol{\Theta}_{i'i}{=}\begin{bmatrix} \mathbf{0}_L & \cdots & \mathbf{0}_L & \cdots & \mathbf{0}_L \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_L & \cdots & \boldsymbol{\theta}_{i'i} & \cdots & \mathbf{0}_L \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0}_L & \cdots & \mathbf{0}_L & \cdots & \mathbf{0}_L \end{bmatrix}, \qquad \boldsymbol{\theta}_{i'i}{=}\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \tag{5.78}$$

In this case, the matrix $\boldsymbol{\Psi}_{i'i}$ must be redefined as $\boldsymbol{\Psi}_{i'}$, i.e., a $[QN\times QN]$ diagonal matrix with only a non-zero matrix at the $(1{+}L(i'{-}1){:}i'L, 1{+}L(i'{-}1){:}i'L)$ position,

$\boldsymbol{\psi}_{i'i'}$, which is defined as an identity matrix of size $[Q\times Q]$,

$$\boldsymbol{\Psi}_{i'}=\begin{bmatrix}\mathbf{0}_Q & \cdots & \mathbf{0}_Q & \cdots & \mathbf{0}_Q\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ \mathbf{0}_Q & \cdots & \boldsymbol{\psi}_{i'i'} & \cdots & \mathbf{0}_Q\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ \mathbf{0}_Q & \cdots & \mathbf{0}_Q & \cdots & \mathbf{0}_Q\end{bmatrix}, \qquad \boldsymbol{\psi}_{i'i'}=\begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & 1 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 1\end{bmatrix}, \tag{5.79}$$

$\mathbf{S}(n)$ must be designed as an identity matrix of size $[LN\times LN]$ and $\boldsymbol{\phi}_{i'}$ must be equal to (5.60). Finally, considering both $\boldsymbol{\beta}$ and $\boldsymbol{\Gamma}$ as null matrices, Q=1 and ϵ_k=0, (5.2) can be arranged as

$$[\hat{\mathbf{U}}]_c(n)=\begin{bmatrix}\alpha_{11}\mathbf{u}_{11}(n) & \alpha_{12}\mathbf{u}_{12}(n) & \cdots & \alpha_{1N}\mathbf{u}_{1N}(n)\\ \alpha_{21}\mathbf{u}_{21}(n) & \alpha_{22}\mathbf{u}_{22}(n) & \cdots & \alpha_{2N}\mathbf{u}_{2N}(n)\\ \vdots & \vdots & \ddots & \vdots\\ \alpha_{N1}\mathbf{u}_{N1}(n) & \alpha_{N2}\mathbf{u}_{N2}(n) & \cdots & \alpha_{NN}\mathbf{u}_{NN}(n)\end{bmatrix} \tag{5.80}$$

and therefore, the generic updating equation can be expressed as

$$\mathbf{w}(n+1)=\mathbf{w}(n)\ -2\mu[\hat{\mathbf{U}}]_c(n)\mathbf{e}(n), \tag{5.81}$$

A distributed version of (5.81) can also be used to fully benefit from the advantages related to the distributed processing previously commented. If we consider a ring topology with incremental communication between the nodes, the data exchange is carried out in a consecutive order, as shown in Figure 5.2.(b). In this way, each node of the ASN described in Figure 5.25 computes its term of the summation in (5.81). Considering the local version of $\mathbf{w}(n)$ at the kth node, $\mathbf{w}^k(n)$, from node k=1 to node k=N, we can split up the contribution of each node in (5.81) as

$$\mathbf{w}^k(n)=\mathbf{w}^{k-1}(n)\ -2\mu\tilde{\mathbf{u}}_k(n)e_k(n) \tag{5.82}$$

Note that this distributed solution, which we denote as distributed MEFxLMS applying a coupling mask (cm-DMEFxLMS), it is not scalable in terms of communication and computational demands because the dimension of $\mathbf{w}^k(n)$, with size $LN\times 1$, depends on the total number of nodes of the network.

In order to reduce both computation and communication requirements of the previous algorithm, a possible strategy consists on creating subsets of nodes acoustically coupled where each node only exchanges information with the nodes of its defined subset (see Figure 5.25) in order to obtain their group estimate as a part of the global network solution. We assume that subsets of nodes are not disjoint, i.e., at least one node of each subset is acoustically coupled with at least some node of another subset.

Note that, due to how is formed $\tilde{\mathbf{u}}_k(n)$ (and depending on the value of α_{jk}), only N_k vectors of $\mathbf{w}^k(n)$ are updated at the k-th node when (5.82) is performed,

$$\begin{bmatrix} \mathbf{w}_1^k(n) \\ \mathbf{w}_2^k(n) \\ \vdots \\ \mathbf{w}_N^k(n) \end{bmatrix} = \begin{bmatrix} \mathbf{w}_1^{k-1}(n) \\ \mathbf{w}_2^{k-1}(n) \\ \vdots \\ \mathbf{w}_N^{k-1}(n) \end{bmatrix} - 2\mu \begin{bmatrix} \alpha_{1k}\mathbf{h}_{1k}^T\mathbf{X}^T(n)e_k(n) \\ \alpha_{2k}\mathbf{h}_{2k}^T\mathbf{X}^T(n)e_k(n) \\ \vdots \\ \alpha_{Nk}\mathbf{h}_{Nk}^T\mathbf{X}^T(n)e_k(n) \end{bmatrix} \tag{5.83}$$

Therefore, defining the vector $\hat{\mathbf{w}}^k(n)$= col$[\mathbf{w}_l^k(n)]_{l\in C_k}$ of size $[LN_k\times 1]$ as the adaptive filters of the nodes acoustically coupled to the k-th node (for l=$1, 2, \ldots, N_k$) and $\hat{\mathbf{u}}_k(n)$= col$[\mathbf{x}(n)\mathbf{h}_{lk}]_{l\in C_k}$ as a vector of size $[LN_k\times 1]$, we obtain the updating equation of the distributed non-disjoint clustering algorithm (cl-DMEFxLMS) for ANC at node k-th as follows

$$\hat{\mathbf{w}}^k(n) = \hat{\mathbf{w}}^{k-1}(n) \; - 2\mu_k\hat{\mathbf{u}}_k(n)e_k(n) \tag{5.84}$$

where

$$\begin{aligned} \hat{\mathbf{w}}^k(n)&=[\,[\mathbf{w}^k_{C_k(1)}(n)]^T \;\; [\mathbf{w}^k_{C_k(2)}(n)]^T \;\cdots\; [\mathbf{w}^k_{C_k(N_k)}(n)]^T\,]^T, \\ \hat{\mathbf{w}}^{k-1}(n)&=[\,[\mathbf{w}^{k_1}_{C_k(1)}(n)]^T \;\; [\mathbf{w}^{k_2}_{C_k(2)}(n)]^T \;\cdots\; [\mathbf{w}^{k_{N_k}}_{C_k(N_k)}(n)]^T\,]^T, \\ \hat{\mathbf{u}}_k(n)&=[\mathbf{h}^T_{C_k(1)k}\mathbf{x}^T(n) \;\; \mathbf{h}^T_{C_k(2)k}\mathbf{x}^T(n) \;\cdots\; \mathbf{h}^T_{C_k(N_k)k}\mathbf{x}^T(n)\,]^T, \end{aligned} \tag{5.85}$$

being

$$\mathbf{w}^{k_l}_{C_k(l)}(n) = \begin{cases} \mathbf{w}^{\max\{C_l\}}_{C_k(l)}(n-1) & \text{if } C_{kl}=\emptyset \\ \mathbf{w}^{\max\{C_{kl}\}}_{C_k(l)}(n) & \text{otherwise} \end{cases}, \tag{5.86}$$

with $C_k(l)$ refers to the l-th node within the set C_k (for l=$1, 2, \ldots, N_k$), C_{kl} = $\{j\in C_l : j<k\}$, $\emptyset$ means empty set and the set C_l=$\{k : l\in C_k\}$ denoting the ordered set of indices associated with those nodes acoustically coupled at node k (for k = $1, 2, \ldots, N$). The set C_{kl} is composed by all indices p of C_l for which p<k. If there exist some index which fulfill this condition, then k_l=max$\{C_{kl}\}$. If the set C_{kl} is empty, then k_l=max$\{C_l\}$. Note that only the local information is needed to generate the kth node output signal $y_k(n)$,

$$y_k(n) = [\mathbf{w}_k^k(n)]^T[\mathbf{X}(n)]_{(:,1)} \tag{5.87}$$

It is important to note that (5.84), which is the same solution as propposed in [36], requires less computation and communication demands than (5.82) since each node k only needs to update and transmit N_k adaptive filters whose size is independent of the total number of nodes (N).

5.6.2 Convergence analysis

In this section, we study the necessary conditions to guarantee the stability of the following algorithms: we start from the cm-MEFxLMS approach, passing through its distributed version (cm-DMEFxLMS) to derive into the proposed cl-DMEFxLMS algorithm. In this way, the same conditions as presented in Section 5.2.2 are considered.

From (5.17), it is easy to note that, the evolutionary behavior of the mean values of the errors in the global adaptive filter $\underline{\mathbf{W}}$ evaluated at the frequency domain for the cm-MEFxLMS algorithm can be expressed as

$$\underline{\mathbf{V}}(n+1) = (I_N - 2\mu\hat{\underline{\mathbf{S}}}\underline{\mathbf{H}}^H)\underline{\mathbf{V}}(n). \tag{5.88}$$

where $\hat{\underline{\mathbf{S}}}$ is a $[N\times N]$ matrix arranged as

$$\hat{\underline{\mathbf{S}}} = \begin{bmatrix} \hat{\underline{S}}_{11} & \hat{\underline{S}}_{12} & \cdots & \hat{\underline{S}}_{1N} \\ \hat{\underline{S}}_{21} & \hat{\underline{S}}_{22} & \cdots & \hat{\underline{S}}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\underline{S}}_{N1} & \hat{\underline{S}}_{N2} & \cdots & \hat{\underline{S}}_{NN} \end{bmatrix}, \tag{5.89}$$

with $\hat{\underline{S}}_{jk}{=}\alpha_{jk}\underline{H}_{jk}$. And therefore, $\underline{\mathbf{V}}(n)$ will tend to zero, if the step-size μ is selected so that

$$0 < \mu_{max} < \frac{1}{\lambda_{max}(\hat{\underline{\mathbf{S}}}\underline{\mathbf{H}}^H)}, \tag{5.90}$$

where $\lambda_{max}(\hat{\underline{\mathbf{S}}}\underline{\mathbf{H}}^H)$ is the maximum of the eigenvalues of $\hat{\underline{\mathbf{S}}}\underline{\mathbf{H}}^H$.

Consider now to examinate the stability conditions of the distributed case. Similarly, from (5.33), the stability of the cm-DMEFxLMS algorithm can be assessed by using the following update equation

$$\underline{\mathbf{V}}(n) = \left[I_N - 2\mu\sum_{k=1}^{N}\hat{\underline{\mathbf{S}}}_k\underline{\mathbf{H}}_k^H\right]\underline{\mathbf{V}}(n-1), \tag{5.91}$$

where $\hat{\underline{\mathbf{S}}}_k = [\hat{\underline{S}}_{1k}\hat{\underline{S}}_{2k}\ldots\hat{\underline{S}}_{Nk}]^T$. Since $\sum_{k=1}^{N}\hat{\underline{\mathbf{S}}}_k\underline{\mathbf{H}}_k^H{=}\hat{\underline{\mathbf{S}}}\underline{\mathbf{H}}^H$ (as similarly proved in (5.37)), (5.91) can be expressed as

$$\underline{\mathbf{V}}(n) = (I - 2\mu\hat{\underline{\mathbf{S}}}\underline{\mathbf{H}}^H)\underline{\mathbf{V}}(n-1), \tag{5.92}$$

which is the same expression as (5.88) in the centralized approach, and consequently, both strategies cm-MEFxLMS and cm-DMEFxLMS algorithms achieve the same stability condition. Note that (5.92) is equal to (5.17) when all values of α_{jk} are equal to 1. But if any α_{jk} is 0 in matrix $\hat{\underline{\mathbf{S}}}$, the system stability can not be assured.

Therefore, the main challenge here is to determine how values of α_{jk} in matrix $\hat{\underline{\mathbf{S}}}$ affect on system convergence. However, this is not a trivial problem, as discussed in Appendix A.

In other hand and with the aim to analyze the equivalence between the stability conditions of cm-DMEFxLMS and cl-DMEFxLMS strategies, two cases are considered:

- *Case a)* : first, the last node of each cluster will tend to converge to the last node of the global network (in the equivalent cm-DMEFxLMS case) and therefore, to the optimal global solution of the network.
- *Case b)* : and second, the local version of the global solution reached for node k-th by using the cl-DMEFxLMS strategy is the same as the local version of the global cm-DMEFxLMS solution at the same node.

The updating equation of the cl-DMEFxLMS algorithm at node k-th defined in (5.84) can be expressed in the frequency domain as

$$\hat{\underline{\mathbf{W}}}^{k}(n) = \hat{\underline{\mathbf{W}}}^{k-1}(n) \; - 2\mu\hat{\underline{\mathbf{H}}}_k\underline{E}_k(n), \tag{5.93}$$

where $\hat{\underline{\mathbf{H}}}_k$=$[\underline{H}^T_{C_k(1)k}\underline{H}^T_{C_k(2)k}\cdots\; \underline{H}^T_{C_k(N_k)k}\,]^T$. Starting from the *Case a*, from (5.93) and considering (5.85), we define $\hat{\underline{\mathbf{W}}}(n)$ as the global vector which contains the filter coefficient of the last node of the N subsets as

$$\hat{\underline{\mathbf{W}}}(n)=[\,[\underline{W}_1^{C_1(N_1)}(n)]^T[\underline{W}_2^{C_2(N_2)}(n)]^T\cdots[\underline{W}_N^{C_N(N_N)}(n)]^T\,]^T, \tag{5.94}$$

where $\underline{W}_k^{C_k(N_k)}(n)$ is the updated filter coefficient of the last node at the n-th iteration which belongs to the subset C_k. In order to prove that, when $f\to\infty$, $\underline{W}_k^{C_k(N_k)}(\infty) = \underline{W}_k^{N}(\infty)$ and then the optimal solution of (5.94), $\hat{\underline{\mathbf{W}}}_o$ will tend to $\underline{\mathbf{W}}_\infty$, an example of an ASN composed of four nodes comparing the results obtained from both cm-DMEFxLMS and cl-DMEFxLMS algorithms is detailed below.

Consider a four-node ASN where nodes have been clustered as follows: $C_1 = \{1,2\}$, C_2=$\{1,2,3\}$, C_3=$\{2,3,4\}$, C_4=$\{3,4\}$. This means that the updating filter equation of each node by using the cm-DMEFxLMS algorithm can be represented as

$$\begin{bmatrix}\underline{W}_1^1(n)\\ \underline{W}_2^1(n)\\ \underline{W}_3^1(n)\\ \underline{W}_4^1(n)\end{bmatrix} = \begin{bmatrix}\underline{W}_1^0(n)\\ \underline{W}_2^0(n)\\ \underline{W}_3^0(n)\\ \underline{W}_4^0(n)\end{bmatrix} -2\mu \begin{bmatrix}\underline{H}_{11}\underline{E}_1(n)\\ \underline{H}_{21}\underline{E}_1(n)\\ 0\\ 0\end{bmatrix}, \tag{5.95}$$

$$\begin{bmatrix}\underline{W}_1^2(n)\\ \underline{W}_2^2(n)\\ \underline{W}_3^2(n)\\ \underline{W}_4^2(n)\end{bmatrix} = \begin{bmatrix}\underline{W}_1^1(n)\\ \underline{W}_2^1(n)\\ \underline{W}_3^1(n)\\ \underline{W}_4^1(n)\end{bmatrix} -2\mu \begin{bmatrix}\underline{H}_{12}\underline{E}_2(n)\\ \underline{H}_{22}\underline{E}_2(n)\\ \underline{H}_{32}\underline{E}_2(n)\\ 0\end{bmatrix}, \tag{5.96}$$

$$\begin{bmatrix} \underline{W}_1^3(n) \\ \underline{W}_2^3(n) \\ \underline{W}_3^3(n) \\ \underline{W}_4^3(n) \end{bmatrix} = \begin{bmatrix} \underline{W}_1^2(n) \\ \underline{W}_2^2(n) \\ \underline{W}_3^2(n) \\ \underline{W}_4^2(n) \end{bmatrix} - 2\mu \begin{bmatrix} 0 \\ \underline{H}_{23}\underline{E}_3(n) \\ \underline{H}_{33}\underline{E}_3(n) \\ \underline{H}_{43}\underline{E}_3(n) \end{bmatrix}, \tag{5.97}$$

$$\begin{bmatrix} \underline{W}_1^4(n) \\ \underline{W}_2^4(n) \\ \underline{W}_3^4(n) \\ \underline{W}_4^4(n) \end{bmatrix} = \begin{bmatrix} \underline{W}_1^3(n) \\ \underline{W}_2^3(n) \\ \underline{W}_3^3(n) \\ \underline{W}_4^3(n) \end{bmatrix} - 2\mu \begin{bmatrix} 0 \\ 0 \\ \underline{H}_{34}\underline{E}_4(n) \\ \underline{H}_{44}\underline{E}_4(n) \end{bmatrix}. \tag{5.98}$$

Note that the updated filter coefficient vector of the last node of the network at iteration n-th, $\underline{W}^4(n)$, can be computed as follows

$$\begin{aligned} \underline{W}_1^4(n) &= \underline{W}_1^0(n) - 2\mu(\underline{H}_{11}\underline{E}_1(n) + \underline{H}_{12}\underline{E}_2(n)), \\ \underline{W}_2^4(n) &= \underline{W}_2^0(n) - 2\mu(\underline{H}_{21}\underline{E}_1(n) + \underline{H}_{22}\underline{E}_2(n) + \underline{H}_{23}\underline{E}_3(n)), \\ \underline{W}_3^4(n) &= \underline{W}_3^0(n) - 2\mu(\underline{H}_{32}\underline{E}_2(n) + \underline{H}_{33}\underline{E}_3(n) + \underline{H}_{34}\underline{E}_4(n)), \\ \underline{W}_4^4(n) &= \underline{W}_4^0(n) - 2\mu(\underline{H}_{43}\underline{E}_3(n) + \underline{H}_{44}\underline{E}_4(n)). \end{aligned} \tag{5.99}$$

On the other hand, by using the proposed cl-DMEFxLMS approach, (5.84) can be calculated as

$$\begin{bmatrix} \underline{W}_1^1(n) \\ \underline{W}_2^1(n) \end{bmatrix} = \begin{bmatrix} \underline{W}_1^2(n-1) \\ \underline{W}_2^3(n-1) \end{bmatrix} - 2\mu \begin{bmatrix} \underline{H}_{11}\underline{E}_1(n) \\ \underline{H}_{21}\underline{E}_1(n) \end{bmatrix}, \tag{5.100}$$

$$\begin{bmatrix} \underline{W}_1^2(n) \\ \underline{W}_2^2(n) \\ \underline{W}_3^2(n) \end{bmatrix} = \begin{bmatrix} \underline{W}_1^1(n) \\ \underline{W}_2^1(n) \\ \underline{W}_3^4(n-1) \end{bmatrix} - 2\mu \begin{bmatrix} \underline{H}_{12}\underline{E}_2(n) \\ \underline{H}_{22}\underline{E}_2(n) \\ \underline{H}_{32}\underline{E}_2(n) \end{bmatrix}, \tag{5.101}$$

$$\begin{bmatrix} \underline{W}_2^3(n) \\ \underline{W}_3^3(n) \\ \underline{W}_4^3(n) \end{bmatrix} = \begin{bmatrix} \underline{W}_2^2(n) \\ \underline{W}_3^2(n) \\ \underline{W}_4^4(n-1) \end{bmatrix} - 2\mu \begin{bmatrix} \underline{H}_{23}\underline{E}_3(n) \\ \underline{H}_{33}\underline{E}_3(n) \\ \underline{H}_{43}\underline{E}_3(n) \end{bmatrix}, \tag{5.102}$$

$$\begin{bmatrix} \underline{W}_3^4(n) \\ \underline{W}_4^4(n) \end{bmatrix} = \begin{bmatrix} \underline{W}_3^3(n) \\ \underline{W}_4^3(n) \end{bmatrix} - 2\mu \begin{bmatrix} \underline{H}_{34}\underline{E}_4(n) \\ \underline{H}_{44}\underline{E}_4(n) \end{bmatrix}. \tag{5.103}$$

Considering that each subset of nodes follows an incremental learning, it is fulfilled that $\underline{W}_1^2(n-1) = \underline{W}_1^0(n)$, $\underline{W}_2^3(n-1) = \underline{W}_2^0(n)$, $\underline{W}_3^4(n-1) = \underline{W}_3^0(n)$ and

$\underline{W}_4^4(n-1) = \underline{W}_4^0(n)$, Then, the updated filter coefficient vector of the last node of each of the four subsets at iteration n-th can be calculated as

$$\begin{aligned}
\underline{W}_1^2(n)&=\underline{W}_1^0(n)-2\mu(\underline{H}_{11}\underline{E}_1(n)+\underline{H}_{12}\underline{E}_2(n)),\\
\underline{W}_2^3(n)&=\underline{W}_2^0(n)-2\mu(\underline{H}_{21}\underline{E}_1(n)+\underline{H}_{22}\underline{E}_2(n)+\underline{H}_{23}\underline{E}_3(n)),\\
\underline{W}_3^4(n)&=\underline{W}_3^0(n)-2\mu(\underline{H}_{32}\underline{E}_2(n)+\underline{H}_{33}\underline{E}_3(n)+\underline{H}_{34}\underline{E}_4(n)),\\
\underline{W}_4^4(n)&=\underline{W}_4^0(n)-2\mu(\underline{H}_{43}\underline{E}_3(n)+\underline{H}_{44}\underline{E}_4(n)),
\end{aligned} \tag{5.104}$$

that it is equal to the solution obtained in (5.99). Note that, the global vector $\underline{\hat{\mathbf{W}}}(n)$ is calculated as $\underline{\hat{\mathbf{W}}}(n)=[\underline{W}_1^2(n)\ \underline{W}_2^3(n)\ \underline{W}_3^4(n)\ \underline{W}_4^4(n)]^T$. Therefore, we can conclude that, in the cl-DMEFxLMS strategy, the last node of each subset will tend to converge to the last node of the global network (in the equivalent cm-DMEFxLMS case) and therefore, to the optimal global solution of the network.

Following with the *Case b*, considering that convergence is achieved, the error signal at the k-th node can be defined in the steady-state as

$$\underline{E}_k(n) = \underline{D}_k(n) + \underline{\mathbf{H}}_k^H \underline{\hat{\mathbf{W}}}(n). \tag{5.105}$$

Under these conditions, (5.93) can be expressed as

$$\underline{\hat{\mathbf{W}}}^k(n)=\underline{\hat{\mathbf{W}}}^{k-1}(n)-2\mu(\underline{\hat{\mathbf{H}}}_k\underline{D}_k(n)+\underline{\hat{\mathbf{H}}}_k\underline{\mathbf{H}}_k^H\underline{\hat{\mathbf{W}}}(n-1)). \tag{5.106}$$

Assuming that, when $f\to\infty$, it is fulfilled that $\underline{\hat{\mathbf{W}}}_o^k=\underline{\hat{\mathbf{W}}}_o^{k-1}$ and therefore, $\underline{\hat{\mathbf{W}}}_o^k\to\underline{\hat{\mathbf{W}}}_o$. Consequently, as proved in *Case A*, $\underline{\hat{\mathbf{W}}}_o\to\mathbf{W}_o$, and then

$$\underline{\hat{\mathbf{H}}}_k\underline{D}_k(n) = -\underline{\hat{\mathbf{H}}}_k\underline{\mathbf{H}}_k^H\underline{\hat{\mathbf{W}}}_o. \tag{5.107}$$

Under convergence conditions, substracting $\underline{\hat{\mathbf{W}}}_o$ from both sides of (5.106) and using (5.107), we obtain

$$\underline{\hat{\mathbf{W}}}^k(n)-\underline{\hat{\mathbf{W}}}_o=\underline{\hat{\mathbf{W}}}^{k-1}(n)-\underline{\hat{\mathbf{W}}}_o-2\mu\underline{\hat{\mathbf{H}}}_k\underline{\mathbf{H}}_k^H(\underline{\hat{\mathbf{W}}}(n-1)-\underline{\hat{\mathbf{W}}}_o). \tag{5.108}$$

Considering that, in an incremental strategy, it is fulfilled that $\underline{\hat{\mathbf{W}}}(n-1)=\underline{\hat{\mathbf{W}}}^0(n)$ where $\underline{\hat{\mathbf{W}}}^0(n)$ is the vector which contains the first node of the N subsets,

$$\underline{\hat{\mathbf{W}}}^0(n)=[\ [\underline{W}_1^{C_1(1)}(n)]^T[\underline{W}_2^{C_2(1)}(n)]^T\cdots[\underline{W}_N^{C_N(1)}(n)]^T\ ]^T. \tag{5.109}$$

Defining $\underline{\hat{\mathbf{V}}}^k(n)=E\{\underline{\hat{\mathbf{W}}}^k(n)-[\underline{\hat{\mathbf{W}}}]_o\}$, (5.108) can be defined as

$$\underline{\hat{\mathbf{V}}}^k(n) = \underline{\hat{\mathbf{V}}}^{k-1}(n) - 2\mu\underline{\hat{\mathbf{H}}}_k\underline{\mathbf{H}}_k^H\underline{\hat{\mathbf{V}}}^0(n). \tag{5.110}$$

Taking into account (5.85) and (5.109), note that the previous equation can be computed as

$$\begin{bmatrix} \underline{V}^{k}_{C_k(1)}(n) \\ \underline{V}^{k}_{C_k(2)}(n) \\ \vdots \\ \underline{V}^{k}_{C_k(N_k)}(n) \end{bmatrix} = \begin{bmatrix} \underline{V}^{k_1}_{C_k(1)}(n) \\ \underline{V}^{k_2}_{C_k(2)}(n) \\ \vdots \\ \underline{V}^{k_{N_k}}_{C_k(N_k)}(n) \end{bmatrix} - 2\mu \begin{bmatrix} \underline{H}_{C_k(1)}\underline{H}^{H}_{1k} & \underline{H}_{C_k(1)}\underline{H}^{H}_{2k} & \cdots & \underline{H}_{C_k(1)}\underline{H}^{H}_{Nk} \\ \underline{H}_{C_k(2)}\underline{H}^{H}_{1k} & \underline{H}_{C_k(2)}\underline{H}^{H}_{2k} & \cdots & \underline{H}_{C_k(2)}\underline{H}^{H}_{Nk} \\ \vdots & \vdots & \ddots & \vdots \\ \underline{H}_{C_k(N_k)}\underline{H}^{H}_{1k} & \underline{H}_{C_k(N_k)}\underline{H}^{H}_{2k} & \cdots & \underline{H}_{C_k(N_k)}\underline{H}^{H}_{Nk} \end{bmatrix} \begin{bmatrix} \underline{V}^{C_1(1)}_{1}(n) \\ \underline{V}^{C_2(1)}_{2}(n) \\ \vdots \\ \underline{V}^{C_N(1)}_{N}(n) \end{bmatrix} \tag{5.111}$$

Therefore, the solution for the l-th node of the subset C_k, denoted as $\underline{V}^{k}_{C_k(l)}(n)$, can be expressed as

$$\underline{V}^{k}_{C_k(l)}(n) = \underline{V}^{k_l}_{C_k(l)}(n) - 2\mu \underline{H}_{C_k(l)k} \sum_{i=1}^{N} \underline{H}^{H}_{ik} \underline{V}^{C_i(1)}_{i}(n). \tag{5.112}$$

On the other hand, consider now the solution for the cm-DMEFxLMS algorithm, which can be obtained from (5.29) as

$$\underline{\mathbf{V}}^{k}(n) = \underline{\mathbf{V}}^{k-1}(n) - 2\mu \hat{\underline{\mathbf{S}}}_k \underline{\mathbf{H}}^{H}_{k} \underline{\mathbf{V}}^{0}(n). \tag{5.113}$$

and it can be split as

$$\begin{bmatrix} \underline{V}^{k}_{1}(n) \\ \underline{V}^{k}_{2}(n) \\ \vdots \\ \underline{V}^{k}_{N}(n) \end{bmatrix} = \begin{bmatrix} \underline{V}^{k-1}_{1}(n) \\ \underline{V}^{k-1}_{2}(n) \\ \vdots \\ \underline{V}^{k-1}_{N}(n) \end{bmatrix} - 2\mu \begin{bmatrix} \alpha_{1k}\underline{H}_{1k}\underline{H}^{H}_{1k} & \alpha_{1k}\underline{H}_{1k}\underline{H}^{H}_{2k} & \cdots & \alpha_{1k}\underline{H}_{1k}\underline{H}^{H}_{Nk} \\ \alpha_{2k}\underline{H}_{2k}\underline{H}^{H}_{1k} & \alpha_{2k}\underline{H}_{2k}\underline{H}^{H}_{2k} & \cdots & \alpha_{2k}\underline{H}_{2k}\underline{H}^{H}_{Nk} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{Nk}\underline{H}_{Nk}\underline{H}^{H}_{1k} & \alpha_{Nk}\underline{H}_{Nk}\underline{H}^{H}_{2k} & \cdots & \alpha_{Nk}\underline{H}_{Nk}\underline{H}^{H}_{Nk} \end{bmatrix} \begin{bmatrix} \underline{V}^{0}_{1}(n) \\ \underline{V}^{0}_{2}(n) \\ \vdots \\ \underline{V}^{0}_{N}(n) \end{bmatrix} \tag{5.114}$$

where the contribution of node k to the calculation of the coefficients to be used by node j is

$$\underline{V}^{k}_{j}(n) = \underline{V}^{k-1}_{j}(n) - 2\mu\alpha_{jk}\underline{H}_{jk} \sum_{i=1}^{N} \underline{H}^{H}_{ik} \underline{V}^{0}_{i}(n), \tag{5.115}$$

Comparing (5.112) and (5.115) and with the aim that both approaches reach the same expression, i.e., $\underline{V}^{k}_{C_k(l)}(n)=\underline{V}^{k}_{j}(n)$, we must prove that

$$\begin{array}{c} \underline{V}^{k_l}_{C_k(l)}(n)=\underline{V}^{k-1}_{j}(n), \\ \alpha_{jk}=1\,, \\ \underline{V}^{C_i(1)}_{i}(n)=\underline{V}^{0}_{i}(n). \end{array} \tag{5.116}$$

To fulfill these expressions, the following assumptions must be considered:

- To prove the expression $\underline{V}^{k_l}_{C_k(l)}(n)=\underline{V}^{k-1}_{j}(n)$, we can make use of the example previously described. Assume that node $C_k(l)$ and node j are the same, and, for instance, we select $\underline{W}^{k}_{j}(n)=\underline{W}^{k}_{C_k(l)}(n)=\underline{W}^{2}_{3}(n)$. Then, from (5.96) and (5.95), we compute that $\underline{W}^{k-1}_{j}(n)=\underline{W}^{1}_{3}(n)=\underline{W}^{0}_{3}(n)$. From (5.101), see that $\underline{W}^{k_l}_{C_k(l)}(n)=\underline{W}^{4}_{3}(n-1)$ is obtained. Since it has been proved in (5.104) that $\underline{W}^{4}_{3}(n-1)=\underline{W}^{0}_{3}(n)=$, it is fulfilled that $\underline{W}^{k_l}_{C_k(l)}(n)=\underline{W}^{k-1}_{j}(n)$.

- Since in (5.112), we consider that the node $C_k(l)$ is acoustically coupled with node k, then, we may assume that in (5.115), node j is within C_k, and then $\alpha_{jk}=1$.

- Note that, in (5.112), $\underline{V}^{C_i(1)}_{i}(n)=\underline{V}^{C_i(N_i)}_{i}(n-1)$ ant taking into account that, when convergence is achieved, the last node of the subset C_i will tend to converge to the last node of the global network, then it is fulfilled that $\underline{V}^{C_i(1)}_{i}(n) = \underline{V}^{N}_{i}(n-1)$. From (5.115), note that $\underline{V}^{0}_{i}(n)=\underline{V}^{N}_{i}(n-1)$. This means that the expression $\underline{V}^{C_i(1)}_{i}(n)=\underline{V}^{0}_{i}(n)$ can be considered.

Therefore, it has been proved that the local version of the global solution reached for node k-th by using the cl-DMEFxLMS strategy is the same as the local version of the global cm-DMEFxLMS solution at the same node. This means that both approaches achieve the same stability conditon.

5.6.3 Simulation results

In this section, we present the simulations carried out to evaluate the performance of the proposed approach over distributed networks with no communication constraints. To this end, we analyze the behavior of the network in three cases: 1) when nodes are updated only with its own information, 2) when nodes are updated with the information of the whole network and finally, 3) when nodes are updated only with the information coming from a subset of nodes. More specifically, we have compared the NC-DFxLMS, the DMEFxLMS and the cl-DMEFxLMS algorithms, respectively, in

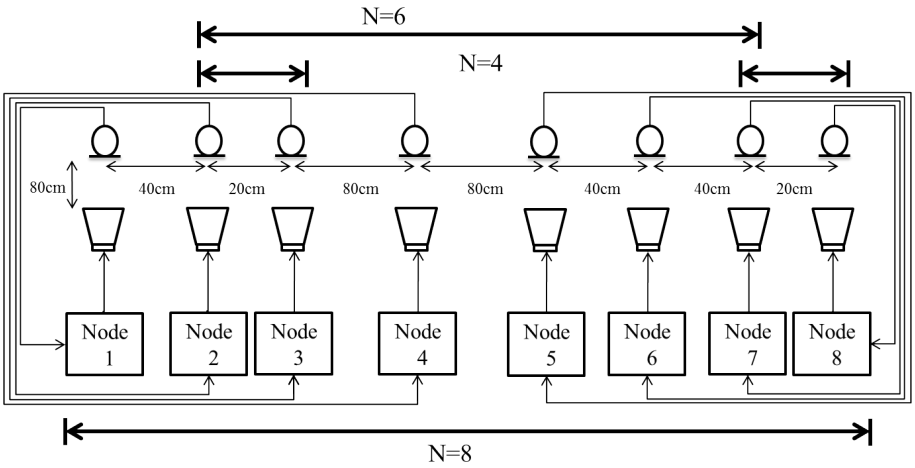


Figure 5.26: Different ASN of N-nodes for an ANC system. Nodes selected for each presented ASN are indicated.

terms of final noise reduction, convergence behavior, computational complexity and communication requirements. All the algorithms have been tested in ASNs composed of four, six and eight single-channel nodes, respectively. The configuration of the simulated ASNs is depicted in Figure 5.26. The nodes selected for each presented ASN are: nodes $2, 3, 7$ and 8 for N=4, nodes 2 to 7 for N=6, and nodes 1 to 8 for N=8. A ring topology with incremental learning has been used at each ASN. Only the nodes with the best and the worst performance are shown in the simulations in order to assess the behavior of the ASNs. The performance of the other nodes remains within this range. For the cl-DMEFxLMS algorithm results, the different subset of nodes for the three ASNs are designed based on the method 4 depicted in Appendix A, where nodes acoustically coupled are clustered. For the designed ASNs, we consider the simulation configuration explained in Section 3.3. Regarding the step-size parameter at each node, with the aim to ensure similar convergence rates for the evaluated solutions, we consider that μ_k=μ/N_k where μ is the step-size used by the NC-DFxLMS algorithm for the estimation of $w_k(n)$. By trial and error, a value of μ=0.05 has been considered as the highest value that ensures the stability of the ANC system in all the ASNs.

In the first simulation, the performance of the proposed cl-DMEFxLMS algorithm over a four-node ASN compared to the NC-DFxLMS and the DMEFxLMS algorithms has been evaluated. Based on the collaborative method depicted in Section 3.1, the creation of the subsets of nodes is defined as follows: $C_2 = \{2, 3\}$, $C_3 = \{2, 3\}$, $C_7 = \{7, 8\}$ and $C_8 = \{7, 8\}$. Note that, due to the acoustical interac-

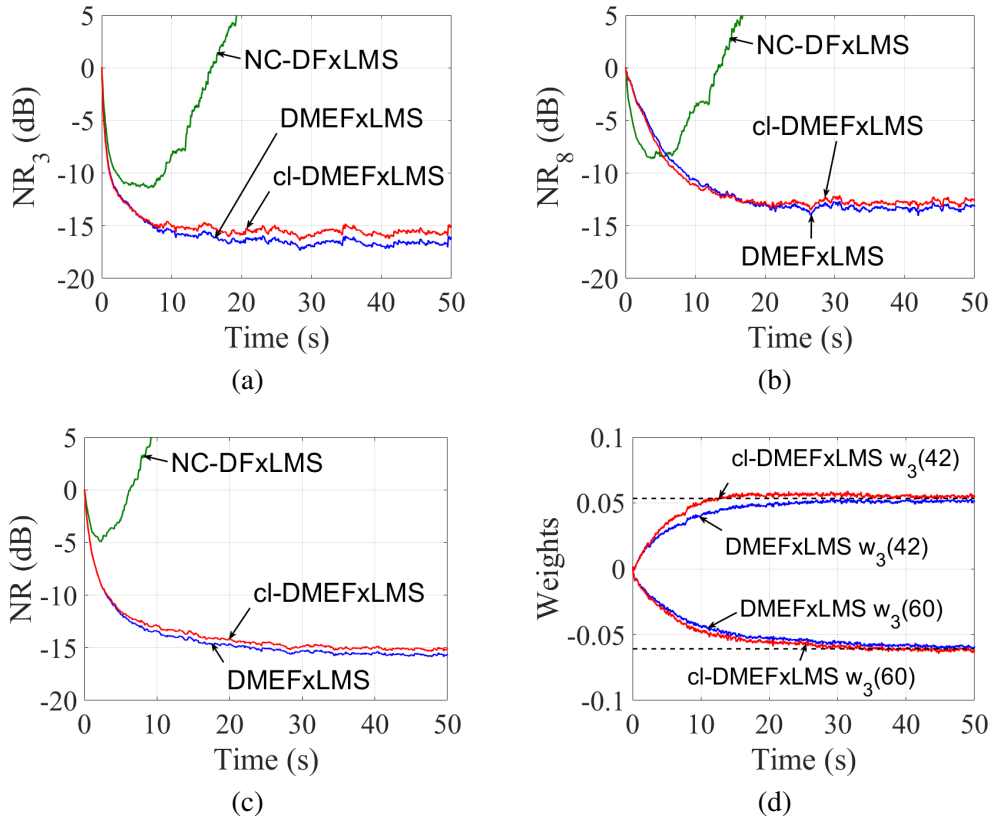


Figure 5.27: Behavior of the NC-DFxLMS, the DMEFxLMS, and the cl-DMEFxLMS algorithms in a four-node ASN: Time evolution of the noise reduction obtained (a) for the best node, (b) for the worst node and (c) in the network. (d) Time evolution of two filter coefficients at the node with the best performance.

tion, both subsets may be viewed as two independent non-collaborative subsystems. Figures 5.27.(a) and 5.27.(b) show the $NR_k(n)$ for the distributed algorithms at the nodes with the best and the worst performance, respectively. It can be observed that the non-collaborative algorithm makes the ANC system unstable while both collaborative strategies show a stable and similar performance (not exceeding 1.2 dB of difference), providing around 16 dB of noise reduction for the best node and more than 13 dB for the worst. In Figure 5.27.(c), the $NR(n)$ in the network for the NC-DFxLMS, the DMEFxLMS and the cl-DMEFxLMS algorithms is presented. The global performance of the algorithms presents similar results as the previous figures. Moreover, in order to analyze the convergence speed of the collaborative strategies, the time evolution of two filter coefficients at the node with the best performance is shown in 5.27.(d). Dashed lines represent the theoretical coefficients. Since the

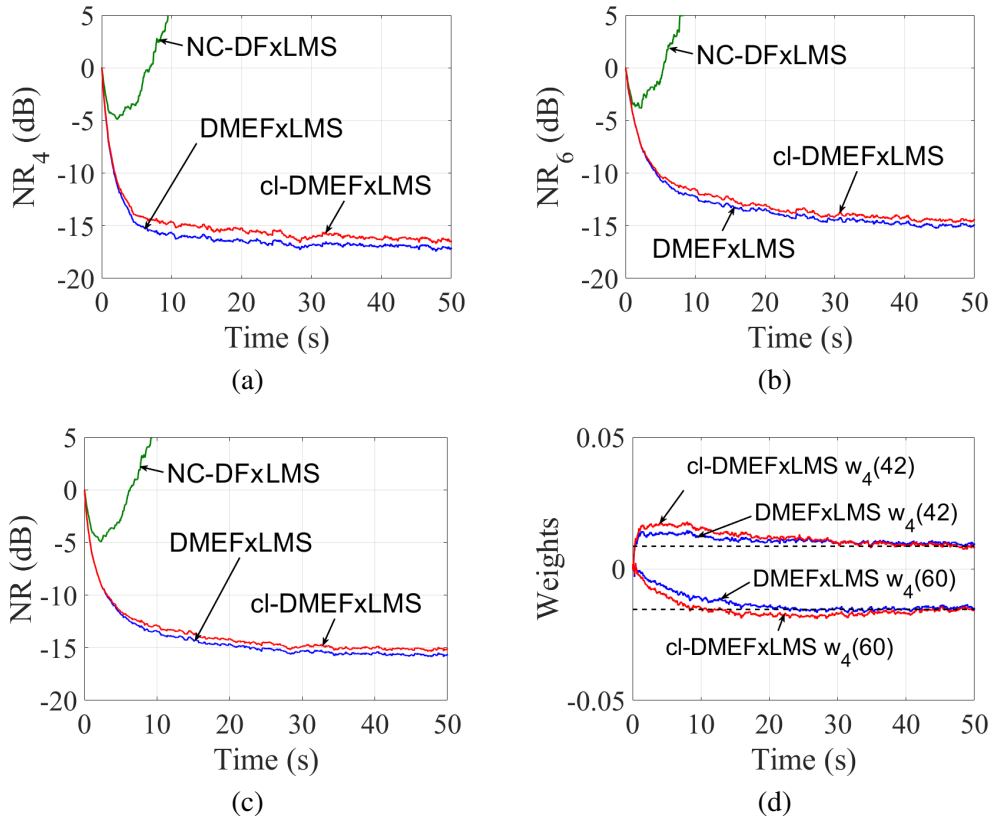


Figure 5.28: Behavior of the NC-DFxLMS, the DMEFxLMS, and the cl-DMEFxLMS algorithms in a six-node ASN: Time evolution of the noise reduction obtained (a) for the best node, (b) for the worst node and (c) in the network. (d) Time evolution of two filter coefficients at the node with the best performance.

NC-DFxLMS algorithm exhibits an unstable behavior, their results have not been included in 5.27.(d) (neither in the following figures related to the weights evolution). It can be seen that the proposed cl-DMEFxLMS strategy shows a slighty improvement in the convergence compared to DMEFxLMS.

In the second simulation, we evaluate the noise reduction and the convergence speed of the distributed ANC algorithms over a six-node ASN. In this case, the subsets of nodes are defined as follows: $C_2 = \{2,3,4\}$, $C_3 = \{2,3,4\}$, $C_4 = \{2,3,4,5,6\}$, $C_5 = \{4,5,6,7\}$, $C_6 = \{4,5,6,7\}$ and $C_7 = \{5,6,7\}$. As it can be seen in Figure 5.28.(a) and 5.28.(b), while both the DMEFxLMS and cl-DMEFxLMS algorithms exhibit a robust and stable convergence, the NC-DFxLMS algorithm diverges for the best and the worst node of the network. Collaborative algorithms present almost the same behaviour in terms of noise attenuation levels and conver-

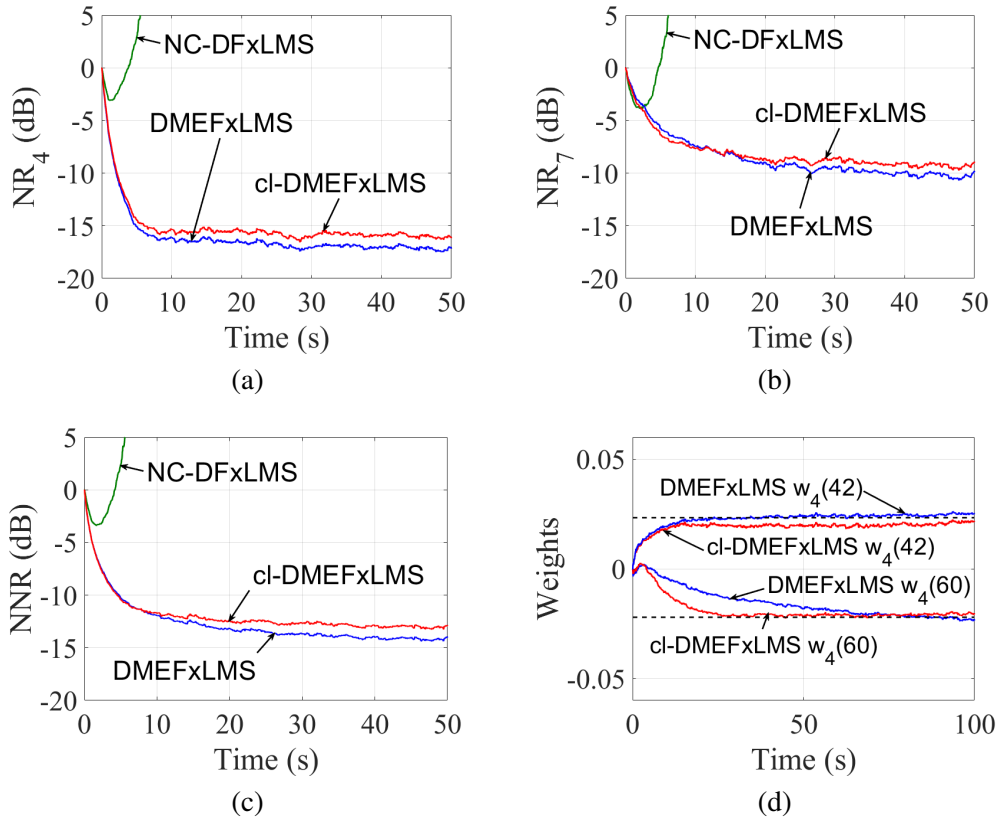


Figure 5.29: Behavior of the NC-DFxLMS, the DMEFxLMS, and the cl-DMEFxLMS algorithms in a eight-node ASN: Time evolution of the noise reduction obtained (a) for the best node, (b) for the worst node and (c) in the network. (d) Time evolution of two filter coefficients at the node with the best performance.

gence speed achieving around 17 dB and 14 dB of noise reduction at the nodes with the best and the worst performance, respectively. Similar results are obtained from Figure 5.28.(c) whose $NR(n)$ fluctuates around 15 dB. Regarding the time evolution of the filter coefficients, as it can be seen in Figure 5.28.(d), both the DMEFxLMS and cl-DMEFxLMS approaches achieve similar results, with as slight improvement in the convergence of DMEFxLMS compared to cl-DMEFxLMS.

Finally, in the third simulation, we present the results obtained for the NC-DFxLMS, the DMEFxLMS, and the cl-DMEFxLMS algorithms considering the eigth-node ASN. Nodes have been clustered as follows: $C_1 = \{1, 2, 3\}$, $C_2 = \{1, 2, 3, 4\}$, $C_3 = \{1, 2, 3, 4\}$, $C_4 = \{2, 3, 4, 5, 6\}$, $C_5 = \{4, 5, 6, 7\}$, $C_6 = \{4, 5, 6, 7, 8\}$, $C_7 = \{5, 6, 7, 8\}$ and $C_8 = \{6, 7, 8\}$. As the previous simulations, the NC-DFxLMS algorithm presents an unstable behaviour for each node of the network, as shown in

Table 5.4: Total number of multiplications (MUX) and data transfer per iteration regarding (1) the computational complexity and (2) the communication requirements of the algorithms, respectively. L: length of the adaptive filters; M: length of the acoustic paths; N: number of nodes; N_k: number of nodes associated with node k. As example, ASNs depicted in Figure 5.26 with L=150, M=256 have been evaluated.

	Algorithms	Generic (at node k-th)	Four-node ASN	Six-node ASN	Eight-node ASN
(1)	DMEFxLMS	$L(1+N)+MN+1$	7100	15522	27192
	cl-DMEFxLMS	$L(1+N_k)+MN_k+1$	3852	9838	14200
(2)	DMEFxLMS	$2LN(N-1)$	3600	9000	16800
	cl-DMEFxLMS	$\sum_{k=1}^{N} LN_k$	1200	3300	4800

Figures 5.29.(a), 5.29.(b) and 5.29.(c). We can see in Figure 5.29.(a) that both collaborative approaches present a good performance at the transient state for the best node of the network achieving a final residual noise around 17 dB. As shown in Figure 5.29.(b), the worst node of the ASN presents performance degradation for both collaboratives strategies in comparison to the previous simulations, providing an attenuation up to 10 dB. Figure 5.29.(c) shows the $NR(n)$ obtained for the eight-node ASN. As previous cases, the cl-DMEFxLMS algorithm achieves a robust behaviour, similar as the DMEFxLMS approach in terms of noise reduction. Regarding the convergence speed, both algorithms present a similar behaviour in general terms although in Figure 5.29.(d) it can be seen that the weights of the proposed algorithm reaches the steady-state a slighty faster than the coefficients of the DMEFxLMS algorithm.

Table 5.4 compares the computational complexity (in terms of multiplications per iteration) and the communication requirements (data transfer) of the distributed ANC algorithms. The transmitted filter coefficients will be proportionally related to the transmitted bits depending on the used coding. The NC-DFxLMS algorithm has not been included in the table because of its unstability performance in all the designed ASNs. To this end, we consider a network of N single-channel nodes. The reference signal has not been considered in the calculation of the data transfer. Note that, for the DMEFxLMS algorithm, the computational complexity depends on L, M and N while the communication requirements only depends on L and N. However, the cl-DMEFxLMS algorithm does not depend on the size of the network, N, but on the size of each subset of nodes, N_k. Both implementation aspects are particularized for the four-node, six-node and eight-node ASNs (N=4, N=6 and N=8 respectively). The values of N_k for each ASNs can be extracted from the definition of subsets commented in the previous simulations. Results show that the computational cost of the DMEFxLMS algorithm is higher than the proposed version increasing significatively

with the number of nodes. Considering the cl-DMEFxLMS algorithm, note that, at each time instant, each node k only needs to update N_k vectors whose dimensions are independent of N. Regarding the communication needs, the DMEFxLMS strategy also requires higher requirements as shown in Table 5.4. As an example, for an incremental-learning N-node network, the DMEFxLMS method needs that every node transfers $LN\times1$ coefficients to the following node $2(N-1)$ times in each iteration (see Figure 5.2.(b) with $N=6$). However, using the cl-DMEFxLMS algorithm, only a data stream of LN_k samples is exchanged between the nodes that belong to each cluster, reducing the data transfer of the network, as shown in Figure 5.30. In other words, at each time instant, each node k only transmits the local estimates of N_k ANC filters, whose dimensions again do not depend on the number of nodes. Following a practical implementation example, we consider the incremental ASN composed of 6 nodes ($N=6$). In the DMEFxLMS strategy, every node must transfer $L\times N$ coefficients to the following node and this has to be repeated along the network $2(N-1)$ times in each sample iteration, being L the size of the adaptive filter. Using the single-precision floating-point format (that occupies 4 bytes) and considering a real scenario (as described in Section 5.2.3) where $L=4096$, we obtain the amount of transferred data of the DMEFxLMS algorithm through four nodes, t_d^D , as follows: $t_d^D=(L\times N)\times2(N-1)\times4$ bytes $=384$ kB (kilobytes). However, note that, the transferred data of the clustering strategy, t_d^{cD}, is reduced as follows: $t_d^{cD}=\sum_{k=1}^{N} LN_k\times4$ bytes $=128$ kB. Considering the 44.1 kHz audio sampling rate and a size block of 2048 samples, note that the DMEFxLMS algorithm would need a transfer rate of 40.4 megabytes per second (MBps), on the incremental six-node ASN. However, the cl-DMEFxLMS algorithm would need a transfer rate of at least 14.8 MBps. Therefore, in the last case, we could use a network of 15 MBps to perform the required data transfer among the nodes.

5.6.4 Conclusions

In this section, with the aim to reduce the communication and computational network requirements of distributed ANC systems, a new approach has been derived from the DMEFxLMS algorithm to ensure ANC system stability in acoustically coupled sensor networks. This alternative strategy, denoted as cl-DMEFxLMS algorithm, is based on the NSPE technique and brings together only the acoustically coupled nodes of distributed networks without communication constraints. A theoretical analysis of the stability of the distributed algorithms using the node-specific technique for ANC systems has been carried out. More specifically, the mean performance of the proposed algorithm at each node has been analyzed for a network with a ring topology and with no communication constraints. It has been proved that, if the positive step-sizes fulfills several assumptions, the resulting estimates of the proposed strategy converge in the mean to the centralized solution. We have carried out simulations to

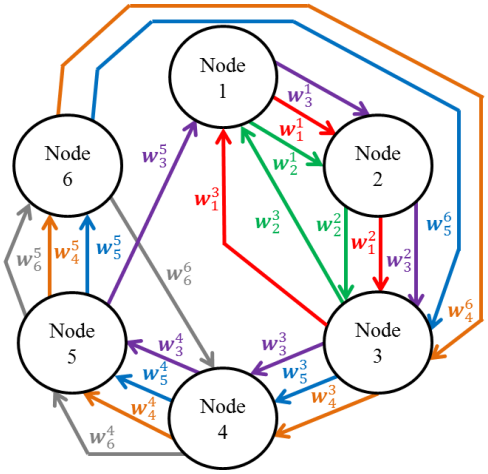


Figure 5.30: A ring topology distributed six-node ASN with incremental communication using the cl-DMEFxLMS algorithm. Data transfer rounds between nodes of the same subset are represented with different colour of lines.

show the behaviour of distributed approaches, such as the NC-DFxLMS, the DMEFxLMS and the cl-DMEFxLMS algorithms, in different ideal ASNs. Results show that collaboration is required in order to ensure the stability of the distributed ANC system over the selected partially-coupled networks. The proposed strategy exhibits similar performance as the DMEFxLMS algorithm in terms of noise reduction reducing significantly the computational complexity and the data transfer of the network. Moreover, unlike any other distributed algorithm for ANC, it has been proved that the cl-DMEFxLMS algorithm is scalable with the network size in terms of computational cost and communication requirements. It can also be observed that, depending on the nodes location and the number of nodes, the solution reached for each subset of nodes by using the cl-DMEFxLMS strategy may slightly outperform the DMEFxLMS solution in terms of convergence speed, improving the behaviour of the global ANC system. However, note that the advantages of this technique will depend on how the loudspeakers and microphones of each node are located and therefore, how the network is defined. In addition, a more selective routing management of shared data may be required. These aspects must be considered for future practical implementations.

5.7 Collaborative distributed algorithm using remote sensing technique

ANC headrest systems have been developed to minimize the noise level at listener's ears avoiding the comfort problem of active headphones [128–130]. Generally, two actuators are located close to the headrest to generate the anti-noise signals while, in order to capture the error signals, two sensors located close to the listener's ears are used. Depending on the control strategy, a reference microphone signal (feedforward control) [131] or an internally estimated signal (feedback control) [132] is used to monitor the undesired noise. The resulting quiet zones located around the error sensors are generally limited in size and they are approximated as a sphere of diameter one tenth of the wavelength of the highest frequency of the noise [66]. Out of these zones, noise level may even increase [67]. This requires the error sensors be located as close as possible to the listener's ears, which is often uncomfortable. With the aim to avoid the direct installation of the error sensors in the local quiet zone, the addition of the remote microphone technique [133] to the ANC headrest systems has been recently analyzed in several works [134–137] as well as it has been investigated during the last decades by several authors, as reviewed in [138]. The remote microphone (RM) technique was developed to generate quiet zones at virtual locations. To this end, the control system minimizes the signals that would be picked up at the virtual sensors by using the signals captured from a set of physical sensors, called monitoring sensors and located out of the control zone. By using pre-measured transfer functions, the RM technique is able to estimate the primary disturbance at the virtual sensors from the primary disturbance at the monitoring sensors.

On the other hand, the final objective of ANC headrest systems is to be installed in the cabin of a public transport to create quiet zones in all the occupants positions. Due to the use of many of these systems working at the same time, a distributed approach can be desirable in order to provide more flexible, versatile, and scalable ANC systems. Consequently, we consider a feedforward personal active control (PAC) system, i.e., a two-channel distributed ANC system composed of a car seat and a sound control node (SCN). A SCN is an acoustic device composed of two loudspeakers to generate acoustic signals, two microphones to obtain acoustical information, and one processor to individually process that information and to interchange it with the other SCN using a suitable communication network. Collaboration among SCN might be needed with the aim to minimize the effects of the acoustic coupling and ensuring the stability of several multichannel ANC systems working together. However, the implementation of an ANC headrest system using the RM technique over distributed networks has not been already aborded and it requires certain considerations to be addressed. For this purpose, we present a distributed ANC headrest system based on the MEFxLMS algorithm combined with the RM technique and using an incremental communication strategy in a ring network. This strategy is denoted as RM-

Table 5.5: Notation for the remote microphone technique

$e_{\mathrm{m},\mathrm{k_m}}(n)$:	error signal at the monitoring sensor k_m.
$e_{\mathrm{v},\mathrm{k_v}}(n)$:	error signal at the virtual sensor k_v.
$d_{\mathrm{m},\mathrm{k_m}}(n)$:	acoustic noise signal at the monitoring sensor k_m.
$d_{\mathrm{v},\mathrm{k_v}}(n)$:	acoustic noise signal at the virtual sensor k_v.
$\mathbf{p}_{\mathrm{m},\mathrm{k_m}}$, $\mathbf{p}_{\mathrm{v},\mathrm{k_v}}$:	acoustic channels between the noise source and sensors k_m and k_v, respectively.
$\mathbf{h}_{\mathrm{m},\mathrm{jk_m}}$, $\mathbf{h}_{\mathrm{v},\mathrm{jk_v}}$:	acoustic channels between actuator j and sensors k_m and k_v, respectively.
$\tilde{\mathbf{h}}_{\mathrm{m},\mathrm{jk_m}}$, $\tilde{\mathbf{h}}_{\mathrm{v},\mathrm{jk_v}}$:	impulse responses defined as FIR filters that estimates $\mathbf{h}_{\mathrm{m},\mathrm{jk_m}}$ and $\mathbf{h}_{\mathrm{v},\mathrm{jk_v}}$, respectively.
K_m:	number of monitoring sensors.
K_v:	number of virtual sensors.

DMEFxLMS algorithm. In order to obtain the same performance as the centralized fashion (RM-MEFxLMS), every node in the distributed network must have access to the control signals of the rest of the nodes, which implies not fully distributed and independent processing at all. In the proposed RM-DMEFxLMS algorithm, each node estimates the control signals of the network by using a local version of the global adaptive filter, which is updated and exchanged between the nodes following an incremental learning. The performance of the distributed implementation compared to its centralized version over a network composed of two PAC systems is evaluated. A study of implementation aspects such as computational complexity and communication capabilities among the nodes in the network is also presented. In addition and in order to save computational load and to reduce the data transfer of the network, two variations of the distributed algorithm are proposed.

It should be noted that, since the application of this technique over distributed ANC systems requires the use of a different structure, its derivation from the generic formulation presented in 3.2 has not been aborded.

5.7.1 Distributed RM-MEFxLMS algorithms

Let us consider a multichannel feedforward ANC system combined with the remote microphone technique working over a homogeneous acoustic sensor network (ASN) of N nodes spatially distributed in some area, as shown in Figure 5.31. We assume that all nodes are composed of a single actuator, execute the same algorithm, and share the same reference signal, $x(n)$, captured by a reference sensor used to detect a single disturbance noise at the discrete time instant n. Initially, we consider that each node receives the information picked up by one or several monitoring sensors and aims to control the sound field in one or several virtual sensors. In this regard, a total of K_m monitoring sensors and K_v virtual sensors are controlled by all the nodes of the network. In a preliminary stage, nodes will gather the information measured

in K_{m} monitoring sensors and in the physical location where K_{v} virtual sensors are desired to be positioned. The reason will be explained below. Our objective is to estimate an adaptive filter $\mathbf{w}_j(n)$ at every node to cancel the acoustic noise signal at the virtual sensor locations, $\mathbf{d}_{\mathrm{v}}(n) = [d_{\mathrm{v},1}(n)\ d_{\mathrm{v},2}(n)\ \ldots\ d_{\mathrm{v},\mathrm{K_v}}(n)\ ]^T$ where $j = 1, 2, \ldots, N$. To that end, the control signals $y_j(n)$, emitted by the actuators and filtered through the acoustic system, are designed to minimize the signals that would be picked up at the virtual sensors, called virtual error signals and denoted by $\mathbf{e}_{\mathrm{v}}(n) = [e_{\mathrm{v},1}(n)\ e_{\mathrm{v},2}(n)\ \ldots\ e_{\mathrm{v},\mathrm{K_v}}(n)\ ]^T$. Since the actual error signals at the virtual error sensors cannot be directly measured, the remote microphone technique considers that the virtual sensor signals can be estimated from the signals at a set of monitoring sensors, $\mathbf{e}_{\mathrm{m}}(n) = [e_{\mathrm{m},1}(n)\ e_{\mathrm{m},2}(n)\ \ldots\ e_{\mathrm{m},\mathrm{K_m}}(n)\ ]^T$. $\mathbf{h}_{\mathrm{m,j}\mathrm{k_m}}$ and $\mathbf{h}_{\mathrm{v,j}\mathrm{k_v}}$ are the acoustic channels between actuator j and sensors k_{m} and k_{v}, respectively (where $k_{\mathrm{m}} = 1, 2, \ldots, K_{\mathrm{m}}$ and $k_{\mathrm{v}} = 1, 2, \ldots, K_{\mathrm{v}}$). Table 5.5 summarizes the additional notation for the remote microphone technique applied to ANC. With the aim to achieve this objective, we start from the known centralized adaptive approach to derive into the proposed distributed solution using an incremental strategy of the data exchange.

5.7.1.1. Centralized RM-MEFxLMS algorithm

Figure 5.32 illustrates the block diagram of a multichannel centralized ANC system based on the FxLMS algorithm and combined with the remote microphone technique. $\mathbf{p}_{\mathrm{v}}$ and $\mathbf{p}_{\mathrm{m}}$ are the acoustic paths that link the noise source with all the virtual sensors and with all the monitoring sensors, respectively. In the same way, the acoustic paths that link all the actuators with all the virtual sensors and with all the monitoring sensors are depicted as $\mathbf{h}_{\mathrm{v}}$ and $\mathbf{h}_{\mathrm{m}}$, respectively.

Since the actual information captured by the virtual sensors, $\mathbf{e}_{\mathrm{v}}(n)$, is not available, it can be estimated from the signals captured by the monitoring error sensors, $\mathbf{e}_{\mathrm{m}}(n)$, defined as

$$\mathbf{e}_{\mathrm{m}}(n) = \mathbf{d}_{\mathrm{m}}(n) + \mathbf{U}_{\mathrm{m}}^T(n)\mathbf{w}(n), \tag{5.117}$$

with

$$\begin{aligned} &\mathbf{d}_{\mathrm{m}}(n) = [d_{\mathrm{m},1}(n)\ d_{\mathrm{m},2}(n)\ \ldots\ d_{\mathrm{m},\mathrm{K_m}}(n)\ ]^T \text{ of size } [K_{\mathrm{m}}\times 1], \\ &\mathbf{U}_{\mathrm{m}}(n) = [\mathbf{u}_{\mathrm{m},1}(n)\ \mathbf{u}_{\mathrm{m},2}(n)\ \ldots\ \mathbf{u}_{\mathrm{m},\mathrm{K_m}}(n)\ ] \text{ of size } [LN\times K_{\mathrm{m}}], \end{aligned} \tag{5.118}$$

where $\mathbf{d}_{\mathrm{m}}(n)$ is a K_{m}-lenght vector of disturbance signals at the monitoring error sensors and $\mathbf{u}_{\mathrm{m},\mathrm{k_m}}(n)$=$[\mathbf{h}_{\mathrm{m},1\mathrm{k_m}}^T\mathbf{X}^T(n),\ \mathbf{h}_{\mathrm{m},2\mathrm{k_m}}^T\mathbf{X}^T(n),\ \ldots, \mathbf{h}_{\mathrm{m},\mathrm{Nk_m}}^T\mathbf{X}^T(n)]^T$ is a $[LN\times 1]$ vector composed of the reference signal $x(n)$ filtered through the acoustic channels between the N loudspeakers and the k_{m}-th monitoring sensor, $\mathbf{h}_{\mathrm{m,j}\mathrm{k_m}} = [h_{\mathrm{m,j}\mathrm{k_m},1}\ h_{\mathrm{m,j}\mathrm{k_m},2}\ \ldots\ h_{\mathrm{m,j}\mathrm{k_m},\mathrm{M}}\ ]^T$, being $\mathbf{X}(n)$ as defined in (3.3). Vector $\mathbf{w}(n)$ concatenates the N vectors $\mathbf{w}_j(n)$, which contains the L filter coefficients of the jth node.

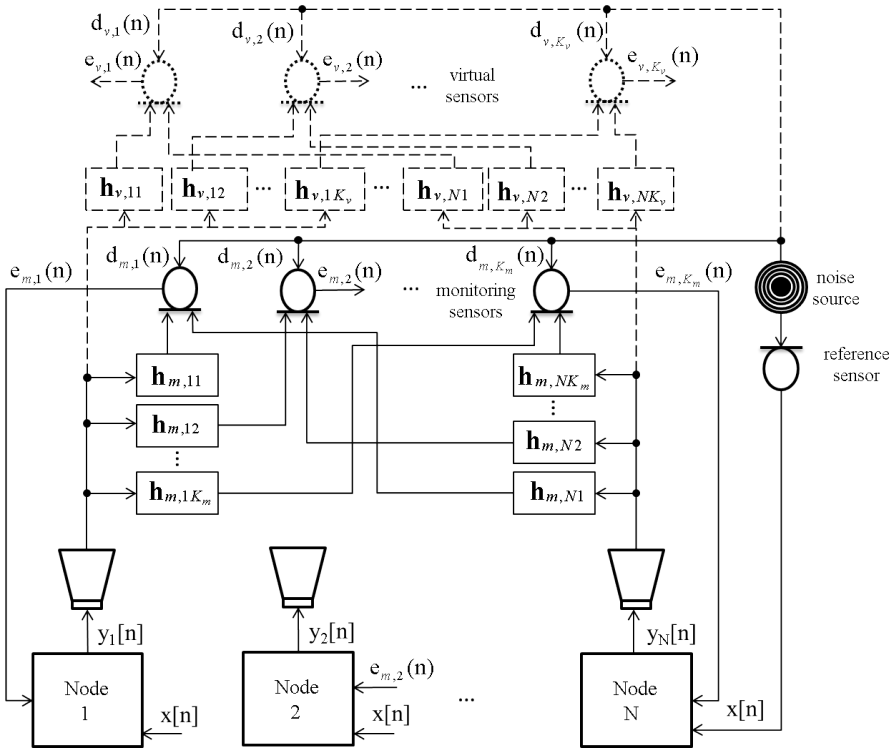


Figure 5.31: Multichannel feedforward ANC system combined with the remote microphone technique working over an ASN of N nodes.

In practice, estimates of the real acoustic channels $\mathbf{h}_{\mathrm{m,j}\underset{\sim}{\mathrm{k}}_{\mathrm{m}}}$ and $\mathbf{h}_{\mathrm{v,j}\underset{\sim}{\mathrm{k}}_{\mathrm{v}}}$ are previously measured before the ANC application and denoted as $\tilde{\mathbf{h}}_{\mathrm{m,j}\underset{\sim}{\mathrm{k}}_{\mathrm{m}}}$ and $\tilde{\mathbf{h}}_{\mathrm{v,j}\underset{\sim}{\mathrm{k}}_{\mathrm{v}}}$, respectively. Similarly, estimates of $\mathbf{d}_{\mathrm{v}}(n)$ and $\mathbf{d}_{\mathrm{m}}(n)$ are denoted as $\tilde{\mathbf{d}}_{\mathrm{v}}(n)$ and $\tilde{\mathbf{d}}_{\mathrm{m}}(n)$ respectively, with $\mathbf{d}_{\mathrm{v}}(n)$ as a K_{v}-lenght vector of disturbance signals at the virtual error sensors. As previously commented, the signals recorded at the monitoring sensors are used to estimate the virtual error signals. Thus, the estimated error signals at the virtual error sensors, $\tilde{\mathbf{e}}_{\mathrm{v}}(n)$, can be calculated as

$$\tilde{\mathbf{e}}_{\mathrm{v}}(n) = \tilde{\mathbf{d}}_{\mathrm{v}}(n) + \tilde{\mathbf{U}}_{\mathrm{v}}^{T}(n)\mathbf{w}(n) = \tilde{\mathbf{O}}\tilde{\mathbf{d}}'_{\mathrm{m}}(n) + \tilde{\mathbf{U}}_{\mathrm{v}}^{T}(n)\mathbf{w}(n), \tag{5.119}$$

where $\tilde{\mathbf{O}}$ is a $[K_{\mathrm{v}} \times PK_{\mathrm{m}}]$ matrix of FIR filters defined as

$$\begin{aligned} \tilde{\mathbf{O}} &= [\mathbf{O}_1(n)\ \mathbf{O}_2(n)\ \ldots\ \mathbf{O}_{K_{\mathrm{v}}}(n)\ ]^T, \\ \mathbf{O}_{k_{\mathrm{v}}}(n) &= [\mathbf{O}_{k_{\mathrm{v}},1}(n)\ \mathbf{O}_{k_{\mathrm{v}},2}(n)\ \ldots\ \mathbf{O}_{k_{\mathrm{v}},K_{\mathrm{m}}}(n)\ ]^T, \\ \mathbf{O}_{k_{\mathrm{v}},k_{\mathrm{m}}}(n) &= [O_{k_{\mathrm{v}},k_{\mathrm{m}},1}\ O_{k_{\mathrm{v}},k_{\mathrm{m}},2}\ \ldots\ O_{k_{\mathrm{v}},k_{\mathrm{m}},P}\ ]^T, \end{aligned} \tag{5.120}$$

being $\mathbf{O}_{k_{\mathrm{v}},k_{\mathrm{m}}}(n)$ the vector of P filter coefficients that model the transfer function between the primary disturbance at the k_{v}-th virtual sensor, $d_{\mathrm{v},\mathrm{k_v}}(n)$, from the primary

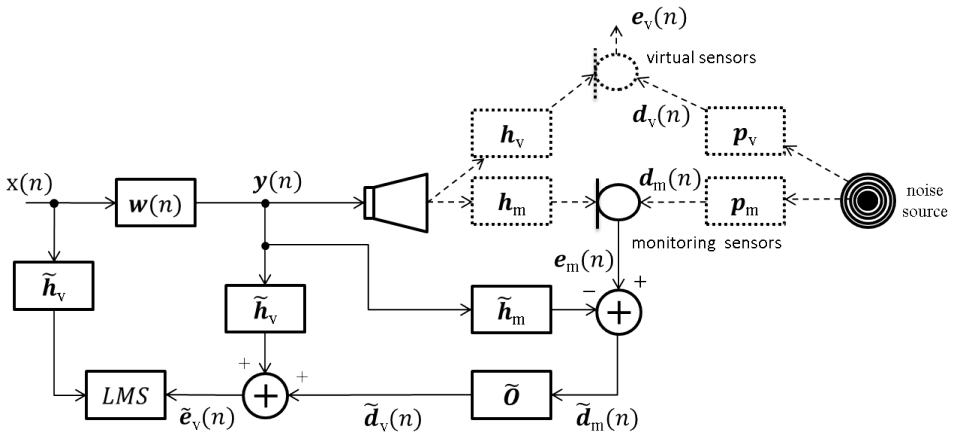


Figure 5.32: Block diagram of a centralized ANC system using the remote microphone technique.

disturbance at the k_{m}-th monitoring sensor, $d_{\mathrm{m},\mathrm{k_m}}(n)$. Therefore, $\tilde{\mathbf{O}}$ may be defined as the estimated matrix of filters used to calculate $\mathbf{d}_{\mathrm{v}}(n)$ from $\tilde{\mathbf{d}}_{\mathrm{m}}(n)$. The $[PK_{\mathrm{m}}\times 1]$ vector $\tilde{\mathbf{d}}'_{\mathrm{m}}(n)=[\tilde{\mathbf{d}}^T_{\mathrm{m}}(n)\ \tilde{\mathbf{d}}^T_{\mathrm{m}}(n-1)\ \ldots\ \tilde{\mathbf{d}}^T_{\mathrm{m}}(n-P+1)\ ]^T$ contains the last P samples of the estimated disturbance noise at the K_{m} monitoring sensors, $\tilde{\mathbf{d}}_{\mathrm{m}}(n)$, calculated as $\tilde{\mathbf{d}}_{\mathrm{m}}(n)=\mathbf{e}_{\mathrm{m}}(n)-\tilde{\mathbf{U}}^T_{\mathrm{m}}(n)\mathbf{w}(n)$, as it can be seen from Figure 5.32. Matrix $\tilde{\mathbf{U}}_{\mathrm{m}}(n)$ of size $[LN\times K_{\mathrm{m}}]$ and matrix $\tilde{\mathbf{U}}_{\mathrm{v}}(n)$ of size $[LN\times K_{\mathrm{v}}]$ are similarly defined as $\mathbf{U}_{\mathrm{m}}(n)$ in (5.118) but using $\tilde{\mathbf{h}}_{\mathrm{m,jk_m}}$ and $\tilde{\mathbf{h}}_{\mathrm{v,jk_v}}$ instead of $\mathbf{h}_{\mathrm{m,jk_m}}$, respectively.

Note that, the estimation error between $\tilde{\mathbf{d}}_{\mathrm{v}}(n)$ and $\mathbf{d}_{\mathrm{v}}(n)$ will affect to the behaviour of the ANC system. Therefore, the optimal value of $\tilde{\mathbf{O}}$, denoted as $\mathbf{O}_{opt}$, can be calculated by minimizing the following cost function

$$\mathbf{J}_1 = E\{(\mathbf{d}_{\mathrm{v}}(n)-\tilde{\mathbf{d}}_{\mathrm{v}}(n))(\mathbf{d}_{\mathrm{v}}(n)-\tilde{\mathbf{d}}_{\mathrm{v}}(n))^T\}. \tag{5.121}$$

As demonstrated in [139], the matrix of optimal observation filters, $\mathbf{O}_{opt}$, can be obtained as

$$\mathbf{O}_{opt} = ((\mathbf{R}_{mm}+\beta \boldsymbol{I}_{PK_{\mathrm{m}}})^{-1}\ \mathbf{R}_{mv})^T, \tag{5.122}$$

where $\mathbf{R}_{mm}=E\{\mathbf{d}'_{\mathrm{m}}(n)\mathbf{d}'^T_{\mathrm{m}}(n)\}$, $\mathbf{R}_{mv}=E\{\mathbf{d}'_{\mathrm{m}}(n)\mathbf{d}^T_{\mathrm{v}}(n)\}$ being $\mathbf{d}'_{\mathrm{m}}(n)$ a $[PK_{\mathrm{m}}\times 1]$ vector with a similar form to $\tilde{\mathbf{d}}'_{\mathrm{m}}(n)$ except that the actual disturbance signal, $d_{\mathrm{m,k_m}}(n)$, is used instead of its estimates $\tilde{d}_{\mathrm{m,k_m}}(n)$. β is a positive real regularization factor included to avoid inversion problems, $\boldsymbol{I}_{PK_{\mathrm{m}}}$ is a identity matriz of size $[PK_{\mathrm{m}}\times PK_{\mathrm{m}}]$ and $E\{.\}$ denotes the expectation operator. Note that $\mathbf{O}_{opt}$ can be calculated before control since $\mathbf{d}_{\mathrm{v}}(n)$ and $\mathbf{d}_{\mathrm{m}}(n)$ can be also measured in the previous identification stage previously commented by using a random excitation signal.

On the other hand, the multichannel ANC system aims to estimate $\mathbf{w}(n)$ that minimizes a cost function that depends on the acoustic field to be controlled, i.e., the virtual error signals. This cost function can be minimised using the MEFxLMS algorithm, as follows

$$\mathbf{J}_2 = \tilde{\mathbf{e}}_{\mathrm{v}}^T(n)\tilde{\mathbf{e}}_{\mathrm{v}}(n) = \sum_{k_{\mathrm{v}}=1}^{K_{\mathrm{v}}} \tilde{e}_{\mathrm{v},\mathrm{k_v}}^2(n) = \sum_{k_{\mathrm{v}}=1}^{K_{\mathrm{v}}} |\tilde{d}_{\mathrm{v},\mathrm{k_v}}(n) + \tilde{\mathbf{u}}_{\mathrm{v},\mathrm{k_v}}^T(n)\mathbf{w}(n)|^2, \quad (5.123)$$

where $\tilde{e}_{\mathrm{v},\mathrm{k_v}}(n)$ is the estimated error signal at the k_{v}-th virtual sensor, $\tilde{d}_{\mathrm{v},\mathrm{k_v}}(n)$ is the estimated primary disturbance at the k_{v}-th virtual sensor and $\tilde{\mathbf{u}}_{\mathrm{v},\mathrm{k_v}}(n) = [\tilde{\mathbf{h}}_{\mathrm{v},1\mathrm{k_v}}^T\mathbf{X}^T(n),\ \tilde{\mathbf{h}}_{\mathrm{v},2\mathrm{k_v}}^T\mathbf{X}^T(n),\ \ldots,\ \tilde{\mathbf{h}}_{\mathrm{v},\mathrm{Nk_v}}^T\mathbf{X}^T(n)]^T$. Applying the gradient to (5.123), the central unit uses a gradient-descent method to estimate $\mathbf{w}(n)$ in an iterative manner. Thus, the global filter updating equation of the centralized MEFxLMS algorithm considering the remote microphone method (RM-MEFxLMS) is stated as follows:

$$\mathbf{w}(n) = \mathbf{w}(n-1) - 2\mu \sum_{k_{\mathrm{v}}=1}^{K_{\mathrm{v}}} \tilde{\mathbf{u}}_{\mathrm{v},\mathrm{k_v}}(n)\tilde{e}_{\mathrm{v},\mathrm{k_v}}(n), \quad (5.124)$$

where μ is the step size parameter. However, due to the well-known problems of the centralized solution, the use of a distributed network is required. Hence, the proposed implementation of (5.124) over a network of distributed nodes will be presented in the following section.

5.7.1.2. Distributed RM-MEFxLMS algorithm

In the distributed case, as shown in Figure 5.33, we consider that node j is composed of one actuator (j-th loudspeaker), K_{m}^j monitoring sensors and K_{v}^j virtual sensors. For simplicity, we consider that each node has its unique monitoring and virtual sensors and consequently, we assume that $\sum_{j=1}^{N} K_{\mathrm{m}}^j = K_{\mathrm{m}}$ and $\sum_{j=1}^{N} K_{\mathrm{v}}^j = K_{\mathrm{v}}$. Now, the adaptive processing is carried out in a distributed way over a ring topology with incremental communication between the nodes where the global filter $\mathbf{w}(n)$ is splitted into local updates. Considering $\mathbf{w}^j(n)$ as a local version of $\mathbf{w}(n)$ at the jth node, from (5.124), we can derive the filter updating equation of the RM-DMEFxLMS algorithm using an incremental strategy on the network communication as

$$\mathbf{w}^j(n) = \mathbf{w}^{j-1}(n) - 2\mu\, \tilde{\mathbf{U}}_{\mathrm{v}}^j(n)\tilde{\mathbf{e}}_{\mathrm{v}}^j(n), \quad (5.125)$$

being

$$\begin{aligned} &\tilde{\mathbf{U}}_{\mathrm{v}}^j(n) = [\tilde{\mathbf{u}}_{\mathrm{v},1^{\mathrm{j}}}(n)\tilde{\mathbf{u}}_{\mathrm{v},2^{\mathrm{j}}}(n)\ \ldots\ \tilde{\mathbf{u}}_{\mathrm{v},\mathrm{K_v^j}}(n)\ ]^T \text{ of size } [LN\times K_{\mathrm{v}}^j],\\ &\tilde{\mathbf{u}}_{\mathrm{v},\mathrm{k_v^j}}(n) = [\tilde{\mathbf{h}}_{\mathrm{v},1\mathrm{k_v^j}}^T\mathbf{X}^T(n),\ \tilde{\mathbf{h}}_{\mathrm{v},2\mathrm{k_v^j}}^T\mathbf{X}^T(n),\ \ldots,\ \tilde{\mathbf{h}}_{\mathrm{v},\mathrm{Nk_v^j}}^T\mathbf{X}^T(n)]^T \text{ of size } [LN\times 1], \end{aligned} \quad (5.126)$$

where $\tilde{\mathbf{h}}_{\mathrm{v},\mathrm{kk_v^j}}^T$ is the estimated acoustic channel between the loudspeaker of the k-th node and the k_{v}-th virtual sensor of node j. The $[K_{\mathrm{v}}^j\times 1]$ vector $\tilde{\mathbf{e}}_{\mathrm{v}}^j(n)$ will be defined

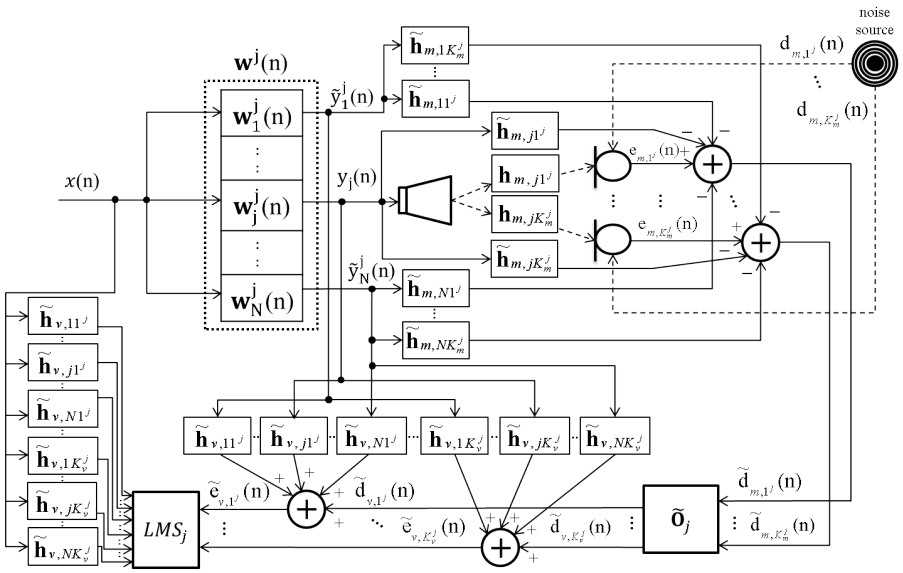


Figure 5.33: Block diagram of the j-th node of a distributed ANC system using the remote microphone technique.

later. It should be noted that only the local information is needed to generate the jth node output signal, $y_j(n)$,

$$y_j(n) = [\mathbf{w}_j(n)]^T [\mathbf{X}(n)]_{(:,1)}. \tag{5.127}$$

However, the control signals of the rest of the $N-1$ nodes are necessary in order to properly calculate the estimated disturbance signals captured by the K_{m}^j monitoring error sensors at j-th node. Since that information is not available to the nodes of a fully distributed network, each node must estimate the control signals of the network by using its local version of the global adaptive filter. To this end, we define the estimated control signal at node j from node k, $\tilde{y}_k^j(n)$, as

$$\tilde{y}_k^j(n) = [\mathbf{w}_k^j(n)]^T [\mathbf{X}(n)]_{(:,1)}, \tag{5.128}$$

with $\mathbf{w}_k^j(n)$=$\mathbf{w}^j(n-1)_{(1+L(k-1):Lk)}$, $\tilde{y}_j^j(n)$=$y_j(n)$ and $\mathbf{w}_j^j(n)$=$\mathbf{w}_j(n-1)$ for $k = 1, 2, \ldots, N$. Back to (5.125), $\tilde{\mathbf{e}}_{\mathrm{v}}^j(n)$=$[\tilde{e}_{\mathrm{v},1^{\mathrm{j}}}(n)\ \tilde{e}_{\mathrm{v},2^{\mathrm{j}}}(n)\ \ldots\ \tilde{e}_{\mathrm{v},\mathrm{K}_{\mathrm{v}}^{\mathrm{j}}}(n)\ ]^T$ is a vector of size $[K_{\mathrm{v}}^j \times 1]$ composed of the estimated error signals at the K_{v}^j-th virtual sensors, with $\tilde{e}_{\mathrm{v},\mathrm{k}_{\mathrm{v}}^{\mathrm{j}}}(n)$ as the estimated error signal at the k_{v}-th virtual sensor of node j.

In order to obtain a similar performance as the obtained by the RM-MEFxLMS algorithm, $\tilde{\mathbf{e}}_{\mathrm{v}}^j(n)$ must be calculated in a distributed manner, in a similar way to the

Algorithm 8: *RM-DMEFxLMS algorithm for N-nodes ASN.*

for all *node* $1 \leq\ j \leq N$ **do**
  $\mathbf{w}^j(n) = \mathbf{w}^{j-1}(n)$ % Copy local state of previous node
  for all $1 \leq\ k \leq N$ **do**
    $\mathbf{w}_k^j(n) = \mathbf{w}^j(n)_{(1+L(k-1):Lk)}$ % Obtain local coefficients related to the k-th node.
    $\tilde{y}_k^j(n) = [\mathbf{w}_k^j(n)]^T[\mathbf{X}(n)]_{(:,1)}$ % k-th estimated output signal
    $\tilde{\mathbf{y}}_{Mk}^j(n) = [\tilde{y}_k^j(n)\ \tilde{y}_k^j(n-1)\ \ldots\ \tilde{y}_k^j(n-M-1)\,]^T$ % Last M samples of $\tilde{y}_k^j(n)$
  end for
  $y_j(n) = \tilde{y}_j^j(n)$ % Output signal
  $\hat{\mathbf{d}}_{\mathrm{v}}^0(n) = \mathbf{0}_{[K_{\mathrm{v}}\times 1]}$ % Initialize the local version of $\tilde{\mathbf{d}}_{\mathrm{v}}(n)$
  for all *node* $1 \leq\ k \leq N$ **do**
    for all $1^k \leq\ k_{\mathrm{m}}^k \leq K_{\mathrm{m}}^k$ **do**
      $\tilde{d}_{\mathrm{m,k_m^k}}(n)=\tilde{e}_{\mathrm{m,k_m^k}}(n)-\sum_{q=1}^{N}[\tilde{\mathbf{y}}_{Mq}^j(n)]^T\tilde{\mathbf{h}}_{\mathrm{m,qk_m^k}}$ % Estimated primary disturbance at the k_{m}-th monitoring sensor
      $\tilde{\mathbf{d}}'_{\mathrm{m,k_m^k}}(n)=[\tilde{d}_{\mathrm{m,k_m^k}}(n)\ \tilde{d}_{\mathrm{m,k_m^k}}(n-1)\ \ldots\ \tilde{d}_{\mathrm{m,k_m^k}}(n-P-1)\,]^T$ % Last P samples of $\tilde{d}_{\mathrm{m,k_m^k}}(n)$
    end for
    $\tilde{\mathbf{d}}'^k_{\mathrm{m}}(n) = [\tilde{\mathbf{d}}'^T_{\mathrm{m,1^k}}(n)\ \tilde{\mathbf{d}}'^T_{\mathrm{m,2^k}}(n)\ \ldots\ \tilde{\mathbf{d}}'^T_{\mathrm{m,K_m^k}}(n)\,]^T$
    $\tilde{\mathbf{O}}^k = \tilde{\mathbf{O}}_{(:,1+\hat{K}_{\mathrm{m}}:K_{\mathrm{m}}^k+\hat{K}_{\mathrm{m}})}$ % where $\hat{K}_{\mathrm{m}} = \sum_{q=1}^{k} K_{\mathrm{m}}^{(q-1)}$ with $K_{\mathrm{m}}^0 = 0$
    $\hat{\mathbf{d}}_{\mathrm{v}}^k(n) = \hat{\mathbf{d}}_{\mathrm{v}}^{k-1}(n) + \tilde{\mathbf{O}}^k\tilde{\mathbf{d}}'^k_{\mathrm{m}}(n)$ % Update local version of $\tilde{\mathbf{d}}_{\mathrm{v}}(n)$
  end for
  $\tilde{\mathbf{d}}_{\mathrm{v}}(n) = \hat{\mathbf{d}}_{\mathrm{v}}^N(n)$ % Update estimated primary disturbances at the K_{v}-th virtual sensors, $\tilde{\mathbf{d}}_{\mathrm{v}}(n)$
  $\tilde{\mathbf{d}}_{\mathrm{v}}^j(n) = \tilde{\mathbf{d}}_{\mathrm{v}}(n)_{(1+\hat{K}:K_{\mathrm{v}}^j+\hat{K})}$ % where $\hat{K} = \sum_{k=1}^{j} K_{\mathrm{v}}^{(k-1)}$ with $K_{\mathrm{v}}^0 = 0$
  for all $1^j \leq\ k_{\mathrm{v}}^j \leq K_{\mathrm{v}}^j$ **do**
    $\tilde{e}_{\mathrm{v,k_v^j}}(n) = \tilde{d}_{\mathrm{v,k_v^j}}(n) + \sum_{k=1}^{N}[\tilde{\mathbf{y}}_{Mk}^j(n)]^T\tilde{\mathbf{h}}_{\mathrm{v,kk_v^j}}$ % Estimated error signal at the k_{v}-th virtual sensor
    $\tilde{\mathbf{u}}_{\mathrm{v,k_v^j}}(n) = [\tilde{\mathbf{h}}^T_{\mathrm{v,1k_v^j}}\mathbf{X}^T(n),\ \tilde{\mathbf{h}}^T_{\mathrm{v,2k_v^j}}\mathbf{X}^T(n),\ \ldots,\ \tilde{\mathbf{h}}^T_{\mathrm{v,Nk_v^j}}\mathbf{X}^T(n)]^T$
  end for
  $\tilde{\mathbf{e}}_{\mathrm{v}}^j(n) = [\tilde{e}_{\mathrm{v,1^j}}(n)\ \tilde{e}_{\mathrm{v,2^j}}(n)\ \ldots\ \tilde{e}_{\mathrm{v,K_v^j}}(n)\,]^T$
  $\tilde{\mathbf{U}}_{\mathrm{v}}^j(n) = [\tilde{\mathbf{u}}_{\mathrm{v,1^j}}(n)\tilde{\mathbf{u}}_{\mathrm{v,2^j}}(n)\ \ldots\ \tilde{\mathbf{u}}_{\mathrm{v,K_v^j}}(n)\,]^T$
  $\mathbf{w}^j(n) = \mathbf{w}^{j-1}(n)\ -2\mu\,\tilde{\mathbf{U}}_{\mathrm{v}}^j(n)\tilde{\mathbf{e}}_{\mathrm{v}}^j(n)$ % Update local state
end for
$\mathbf{w}(n) = \mathbf{w}^N(n)$ % Updated global state of the network
for all *node* $1 \leq\ k \leq (N-1)$ **do**
  $\mathbf{w}^k(n) = \mathbf{w}(n)$ % Disseminate global state of the network
end for

update procedure of $\mathbf{w}^j(n)$. To this end, we must define $\hat{\mathbf{d}}_{\mathrm{v}}^k(n)$ as a local version at each node of $\tilde{\mathbf{d}}_{\mathrm{v}}(n)$ of size $[K_{\mathrm{v}}\times 1]$. In the same way, we define $\tilde{\mathbf{O}}^k$ as the matrix of size $[K_{\mathrm{v}}\times PK_{\mathrm{m}}^k]$ defined as the part of $\tilde{\mathbf{O}}$ (defined in (5.120)) corresponding to the node k. In this way, if we go through the nodes in an incremental way, summing at each node the product of $\tilde{\mathbf{d}}'^k_{\mathrm{m}}(n)$ and $\tilde{\mathbf{O}}^k$ and passes the result to the following one, the last node of the network will obtain the updated version of $\tilde{\mathbf{d}}_{\mathrm{v}}(n)$, which is identical to the one obtained with the centralized scheme. Once the vector $\tilde{\mathbf{d}}_{\mathrm{v}}(n)$ is updated, i.e., $\tilde{\mathbf{d}}_{\mathrm{v}}(n)=\hat{\mathbf{d}}_{\mathrm{v}}^N(n)$, only the local vector that contains the information of the K_{v}^j virtual sensor locations is needed to calculate $\tilde{\mathbf{e}}_{\mathrm{v}}^j(n)$,

$$\tilde{\mathbf{d}}_{\mathrm{v}}^j(n)=\tilde{\mathbf{d}}_{\mathrm{v}}(n)_{(1+\hat{K}:K_{\mathrm{v}}^j+\hat{K})}, \tag{5.129}$$

Algorithm 9: *1r-RM-DMEFxLMS algorithm for N-nodes ASN.*

1: $\hat{\mathbf{d}}_{\mathrm{v}}^{0}(n) = \mathbf{0}_{[K_{\mathrm{v}}\times 1]}$ % Initialize the local version of $\tilde{\mathbf{d}}_{\mathrm{v}}(n)$
2: **for all** *node* $1 \leq j \leq N$ **do**
3: $\mathbf{w}^{j}(n) = \mathbf{w}^{j-1}(n)$ % Copy local state of previous node
4: **for all** $1 \leq k \leq N$ **do**
5: $\mathbf{w}_{k}^{j}(n) = \mathbf{w}^{j}(n)_{(1+L(k-1):Lk)}$ % Obtain local coefficients related to the k-th node.
6: $\tilde{y}_{k}^{j}(n) = [\mathbf{w}_{k}^{j}(n)]^{T}[\mathbf{X}(n)]_{(:,1)}$ % k-th estimated output signal
7: $\tilde{\mathbf{y}}_{Mk}^{j}(n) = [\tilde{y}_{k}^{j}(n)\ \tilde{y}_{k}^{j}(n-1)\ \ldots\ \tilde{y}_{k}^{j}(n-M-1)\,]^{T}$ % Last M samples of $\tilde{y}_{k}^{j}(n)$
8: **end for**
9: $y_{j}(n) = \tilde{y}_{j}^{j}(n)$ % Output signal
10: **for all** $1^{j} \leq k_{\mathrm{m}}^{j} \leq K_{\mathrm{m}}^{j}$ **do**
11: $\tilde{d}_{\mathrm{m},\mathrm{k}_{\mathrm{m}}^{\mathrm{j}}}(n) = \tilde{e}_{\mathrm{m},\mathrm{k}_{\mathrm{m}}^{\mathrm{j}}}(n) - \sum_{k=1}^{N} [\tilde{\mathbf{y}}_{Mk}^{j}(n)]^{T}\tilde{\mathbf{h}}_{\mathrm{m},\mathrm{kk}_{\mathrm{m}}^{\mathrm{j}}}$ % Estimated primary disturbance at the k_{m}-th monitoring sensor
12: $\tilde{\mathbf{d}}'_{\mathrm{m},\mathrm{k}_{\mathrm{m}}^{\mathrm{j}}}(n) = [\tilde{d}_{\mathrm{m},\mathrm{k}_{\mathrm{m}}^{\mathrm{j}}}(n)\ \tilde{d}_{\mathrm{m},\mathrm{k}_{\mathrm{m}}^{\mathrm{j}}}(n-1)\ \ldots\ \tilde{d}_{\mathrm{m},\mathrm{k}_{\mathrm{m}}^{\mathrm{j}}}(n-P-1)\,]^{T}$ % Last P samples of $\tilde{d}_{\mathrm{m},\mathrm{k}_{\mathrm{m}}^{\mathrm{j}}}(n)$
13: **end for**
14: $\tilde{\mathbf{d}}_{\mathrm{m}}^{\prime j}(n) = [\tilde{\mathbf{d}}'^{T}_{\mathrm{m},1^{\mathrm{j}}}(n)\ \tilde{\mathbf{d}}'^{T}_{\mathrm{m},2^{\mathrm{j}}}(n)\ \ldots\ \tilde{\mathbf{d}}'^{T}_{\mathrm{m},\mathrm{K}_{\mathrm{m}}^{\mathrm{j}}}(n)\,]^{T}$
15: $\tilde{\mathbf{O}}^{j} = \tilde{\mathbf{O}}_{(:,1+\hat{K}_{\mathrm{m}}:K_{\mathrm{m}}^{j}+\hat{K}_{\mathrm{m}})}$ % where $\hat{K}_{\mathrm{m}} = \sum_{k=1}^{j} K_{\mathrm{m}}^{(k-1)}$ with $K_{\mathrm{m}}^{0} = 0$
16: $\hat{\mathbf{d}}_{\mathrm{v}}^{j}(n) = \hat{\mathbf{d}}_{\mathrm{v}}^{j-1}(n) + \tilde{\mathbf{O}}^{j}\tilde{\mathbf{d}}_{\mathrm{m}}^{\prime j}(n)$ % Update local version of $\tilde{\mathbf{d}}_{\mathrm{v}}(n)$
17: $\tilde{\mathbf{d}}_{\mathrm{v}}^{j}(n) = \hat{\mathbf{d}}_{\mathrm{v}}^{j}(n)_{(1+\hat{K}_{\mathrm{v}}:K_{\mathrm{v}}^{j}+\hat{K}_{\mathrm{v}})}$ % where $\hat{K}_{\mathrm{v}} = \sum_{k=1}^{j} K_{\mathrm{v}}^{(k-1)}$ with $K_{\mathrm{v}}^{0} = 0$
18: **for all** $1^{j} \leq k_{\mathrm{v}}^{j} \leq K_{\mathrm{v}}^{j}$ **do**
19: $\tilde{e}_{\mathrm{v},\mathrm{k}_{\mathrm{v}}^{\mathrm{j}}}(n) = \tilde{d}_{\mathrm{v},\mathrm{k}_{\mathrm{v}}^{\mathrm{j}}}(n) + \sum_{k=1}^{N} [\tilde{\mathbf{y}}_{Mk}^{j}(n)]^{T}\tilde{\mathbf{h}}_{\mathrm{v},\mathrm{kk}_{\mathrm{v}}^{\mathrm{j}}}$ % Estimated error signal at the k_{v}-th virtual sensor
20: $\tilde{\mathbf{u}}_{\mathrm{v},\mathrm{k}_{\mathrm{v}}^{\mathrm{j}}}(n) = [\tilde{\mathbf{h}}^{T}_{\mathrm{v},1\mathrm{k}_{\mathrm{v}}^{\mathrm{j}}}\mathbf{X}^{T}(n),\ \tilde{\mathbf{h}}^{T}_{\mathrm{v},2\mathrm{k}_{\mathrm{v}}^{\mathrm{j}}}\mathbf{X}^{T}(n),\ \ldots,\ \tilde{\mathbf{h}}^{T}_{\mathrm{v},\mathrm{Nk}_{\mathrm{v}}^{\mathrm{j}}}\mathbf{X}^{T}(n)]^{T}$
21: **end for**
22: $\tilde{\mathbf{e}}_{\mathrm{v}}^{j}(n) = [\tilde{e}_{\mathrm{v},1^{\mathrm{j}}}(n)\ \tilde{e}_{\mathrm{v},2^{\mathrm{j}}}(n)\ \ldots\ \tilde{e}_{\mathrm{v},\mathrm{K}_{\mathrm{v}}^{\mathrm{j}}}(n)\,]^{T}$
23: $\tilde{\mathbf{U}}_{\mathrm{v}}^{j}(n) = [\tilde{\mathbf{u}}_{\mathrm{v},1^{\mathrm{j}}}(n)\tilde{\mathbf{u}}_{\mathrm{v},2^{\mathrm{j}}}(n)\ \ldots\ \tilde{\mathbf{u}}_{\mathrm{v},\mathrm{K}_{\mathrm{v}}^{\mathrm{j}}}(n)\,]^{T}$
24: $\mathbf{w}^{j}(n) = \mathbf{w}^{j-1}(n) - 2\mu\,\tilde{\mathbf{U}}_{\mathrm{v}}^{j}(n)\tilde{\mathbf{e}}_{\mathrm{v}}^{j}(n)$ % Update local state
25: **end for**
26: $\mathbf{w}(n) = \mathbf{w}^{N}(n)$ % Updated global state of the network
27: **for all** *node* $1 \leq k \leq (N-1)$ **do**
28: $\mathbf{w}^{k}(n) = \mathbf{w}(n)$ % Disseminate global state of the network
29: **end for**

where $\tilde{\mathbf{d}}_{\mathrm{v}}^{j}(n)$ is the estimated disturbance signal at the K_{v}^{j} virtual sensor locations for the j-th node, $\tilde{\mathbf{d}}_{\mathrm{v}}^{j}(n) = [\tilde{\mathbf{d}}^{T}_{\mathrm{v},1^{\mathrm{j}}}(n)\ \tilde{\mathbf{d}}^{T}_{\mathrm{v},2^{\mathrm{j}}}(n)\ \ldots\ \tilde{\mathbf{d}}^{T}_{\mathrm{v},\mathrm{K}_{\mathrm{v}}^{\mathrm{j}}}(n)\,]^{T}$ with $\tilde{\mathbf{d}}_{\mathrm{v},\mathrm{k}_{\mathrm{v}}^{\mathrm{j}}}(n)$ as the estimated primary disturbance at the k_{v}-th virtual sensor of node j and $\hat{K} = \sum_{k=1}^{j} K_{\mathrm{v}}^{(k-1)}$ with $K_{\mathrm{v}}^{0}=0$. A summary of the RM-DMEFxLMS algorithm pseudocodes executed per sample time at each node is given in ***Algorithm 8***. Note that an data transfer round (for loop of lines 11-18) is required in order to obtain the proper updated version of $\hat{\mathbf{d}}_{\mathrm{v}}^{k}(n)$ at the last node of the network. To avoid this extra round, two versions of the RM-DMEFxLMS algorithm that require less communication and computing demands are proposed below.

5.7.1.3. Distributed 1r-RM-MEFxLMS algorithm

The one-round strategy (1r-RM-DMEFxLMS algorithm) aims to reduce both the communications demands of the network and the computational cost of the algorithm

Algorithm 10: *pnc-RM-DMEFxLMS algorithm for N-nodes ASN.*

1: **for all** $node\ 1 \leq\ j \leq N$ **do**
2: $\mathbf{w}^j(n) = \mathbf{w}^{j-1}(n)$ % Copy local state of previous node
3: **for all** $1 \leq\ k \leq N$ **do**
4: $\mathbf{w}_k^j(n) = \mathbf{w}^j(n)_{(1+L(k-1):Lk)}$ % Obtain local coefficients related to the k-th node.
5: $\tilde{y}_k^j(n) = [\mathbf{w}_k^j(n)]^T[\mathbf{X}(n)]_{(:,1)}$ % k-th estimated output signal
6: $\tilde{\mathbf{y}}_{Mk}^j(n) = [\tilde{y}_k^j(n)\ \tilde{y}_k^j(n-1)\ \ldots\ \tilde{y}_k^j(n-M-1)\,]^T$ % Last M samples of $\tilde{y}_k^j(n)$
7: **end for**
8: $y_j(n) = \tilde{y}_j^j(n)$ % Output signal
9: **for all** $1^j \leq\ k_{\mathrm{m}}^j \leq K_{\mathrm{m}}^j$ **do**
10: $\tilde{d}_{\mathrm{m,k_m^j}}(n) = \tilde{e}_{\mathrm{m,k_m^j}}(n) - \sum_{k=1}^{N} [\tilde{\mathbf{y}}_{Mk}^j(n)]^T \tilde{\mathbf{h}}_{\mathrm{m,kk_m^j}}$ % Estimated primary disturbance at the k_{m}-th monitoring sensor
11: $\tilde{\mathbf{d}}'_{\mathrm{m,k_m^j}}(n) = [\tilde{d}_{\mathrm{m,k_m^j}}(n)\ \tilde{d}_{\mathrm{m,k_m^j}}(n-1)\ \ldots\ \tilde{d}_{\mathrm{m,k_m^j}}(n-P-1)\,]^T$ % Last P samples of $\tilde{d}_{\mathrm{m,k_m^j}}(n)$
12: **end for**
13: $\tilde{\mathbf{d}}_{\mathrm{m}}^{\prime j}(n) = [\tilde{\mathbf{d}}'^T_{\mathrm{m,1^j}}(n)\ \tilde{\mathbf{d}}'^T_{\mathrm{m,2^j}}(n)\ \ldots\ \tilde{\mathbf{d}}'^T_{\mathrm{m,K_m^j}}(n)\,]^T$
14: $\tilde{\mathbf{d}}_{\mathrm{v}}^j(n) = \tilde{\mathbf{O}}_j \tilde{\mathbf{d}}_{\mathrm{m}}^{\prime j}(n)$ % Estimated primary disturbances at the K_{v}-th virtual sensors
15: **for all** $1^j \leq\ k_{\mathrm{v}}^j \leq K_{\mathrm{v}}^j$ **do**
16: $\tilde{e}_{\mathrm{v,k_v^j}}(n) = \tilde{d}_{\mathrm{v,k_v^j}}(n) + \sum_{k=1}^{N} [\tilde{\mathbf{y}}_{Mk}^j(n)]^T \tilde{\mathbf{h}}_{\mathrm{v,kk_v^j}}$ % Estimated error signal at the k_{v}-th virtual sensor
17: $\tilde{\mathbf{u}}_{\mathrm{v,k_v^j}}(n) = [\tilde{\mathbf{h}}^T_{\mathrm{v,1k_v^j}}\mathbf{X}^T(n),\ \tilde{\mathbf{h}}^T_{\mathrm{v,2k_v^j}}\mathbf{X}^T(n),\ \ldots,\ \tilde{\mathbf{h}}^T_{\mathrm{v,Nk_v^j}}\mathbf{X}^T(n)]^T$
18: **end for**
19: $\tilde{\mathbf{e}}_{\mathrm{v}}^j(n) = [\tilde{e}_{\mathrm{v,1^j}}(n)\ \tilde{e}_{\mathrm{v,2^j}}(n)\ \ldots\ \tilde{e}_{\mathrm{v,K_v^j}}(n)\,]^T$
20: $\tilde{\mathbf{U}}_{\mathrm{v}}^j(n) = [\tilde{\mathbf{u}}_{\mathrm{v,1^j}}(n)\tilde{\mathbf{u}}_{\mathrm{v,2^j}}(n)\ \ldots\ \tilde{\mathbf{u}}_{\mathrm{v,K_v^j}}(n)\,]^T$
21: $\mathbf{w}^j(n) = \mathbf{w}^{j-1}(n) - 2\mu\ \tilde{\mathbf{U}}_{\mathrm{v}}^j(n)\tilde{\mathbf{e}}_{\mathrm{v}}^j(n)$ % Update local state
22: **end for**
23: $\mathbf{w}(n) = \mathbf{w}^N(n)$ % Updated global state of the network
24: **for all** $node\ 1 \leq\ k \leq (N-1)$ **do**
25: $\mathbf{w}^k(n) = \mathbf{w}(n)$ % Disseminate global state of the network
26: **end for**

by eliminating the **for** loop of lines 11-18 in ***Algorithm 8*** and, consequently, the update of $\tilde{\mathbf{d}}_{\mathrm{v}}(n)$ (line 19). In this case, in one round each node uses the information received from its precedent node $\hat{\mathbf{d}}_{\mathrm{v}}^{j-1}(n)$ and its calculation of $\tilde{\mathbf{d}}_{\mathrm{m}}^{\prime j}(n)\tilde{\mathbf{O}}^j$ to update its local version of $\tilde{\mathbf{d}}_{\mathrm{v}}(n)$, $\hat{\mathbf{d}}_{\mathrm{v}}^j(n)$. This means that the noise signal that would reach to the virtual sensor of a node is approximated by the noise signal of its previous node. This result will be supplied to the following node since an incremental strategy is considered. Therefore, instead of using the updated information of the last node, $\hat{\mathbf{d}}_{\mathrm{v}}^N(n)$, each node uses its local version of $\tilde{\mathbf{d}}_{\mathrm{v}}(n)$ and $\hat{\mathbf{d}}_{\mathrm{v}}^j(n)$, to calculate $\tilde{\mathbf{d}}_{\mathrm{v}}^j(n)$ and therefore, the estimated error signal at its virtual sensors, $\tilde{\mathbf{e}}_{\mathrm{v}}^j(n)$. Thus, the round to calculate the updated information of the last node, $\hat{\mathbf{d}}_{\mathrm{v}}^N(n)$, is avoided and it is possible to reduce the data transfer among the nodes. A summary of the 1r-RM-DMEFxLMS algorithm pseudocodes executed per sample time at each node is given in ***Algorithm 9***.

5.7.1.4. Distributed pnc-RM-MEFxLMS algorithm

However, the distributed calculation of $\tilde{\mathbf{e}}_{\mathrm{v}}^j(n)$ can be also avoided if each node

uses its own information to update $\tilde{\mathbf{d}}_{\mathrm{v}}^{j}(n)$,

$$\tilde{\mathbf{d}}_{\mathrm{v}}^{j}(n) = \tilde{\mathbf{O}}_{j}\tilde{\mathbf{d}}_{\mathrm{m}}^{\prime j}(n), \tag{5.130}$$

where $\tilde{\mathbf{O}}_{j}$ is the estimated matrix of filters of size $[K_{\mathrm{v}}^{j}\times PK_{\mathrm{m}}^{j}]$ defined similarly as $\tilde{\mathbf{O}}$ but changing the dimensions K_{m} and K_{v} by K_{m}^{j} and K_{v}^{j} respectively. In other words, instead of using the transfer functions between the virtual sensors of the node and the monitoring sensors of the rest of the nodes, an estimation of $\tilde{\mathbf{O}}$ composed of the transfer functions between the virtual and monitoring sensors of the node is considered. Therefore, this strategy works in a non-collaborative manner to calculate $\tilde{\mathbf{e}}_{\mathrm{v}}^{j}(n)$ and with a collaborative strategy to update $\mathbf{w}^{j}(n)$. Thus, we denote this partially non-collaborative method as pnc-RM-DMEFxLMS algorithm. A summary of the pnc-RM-DMEFxLMS algorithm pseudocodes executed per sample time at each node is given in ***Algorithm 10***. Following a practical example, consider the case where the SCNs of two PAC systems are located at such a distance that makes the acoustic system is coupled (as it can occur in the cabin of a public transport). Each SCN is composed of two loudspeakers to generate acoustic signals, four monitoring sensors to estimate the acoustical information that would be captured by two sensors virtually located at the listener ears and one processor with communication capability. Therefore, $K_{\mathrm{v}}^{1}{=}K_{\mathrm{v}}^{2}{=}N{=}2$. In those cases, the filter updating equation of the pnc-RM-DMEFxLMS algorithm can be stated for the two SCNs as follows

$$\begin{bmatrix}\mathbf{w}_{1}^{1}(n)\\ \mathbf{w}_{2}^{1}(n)\\ \mathbf{w}_{3}^{1}(n)\\ \mathbf{w}_{4}^{1}(n)\end{bmatrix} = \begin{bmatrix}\mathbf{w}_{1}^{2}(n-1)\\ \mathbf{w}_{2}^{2}(n-1)\\ \mathbf{w}_{3}^{2}(n-1)\\ \mathbf{w}_{4}^{2}(n-1)\end{bmatrix} - \mu \begin{bmatrix}\tilde{\mathbf{u}}_{\mathrm{v},11}(n) & \tilde{\mathbf{u}}_{\mathrm{v},12}(n)\\ \tilde{\mathbf{u}}_{\mathrm{v},21}(n) & \tilde{\mathbf{u}}_{\mathrm{v},22}(n)\\ \tilde{\mathbf{u}}_{\mathrm{v},31}(n) & \tilde{\mathbf{u}}_{\mathrm{v},32}(n)\\ \tilde{\mathbf{u}}_{\mathrm{v},41}(n) & \tilde{\mathbf{u}}_{\mathrm{v},42}(n)\end{bmatrix} \begin{bmatrix}\tilde{e}_{\mathrm{v},1}(n)\\ \tilde{e}_{\mathrm{v},2}(n)\end{bmatrix}, \tag{5.131}$$

$$\begin{bmatrix}\mathbf{w}_{1}^{2}(n)\\ \mathbf{w}_{2}^{2}(n)\\ \mathbf{w}_{3}^{2}(n)\\ \mathbf{w}_{4}^{2}(n)\end{bmatrix} = \begin{bmatrix}\mathbf{w}_{1}^{1}(n)\\ \mathbf{w}_{2}^{1}(n)\\ \mathbf{w}_{3}^{1}(n)\\ \mathbf{w}_{4}^{1}(n)\end{bmatrix} - \mu \begin{bmatrix}\tilde{\mathbf{u}}_{\mathrm{v},13}(n) & \tilde{\mathbf{u}}_{\mathrm{v},14}(n)\\ \tilde{\mathbf{u}}_{\mathrm{v},23}(n) & \tilde{\mathbf{u}}_{\mathrm{v},24}(n)\\ \tilde{\mathbf{u}}_{\mathrm{v},33}(n) & \tilde{\mathbf{u}}_{\mathrm{v},34}(n)\\ \tilde{\mathbf{u}}_{\mathrm{v},43}(n) & \tilde{\mathbf{u}}_{\mathrm{v},44}(n)\end{bmatrix} \begin{bmatrix}\tilde{e}_{\mathrm{v},3}(n)\\ \tilde{e}_{\mathrm{v},4}(n)\end{bmatrix}. \tag{5.132}$$

Note that, from the previous example, $\tilde{\mathbf{u}}_{\mathrm{v}}^{j}(n)$ and $\tilde{\mathbf{e}}_{\mathrm{v}}^{j}(n)$ may be redefined as

$$\begin{aligned}\tilde{\mathbf{u}}_{\mathrm{v}}^{j}(n){=}&\tilde{\mathbf{u}}_{\mathrm{v}}(n)_{(:,1+\hat{K}:K_{\mathrm{v}}^{j}+\hat{K})}\\ \tilde{\mathbf{e}}_{\mathrm{v}}^{j}(n){=}&\tilde{\mathbf{e}}_{\mathrm{v}}(n)_{(:,1+\hat{K}:K_{\mathrm{v}}^{j}+\hat{K})}.\end{aligned} \tag{5.133}$$

It should be noted that, since the updated version of the estimated disturbance signals at the virtual error sensors, $\tilde{\mathbf{d}}_{\mathrm{v}}(n)$, are fully shared among the nodes, the RM-DMEFxLMS algorithm will achieve a similar performance as centralized version, the RM-MEFxLMS algorithm. However, the behaviour of its two versions, the 1r-RM-DMEFxLMS and pnc-RM-DMEFxLMS algorithms, are not exactly the same as the

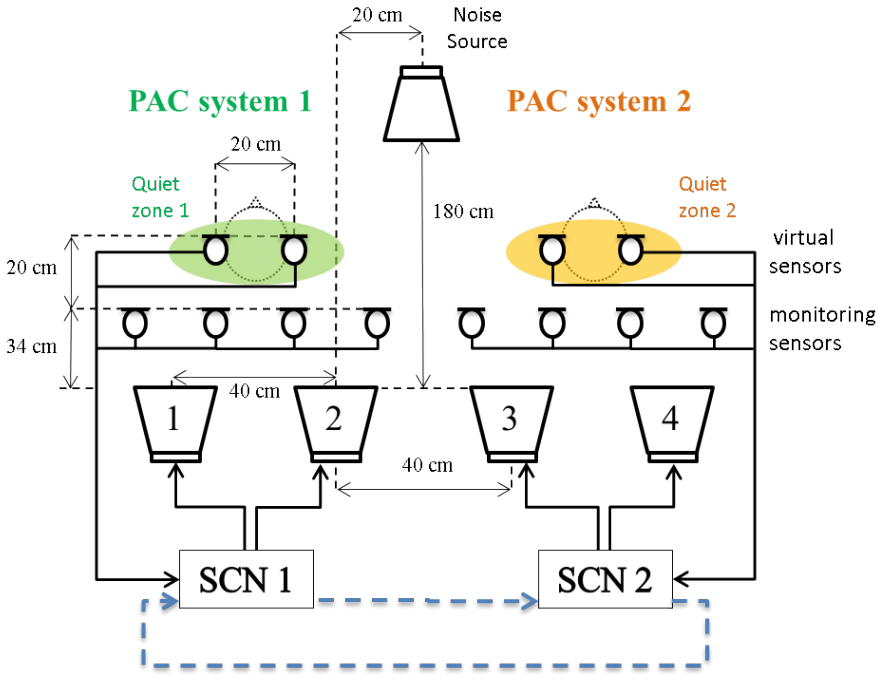


Figure 5.34: Distributed ANC headrest system composed of two PAC systems. Dashed blue lines represent the incremental communication strategy. Monitoring microphones are located in front of the loudspeakers at each SCN.

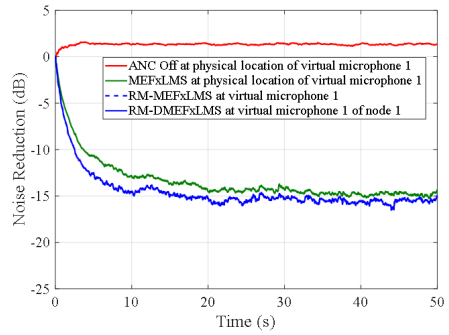
RM-DMEFxLMS approach. Since the data exchange is carried out in a consecutive order, nodes closer to the last node will have a more accurate estimation of $\tilde{\mathbf{d}}_{\mathrm{v}}(n)$. On the contrary, at the first nodes of the network, the estimation of $\tilde{\mathbf{d}}_{\mathrm{v}}(n)$ will be less accurate. This may lead to increase the estimation error between $\tilde{\mathbf{d}}_{\mathrm{v}}(n)$ and $\mathbf{d}_{\mathrm{v}}(n)$, affecting to the system behaviour. Therefore, differences between methods may result more relevant in larger ASN. However, in the case of ASN composed of few nodes, all distributed RM methods will obtain a similar performance in terms of convergence speed and noise reduction. A performance comparison between all the proposed algorithms is presented in the following section.

5.7.2 Simulation results

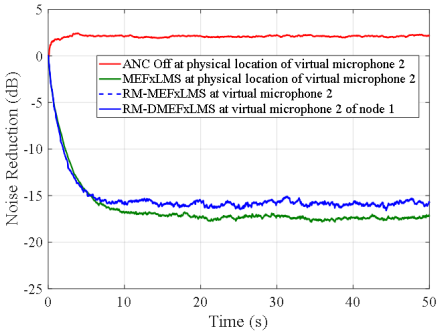
In this section, we present the simulations carried out to evaluate the performance of the proposed algorithms over a distributed ANC headrest system with no communication constraints. Firstly, we have compared the ANC system when control in virtual microphones is off (but is on in the monitoring microphones) with the MEFxLMS,

RM-MEFxLMS and RM-DMEFxLMS algorithms in terms of stability, noise reduction and convergence rate. Secondly, we have justified the use of the two versions of the fully distributed RM implementation, the 1r-RM-DMEFxLMS and the pnc-RM-DMEFxLMS algorithms with the aim to reduce the communication demands of the network and the computational load of the RM-DMEFxLMS algorithm. To this end, we have compared the RM-DMEFxLMS, the 1r-RM-DMEFxLMS and the pnc-RM-DMEFxLMS algorithms in terms of final noise reduction, convergence behavior, computational complexity and communication requirements. All algorithms have been tested in two simulated PAC systems composed of two-channel SCNs each of them, as shown in Figure 5.34. For the designed scheme, we use the real acoustic channels depicted in Section 3.3. In the same identification stage prior to control, the matrix $\tilde{\mathbf{O}}$ were modelled as 64 coefficient FIR filters (P=64). This value has been chosen by trial and error and it has been estimated as appropriate. The configuration of the simulated scenario is depicted as follows: two SCNs composed by two loudspeakers, four monitoring microphones and two virtual microphones were considered. An equal separation of 40 cm between adjacent loudspeakers were selected. At each SCN, the monitoring microphones were separated 34 cm away from the loudspeakers and the distance between the virtual and monitoring microphones was 20 cm. The separation between each microphone was also 20 cm. The disturbance signal is emitted by a loudspeaker located in front of the PAC sytems with a separation of 180 cm away from SNC loudspeakers. All microphones and loudspeakers involved were located at a height of 147 cm. The tested distribution tries to emulate a real ANC headrest application where we would seek to create local quiet zones in a cabin of a public transport. We have considered a wideband zero-mean Gaussian white noise with unit variance as disturbance signal and an adaptive filter length of L=150 coefficients. A step size parameter of μ=0.01 as the highest value that ensures the stability of the algorithms, has been used in the simulations.

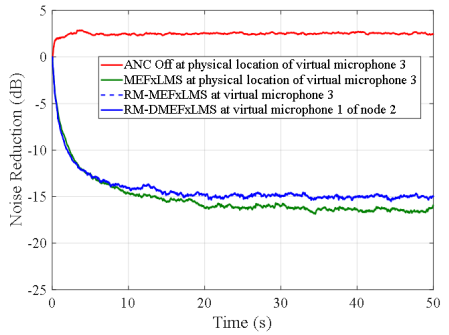
Figure 5.35 illustrates the time evolution of the $NR_k(n)$ of the distributed ANC headrest system in dB for centralized and distributed algorithms with and without the use of the remote microphone technique. As expected, when the ANC system is off, there is no attenuation (even some amplification, up to $3,5$ dB at the PAC 2) at the physical location of the virtual microphones. Comparing the performance of the MEFxLMS algorithm with and without the remote microphone technique, the results can be summarized as follows: in the best case, at the virtual microphone 4 (see Figure 5.35.(d)), both MEFxLMS and RM-MEFxLMS algoritms present the same attenuation, 13.5 dB and in the worst case, at the virtual microphone 2 (see Figure 5.35.(b)), the attenuation difference between both strategies is around 1 dB. At the virtual microphones 1 and 3 (see Figure 5.35.(a) and 5.35.(c), respectively), differences between the noise reduction achieved by the MEFxLMS and RM-MEFxLMS algoritms are $0,3$ dB and 0.8 dB respectively. Finally, the proposed RM-DMEFxLMS algorithm presents the same results as its centralized version



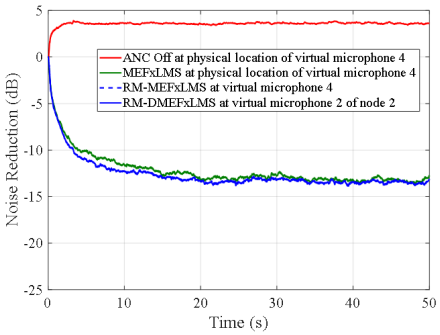
(a) Virtual microphone 1 (left) at PAC 1



(b) Virtual microphone 2 (right) at PAC 1



(c) Virtual microphone 1 (left) at PAC 2

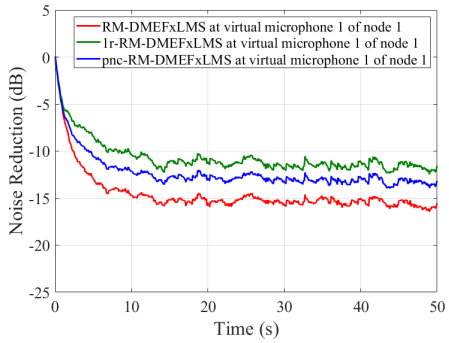


(d) Virtual microphone 2 (right) at PAC 2

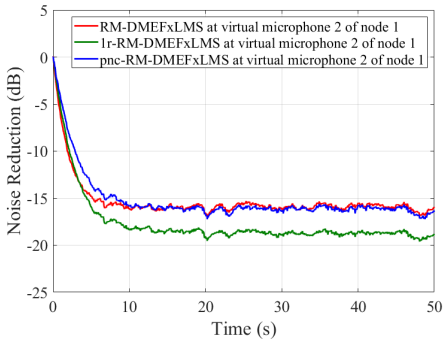
Figure 5.35: Noise reduction obtained by the distributed ANC headrest system for the different algorithms.

achieving a stable and robust behaviour and providing 15.3 dB, 15, 9 dB, 13 dB and 13.5 dB of noise reduction at the four virtual microphones, respectively. Regarding the covergence speed, there are no remarkable differences except the first case (see Figure 5.35.(a)) where the MEFxLMS algorithm converges lightly slowly than, for example, the RM-DMEFxLMS algorithm. Note that this may be caused by the particular setting of the SCNs since the performance of the RM algorithms may vary if another location of the monitoring microphones is selected, as mentioned in [135]. Therefore, considering that the RM technique involves the non-use of physical microphones located close to the listener ears and based on the results obtained, its use in distributed ANC headrest systems is justified.

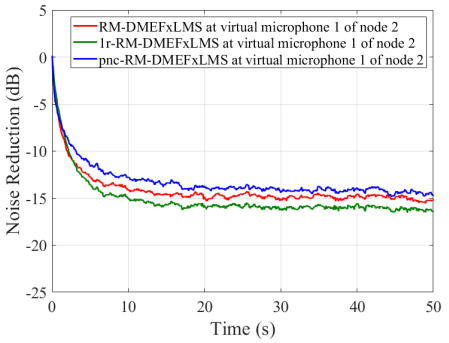
Secondly, to compare the behaviour of the several versions of the RM distributed strategy, Figure 5.36 shows the time evolution of the noise reduction for the RM-DMEFxLMS algorithm, the 1r-RM-DMEFxLMS algorithm and the pnc-RM-DMEFxLMS algorithm for the two PACS system described in Figure 5.34. It can be seen from Figure 5.36.(a),(b),(c) and (d) that all the three RM distributed methods ensure the sys-



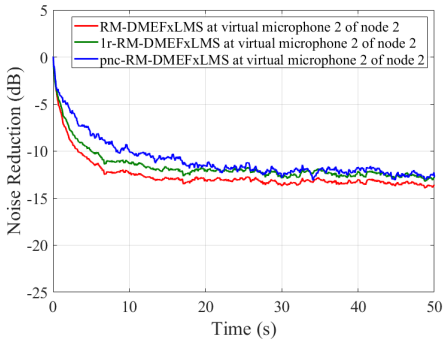
(a) Virtual microphone 1 (left) at PAC 1



(b) Virtual microphone 2 (right) at PAC 1



(c) Virtual microphone 1 (left) at PAC 2



(d) Virtual microphone 2 (right) at PAC 2

Figure 5.36: Noise reduction obtained by the distributed ANC headrest system for the proposed distributed RM algorithms.

tem stability for all the virtual microphones location of the two PAC systems. More specifically, for the virtual microphone 1 at PAC 1 (see Figure 5.36.(a)), the RM-DMEFxLMS algorithm outperforms the attenuation of the other two strategies providing an attenuation around 15.7 dB, while the 1r-RM-DMEFxLMS algorithm and the pnc-RM-DMEFxLMS algorithm achieve 11.7 dB and 13.3 dB, respectively. The noise reduction obtained at the virtual microphone 2 of PAC 1 for the three distributed RM methods is depicted in Figure 5.36.(b). In this case, the 1r-RM-DMEFxLMS algorithm exhibits the best overall performance, providing the higher noise reduction, 18.8 dB. The RM-DMEFxLMS algorithm and the pnc-RM-DMEFxLMS algorithm provide both an attenuation of 2.5 dB below the one round strategy (16.3 dB). As it can be seen in Figure 5.36.(c), the two approaches, 1r-RM-DMEFxLMS algorithm and the pnc-RM-DMEFxLMS algorithm, present similar results as the RM-DMEFxLMS method at virtual microphone 1 of PAC 2 in terms of final residual noise, since the difference between the best and worst attenuation achieved by the three methods is just 1.6 dB. More specifically, the RM-DMEFxLMS algorithm, the

Table 5.6: Total number of multiplications (MUX) and data transfer per iteration regarding (1) the computational complexity and (2) the communication requirements of the algorithms, respectively. L: length of the adaptive filters; M: length of the acoustic paths; N: number of nodes. As example, some typical cases considering L=150, M=256, K_{m}^k=4, K_{v}^k=2 and N=2, 4 and 8 nodes, have been evaluated. Note that we consider K_{m}=$K_{\mathrm{m}}^k N$ and K_{v}=$K_{\mathrm{v}}^k N$for each case.

	Algorithms	**Generic**	***N* = 1**	***N* = 4**	***N* = 8**
(1)	RM-MEFxLMS	$1{+}N(L{+}K_m(M{+}P/N){+}K_v(2M{+}L{+}1/N))$	2629	38689	152513
	RM-DMEFxLMS	$1{+}N(L{+}K_{\mathrm{m}}^k(MN{+}P){+}K_{\mathrm{v}}^k(2M{+}2L){+}Kv)$	2629	22825	78481
	1r-RM-DMEFxLMS	$1{+}N(L{+}K_{\mathrm{m}}^k(M{+}P/N){+}K_{\mathrm{v}}^k(3M{+}2L){+}Kv/N)$	3141	12177	24225
	pnc-RM-DMEFxLMS	$1{+}N(L{+}K_{\mathrm{m}}^k(M{+}P/N){+}K_{\mathrm{v}}^k(3M{+}2L+1/N))$	3141	12171	24211
	Algorithms	**Generic**	***N* = 2**	***N* = 4**	***N* = 8**
(2)	RM-DMEFxLMS	$2LN(N{-}1)+K_{\mathrm{v}}N^2$	616	3728	17824
	1r-RM-DMEFxLMS	$2LN(N{-}1)+K_{\mathrm{v}}N$	608	3632	16928
	pnc-RM-DMEFxLMS	$2LN(N{-}1)$	600	3600	16800

1r-RM-DMEFxLMS algorithm and the pnc-RM-DMEFxLMS algorithm obtain an attenuation up to 15.3 dB, 16.2 dB and 14.6 dB respectively. Finally, at the virtual microphone 2 at PAC 2, the fully distributed strategy outperforms the attenuation of its two versions obtaining a noise reduction of 13.7 dB, as it is shown in Figure 5.36.(d). It can be seen that both the one round and the partially non-collaborative strategies achieve a similar attenuation of 12.5 dB. Note that it would be expected that both 1r-RM-DMEFxLMS algorithm and the pnc-RM-DMEFxLMS algorithm would show worse performance at the two SCNs in terms of convergence speed or noise reduction than the RM-DMEFxLMS algorithm since they are approximations of the exact distributed version. However, as previously commented, this may be caused by the particular configuration of the SCNs within the enclosure.

Table 5.6 compares the computational complexity (in terms of multiplications per iteration) of the centralized and distributed RM algorithms. Similarly, the communication requirements (data transfer) of the distributed RM algorithms are also analyzed. To this end, we consider a network of N single-channel nodes. For simplicity, we assume that each node has access to $x(n)$ through an alternative broadcast channel. Therefore, the reference signal has not been considered in the calculation of the data transfer. Note that, for all the algorithms, the computational complexity depends on L, N, M, P and K_{v}. In addition, the distributed methods also depend on K_{m}^k and K_{v}^k. On the other hand, the communication requirements only depends on L, N and K_{v}. As an exception, the pnc-RM-DMEFxLMS algorithm does not depend on the number of virtual microphones, K_{v} neither in the calculation of the total number of multiplications nor in the calculation of data transfer. Both implementation aspects are particularized for N=2, N=4 and N=8 nodes. As expected, when N=1, The

RM-DMEFxLMS algorithm has exactly the same computational complexity as the RM-MEFxLMS strategy. However, as N increases, Table 5.6 shows that the computational cost of the centralized algorithm is higher than the distributed approach whose complexity at each node is not so significant. Moreover, it can be seen that the computational cost of the RM-DMEFxLMS algorithm is higher than its two approaches which have a similar number of multiplications per iteration. Regarding the communication demands, there are no significant differences between the three distributed RM methods. The RM-DMEFxLMS has slightly higher requirements in comparison with the rest, followed by the 1r-RM-DMEFxLMS method and the pnc-RM-DMEFxLMS method respectively. Differences between methods are slightly greater as the number of nodes increases also because of the dependence of K_{v}, on the first two strategies. For an incremental-learning N-node ASN, every node must transfer $LN\times 1$ coefficients to the following node $2(N-1)$ times in each iteration. This is fulfilled in all three distributed RM algorithms, as shown in Table 2. In addition, in the fully distributed RM strategy, a data stream of $K_{\mathrm{v}}N^2$ samples is also propagated through the nodes. By using the one-round distributed RM strategy, this extra data transfer is reduced by N while, for the partially non-collaborative distributed RM method, this data exchange is avoided. This is because the pnc-RM-DMEFxLMS algorithm does not depend on K_{v}. Therefore, it is possible to reduce the data transfer of the network by using this approach in the case of enlarging the quiet zone were required. Note that, for all RM strategies, both implementation aspects increase significantly with the number of nodes.

5.8.2.1. Simulated acoustic channels

With the aim to try to avoid the influence of the room acoustics on the performance of the algorithms, in this subsection, we present simulation results obtained for the distributed methods considering two configurations of the monitoring microphones and using simulated acoustic channels. Since the purpose of the first simulation described in the previous section was to prove that the distributed version achieves the same behavior as the centralized one, the performance of those algorithms is omitted in this subsection. We focus on the performance comparison of the three distributed RM algorithms in a simulated soundfield. We have tried to simulate a quasi-anechoic environment by designing simulated acoustic channels thanks to a room impulse response tool [140] based on the image method for simulating small-room acoustics [141]. A virtual room with the same dimensions as the listening room located at the Audio Processing Laboratory has been simulated. With the aim to characterize and compare the acoustic response of the real and the virtual room, a simulated impulse response and a real acoustic channel of one loudspeaker-microphone pair used at the previous section are depicted in Figure 5.37(a) and Figure 5.37(c), respectively. The frequency response curve of both acoustic channels are also shown in Figure 5.37(b) and Figure 5.37(d), respectively. Both responses can be considered as representative measures of the whole (two PACs) acoustic system at each case.

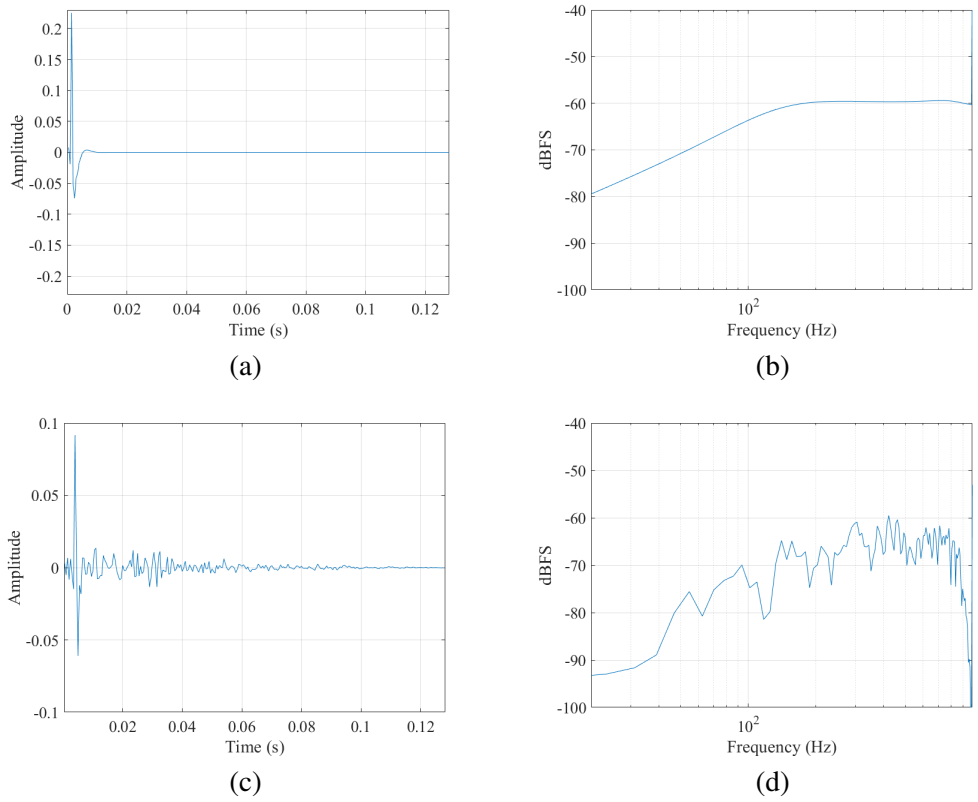
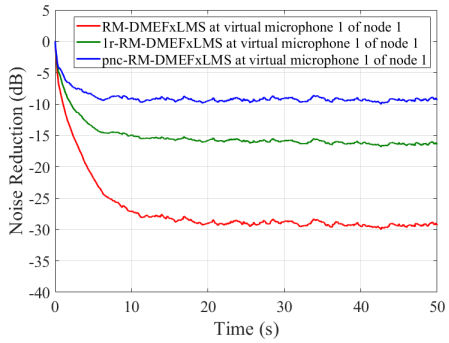
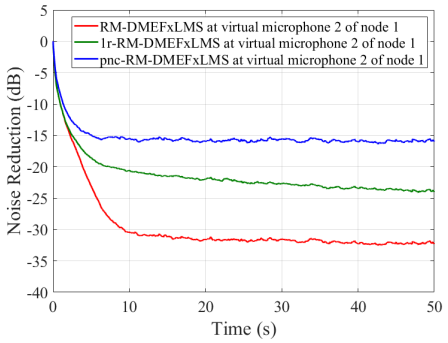


Figure 5.37: Simulated (a) and real measured (c) acoustic impulse response with their correspondig frequency responses (b) and (d) respectively, of a loudspeaker-microphone pair.

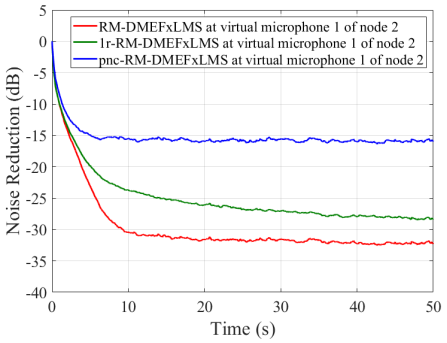
Regarding the configurations of the SCNs, first, we have designed a scheme of two simulated PAC systems with the same location of the transducers as shown in Figure 5.34 with the aim to compare simulation results. In the second scheme, the monitoring microphones are arranged in a linear array between the lousdspeakers at each SCN, as shown in Figure 5.39. At each SCN, the separation distance between the monitoring microphones is 8 cm and the virtual microphones were separated 20 cm away from the loudspeakers. Figure 5.38 illustrates the time evolution of the noise reduction of the distributed ANC headrest system in dB for the three distributed RM algorithms using simulated acoustic channels. As initially expected, in all the virtual microphones positions, the RM-DMEFxLMS algorithm outperforms the behaviour of its two approaches providing 28.8 dB, 31.9 dB, 31.9 dB and 28.9 dB respectively. It is interesting to note that, as expected, the performance of the 1r-RM-DMEFxLMS algorithm improves as we get closer to the last node of the network since, as previously commented, it will have the more accurate estimation of the disturbance signals



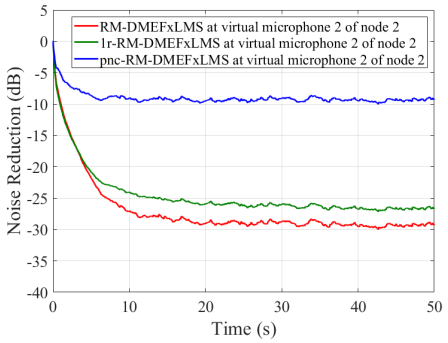
(a) Virtual microphone 1 (left) at PAC 1



(b) Virtual microphone 2 (right) at PAC 1



(c) Virtual microphone 1 (left) at PAC 2



(d) Virtual microphone 2 (right) at PAC 2

Figure 5.38: Noise reduction obtained by the distributed ANC headrest system for the proposed distributed RM algorithms.

at the virtual error sensors. The attenuation obtained for this strategy is 16.6 dB and 23.7 dB at the virtual microphones 1 and 2 of the first SNC, and 28.1 dB and 26.2 dB at the virtual microphones 1 and 2 of the second SNC, respectively. Finally, the pnc-RM-DMEFxLMS algorithm gets the worst performance in both control zones obtainig a noise reduction of 8.9 dB in both the left virtual microphone of the PAC 1 and the right virtual microphone of the PAC 2 and a noise reduction of 15.7 dB in both the right virtual microphone of the PAC 1 and the left virtual microphone of the PAC 2.

However, in order to test whether the position of the monitoring microphones influences the behavior of the distributed RM algorithms, the second scheme depicted in Figure 5.39 is used. The results for this scheme are depicted in Figure 5.40. In this case, the noise reduction curves present the same behaviour for the RM-DMEFxLMS and the pnc-RM-DMEFxLMS algoritm providing an attenuation of more than 30 dB for both SCNs. On the other hand, the 1r-RM-DMEFxLMS algoritm presents a behavior similar to the previous scheme, i.e., the closer to the last node, the larger noise

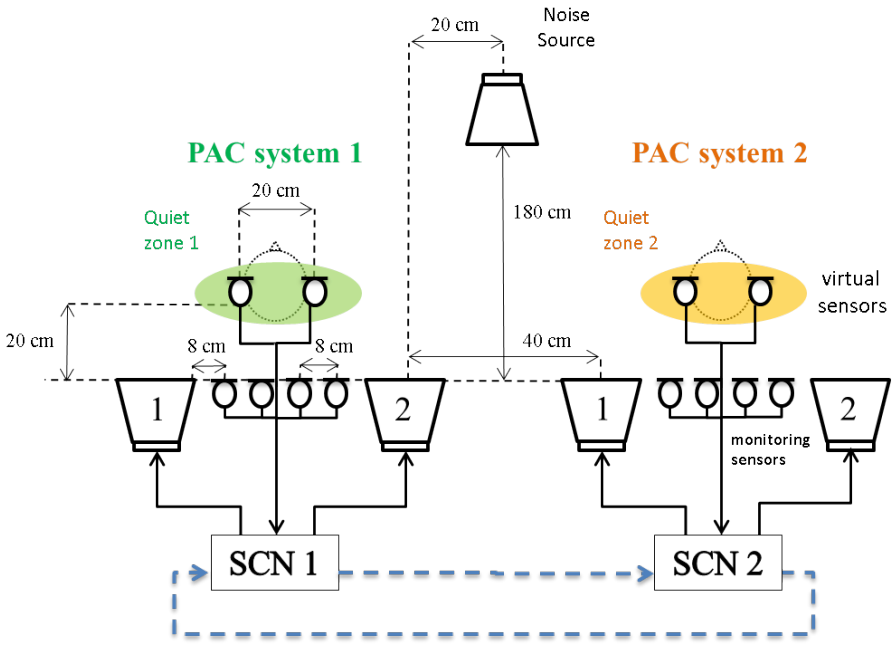
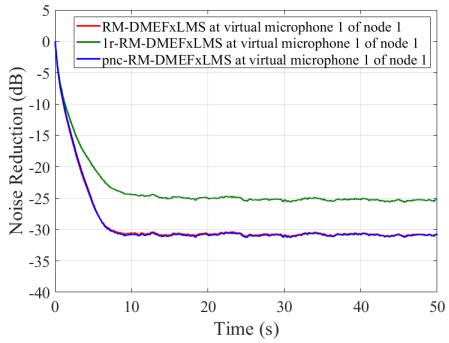


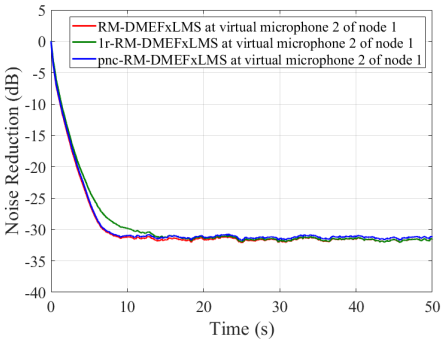
Figure 5.39: Distributed ANC headrest system composed of two simulated PAC systems. Dashed blue lines represent the incremental communication strategy. Monitoring microphones are located between the loudspeakers at each SCN.

attenuation. At the first virtual microphone of PAC 1, the 1r-RM-DMEFxLMS algorithm provides an attenuation of 5.6 dB below the other two algorithms, achieving a noise reduction of 25.2 dB. At the virtual microphone 2 of node 1 and at the virtual microphone 1 of node 2 (closer microphones from different nodes), the one round strategy achieves the same noise reduction as the other two methods (31.2 dB) but converging a litlle bit slower. Finally, At the virtual microphone 2 of PAC 2, the 1r-RM-DMEFxLMS algorithm presents the best performance of the three methods obtaining a noise reduction of 33 dB.

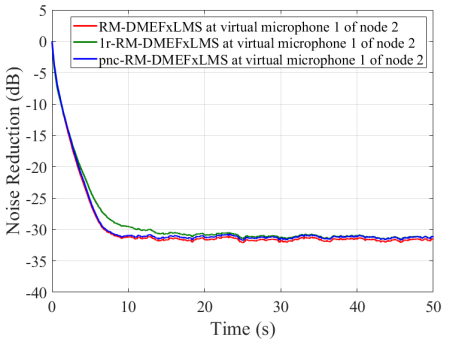
Therefore, as a summary of this subsection, we can conclude that, in ideal conditions, the relative physical location between virtual and monitoring microphones is a critical parameter that influence on the performance of the distributed RM algorithms. In other words, the shapes of the quiet zones around the virtual microphones depend on the location of the monitoring microphones, as demonstrated in [135].



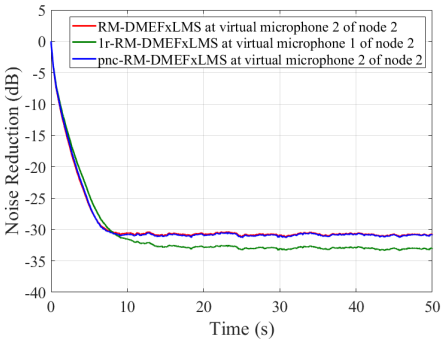
(a) Virtual microphone 1 (left) at PAC 1



(b) Virtual microphone 2 (right) at PAC 1



(c) Virtual microphone 1 (left) at PAC 2



(d) Virtual microphone 2 (right) at PAC 2

Figure 5.40: Noise reduction obtained by the distributed ANC headrest system for the proposed distributed RM algorithms.

5.7.3 Conclusions

In this section, the MEFxLMS algorithm using the RM technique has been implemented on a distributed ANC headrest system over an ideal ASN using a collaborative incremental strategy, denoted as RM-DMEFxLMS algorithm. In addition, two distributed RM approaches have been derived from the RM-DMEFxLMS to ensure ANC headrest system stability while computational and communication demands are reduced. We have carried out simulations to show the performance of the different approaches in a network composed of two PAC systems. Results show that the RM-DMEFxLMS algorithm exhibits the same performance as its centralized version (RM-MEFxLMS) when there are no communication constraints in the network. In addition, we have compared the performance of the RM-DMEFxLMS algorithm with its two versions, called 1r-RM-DMEFxLMS algorithm and pnc-RM-DMEFxLMS algorithm in order to justify its use in certain configurations of distributed ANC headrest systems. To this end, a study of implementation aspects such as computational

complexity and communication capabilities among the nodes in the network for the different distributed strategies is also presented. Results demonstrate that, considering this particular setting, the two proposed variants achieve a stable performance but reducing the computational and communication demands of the RM-DMEFxLMS algorithm. However, the performance of these two approaches may vary depending on the location of the monitoring microphones. Therefore, an analysis of the influence of the microphones location over the behaviour of the proposed distributed RM algorithms in real enviroments is suggested for future works.

5.8 Conclusions

This chapter has described different distributed algorithms considering implementation aspects present in the practical performance of ANC systems over networks composed of collaborative acoustic nodes. To this end, most of these collaborative algorithms have been based on the DMEFxLMS algorithm and how this distributed strategy is derived from its centralized version.

Since acoustic nodes are composed of power constrained actuators, the use of leaky distributed algorithms in practical cases has been justified to keep the output constraint controlled. Furthermore, the proposed 1*r re-scaling l-DMEFxLMS* algorithm has shown a good trade off between control effort and computational cost but reducing significantly the data transfer demands of the network. When higher convergence speed is desired in ASNs, we have proposed the *DFxAPL-I* strategy, i.e., a distributed version of the FxAPL-I algorithm which achieves the same performance as its centralized version showing faster speed of convergenge than the DMEFxLMS algorithm. Moreover, as a previous step to the real-time implementation of distributed ANC systems, the DMEFxLMS algorithm has been implemented in the frequency domain and working with blocks of samples as well as considering both incremental and diffusion mode of cooperation. These strategies are denoted as *incremental FPBFxLMS* and *diffusion FPBFxLMS* algorithms. It has been also proved that the computational complexity of the distributed algorithms increases significantly as the number of nodes increases. Due to this and with the aim to reduce computational and communication burden of ASNs, a clustered version of the DMEFxLMS algorithm has been proposed and denoted as *cl-DMEFxLMS* algorithm. Considering this node-specific-based approach, only a subset of nodes is considered to solve the distributed ANC problem in order to avoid the exchange of redundant information amnog nodes. Finally, to avoid the uncomfortable location of error microphones, a distributed implementation of an ANC system considering the remote microphone technique, called *RM-DMEFxLMS* algorithm, has been proposed. In addition, two proposed variants of this strategy, called *1r-RM-DMEFxLMS* and *pnc-RM-DMEFxLMS* algorithms, have been also presented to reduce the computational and communication demands of the

RM-DMEFxLMS algorithm in ASNs.

Chapter 6

ANE applications over distributed networks

In previous chapters, the use of distributed solutions for ANC over ASN has been demonstrated to be a flexible and efficient solution for the creation of local quiet zones in enclosures. However, in some SFC applications, instead of noise cancelling, it may be desirable to retain a residual noise with a certain spectral shape usually for safety reasons. In that cases, the use of active noise equalization (ANE) systems will allow to modify the spectral content of the disturbance noise at each point where the control is exercised with the aim to create a desired acoustic sound field. In this regard, this chapter deals with the distributed implementation of multichannel ANE algorithms over an ASN with a collaborative learning and composed of acoustic nodes. To our knowledge, no distributed algorithms over ASNs for ANE systems have been already reported. The objective of the nodes is to create personal sound zones with different equalization profiles which is also novel since until now, the same equalization profile used to be considered in the different control zones. To this end, several LMS-based narrowband and broadband approaches are considered. Different simulation results of their performance are also presented in order to verify the theoretical solutions.

6.1 Distributed narrowband ANE algorithms

Consider a SFC application devoted to controlling a narrowband noise composed of periodic components at different points in space for multiple users. The objective is that the tonal noise perceived by each user is equalized according to some defined spectral shape. To this end, we consider an ASN composed of N single-channel nodes as shown in Figure 6.1. We aim at equalizing one periodic noise signal in the sensors location, $d_k(n)$, designing an adaptive filter at each node, $w_k(n)$. For

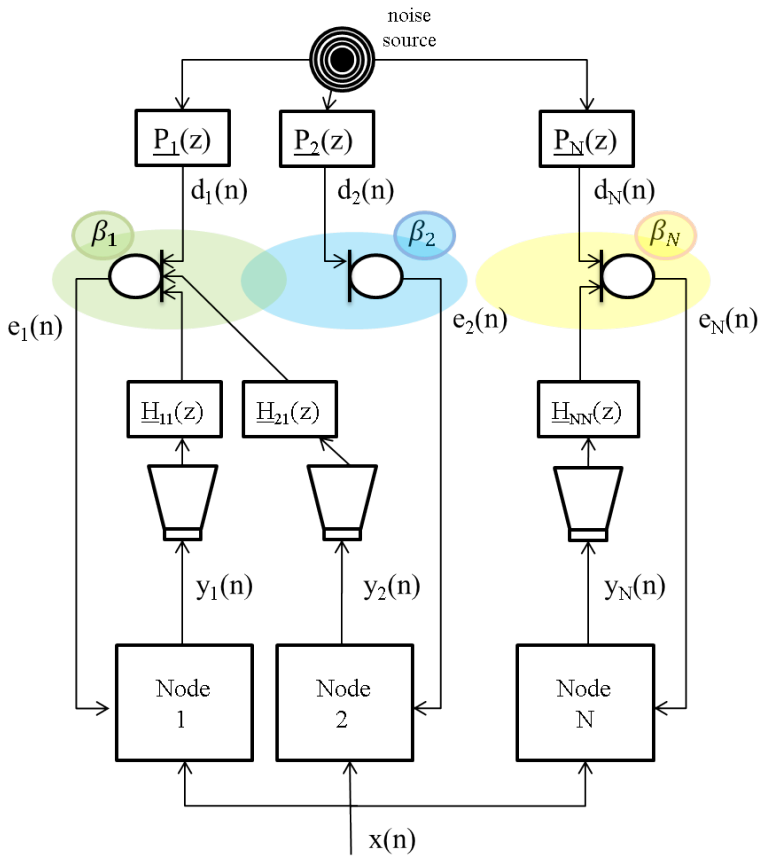


Figure 6.1: Scheme of a single-frequency ANE system working over an ASN of N single-channel nodes with different equalization profiles β_k.

simplicity, we start considering the single-frequency case, i.e. the reference signal $x(n)$ is a tone with a sinusoidal waveform generated internally which has the same frequency, ω_0, as the corresponding harmonic to be controlled. Defining $\underline{P}_k(z)$ as the acoustic transfer function that links the noise source with the k-th monitoring sensor and $\underline{H}_{jk}(z)$ as the acoustic transfer function that links the j-th actuator with the k-th sensor, the k-th error signal is given by

$$\underline{E}_k(z) = \underline{D}_k(z) + \sum_{j=1}^{N} \underline{Y}_j(z)\underline{H}_{jk}(z), \tag{6.1}$$

where $\underline{Y}_j(z)$ is the z-transform of the output signal at the j-th actuator. Our objective is to apply a different equalization profile β_k at each sensor at a single frequency ω_0.

Therefore, for $z{=}e^{i\omega_0}$, we get that

$$\underline{E}_k(e^{i\omega_0}) = \beta_k \underline{D}_k(e^{i\omega_0}), \tag{6.2}$$

and, therefore

$$\underline{D}_k(z)(\beta_k - 1) = \sum_{j=1}^{N} \underline{Y}_j(z)\underline{H}_{jk}(z). \tag{6.3}$$

As previously commented, considering that $x(n)$ is a pure tone at frequency ω_0, it is straightforward to verify that $\underline{D}_k(e^{i\omega_0}) = \underline{X}(e^{i\omega_0})\underline{P}_k(e^{i\omega_0})$, and $\underline{Y}_j(e^{i\omega_0}) = \underline{X}(e^{i\omega_0})\underline{W}_j(e^{i\omega_0})$, being $\underline{X}(z)$ the z-transform of the reference signal and $\underline{W}_j(z)$ the z-transform of the j-th linear controller. Thus, (6.3) can be rewritten as

$$\underline{X}(e^{i\omega_0})\underline{P}_k(e^{i\omega_0})(\beta_k - 1)) = \sum_{j=1}^{N} \underline{X}(e^{i\omega_0})\underline{W}_j(e^{i\omega_0})\underline{H}_{jk}(e^{i\omega_0}), \tag{6.4}$$

where we can obtain $\underline{W}_j(e^{i\omega_0})$ by solving

$$\begin{bmatrix} \underline{H}_{11}(e^{i\omega_0}) & \underline{H}_{21}(e^{i\omega_0}) & \cdots & \underline{H}_{N1}(e^{i\omega_0}) \\ \underline{H}_{12}(e^{i\omega_0}) & \underline{H}_{22}(e^{i\omega_0}) & \cdots & \underline{H}_{N2}(e^{i\omega_0}) \\ & & \ddots & \\ \underline{H}_{1N}(e^{i\omega_0}) & \underline{H}_{2N}(e^{i\omega_0}) & \cdots & \underline{H}_{NN}(e^{i\omega_0}) \end{bmatrix} \begin{bmatrix} \underline{W}_1(e^{i\omega_0}) \\ \underline{W}_2(e^{i\omega_0}) \\ \vdots \\ \underline{W}_N(e^{i\omega_0}) \end{bmatrix} = \begin{bmatrix} (\beta_1 - 1)\underline{P}_1(e^{i\omega_0}) \\ (\beta_2 - 1)\underline{P}_2(e^{i\omega_0}) \\ \vdots \\ (\beta_N - 1)\underline{P}_N(e^{i\omega_0}) \end{bmatrix} \tag{6.5}$$

Note that the feasibility of this system configuration only depends on the acoustic paths $\underline{H}_{jk}(z)$ evaluated at frequency ω_0. If the left matrix in (6.5) is invertible, any equalization profile is theoretically reachable and $\underline{W}_j(e^{i\omega_0})$ can be obtained by solving (6.5). Since we consider the case of a single-frequency tonal noise, note that the control signal generated by the j-th adaptive filter is a real signal which can be calculated as a tonal reference signal $\cos(\omega_0 \mathrm{n})$, filtered by $\underline{W}_j(e^{i\omega_0})$, such as

$$\begin{aligned} y_j(n) &= |\underline{W}_j(e^{i\omega_0})|\cos(\omega_0 n + \angle(\underline{W}_j(e^{i\omega_0}))) \\ &= \mathrm{Re}\{\underline{W}_j(e^{i\omega_0})\}\cos(\omega_0 n) - \mathrm{Im}\{\underline{W}_j(e^{i\omega_0})\}\sin(\omega_0 n), \end{aligned} \tag{6.6}$$

which, considering $w_j(\omega_0){=}\mathrm{Re}\{\underline{W}_j(e^{i\omega_0})\}$ and $\widehat{w}_j(\omega_0){=}-\mathrm{Im}\{\underline{W}_j(e^{i\omega_0})\}$, it can be expressed as

$$\begin{aligned} y_j(n) &= w_j(\omega_0)\cos(\omega_0 n) + \widehat{w}_j(\omega_0)\sin(\omega_0 n) \\ &= w_j(\omega_0)x(n) + \widehat{w}_j(\omega_0)\widehat{x}((n)) \end{aligned} \tag{6.7}$$

However, since $\underline{P}_k(z)$ are usually unknown, adaptive algortihms are used in order to reach the filter coefficients $\underline{W}_j(e^{i\omega_0})$. In the next section, an iterative solution based on the MEFxLMS algorithm to design the adaptive filters which equalize a single frequency at different points in space with different equalization values is proposed.

6.1.1 Single-frequency multichannel ANE algorithm

To control a periodic noise, ANE systems aim to minimize a pseudo-error signal instead of the error signal picked up by the sensors. These systems usually use a gradient-stochastic method to adjust the output signal, $y_j(n)$, generated by its actuator j in a multichannel fashion (see Figure 6.2). Therefore, now the problem is how to define the pseudo-error signal to satisfy the condition depicted in (6.3). Considering perfect estimation, i.e. $\underline{\widetilde{H}}_{jk}(z)=\underline{H}_{jk}(z)$, we define the pseudo-error signal at k-th sensor as

$$\underline{E}'_k(z) = \underline{E}_k(z) + \sum_{j=1}^{N} g_{jk}\underline{Y}_j(z)\underline{H}_{jk}(z), \tag{6.8}$$

where the values of g_{jk} must be calculated to satisfy $\underline{E}_k(z)=\beta_k\underline{D}_k(z)$ for $z=e^{i\omega_0}$. Note that the expressions dependent on β_k are valid for $z=e^{i\omega_0}$ since equalization profiles are evaluated at frequency ω_0. Thus, (6.8) can be expressed as

$$\underline{E}'_k(z) = \underline{E}_k(z) + \beta_k \sum_{j=1}^{N} \frac{g_{jk}}{\beta_k}\underline{Y}_j(z)\underline{H}_{jk}(z). \tag{6.9}$$

When the pseudo-error signal is minimized, $\underline{E}'_k(z)\to 0$, and, therefore

$$\underline{E}_k(z) = -\beta_k \sum_{j=1}^{N} \frac{g_{jk}}{\beta_k}\underline{Y}_j(z)\underline{H}_{jk}(z). \tag{6.10}$$

We must calculate the values of g_{jk} that satisfy $\underline{D}_k(z)=-\sum_{j=1}^{N}\frac{g_{jk}}{\beta_k}\underline{Y}_j(z)\underline{H}_{jk}(z)$. Substituting (6.1) in (6.8), we get that

$$\underline{E}'_k(z) = \underline{D}_k(z) + \sum_{j=1}^{N} (1+g_{jk})\underline{Y}_j(z)\underline{H}_{jk}(z). \tag{6.11}$$

When $\underline{E}'_k(z)\to 0$, $\underline{D}_k(z)=-\sum_{j=1}^{N}(1+g_{jk})\underline{Y}_j(z)\underline{H}_{jk}(z)$, and therefore, for each pair (j,k), the follow expression can be proposed

$$\frac{g_{jk}}{\beta_k} = 1 + g_{jk}, \tag{6.12}$$

and, consequently, we obtain

$$g_{jk} = \frac{\beta_k}{1-\beta_k}. \tag{6.13}$$

Therefore, the new pseudo-error equation at each sensor is stated as follows

$$\underline{E}'_k(z) = \underline{E}_k(z) + \frac{\beta_k}{1-\beta_k} \sum_{j=1}^{N} \underline{Y}_j(z)\underline{H}_{jk}(z). \tag{6.14}$$

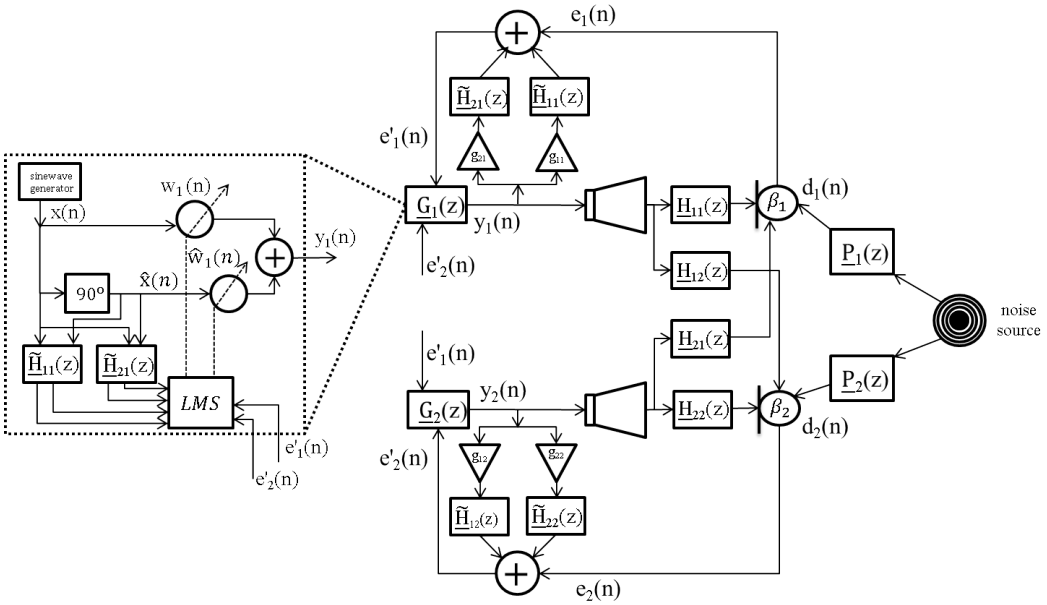


Figure 6.2: 1:2:2 single-frequency ANE based on the MeFxLMS algorithm

If $\underline{E}'_k(z)\rightarrow 0$, it is straightforward to verify that

$$\underline{E}_k(z) = -\beta_k \sum_{j=1}^{N} \frac{1}{1-\beta_k} \underline{Y}_j(z)\underline{H}_{jk}(z), \tag{6.15}$$

and

$$\begin{aligned}\underline{D}_k(z) &= -\sum_{j=1}^{N} (1+\frac{\beta_k}{1-\beta_k})\underline{Y}_j(z)\underline{H}_{jk}(z) = -\sum_{j=1}^{N} \frac{1-\beta_k+\beta_k}{1-\beta_k}\underline{Y}_j(z)\underline{H}_{jk}(z) \\ &= -\sum_{j=1}^{N} \frac{1}{1-\beta_k}\underline{Y}_j(z)\underline{H}_{jk}(z),\end{aligned} \tag{6.16}$$

which proves the desired expression $\underline{E}_k(z)=\beta_k\underline{D}_k(z)$. Equation (6.14) is valid for any equalization profile except for $\beta_k=1$. This means that the algorithm will not converge if the noise signal captured at some sensor is not equalized.

In the following section, we are interested in estimating an adaptive filter at each actuator, $w_j(n)$, that minimizes the pseudo-error signal at each sensor, $e'_k(n)$. To do this, we use an instantaneous gradient-descent method to estimate the coefficients in an iterative manner.

6.1.1.1. Adaptive solution

As depicted in Figure 6.2, our ANE system is based on a two-coefficient adaptive filter which control the in-phase and quadrature components of the reference signal,

$$x(n) = A\cos(\omega_0 \mathrm{n} + \phi_0), \tag{6.17}$$

where A is the amplitude, ω_0 is the frequency and ϕ_0 is the initial phase of sinusoid at the discrete time n. The quadrature reference signal is derived from $x(n)$ by using a 90$^{\mathrm{o}}$ phase shifter,

$$\widehat{x}(n) = A\sin(\omega_0 \mathrm{n} + \phi_0). \tag{6.18}$$

For simplicity, we assume for the following expressions that A=1 and ϕ_0=0. Considering (6.7), note that the control signal generated by the j-th actuator can be expressed as

$$y_j(n) = w_j(n)x(n) + \widehat{w}_j(n)\widehat{x}(n). \tag{6.19}$$

Now, the time domain version of the pseudo-error signal at the k-th sensor can be obtained as

$$e'_k(n) = d_k(n) + \frac{1}{1-\beta_k}\sum_{j=1}^{N}\{y_j(n) * \widetilde{h}_{jk}\}, \tag{6.20}$$

where $\widetilde{h}_{jk}$ is the estimated impulse response that links the j-th actuator with the k-th sensor. Assuming linearity, i.e., acoustic and electrical systems can be exchanged and using the MEFxLMS method on (6.20), the j-th filter updating equation is given by

$$\begin{aligned} w_j(n+1) &= w_j(n) - 2\mu\sum_{k=1}^{N}\frac{1}{1-\beta_k}\, e'_k(n)\{x(n) * \widetilde{h}_{jk}\}, \\ \widehat{w}_j(n+1) &= \widehat{w}_j(n) - 2\mu\sum_{k=1}^{N}\frac{1}{1-\beta_k}\, e'_k(n)\{\widehat{x}(n) * \widetilde{h}_{jk}\}, \end{aligned} \tag{6.21}$$

being μ the step-size parameter of the algorithm. The reference signal $x(n)$ filtered through $\widetilde{h}_{jk}$ at the frequency ω_0, can be calculated as

$$\begin{aligned} v_{jk}(n) &= \widetilde{A}_{jk} cos(w_0 n + \widetilde{\varphi}_{jk}) \\ \widehat{v}_{jk}(n) &= \widetilde{A}_{jk} sin(w_0 n + \widetilde{\varphi}_{jk}), \end{aligned} \tag{6.22}$$

with

$$\begin{aligned} \widetilde{A}_{jk} &= |\underline{\widetilde{H}}_{jk}(e^{jw_0})| \\ \widetilde{\varphi}_{jk} &= \angle(\underline{\widetilde{H}}_{jk}(e^{jw_0})). \end{aligned} \tag{6.23}$$

If we gather the in-phase and quadrature components of (6.21) and (6.22) in the following vectors

$$\mathbf{w}_j(n) = \begin{bmatrix} w_j(n) \\ \widehat{w}_j(n) \end{bmatrix}, \qquad \mathbf{v}_{jk}(n) = \begin{bmatrix} v_{jk}(n) \\ \widehat{v}_{jk}(n) \end{bmatrix}, \tag{6.24}$$

the j-th filter updating equation defined in (6.21), can be rewritten as

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) - 2\mu \sum_{k=1}^{N} \mathbf{v}_{jk}(n) \frac{1}{1-\beta_k}\, e'_k(n). \tag{6.25}$$

Thus, the global filter updating equation of the centralized single-frequency ANE algorithm is stated as follows

$$\mathbf{w}(n+1) = \mathbf{w}(n) - 2\mu \sum_{k=1}^{N} \mathbf{v}_k(n) \frac{1}{1-\beta_k}\, e'_k(n), \tag{6.26}$$

where vector $\mathbf{w}(n)=[\mathbf{w}_1(n)\ \mathbf{w}_2(n)\cdots\ \mathbf{w}_N(n)]^T$ of size $[2N\times 1]$ concatenates the N adaptive filters $\mathbf{w}_j(n)$ and $\mathbf{v}_k(n)$ is a vector of size $2N\times 1$ defined as $\mathbf{v}_k(n) = [\mathbf{v}_{1k}(n)\ \mathbf{v}_{2k}(n)\cdots\ \mathbf{v}_{Nk}(n)]^T$ with $\mathbf{v}_{jk}(n)$ defined in (6.24).

6.1.1.2. Convergence analysis

As it is well known, the convergence characteristics of the MEFxLMS algorithm will depend on the step-size parameter and it can be demonstrated [41] that μ must fulfill the following condition

$$\mu < \frac{1}{\mathrm{Tr}\{\mathbf{R}(\omega_0)\}}, \tag{6.27}$$

where the $[2N\times 2N]$ matrix $\mathbf{R}(\omega_0)$ is defined as

$$\mathbf{R}(\omega_0) = E\left\{ \sum_{k=1}^{N} \mathbf{v}_k(n)\mathbf{v}_k^T(n) \right\} \tag{6.28}$$

with $\mathbf{v}_k(n)$ depending of β_k and redefined as

$$\mathbf{v}_k^T(n) = \left[\frac{1}{1-\beta_k}\mathbf{v}_{1k}^T(n)\ \ \frac{1}{1-\beta_k}\mathbf{v}_{2k}^T(n)\ \ \cdots\ \ \frac{1}{1-\beta_k}\mathbf{v}_{Nk}^T(n) \right]. \tag{6.29}$$

Matrix $\mathbf{R}(\omega_0)$ can be also expressed as

$$\mathbf{R}(\omega_0) = \begin{bmatrix} \mathbf{R}_{11}(\omega_0) & \mathbf{R}_{12}(\omega_0) & \cdots & \mathbf{R}_{1N}(\omega_0) \\ \mathbf{R}_{21}(\omega_0) & \mathbf{R}_{22}(\omega_0) & \cdots & \mathbf{R}_{2N}(\omega_0) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{N1}(\omega_0) & \mathbf{R}_{N2}(\omega_0) & \cdots & \mathbf{R}_{NN}(\omega_0) \end{bmatrix}, \tag{6.30}$$

where the $[2\times 2]$ submatrix located at the f-row and c-column, $\mathbf{R}_{fc}(\omega_0)$, can be calculated as

$$\mathbf{R}_{fc}(\omega_0) = E\left\{ \sum_{k=1}^{N} \frac{1}{(1-\beta_k)^2} \mathbf{v}_{fk}(n)\mathbf{v}_{ck}^T(n). \right\} \tag{6.31}$$

Considering (6.22) and noting that

$$E\{A_1\cos(\omega_0 n+\phi_1)A_2\cos(\omega_0 n+\phi_2)\}=\frac{A_1A_2}{2}\cos(\phi_1-\phi_2), \tag{6.32}$$

$$E\{A_1\sin(\omega_0 n+\phi_1)A_2\sin(\omega_0 n+\phi_2)\}=\frac{A_1A_2}{2}\cos(\phi_2-\phi_1), \tag{6.33}$$

$$E\{A_1\cos(\omega_0 n+\phi_1)A_2\sin(\omega_0 n+\phi_2)\}=\frac{A_1A_2}{2}\sin(\phi_2-\phi_1), \tag{6.34}$$

(6.29) can be redefined as

$$\mathbf{R}_{fc}(w_0)=\sum_{k=1}^{N}\frac{A_{fk}A_{ck}}{2(1-\beta_k)^2}\begin{bmatrix}\cos(\Phi_{fk}-\Phi_{ck}) & \sin(\Phi_{ck}-\Phi_{fk})\\ \sin(\Phi_{fk}-\Phi_{ck}) & \cos(\Phi_{ck}-\Phi_{fk})\end{bmatrix} \tag{6.35}$$

Thus, the sum of the diagonal elements of $\mathbf{R}(\omega_0)$ in (6.27) is given by

$$\mathrm{Tr}\{\mathbf{R}(\omega_0)\}=\sum_{j=1}^{N}\sum_{k=1}^{N}\frac{A_{jk}^2}{(1-\beta_k)^2}. \tag{6.36}$$

and therefore (6.27) can be expressed as

$$\mu<\frac{1}{\sum_{j=1}^{N}\sum_{k=1}^{N}\frac{A_{jk}^2}{(1-\beta_k)^2}}. \tag{6.37}$$

Finally, we aim to implement (6.26) over an ASN composed of collaborative acoustic nodes which are devoted to equalize a tonal noise composed of multiple periodic components in some control area according to β_k. To this end, the multiple-frequency distributed version of (6.26) is outlined below.

6.1.2 Distributed multiple-frequency ANE algorithms

The multiple-frequency extension of (6.26) is firstly addressed. Our objective now is to control I different single-frequency reference signals. As we want to equalize multiple frequencies of the periodic noise, we will have at each sensor a equalization profile as the following vector $\boldsymbol{\beta}_k=[\beta_{1k},\beta_{2k},\ldots\ \beta_{Ik}]$ where β_{ik} is the equalization profile at the error sensor k related with the frequency w_i for i=1, 2, $\ldots$, I.

Now, the error signal at k-th sensor may be expressed as

$$e_k(n)=d_k(n)+\sum_{j=1}^{N}\{y_j(n)*\widetilde{h}_{jk}(n)\}, \tag{6.38}$$

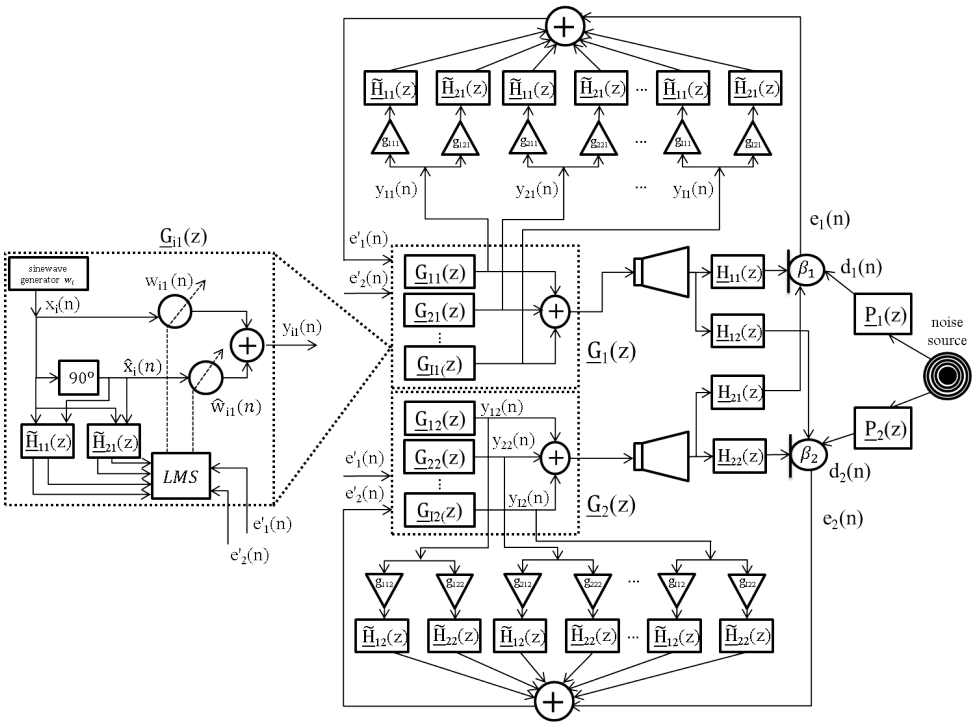


Figure 6.3: 1:2:2 multi-frequency ANE based on the MEFxLMS algorithm

with $y_j(n)$ defined as follows

$$y_j(n) = \sum_{i=1}^{I} y_{ij}(n) \tag{6.39}$$

being $y_{ij}(n)$ the output signal of the j-th adaptive filter which is fed with the i-th reference signal (see Figure 6.3). At this point, two strategies to calculate the pseudo-error signals are presented. In the first method, we use a different pseudo-error signal for each frequency related to the same error sensor (*multiple-pseudo-error strategy* [75]). However, in the second method, we define a only one pseudo-error signal for each error sensor (*common-pseudo-error strategy* [78]).

Therefore, by using the *multiple-pseudo-error* strategy, the i-th pseudo-error signal for each k-th sensor is defined as

$$e'_{ik}(n) = e_k(n) + \frac{\beta_{ik}}{1-\beta_{ik}} \sum_{j=1}^{N} \{y_{ij}(n) * \widetilde{h}_{jk}(n)\} \tag{6.40}$$

for i=1, 2, ..., I.

Let us define $\mathbf{v}_{ijk}(n)$=$[v_{ijk}(n)\ \widehat{v}_{ijk}(n)]^T$ with $v_{jk}(n)$=$\widetilde{A}_{ijk}cos(w_in$+$\widetilde{\varphi}_{ijk})$ and $\widehat{v}_{ijk}(n)$=$\widetilde{A}_{ijk}sin(w_in$+$\widetilde{\varphi}_{ijk})$, being $\widetilde{A}_{ijk}$=$|\underline{\widetilde{H}}_{jk}(e^{jw_i})|$ and $\widetilde{\varphi}_{ijk}$=$\angle(\underline{\widetilde{H}}_{jk}(e^{jw_i}))$. Thus, the update equation of the j-th adaptive filter for the i-th reference signal is given by

$$\mathbf{w}_{ij}(n+1) = \mathbf{w}_{ij}(n) - 2\mu_i \sum_{k=1}^{N} \mathbf{v}_{ijk}(n) \frac{1}{1-\beta_{ik}}\, e'_{ik}(n), \tag{6.41}$$

where $\mathbf{w}_{ij}(n) = [w_{ij}(n)\ \widehat{w}_{ij}(n)]^T$ and μ_i is the step-size parameter for the frequency w_i.

On the contrary, if we uses a common pseudo-error signal for each sensor (*common pseudo-error*), the pseudo-error signal at k-th sensor becomes

$$e'_k(n) = e_k(n) + \sum_{i=1}^{I} \frac{\beta_{ik}}{1-\beta_{ik}} \sum_{j=1}^{N} \{y_{ij}(n) * \widetilde{h}_{jk}\}. \tag{6.42}$$

And, therefore, for the i-th reference signal, the j-th filter updating equation is expressed as

$$\mathbf{w}_{ij}(n+1) = \mathbf{w}_{ij}(n) - 2\mu_i \sum_{k=1}^{N} \mathbf{v}_{ijk}(n) \frac{1}{1-\beta_{ik}}\, e'_{k}(n). \tag{6.43}$$

With the aim to define the global filter updating equations of both strategies, let us define $\mathbf{w}(n)$=$[\mathbf{w}_1^T(n)\ \mathbf{w}_2^T(n) \ldots\ \mathbf{w}_I^T(n)]^T$ as a vector of size [$2IN$×1] which concatenates the I filters vectors of size [$2N$×1] that store all the coefficients of all the nodes that control the frequency i, i.e., $\mathbf{w}_i(n) = [\mathbf{w}_{i1}^T(n)\ \mathbf{w}_{i2}^T(n)\ \ldots\ \mathbf{w}_{iN}^T(n)]^T$. Similarly, we define $\mathbf{v}_k(n) = [\mathbf{v}_{11k}^T(n)\ \mathbf{v}_{12k}^T(n)\ \ldots \mathbf{v}_{1Nk}^T(n)\ \mathbf{v}_{21k}^T(n)\ \ldots\ \mathbf{v}_{INk}^T(n)]^T$ as a vector of size [$2IN$×1] composed of the reference signal $x(n)$ filtered through the acoustic channels between the N actuators and the sensor of node k.

Thus, the global filter updating equation of the centralized multiple-frequency ANE algorithm considering the multiple-pseudo-error strategy is stated as follow

$$\mathbf{w}(n+1) = \mathbf{w}(n) - 2\sum_{k=1}^{N} \boldsymbol{\mu}_{k\beta e} \mathbf{v}_k(n), \tag{6.44}$$

where $\boldsymbol{\mu}_{k\beta e}$ is a diagonal matrix of size $2IN$×$2IN$ whose diagonal elements are the concatenation of I vectors of size $2N$×1 where the vector at the i-th position is $\frac{\mu_i}{1-\beta_{ik}} e'_{ik}$ replicated $2N$ times.

On the other hand, the global filter updating equation of the centralized multiple-frequency ANE algorithm considering the common-pseudo-error strategy is given by

$$\mathbf{w}(n+1) = \mathbf{w}(n) - 2\sum_{k=1}^{N} \boldsymbol{\mu}_{k\beta}\mathbf{v}_k(n)e'_k(n) \tag{6.45}$$

where $\boldsymbol{\mu}_{k\beta}$ is similar defined as $\boldsymbol{\mu}_{k\beta e}$ but considering that the vector at the i-th position is now defined as $\dfrac{\mu_i}{1-\beta_{ik}}$ replicated $2N$ times.

Thus, the pseudo-error signal of both strategies can be calculated as

$$e'_{ik}(n) = e_k(n) + \mathbf{v}_k^T(n)_{[2N(i-1)+1:2Ni]}\ \mathbf{w}_i(n), \qquad \text{(multiple-pseudo-error)} \tag{6.46}$$

$$e'_k(n) = e_k(n) + \mathbf{v}_k^T(n)\mathbf{w}(n). \qquad \text{(common-pseudo-error)} \tag{6.47}$$

In addition, note that, due to the independently control of multiple frequencies, μ_i can be calculated as

$$\mu_i < \frac{1}{\text{Tr}\{\mathbf{R}(\omega_i)\}}, \tag{6.48}$$

where

$$\text{Tr}\{\mathbf{R}(\omega_i)\} = \sum_{j=1}^{N}\sum_{k=1}^{N} \frac{\widetilde{A}_{ijk}^2}{(1-\beta_{ik})^2}. \tag{6.49}$$

Now, the goal is to distribute the calculation of the adaptive filters among the N single-channel nodes of the ANE system described in Figure 6.1 but considering a ring topology with incremental communication. To this end, the distributed version of both strategies is calculated by splitting up the contribution of each node in both (6.44) and (6.45). From (6.40) and (6.42), it can easily be shown that that the calculation of both pseudo-error signals at k-th node should involve to have access to the control signals of the rest of the nodes of the network. Due to this information is not available to the nodes in a fully distributed network, each node must calculate a estimated pseudo-error signal at each node by using a local version of $\mathbf{w}(n)$. Therefore, similarly to the distributed procedure detailed in previous chapters, we consider $\mathbf{w}(n)$ as the global state of the network and $\mathbf{w}^k(n)$ as a local version of $\mathbf{w}(n)$ at the k-th node.

Thus, we get that the updating filter equation of the k-th node by using the distributed multiple-pseudo-error multiple-frequency ANE algorithm is given by

$$\mathbf{w}^k(n) = \mathbf{w}^{k-1}(n) - 2\widetilde{\boldsymbol{\mu}}_{k\beta e}\mathbf{v}_k(n), \tag{6.50}$$

where $\widetilde{\boldsymbol{\mu}}_{k\beta e} = \dfrac{\mu_i}{1-\beta_{ik}}\widetilde{e}'_{ik}(n)$ with $\widetilde{e}'_{ik}(n)$ as the estimated version of $e'_{ik}(n)$ which has access only to its own information at the k-th node for the i-th frequency

$$\widetilde{e}'_{ik}(n) = e_k(n) + \mathbf{v}_k^T(n)_{[2N(i-1)+1:2Ni]}\mathbf{w}^k(n)_{[2N(i-1)+1:2Ni]}\,. \tag{6.51}$$

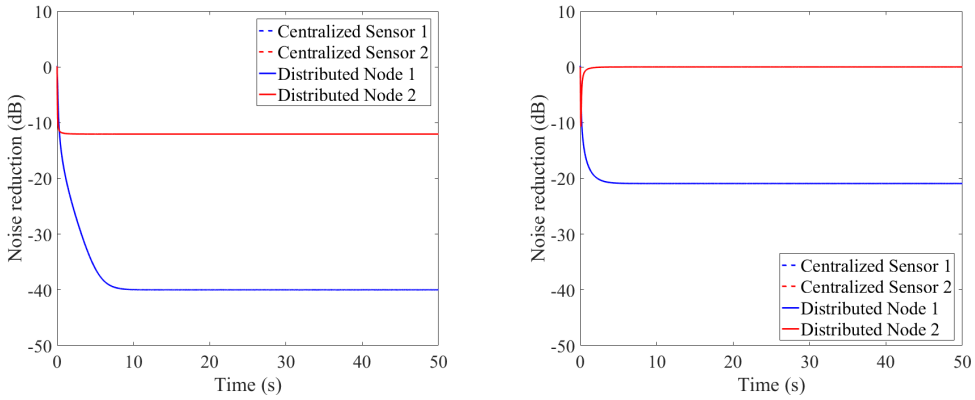


Figure 6.4: Noise reduction obtained by both the two-node distributed common-pseudo-error ANE system (solid line) and the centralized common-pseudo-error ANE system (dashed line) with a 1:2:2 configuration with different equalization profiles: (a) β_1=[0.1 0.1 0.1] and β_2=[0.5 0.5 0.5] and (b) β_1=[0.3 0.3 0.3] and β_2=[0.9 0.9 0.9].

Similarly, the updating filter equation of the distributed common-pseudo-error version can be expressed as

$$\mathbf{w}^k(n) = \mathbf{w}^{k-1}(n) - 2\boldsymbol{\mu}_{k\beta}\mathbf{v}_k(n)\widetilde{e}'_k(n), \tag{6.52}$$

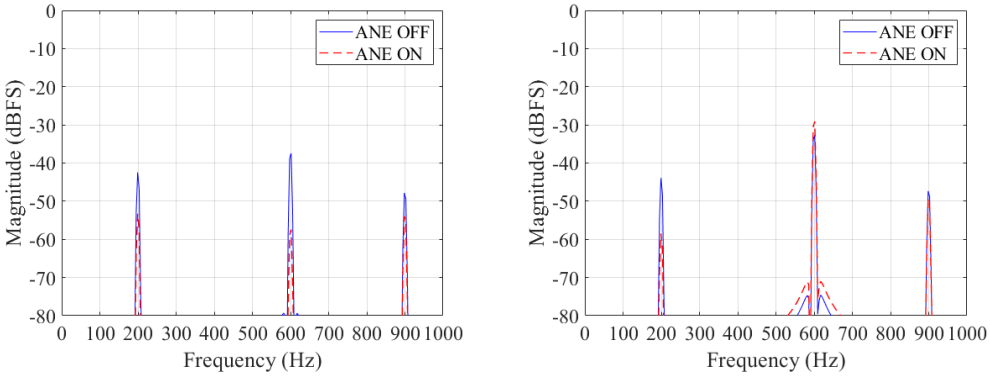
where $\widetilde{e}'_k(n)$ is the estimated version of $e'_k(n)$ which has access only to its own information, defined as

$$\widetilde{e}'_k(n) = e_k(n) + \mathbf{v}_k^T(n)\mathbf{w}^k(n)\cdot \tag{6.53}$$

6.1.3 Simulation results

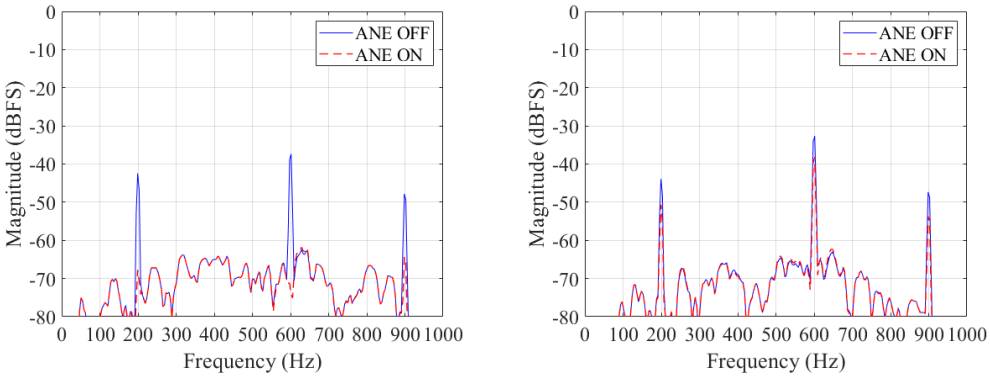
In this section, we have evaluated the performance of a multiple-frequency ANE system over several distributed networks for different equalization profiles. In previous works, it has been demonstrated that the common-pseudo-error method presents a more efficient and robust performance than the multiple-pseudo-error strategy achieving a very similar results in terms of equalization zones [78]. For these reasons, the performance of the multiple-pseudo-error ANE method has not been included in this analysis.

The simulations have benn carried out considering two scenarios. In the first scenario, the two-node ASN depicted in 4.3 is considered in order to control a tone with three harmonics at frequencies 200, 600 and 900 Hz. Moreover, to validate the



(a) $\beta_1(dB)$=[−10.45 −20.00 −6.02] (b) $\beta_2(dB)$=[−13.98 3.52 −1.93]

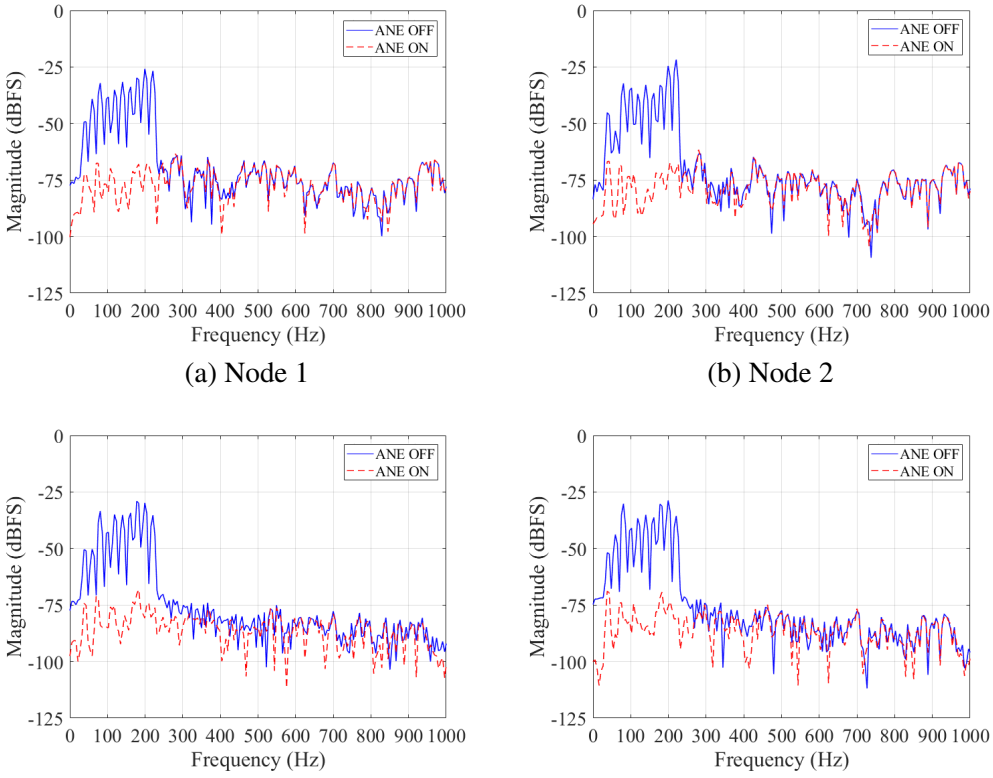
Figure 6.5: Power spectral density of a tone with three harmonics at frequencies 200, 600 and 900 Hz before (blue solid lines) and after (red dashed lines) the distributed common-pseudo-error ANE system for (a) node 1 with $\beta_1(dB)$=[−10.45 −20.00 −6.02] and (b) node 2 with $\beta_2(dB)$=[−13.98 3.52 −1.93].



(a) $\beta_1(dB)$=[−∞ −∞ −∞] (b) $\beta_2(dB)$=[−6.02 −6.02 −6.02]

Figure 6.6: Power spectral density of a noisy tone with three harmonics at frequencies 200, 600 and 900 Hz and SNR=15 dB before (blue solid lines) and after (red dashed lines) the distributed common-pseudo-error ANE system for (a) node 1 with $\beta_1(dB)$=[−∞ −∞ −∞] (β_1=[0 0 0]) and (b) node 2 with $\beta_2(dB)$=[−6.02 −6.02 −6.02] .

robustness of the algorithm, a zero-mean gaussian white noise with a signal-to-noise ratio (SNR), equal to =15 dB is added to the tonal noise in some simulations. To this end, the influence of this ambient noise on the ANE system performance has



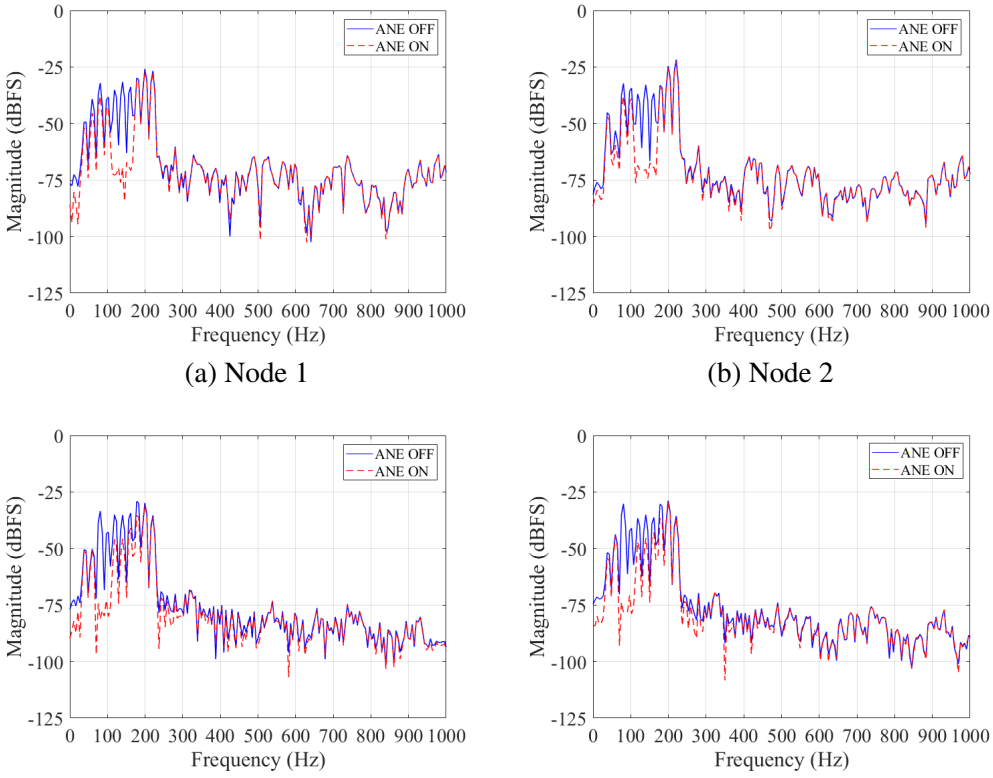
(a) Node 1 (b) Node 2

(c) Node 3 (d) Node 4

Figure 6.7: Power spectral density of a noisy tone with ten harmonics and SNR=15 dB before (blue solid lines) and after (red dashed lines) the distributed common-pseudo-error ANE system for a four-node network with β_k=[0 0 0 0 0 0 0 0 0 0] for all nodes.

been analyzed. In the second scenario, a synthesized noise with ten harmonics at frequencies 40, 60, 80, 100, 120, 140, 160, 180, 200, 220 Hz has been considered as reference signal which is aimed to be equalized by the four-node ASN depicted in 4.3. Regarding the acoustic system, in both scenarios, we consider the real acoustic channels presented in Section 3.3. In addition, the step size parameter is theoretically calculated as explained in (6.48) for each frequency and for each node. Differents values of the equalization profile β_k have been considering to validate the independent control at each node.

Cosidering the first scenario, in the first simulation, the performance comparison between the centralized common-pseudo-error multiple-frequency ANE algorithm and its distributed version is presented. Figure 6.4 show the time evolution of the



(a) Node 1 (b) Node 2

(c) Node 3 (d) Node 4

Figure 6.8: Power spectral density of a noisy tone with ten harmonics and SNR=15 dB before (blue lines) and after (red lines) the distributed common-pseudo-error ANE system for a four-node network with β_1=β_2=[0.5 0.5 0.5 0.5 0 0 0 0.9 0.9 0.9] and β_3=β_4=[0.9 0.9 0 0 0.3 0.3 0.5 0.5 0.9 0.9] .

$NR_k(n)$ for both algorithms at both sensors in the *1:2:2* centralized case and at both nodes in the case of the two-node distributed approach. As expected, the distributed implementation exhibits exactly the same results as its centralized version in terms of convergence speed and final noise reduction achieving an attenuation according to the selected equalization profile. However, in the time domain, it is not possible to appreciate the selective control at each frequency that the ANE system can provide. Therefore, in Figure 6.5, the power spectral density (PSD) of the same reference signal with and without the distributed common-pseudo-error ANE control system is shown. Now, each node has different equalization profiles for the three tonal components, that is, β_1=[0.3 0.1 0.5] and β_2=[0.2 1.5 0.8]. For each node, it can be seen in Figure 6.5.(a) and Figure 6.5.(b) that the value of the selected β_k for each frequency agrees with the attenuation obtained in dB, achieving then the desired equalization at

each node. Simultaneously, the robustness of the ANE system is also evaluated by adding a white noise with a SNR=15 dB with respect to the tonal noise. For simplicity, each node has the same β_k for the three frequencies. At node1, it is intended to cancel the periodic noise signal (β_1=[0 0 0]) while in node 2, it is intended to attenuate the tonal components by half (β_2=[0.5 0.5 0.5]). It can be seen in Figures 6.6(a) and 6.6(b) that the distributed ANE algorithm equalizes perfectly the frequency components to be controlled, mantaining the rest of the spectral components unaltered. Finally, the behaviour of the distributed ANE system is evaluated for different equalization profiles but increasing the number of frequencies to control as well as the number of nodes. To this end, the second scenario is considered. In Figures 6.7 and 6.8, the PSD of the a noisy tone with ten harmonics and SNR=15 dB before and after the control of the distributed common-pseudo-error ANE system for a four-node network and considering different equalization profiles is depicted. In Figure 6.7, the network aims to cancel the multi-frequency tonal noise at the area controlled by the sensors of the four nodes. On the ohter hand, in Figure 6.8, different equalization profiles by pairs of nodes has been selected, β_1=β_2=[0.5 0.5 0.5 0.5 0 0 0 0.9 0.9 0.9] and β_3=β_4=[0.9 0.9 0 0 0.3 0.3 0.5 0.5 0.9 0.9]. From the results obtained in both figures, it can be observed that the distributed common-pseudo-error ANE algorithm achieves to equalize each frequency as desired avoiding to modify the rest of the spectral content.

6.2 Distributed broadband ANE algorithms

As demonstrated in the previous section, the narrowband ANE provides an individually control at each harmonic of a periodic noise. This strategy assumes that the real noise signal and the synthesized reference signal internally generated contains the same number of tonal components at the same frequencies. However, in real scenarios, the tonal noise signal aim to be equalized may contain other uncorrelated noise components. When these undesired interferences appear at frequencies where the magnitude response of the acoustic channels has high gain, the ANE system will suffer from a passband disturbance [142]. In those cases, it is possible that the narroband strategy can be extended to become a broadband ANE [92] using a shaping filter to shape the desired noise spectrum while reducing the passband disturbance caused by uncorrelated noise. In this way, the broadband ANE system can be used for both tonal and broadband noises. Consequently, this section deal with the distributed implementation of broadband ANE systems over networks composed of collaborative acoustic nodes.

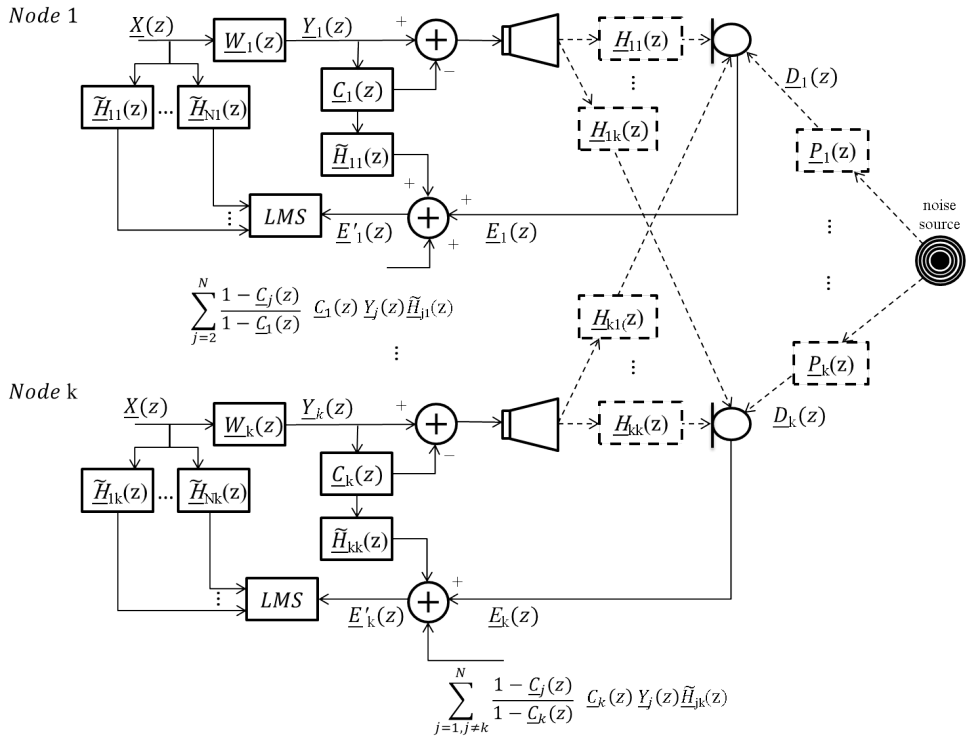


Figure 6.9: Block diagram of the multichannel broadband filtered-x LMS ANE algorithm when $\underline{G}(z)=1-\underline{C}(z)$.

6.2.1 Distributed broadband MEFxLMS ANE algorithm

Starting from the single-channel version of the broadband ANE system explained in Section 2.4.2, its multichannel extension can be directly derived as follows.

Considering a centralized N-node ASN as depicted in Figure 6.9. Then, N sensors, N actuators, N adaptive filters, N shaping filters and $N\times N$ acoustic channels are considered. The objective is that the signal picked up by the k-th error sensor is designed as

$$\underline{E}_k(z)=\underline{C}_k(z)\underline{D}_k(z), \tag{6.54}$$

being $\underline{D}_k(z)$ and $\underline{C}_k(z)$ the noise signal and the shaping filter at the node k, respectively. Note that the information captured by the error sensor at the node k is calculated as

$$\underline{E}_k(z) = \underline{D}_k(z) + \sum_{j=1}^{N} \underline{G}_j(z)\underline{Y}_j(z)\underline{H}_{jk}(z), \tag{6.55}$$

where $\underline{G}_j(z)$ is a generic control filter which affects to the control signal $\underline{Y}_j(z)$ at the node j and $\underline{H}_{jk}(z)$ is the acoustic transfer function that links the actuator of node j with the error sensor of node k.

The objective is to obtain a pseudo error signal that, when minimized, causes the desired equalization effect. To this end, the following pseudo error signal at the k-th node is proposed

$$\underline{E}'_k(z) = \underline{E}_k(z) + f_k\{\underline{Y}_j(z), \underline{H}_{jk}(z), \underline{G}_j(z)\}. \tag{6.56}$$

Therefore, it is required to calculate a function at each node k which depends on the control signals generated by all the nodes as well as some multichannel design parameters that allow us to achieve the desired equalization by minimizing $\underline{E}'_k(z)$. Substituting (6.54) in (6.55), we obtain

$$\underline{D}_k(z) = -\sum_{j=1}^{N} \frac{\underline{G}_j(z)}{1-\underline{C}_k(z)} \underline{Y}_j(z)\underline{H}_{jk}(z). \tag{6.57}$$

Similarly as the single-channel case, multiplying both sides of (6.57) by $\underline{C}_k(z)$ and when $\underline{E}'_k(z) \to 0$, we get

$$f_k\{\underline{Y}_j(z), \underline{H}_{jk}(z), \underline{G}_j(z)\} = \underline{C}_k(z) \sum_{j=1}^{N} \frac{\underline{G}_j(z)}{1-\underline{C}_k(z)} \underline{Y}_j(z)\underline{H}_{jk}(z). \tag{6.58}$$

and therefore, the pseudo-error signals can be obtained as

$$\underline{E}'_k(z) = \underline{E}_k(z) + \underline{C}_k(z) \sum_{j=1}^{N} \frac{\underline{G}_j(z)}{1-\underline{C}_k(z)} \underline{Y}_j(z)\underline{H}_{jk}(z). \tag{6.59}$$

Once $\underline{E}'_k(z)$ has been calculated, the MEFxLMS algorithm is considered to try to minimize a cost function $J(n)$ which depends on the pseudoerror signals, that is, $J(n)=\{\sum_{k=1}^{N} e'^2_k(n)\}$. To this end, substituting (6.55) in (6.59), we obtain that

$$\underline{E}'_k(z) = \underline{D}_k(z) + \sum_{j=1}^{N} \underline{S}_{jk}(z)\underline{Y}_j(z). \tag{6.60}$$

where

$$\underline{S}_{jk}(z) = \frac{\underline{G}_j(z)\underline{H}_{jk}(z)}{1-\underline{C}_k(z)}. \tag{6.61}$$

The time domain version of (6.60) in steady-state can be expressed as

$$e'_k(n) = d_k(n) + \sum_{j=1}^{N} x_{jk}(n) * w_j(n). \tag{6.62}$$

where $x_{jk}(n)=s_{jk}(n)*x(n)$. It should be noted that the calculation of $s_{jk}(n) = (g_j(n)*h_{jk}(n))*(\delta(n)-c_k(n))^{-1}$ can be done in a previously stage. It should be taken into account that, in order to avoid phase problems, $\delta(n)$ should be moved to the sample in which the design of the rest of the filters is causal. To obtain the coefficients of the j-th adaptive filter that minimize $J(n)$, the MEFxLMS strategy is used but applied to the pseudo error signals, that is,

$$\mathbf{w}_j(n) = \mathbf{w}_j(n-1) - \mu\nabla_{\mathbf{w_j}} \sum_{k=1}^{N} e_k'^2(n) \tag{6.63}$$

where

$$\nabla_{\mathbf{w_j}} \sum_{k=1}^{N} e_k'^2(n) = \sum_{k=1}^{N} 2e_k'(n)\nabla_{\mathbf{w_j}} \left(\sum_{k=1}^{N} e_k'(n) \right). \tag{6.64}$$

Considering that (6.62) can be expressed in matricial form as

$$e_k'(n) = d_k(n) + \sum_{j=1}^{N} \mathbf{x}_{jk}^T(n)\mathbf{w}_j(n), \tag{6.65}$$

note that

$$\nabla_{\mathbf{w_j}} \left(\sum_{k=1}^{N} e_k'(n) \right) = \sum_{k=1}^{N} \mathbf{x}_{jk}(n) \tag{6.66}$$

where $\mathbf{x}_{jk}(n)$ is the vector that contain the last L samples of $x_{jk}(n)$. Therefore, the filter updating equation of the centralized broadband MEFxLMS ANE algorithm at the j-th node is given by

$$\mathbf{w}_j(n) = \mathbf{w}_j(n-1) - 2\mu \sum_{k=1}^{N} \mathbf{x}_{jk}(n)e_k'(n). \tag{6.67}$$

In order to implement the broadband ANE algorithm in a collaborative incremental ASN as depicted in Figure 2.2.(b), note that each node k must previously know the desired spectral shape applied to its sensor (whose impulse response is $c_k(n)$), all the control filters, $g_j(n)$ and the acoustic channels that link the actuator of each node with its sensor, $s_{jk}(n)$ (for $j=1, 2, \ldots, N$). From this information and similarly as the centralized system, it will be possible to previously calculate the auxiliary filters required by the proper performance of the algorithm. As developed in previous distributed algorithms, we obtain the formulation for this approach by splitting the sum of the updating equation (5.56) into the contributions of each node,

$$\mathbf{w}^k(n) = \mathbf{w}^{k-1}(n) - \mu\mathbf{x}_{jk}(n)e_k'(n), \tag{6.68}$$

where $\mathbf{w}^k(n)$=[$[\mathbf{w}_1^k(n)]^T$ $[\mathbf{w}_2^k(n)]^T$ ... $[\mathbf{w}_N^k(n)]$]^T is the local version of the global state of the network, $\mathbf{w}(n)$, at node k being $\mathbf{w}_j^k(n)$ the estimated adaptive filter of the node j calculated at node k. A summary of the distributed broadband (broadband DMEFxLMS ANE) algorithm pseudocodes executed per sample time at each node is given in ***Algorithm 11***. Note that the pseudoerror signal must be also calculated in a distributed way. It can be observed in Figure 6.9 that the control signals of the rest of the $N-1$ nodes are required in the pseudo error signal calculation at k-th node. Since in a fully distributed network those signals are not available to the nodes, a estimation of the control signals of the network must be calculated at each node by using its local version of $\mathbf{w}(n)$. To this end, we define $y_j^k(n)$ as the estimated control signal at node k from node j, that is,

$$y_j^k(n) = [\mathbf{w}_j^k(n)]^T[\mathbf{X}(n)]_{(:,1)}, \tag{6.69}$$

with $\mathbf{w}_j^k(n)$=$\mathbf{w}^k(n-1)_{(1+L(j1):Lj)}$, $y_k^k(n)$=$y_k(n)$ and $\mathbf{w}_k^k(n)$=$\mathbf{w}_k(n-1)$. In this way, the output signal generated by the node k can be obtained as $out_k(n) = \mathbf{g}_k^T y_k^k(n)$ where $\mathbf{g}_j(n)$ is the impulse response of the N control filters modeled as a FIR filter of size M_g. In addition, note that, depending on the filters $\underline{C}_k(z)$, the auxiliary filters $\underline{S}_{jk}(z)$ can be approximated by means of a FIR filter, denoted as $\mathbf{s}_{jk}(n)$, and then, the transform domain may be used to simplify its calculation. In that case, the number of the FFT points should be sufficiently high to avoid frequency resolution problems when calculating the new filter, and as small as possible to avoid high order filters that introduce delays and high computational costs. To fulfill this trade-off, a FFT of size $2M_c$ is considered being M_c the size of the FIR filter $\mathbf{c}_k(n)$ that models the impulse response $c_k(n)$. Therefore, the size of these auxiliary filters is $M+2M_c-1$. It should be remembered that M is the size of the acoustic paths $\mathbf{h}_{jk}$. Similarly, considering $s'_{jk}(n)$=$h_{jk}(n)$*$(g_j(n)$*$c_k(n))$*$(\delta(n)-c_k(n))^{-1}$, the vector $\mathbf{s}'_{jk}$ is defined as a FIR filter of size $M+2M_c-1$ that models $s'_{jk}(n)$. Taking this into consideration and defining, $\mathbf{y}_j^{'k}(n)$ as a vector with the last $M+2M_c-1$ samples of $y_j(n)$, the pseudo error signal at the k-th node, $e'_k(n)$, can be now calculated (see Line 8 in ***Algorithm 11***). Finally, with the aim to update the local state of the network at each node, we must define $\mathbf{x}'(n)$ as a vector which contains the last $M+2M_c-1$ samples of $x(n)$. Note that this vector is required to properly filter $x(n)$ by $\mathbf{s}_{jk}(n)$ (see Line 10 in ***Algorithm 11***).

6.2.2 Distributed broadband MEFeLMS ANE algorithm

It should be noted that the use of the broadband DMEFxLMS ANE algorithm may require that IIR filters with the form $(1-\underline{C}(z))^{-1}$ must be used and approximated by FIR filters. In addition, the pseudo-error signals are calculated as an approximation of its centralized exact calculation. Another strategy for shaping the spectrum of the error signal is the use of the filtered-error LMS (FeLMS) algorithm [88]. The

Algorithm 11: *Broadband DMEFxLMS ANE*

$\mathbf{w}^0(n) = \mathbf{w}^N(n-1)$ % Needed at node k=1
for all *node* $1 \leq k \leq N$ **do**
  for all $1 \leq j \leq N$ **do**
    $\mathbf{w}_j^k(n) = \mathbf{w}^k(n-1)_{(1+L(j-1):Lj)}$ % Obtain local coefficients related to the j-th node.
    $y_j^k(n) = [\mathbf{w}_j^k(n)]^T[\mathbf{X}(n)]_{(:,1)}$ % j-th estimated output signal
    $out_k(n) = \mathbf{g}_k^T y_k^k(n)$ % output signal
  end for
  $e'_k(n) = e_k(n) + \sum_{j=1}^{N} \mathbf{s}'^T_{jk}\mathbf{y}'^k_j(n)$, % Calculate the pseudo error signal
  for all $1 \leq j \leq N$ **do**
    $x_{jk}(n) = \mathbf{s}_{jk}^T\mathbf{x}'(n)$ % Vector that contains reference signal filtered by auxiliary filters
    $\mathbf{w}_j^k(n) = \mathbf{w}_j^{k-1}(n) - 2\mu\mathbf{x}_{jk}(n)e'_k(n)$ % Update local state related to the j-th node.
  end for
end for
$\mathbf{w}(n) = \mathbf{w}^N(n)$ % Updated global state of the network
for all *node* $1 \leq k \leq (N-1)$ **do**
  $\mathbf{w}^k(n) = \mathbf{w}(n)$ % Disseminate global state of the network
end for

filtered-error structure allows the residual noise signal spectrum to be modified using a spectral shaping filter. In other words, the existing noise in the system is not directly equalized but the remaining residual noise in the system. This structure can be especially useful in the cases when some part of the noise signal spectrum is aimed to be cancelled maintaining other parts intact. In addition, the FeLMS approach remove the uncorrelated noise components from the signal captured by the sensors, thus reducing the undesired passband amplification [143] as well as the weight vector perturbation [142]. For these reasons, the distributed version of the broadband ANE algorithm based on the filtered-error LMS structure is outlined below.

The single-channel version of the broadband FeLMS algorithm is depicted in Figure 6.10. From which it can be deduced that the final residual error signal spectrum fulfills that,

$$\underline{E}_k(z) = \underline{C}_k^{-1}(z)\underline{E}'_k(z). \tag{6.70}$$

Note that when $|\underline{C}(z)|=1$, the ANE is a cancellation system and $\underline{E}_k(z)=\underline{E}'_k(z)$. Therefore, on those frequencies aim to be cancelled, $|\underline{C}(z)|=1$ must be selected at those frequencies. On the contrary, if some frequencies aim to be inaltered, the maximum attenuation of $|\underline{C}(z)|$ at those frequencies must be introduced. The single-channel implementation of this algorithm will be exactly the same as the FxLMS approach, substituting in (2.18), the error signal $e(n)$ by $e'(n)$ and the filtered reference signal $\mathbf{x}_f(n)$ now must be filtered by $\underline{C}(z)$. Its multichannel extension is also immediate, as depicted in Figure 6.11. The difficulty of this structure lies on the design of $\underline{C}(z)$ to create a more complex desired equalization in the residual signal.

Therefore, the filter updating equation of the multichannel broadband MEFeLMS

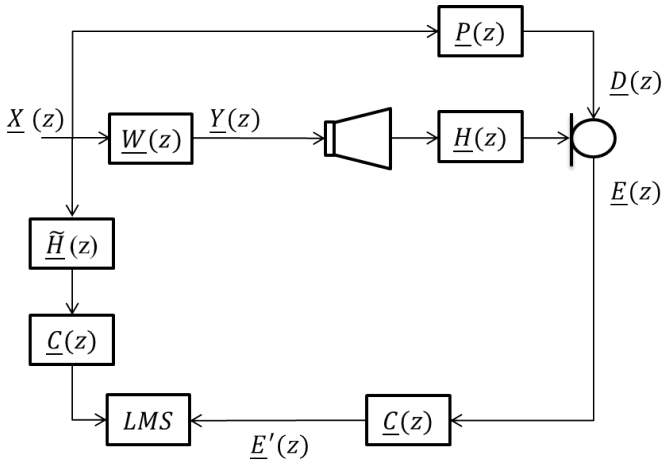


Figure 6.10: Block diagram of the single-channel broadband filtered-error LMS ANE algorithm.

algorithm is given by

$$\mathbf{w}_j(n) = \mathbf{w}_j(n-1) - 2\mu \sum_{k=1}^{N} \bar{\mathbf{x}}_{jk}(n)\bar{e}'_k(n). \tag{6.71}$$

where $\bar{\mathbf{x}}_{jk}(n)$ is the vector that contain the last L samples of $\bar{x}_{jk}(n)$ defined as $\bar{x}_{jk}(n)=\bar{s}_{jk}(n)*x(n)$ being $\bar{s}_{jk}(n)=h_{jk}(n)*c_k(n)$ and $\bar{e}'_k(n)=e_k(n)*c_k(n)$. As the previously case, the calculation of $\bar{s}_{jk}(n)$ can be done in a previously stage.

In this case, the distributed implementation of this algorithm can be also directly calculated as,

$$\mathbf{w}^k(n) = \mathbf{w}^{k-1}(n) - \mu\bar{\mathbf{x}}_{jk}(n)\bar{e}'_k(n). \tag{6.72}$$

In this case and unlike the broadband MEFxLMS-based case, note that the distributed version of the broadband MEFeLMS (broadband DMEFeLMS ANE) algorithm is an exact solution of its centralized version. A summary of the broadband DMEFeLMS ANE algorithm pseudocodes executed per sample time at each node is given in ***Algorithm 12***. To this end, we define $\bar{\mathbf{e}}_k(n)$ as a vector which contains the M_c last samples of $e_k(n)$. In addition, $\bar{\mathbf{x}}'(n)$ is defined as a vector which contains the last $M+M_c-1$ samples of $x(n)$ required to properly filter $x(n)$ by $\bar{\mathbf{s}}_{jk}(n)$ which is defined as a FIR filter of size $M+M_c-1$ that models $\bar{s}_{jk}(n)$.

6.2.3 Simulation results

In this section, some simulations were performed to validate the proposed distributed broadband ANE systems over a four-node ASN with a ring topology and an incre-

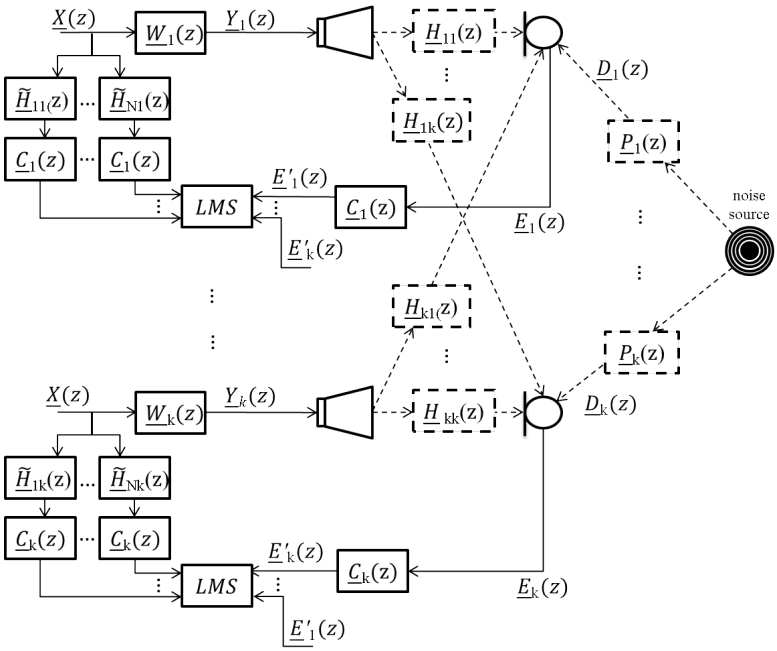
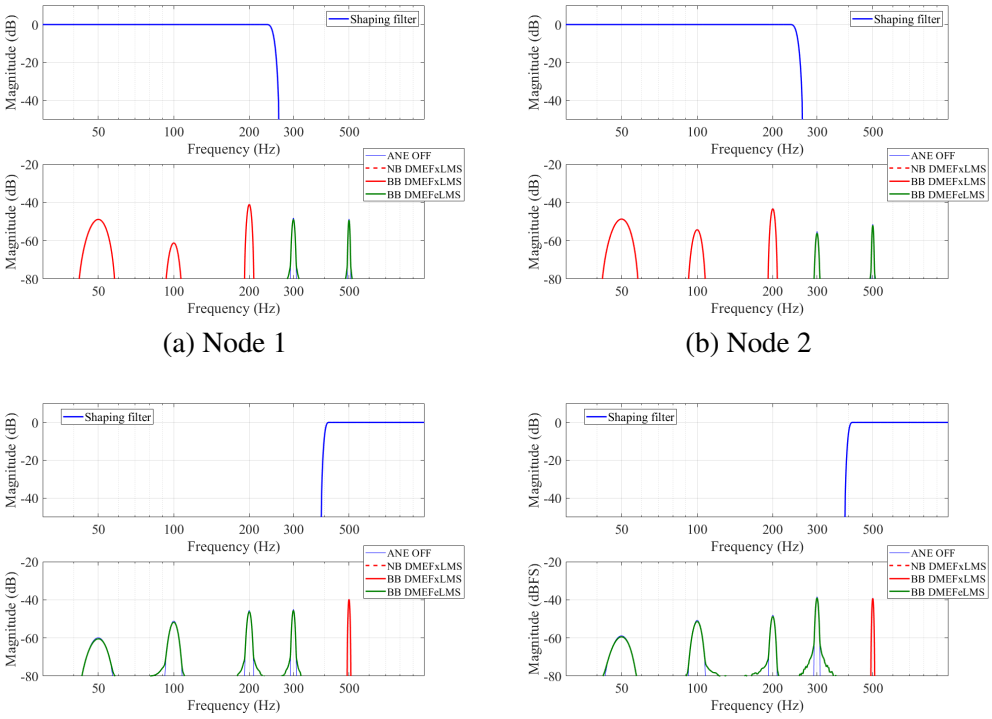


Figure 6.11: Block diagram of the multichannel broadband filtered-error LMS ANE algorithm.

Algorithm 12: *Broadband DMEFeLMS ANE.*

$\mathbf{w}^0(n) = \mathbf{w}^N(n-1)$ % Needed at node k=1
for all *node* $1 \leq k \leq N$ **do**
  $\mathbf{w}_k(n) = [\mathbf{w}^k(n-1)]_{(1+L(k-1):Lk)}$ % Obtain local coefficients.
  $y_k(n) = \mathbf{w}_k^T(n)[\mathbf{X}(n)]_{(:,1)}$ % Output signal
  $\bar{e}'_k(n) = \mathbf{c}_k^T(n)\bar{\mathbf{e}}_k(n)$ % Calculate the pseudo error signal
  for all $1 \leq j \leq N$ **do**
    $\bar{x}_{jk}(n) = \bar{\mathbf{s}}_{jk}^T\bar{\mathbf{x}}'(n)$ % Vector that contains reference signal filtered by auxiliary filters
    $\mathbf{w}_j^k(n) = \mathbf{w}_j^{k-1}(n) - 2\mu\bar{\mathbf{x}}_{jk}(n)\bar{e}'_k(n)$ % Update local state related to the j-th node.
  end for
end for
$\mathbf{w}(n) = \mathbf{w}^N(n)$ % Updated global state of the network
for all *node* $1 \leq k \leq (N-1)$ **do**
  $\mathbf{w}^k(n) = \mathbf{w}(n)$ % Disseminate global state of the network
end for

mental learning. In this regard, the power spectrum of the broadband DMEFeLMS ANE system is evaluated and compared with both the narrowband and the broadband DMEFxLMS ANE algorithms for different spectral shapes. The simulations have been carried out by using the real acoustic channels depicted in Section 3.3 and the four-node ASN depicted in 4.3 devoted to control a periodic noise with 50 Hz, 100



(a) Node 1 (b) Node 2

(c) Node 3 (d) Node 4

Figure 6.12: (top) Frequency response of the shaping filter and (bottom) frequency shape of the residual noise obtained at each node and applying the corresponding shaping filter. NB refers to narrowband ANE and BB refers to broadband ANE.

Hz,200 Hz, 300 Hz and 500 Hz components.

Regarding the equalization profiles, for the broadband ANE strategies, two different shaping filters, $\underline{C}_1(z)$ and $\underline{C}_2(z)$, are considered. $\underline{C}_1(z)$ and $\underline{C}_2(z)$ are designed as a lowpass and a highpass FIR filters of M_c=256 coefficients, respectively. Both fiilters consider a Hamming window with M_c+1 elements. $\underline{C}_1(z)$ uses a cutoff frequency of 250 Hz while $\underline{C}_2(z)$ considers 400 Hz as its cutoff frequency both using the same sampling rate as defined in Section 3.3 to measure the real acoustic channels, that is, f_s=2000 Hz. In addition, an adaptive filter length of L=512 coefficients is considered for the broadband strategies. Once the shaping filters have been designed, the parameter β_k required for the narrowband ANE strategy can be calculated as the value of the modulus of $\underline{C}_k(z)$ evaluated at the frequencies of interest (after converting dB to magnitude). In this regard, we obtain that β_1=β_2=[0.9 0.9 0.9 0 0] and β_3=β_4=[0 0 0 0 0.9]. Regarding the step-size parameter, it has been considered a value of μ=0.002 as the highest value that ensures the stability of the broadband

ANE systems while for the narrowband strategy, a step-size for each frequency and for each node has been calculated as obtained in (6.48).

In Figure 6.12(a).(top) and Figure 6.12(b).(top), the frequency response of $\underline{C}_1(z)$ is depicted. Similarly, the frequency response of $\underline{C}_2(z)$ is shown in Figure 6.12(c).(top) and Figure 6.12(d).(top). Figures 6.12(a).(bottom), 6.12(b).(bottom), 6.12(c).(bottom) and 6.12(d).(bottom) show the PSD of the resulting residual noised for the four nodes respectively. It can be observed that, in all nodes, both the distributed MEFxLMS-based narrowband and broadband ANE algorithms exhibit exactly the same behavior. By applying the lowpass filter at nodes 1 and 2, the higher frequencies, 300 Hz and 500 Hz, are eliminated. However, since a highpass filter have been used at nodes 3 and 4, all frequencies, except 500 Hz, are totally attenuated. On the other hand, the broadband DMEFeLMS ANE algorithm eliminates the frequencies that have been maintained with the other two strategies. This is due to the calculated auxiliary filters contain the inverse frequency response. For the nodes 1 and 2, the frequencies 300 Hz and 500 Hz are maintained intact while the lower frequencies, 50 Hz, 100 Hz and 200 Hz, are totally cancelled. On the contrary, at nodes 3 and 4, only the frequency at 500 Hz is eliminated.

6.3 Conclusions

With the aim to create independent-zone equalization profiles, in this section we have developed distributed ANE systems over networks with collaborative learning and composed of acoustic nodes. More specifically, several LMS-based narrowband and broadband algorithms have been presented

The proposed narrowband algorithm is based on the well-known MEFxLMS method and it has been designed to act over narrowband noises composed of multiple-periodic components. Therefore, a multichannel equalizer system for multifrequency signals has been implemented. Since personal audio control systems are usually acoustically coupled, we use a collaborative strategy among nodes to ensure the stability of the system. In addition, both the multiple-error and the common-error strategies have been also included. In the conventional multiple error method, a different pseudo-error signal for each frequency related to the same error sensor is used. However, in the common-error strategy, only one pseudo-error signal for each error sensor is considered decreasing the computational burden of the ANE system. Due to the advantages of the common error strategy, several simulations have been carried out considering only this common error approach over several distributed networks and for different spectral shapes. The obtained results have proved that the distributed narrowband MEFxLMS ANE algorithm allows us to equalize multifrequency noise signals with different equalization profiles at each control point in a distributed way.

In addition, two distributed broadband ANE algorithms, denoted as distributed

broadband MEFxLMS ANE and distributed broadband MEFeLMS ANE, have been implemented with the aim to shape the desired noise spectrum in the cases where the tonal components of a periodic noise can not be properly synthesized or when uncorrelated noise components appear. Moreover, broadband ANE systems can be deal with both tonal and broadband noises. However, the design of the shaping filters which modify the noise spectrum present some challenges when a more complex equalization is desired. Two different alternatives have been proposed based on the filtered-x and filtered-error structures. Simulations results show that, with a proper choice of the equalization profiles and equivalently proper design of the shaping filters, both the distributed MEFxLMS-based narrowband and broadband ANE algorithms exhibit exactly the same behavior obtaining the desired spectral shape at different control zones. On the other hand, the distributed MEFeLMS ANE algorithm can be especially useful when some part of the noise signal spectrum is aimed to be modified maintaining other parts intact.

Chapter 7

Practical implementation of SFC applications over distributed networks

Finally, some of the proposed distributed SFC algorithms that has been developed in this thesis, have been implemented in a real-time personal audio prototype by using collaborative acoustic nodes. Particularly, in this chapter the design and configuration of the prototype as well as its hardware and software integration to the system are described. It also includes the main constraining conditions related to the practical implementation of both ANC and narrowband ANE applications. In addition, the performance of the distributed algorithms in real scenarios have been evaluated by means of some experiments inside the listening room where the prototype is located.

7.1 Introduction

Practical implementations of noise control applications (either cancellation or equalization) are usually related with the use of active headrest systems. As commented in previously chapters, these systems are thought to be installed in public transportation as trains, airplanes, buses, etc., which means that various of these systems will be working simultaneously trying to create control zones at the passengers positions. As the passenger seats are close to each other, the overlapping between adjacent control zones can cause a significant performance degradation if no collaborative processing is carried out. In the worst case scenario, the interaction between PAC systems may lead to the unstability of the system. In those cases and as we have discussed previously, a collaborative strategy may be desirable to minimize the effects of the acoustic coupling at the same time that the system stability is ensured. A first approach to deal

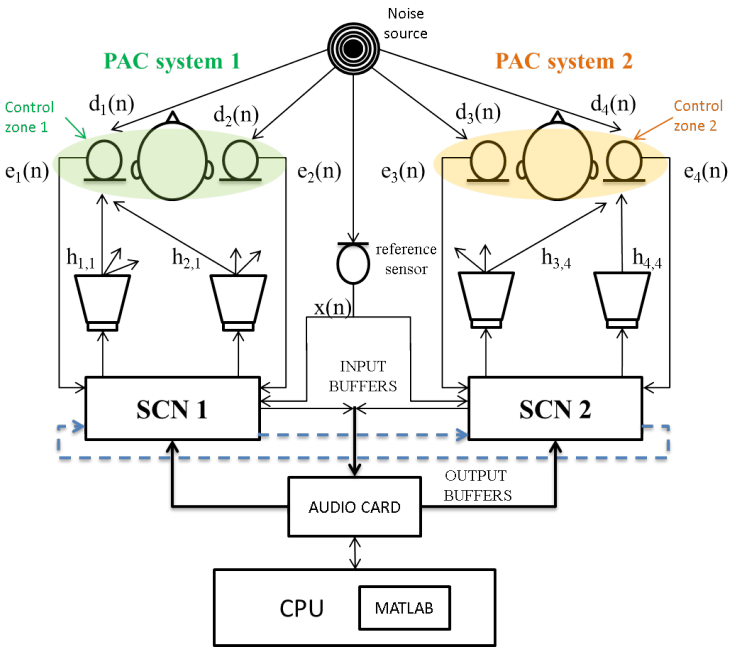


Figure 7.1: Two personal audio chair (PAC) systems composed of two-channel sound control nodes (SCN) each of them. Dashed blue lines represent the incremental communication strategy.

with this problem is to consider the set of active headrest systems as a whole multichannel active headrest system controlled by a central processor. These multichannel system should require a high computational capacity to capture, process and generate all the signals involved on the control process. An alternative approach is to consider each active headrest system as a single node of a distributed network where the computational cost is shared amongst them. In this regard, considering the definition of SNC introduced in Section 5.7, we extend the term of PAC system to a two-channel distributed SFC system composed of a car seat and a SCN. Note that the selection of the network topology will affect how data is processed by each SCN. The PAC systems use a feedforward control hence the undesired noise is monitored by a reference sensor. Stability can also be put in jeopardy due to other uncertainties related to sound field control [51]. As previously commented, for a personal sound control system, at least 11 dB between the different zones is required to provide adequate separation [87]. In [144], a real multichannel ANC application composed of one PAC system was experimentally tested achieving an attenuation of 10 dB around the headrest of a car seat. In addition, a similar real ANE system was presented in [77] achieving an independently control of multiple frequencies in the area analyzed. On

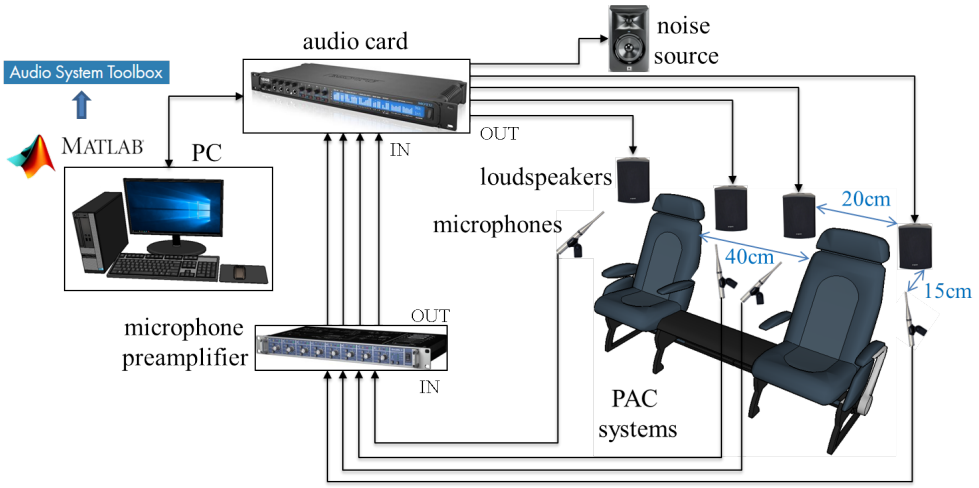


Figure 7.2: Scheme of the ANE prototype.

the other hand, not only objective measurements, but subjective analysis [91, 145] should be carried out to properly evaluate the performance of SFC applications.

Based on the previous concepts, this chapter reports a description of the design and implementation of a distributed SFC system to run both ANC and narrowband ANE applications. The aim of the PAC systems is to create independent control zones in the vicinity of the left and right ears of two listeners by controlling an external disturbance noise at the sensors' location (see Figure 7.1). About the proposed algorithmic approach, since an audio card to run the multichannel real-time audio application are required, we consider the incremental FPBFxLMS depicted in Section 5.5.2 for ANC applications and the block implementation of the distributed common-error ANE algorithm presented in Section 6.1 for narrowband ANE applications. Regarding this last algorithm, the block processing has been carried out based on the Block LMS algorithm presented in Section 2.3.4 but considering the superposition property of the multifrequency ANE algorithm.

7.2 Protoype description

The configuration of the distributed SFC application is depicted in Figure 7.2. Two PAC systems composed of a car seat and a SCN were considered. Each SCN is equipped with two actuators, two sensors and both SCN share the same proccesor (PC) which runs the distributed audio processing and simulates the communication model. The PAC systems have been mounted inside the listening room depicted in

Figure 7.3: Three different views of the distributed SFC prototype mounted in the listening room of the Audio Processing Laboratory of the Polytechnic University of Valencia.

Section 3.3, as shown in Figure 7.3. The SCN actuators consists of two hi-fi autoamplified loudspeakers APART model SDQ5P [146]. The election of these loudspeakers is based on a tradeoff between good low-frequency response, small size, enough sound power level and easy mounting. The loudspeakers were placed adjacent to the car seat in the rear of the headset, as shown in Figure 7.2 and with a separation of 40 cm between the loudspeakers of a same SNC. Two electrec condenser microphones per PAC were selected as error sensors. They are Behringer model ECM8000 [147] with omnidirectional pattern, specially suited for measurements applications. Each microphone was placed in front of each headrest, as close as possible to the listener ears (see Figure 7.3) in order to create the desired personal equalization zones. The microphones were separated 15 cm away from the loudspeakers and the distance between the SNC microphones was 20 cm. The disturbance noise to equalize by the SFC applicationS is emitted by an autoamplifier loudspeaker JBL model LSR305 [148]. Two KU 100 dummy heads manufactured by Neuman [149] have been used to monitor the impact of the system over a listener placed in the control area. It resembles the human head and has two high-quality microphone capsules built into the ears. The error sensors were separated 5 cm away from the microphone capsules of the dummy heads. The acoustical signals picked up by the microphones are sent to a RME OctaMic II [150] preamplifier to increase the microphone signal to the required line level of the next device of the audio chain, the audio card Motu 16 AVB [151]. The outputs of the audio card are connected to the loudpspeakers of

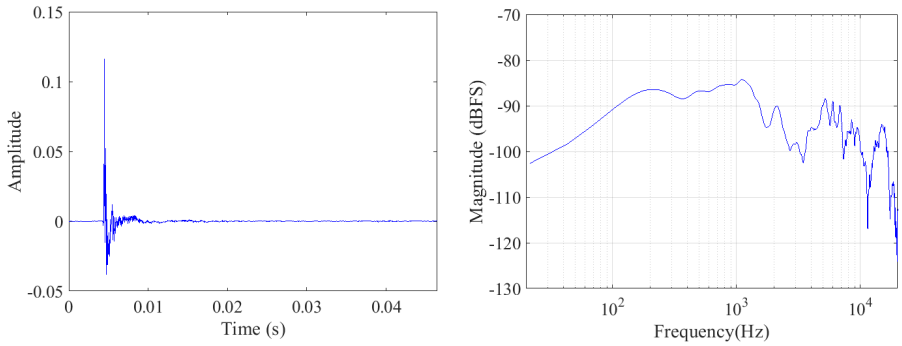


Figure 7.4: Acoustic impulse response (a) and the corresponding frequency response (b) of a representative loudspeaker-microphone pair.

the PAC systems.

With the aim to characterize the acoustic response of sound transducers, the impulse response of one loudspeaker-microphone pair is depicted in Figure 7.4.a. As is well-known, the impulse response of an acoustic system allows to obtain its frequency response curve, shown in Figure 7.4.b. Both responses can be considered as representative measures of the whole (two PACs) acoustic system.

7.3 Hardware and software integration

A diagram of the audio data flow can be seen in Figure 7.5. The audio card stores the input data from the sensor of each node in buffers and sends them to the CPU through the ASIO drivers. The CPU runs the audio processing algorithms in MATLAB®, saves the output data in buffers and sends them back to the audio card through the ASIO drivers, to be reproduced by the loudspeaker of each node. The addition of the recent Audio System Toolbox [152] in the computing environment MATLAB® provides multichannel real-time audio recording, processing and reproduction at low latency. Audio objects based on Object Oriented Programming (OOP) have been optimized for iterative computations that process large streams of audio data. Moreover, ASIO drivers [153] have been incorporated to this software providing a low-latency and high fidelity interface between MATLAB® and the audio card. The communication between the CPU (Intel Core i7 3.07 GHz) and the audio card is performed by ASIO drivers and it is controlled by using the MATLAB® System objects (*AudioDeviceWriter* and *AudioDeviceReader*) provided by the Audio System Toolbox of MATLAB®.

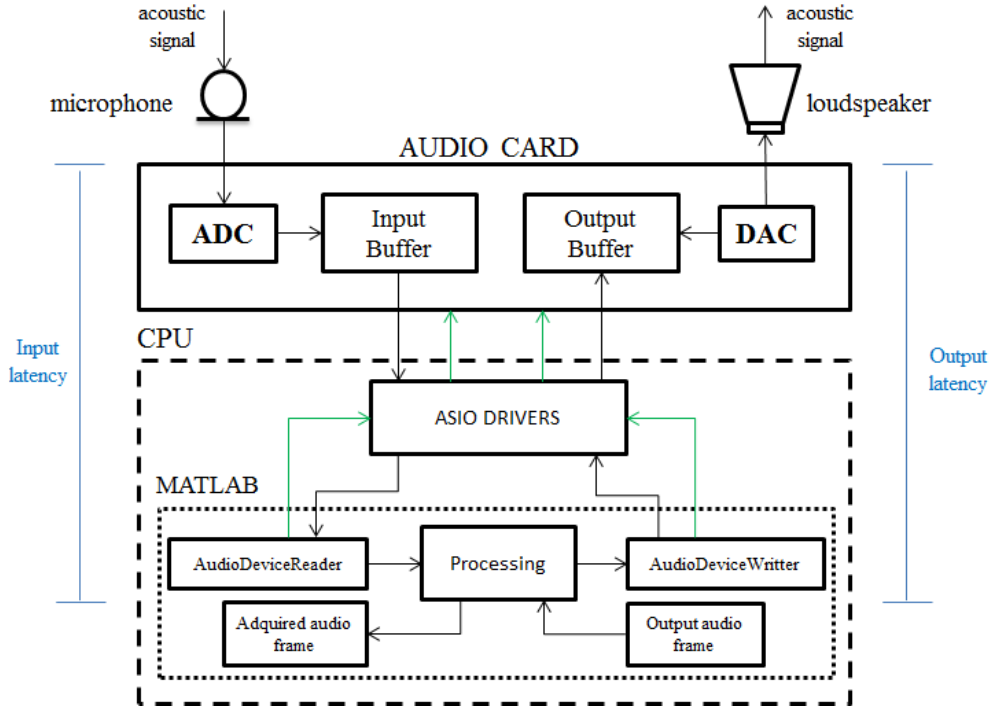


Figure 7.5: Audio data flow. The communication of the configuration parameters is represented by dashed green lines.

A web interface which involves real-time transfer of data from a web browser to Matlab® has been developed. An important benefit brought by such web interface is that it allows us to control independently each PAC system in real time. For example, for the ANC application, it enables to minimize the undesired noise at the right seat control zone while, at the left seat control zone, the noise is present and vice versa. The web interface has been built as an HTTP site with a simple user design (see ANC example in Figure 7.6.a) using HTML language. Node.js® [154] is a stable web serving platform running on Javascript® that offers solutions to building servers and web/mobile applications. By creating a TCP/IP object in Matlab®, it is possible to communicate with the web interface through Node.js® (see Figure 7.6.(b)).

Regarding the ANE application, the software interface provides the user an easy and intuitive tool to switch between different noise equalization profiles in real-time. In this way, each user can control a music player and its PAC profile separately, allowing to manage the listening experience. The users interface for each PAC is shown in Figure 7.7. It is divided in three blocks: the PAC control, the music player

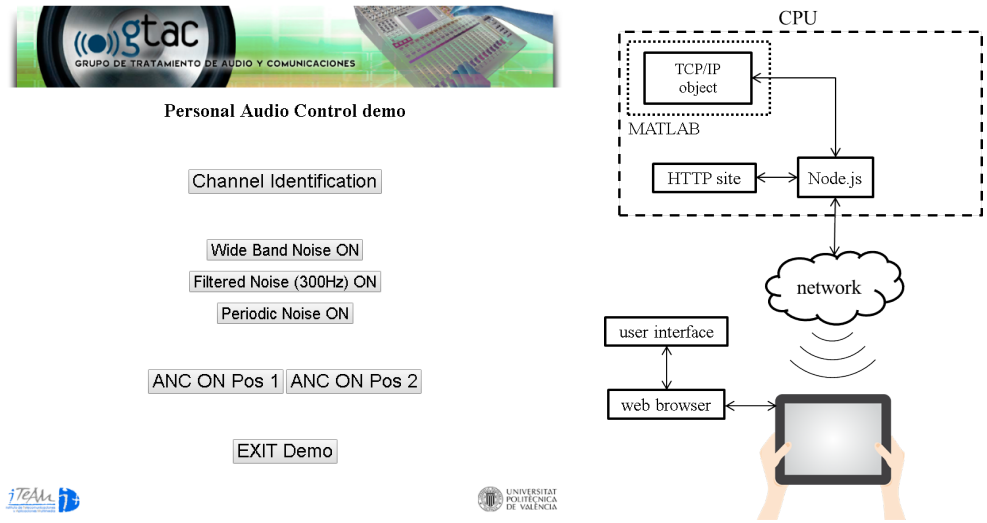


Figure 7.6: (a) ANC user interface design (b) and interface communication of the developed web interface.

and the visualizer. The music player allows to reproduce songs of different music styles as well as volume slider and some play, pause and stop buttons. Regarding the PAC control, it is based on a pop-up menu where the users can select their desired profiles to equalize the noise in real time. The equalization profiles are summarized as follows:

- **Full cancellation**. This profile cancels out or attenuates the noise signal as much as possible for the given setup, performing all available efforts to achieve maximum cancellation.
- **Hearing threshold (HT)**. Based on the threshold of human hearing curve [90], this profile should be equivalent to full cancellation in terms of perception, since the algorithm attenuates the noise until the Sound Pressure Level (SPL) reaches the minimum audibility curve (see Figure 7.8). It can be defined as the lowest sound level that a listener can hear.
- **A-Weighting HT**. Similar to the hearing threshold profile, the unique difference between them is an A-Weighting curve applied to the hearing threshold. Since this equalization profile, also shown in Figure 7.8, takes into account the relative loudness perceived by the human ear, it provides mainly lower cancellation at low frequencies, leaving some residual signal from the noise source.

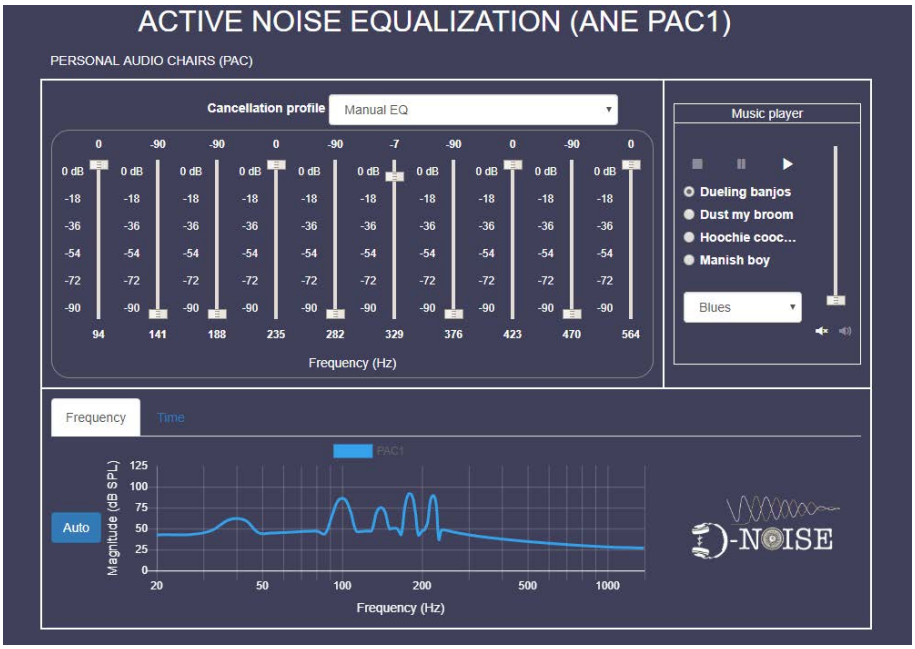


Figure 7.7: ANE user interface design.

- **Manual EQ**. This profile allows the user to make a personalized equalization profile, setting the desired attenuation per frequency. Only ten tonal components are handled due to the working range of the active equalization algorithm.

Both HT and A-Weighting HT curves are defined in the standard ISO 226:2003 concerning equal-loudness-level contours [155]. On the other hand, the visualizer plots a graph in real allowing to visualize both the frequency and time domain characteristics of the data stream captured by the error microphones at the selected ANE PAC. Figure 7.9 illustrates how different users manage and personalize their own active equalization sound profiles by using the developed software interface. In this particular setup, each user is given a tablet device, which is connected to the server via TCP/IP and over a wireless connection. As a result, the user-selected parameters are immediately sent to the PAC signal processing unit, where suitable actions are taken accordingly. Therefore, users can interact with the PAC system during the execution time. It should be noted that both the design of the ANE interface and its implementation within the prototype has been done jointly by several GTAC members as part of the Dnoise project presented in Section 1.1.

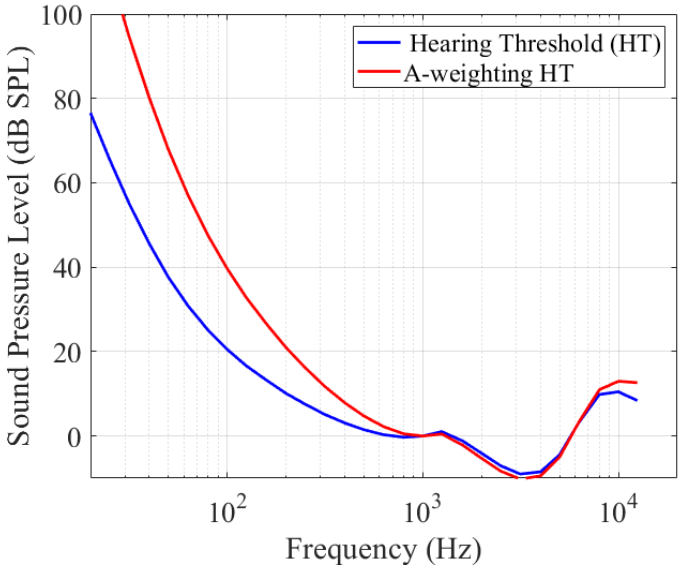


Figure 7.8: Hearing threshold profiles.

7.4 Implementation aspects

The selection of the frame size in the audio card is critical to determine the latency of the algorithm. The latency is the sum of the time spent on storing data in the input buffers, on processing these data and on sending them to the output buffers as well. A real-time application must satisfy that the time spent to fill up the input buffers (buffering time) is higher than the time spent in data processing (processing time). The buffering time is the ratio between the frame size (B) and the sampling rate (f_s) of the audio card, B/fs, with f_s fixed at 44.1 kHz as the lowest possible rate because of the limitations of hardware (though the sample rate could be lower due to the developed application). In a centralized system, the processing time is the time that the algorithm takes to process data. However, in a distributed system, it also includes the time in updating its own global state of the network and in delivering this information among the nodes. Since in distributed narrowband ANE only two coefficients (L=2) are exchanged by each node, we focus on the distributed ANC case. As we consider an incremental network composed of N nodes, every node must transfer $2L\times N$ coefficients (size of $\underline{\hat{\mathbf{W}}}_k[n]$ in (5.72)) to the following node $2(N-1)$ times in each block iteration (see Figure 5.2.(b)). Therefore, the processing time of the whole network at each block iteration (algorithm processing time $+N$ times the updating time of $\underline{\hat{\mathbf{W}}}_k[n]$+$2(N-1)$ times the transmission time of $\underline{\hat{\mathbf{W}}}_k[n]$) has to be less than the buffering time (see Figure 7.10). Considering a similar example as

Figure 7.9: Users while using the software interface.

presented in Section 5.2.3, a transfer rate at least of 16.5 MBps would be necessary with an incremental network composed of four nodes. Therefore, using a standard network of 1 Gbps ($\approx$ 125 MBps), we would have enough rate capacity to perform the required data transfer among the nodes. It should be noted that, if the processing time of the network increases due to the addition of more nodes, the buffering time must be increased in order to satisfy the real-time condition. Hence, if we assume a fixed sampling rate, the block size B must be increased.

Another important aspect that should be guaranteed is the causality of the ANC system. The algorithm has to satisfy that [156]

$$\tau_{sec}^{max} + \tau_{ref}^{max} \leq \tau_{pri}^{min} \tag{7.1}$$

where τ_{sec}^{max} is the sum of the buffering delay and the maximum delay of the acoustic paths that join the actuators with the error sensors, τ_{ref}^{max} is the sum of the buffering delay and the maximum delay of the acoustic path that links the reference sensor and the noise source and finally, τ_{pri}^{min} the minimum delay of the acoustic paths that join the noise source with the error sensors. Causality constraint can be relaxed when a harmonic excitation is considered, but it is important in feedforward broadband noise control. As we perform an ANC system in a acoustic enclosure, where the wave-

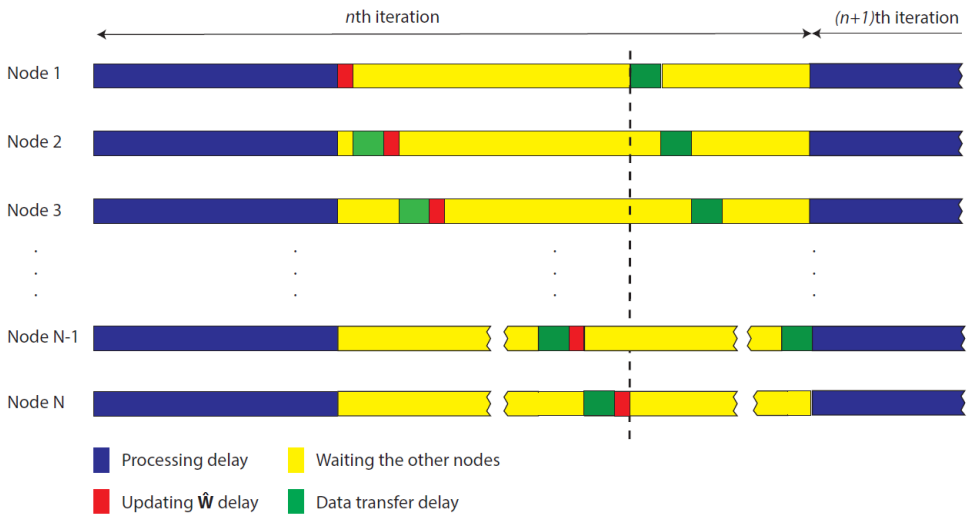


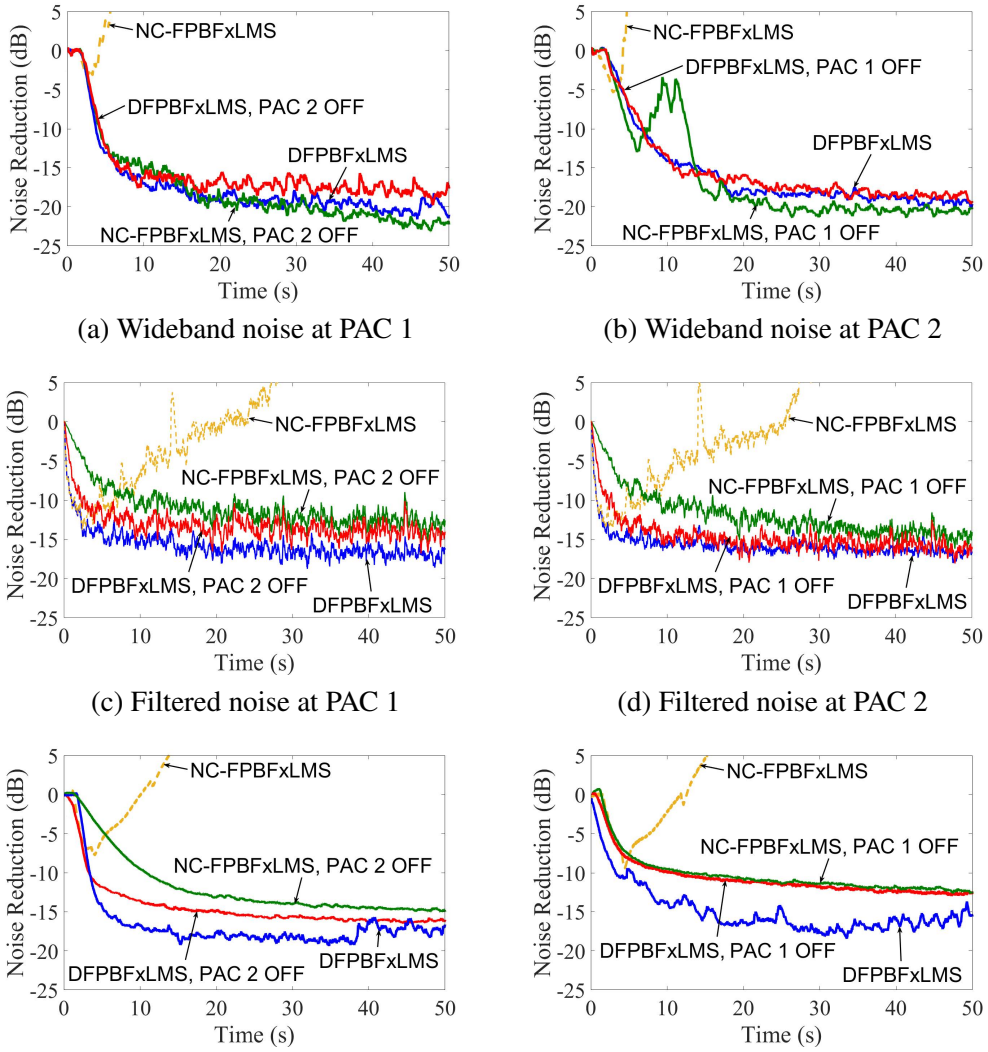
Figure 7.10: Timing diagram of the processes carried out by each node of the network at each block iteration.

length is relatively low in comparison with the physical dimensions of the system, the causality condition is fulfilled by carefully choosing the distances between the noise source, actuators and error sensors. However, if this condition is not met, a decrease in buffering time is required. When these requirements are fulfilled and assuming a network of synchronized nodes, the proposed distributed ANC algorithm can achieve the same performance as the centralized version.

Note that, although each PAC should carry its own processing unit as previously described, we have focused on the experimentation of both the acoustic and the processing models. Therefore, we have simulated the communications model between PACs using the same central processing unit (CPU), as it has been shown in 7.5. All the nodes share the same central processing unit (CPU) but the implemented code allows a distributed and independent processing, simulating the processing carried out in a real distributed ASNs. Therefore, a basic networking hardware dedicated to the information exchange (a physical layer of the network) is not considered. The communication among the nodes is virtual, thanks to the code designed in MATLAB®.

7.5 Experimental results

In this section, we show the experiments carried out to validate the performance of both the distributed ANC and the distributed narrowband ANE applications. To this end, the prototype described in previous sections has been considered. For both ap-



(a) Wideband noise at PAC 1

(b) Wideband noise at PAC 2

(c) Filtered noise at PAC 1

(d) Filtered noise at PAC 2

(e) Periodic noise at PAC 1

(f) Periodic noise at PAC 2

Figure 7.11: Noise reduction obtained by both the decentralized PAC systems and the distributed PAC systems for different disturbance noises.

plications, real acoustic responses between all the loudspeakers and all microphones are identified off-line using adaptive methods and modeled as FIR filters of $M{=}2048$ coefficients at a sampling rate of 44.1 kHz. The audio card offers different values of frame size between 8 and 1024 samples. Due to the real-time condition explained previously, we have selected 1024 as a tradeoff between the processing time and the buffering time.

Table 7.1: Values of G and β_k calculated at each frequency depending of the selected Hearing Threshold (HT) profile. Note that when is G=0 dB, instead of 1, a β_k=0.99 is considered.

HT profile	Parameter	40 Hz 140 Hz	60 Hz 160 Hz	80 Hz 180 Hz	100 Hz 200 Hz	120 Hz 220 Hz
HT	$G(dB)$	-27.3657	-41.9925	-47.8686	-52.3666	-56.2342
		-56.2443	-59.7463	-62.8100	-62.7507	-62.8129
	β_k	0.0428	0.0080	0.0040	0.0024	0.0015
		0.0015	0.0010	0.0007	0.0007	0.0007
A-weighting HT	$G(dB)$	0	-15.7696	-25.4710	-33.2217	-40.0445
		-40.0547	-46.5019	-51.9630	-51.9037	-51.9658
	β_k	0.9900	0.1627	0.0533	0.0218	0.0099
		0.0099	0.0047	0.0025	0.0025	0.0025

7.5.1 ANC application

Firstly, multiple measurements have been carried out by using different disturbance noises to evaluate the performance of several distributed ANC algorithms in terms of stability and convergence rate. We have considered three diferent types of synthesized noises as disturbance signals: 1) a wideband zero-mean Gaussian white noise with unit variance, 2) a zero-mean Gaussian white noise with unit variance limited to 200 Hz, and 3) a periodic noise of a fundamental tone of 20 Hz and a five harmonics, trying to simulate an engine noise. All the disturbance signals are emitted by a loudspeaker located in front of the PAC sytems. Furthermore, we have considered a FFT of the adaptive filter with a size of $2B$. A constant step size parameter of μ_1=$3\cdot10^{-5}$, μ_2=$6\cdot10^{-5}$ and μ_3=$10\cdot10^{-5}$ for the three type of noises respectively, as the highest value that ensures the stability of the algorithms, have been used.

Figure 7.11 illustrates the time evolution of the $NR(n)$ of the ANC system in dB for the non-collaborative DFPBFxLMS and incremental DFPBFxLMS algorithms using the different disturbance noises and considering the signals captured by the dummy head sensors. As expected, in a coupled environment, the non-collaborative algorithm makes the PAC systems unstable for the three type of noises. However, if both SCNs collaborate, the PAC systems achieve the stability showing a robust and stable performance for all the disturbance noises. In the cases where only one of the PAC systems is trying to cancel the noise while the other one is off, for all the cases, both collaborative and non-collaborative algorithms present similar results achieving a stable behavior. In summary, the performance of the coupled PAC systems improves when there is collaboration between the SNC's.

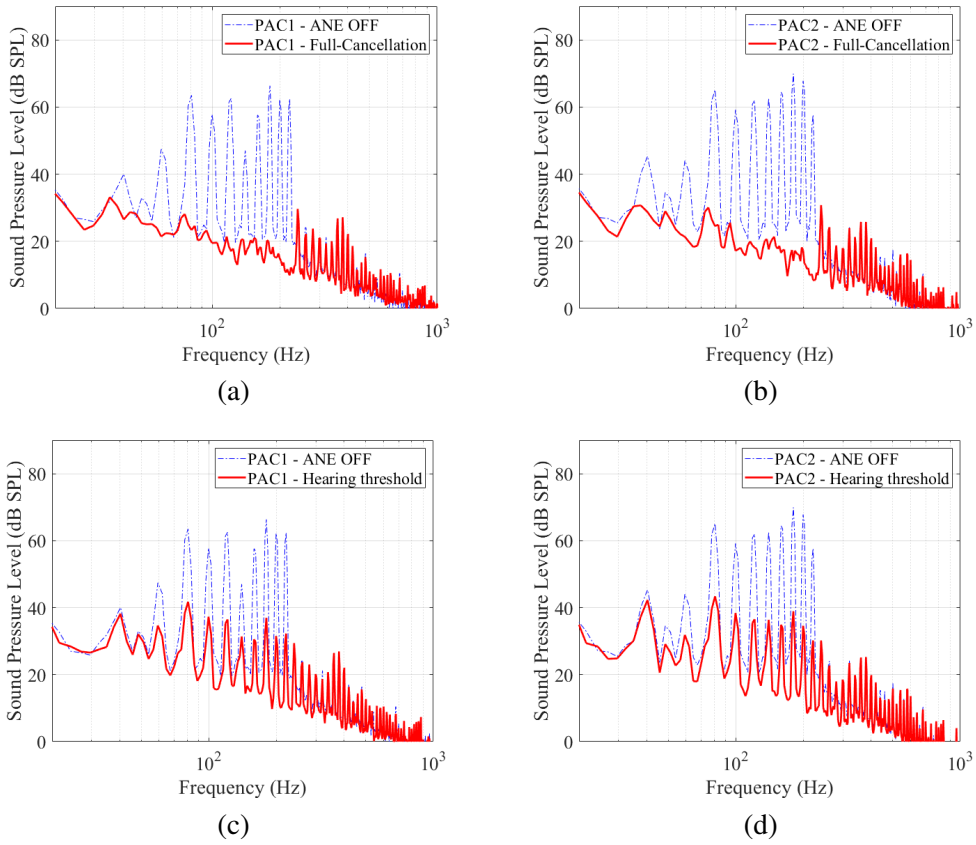


Figure 7.12: Sound pressure Level measured at (a)(c) PAC 1 and (b)(d) PAC 2 by using both the full cancellation and the hearing threshold profiles, respectively.

7.5.2 Narrowband ANE application

Performance evaluation of the block implementation of the distributed ANE algorithm for different equalization profiles is outlined below. As reference signal, a synthesized noise with ten tonal components at frequencies 40, 60, 80, 100, 120, 140, 160, 180, 200, 220 Hz has been used. Four different equalization profiles have been considered at both PACs: 1) full cancellation, 2) hearing threshold (HT), 3) A-weighting HT curve and 4) a specific profile where the ANE control at PAC 1 is off while full cancellation is applied at PAC 2. Note that these profiles are related to the value of the β parameter which is our target equalization profile. In our ANE system, we consider as many betas as frequencies for each node. Therefore, we consider for the equalization profile 1, β_k=0 for all frequencies and for all nodes. In the equaliza-

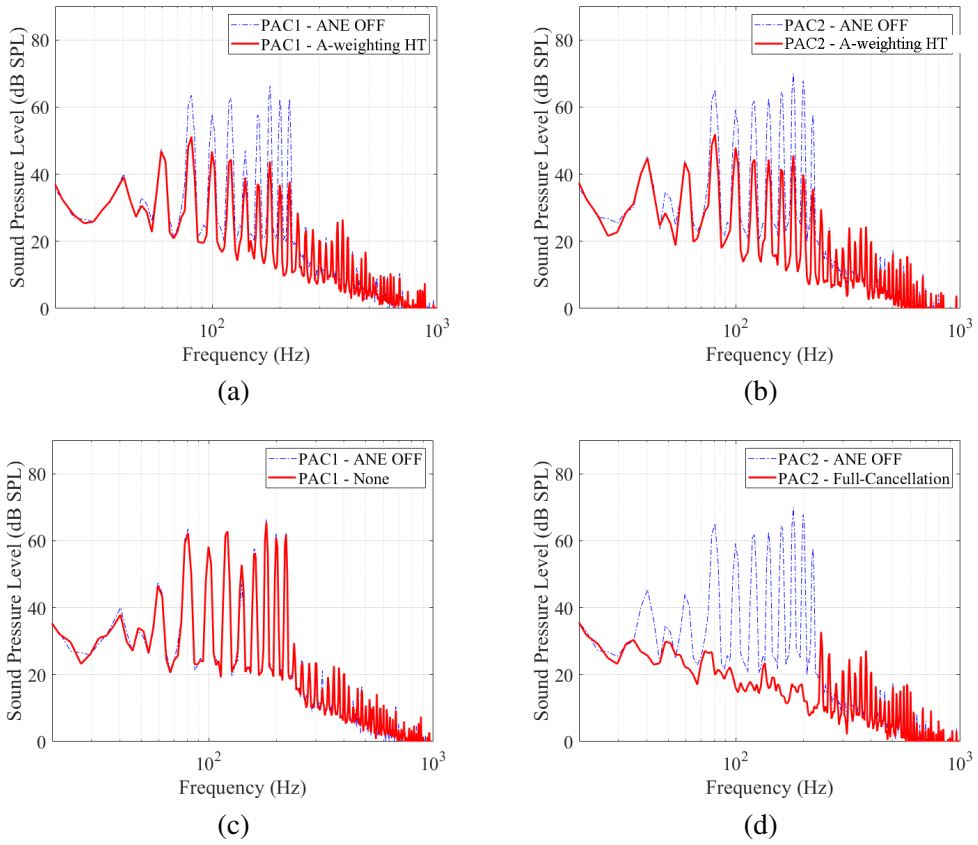
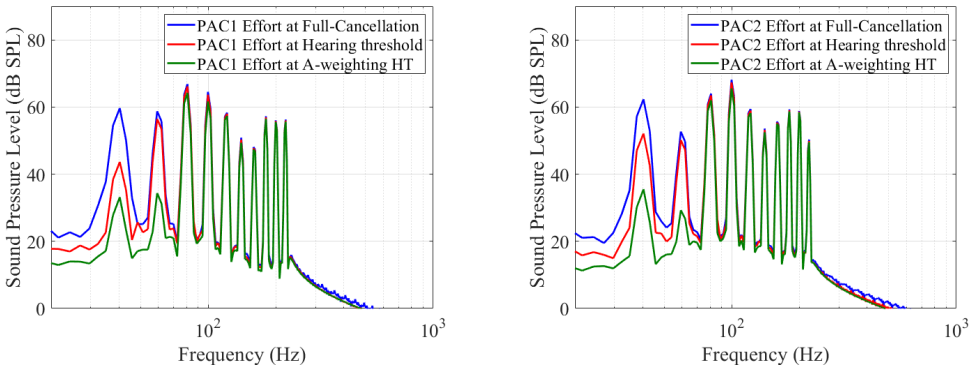


Figure 7.13: Sound pressure Level measured at (a)(c) PAC 1 and (b)(d) PAC 2 by using the A-weighting hearing threshold profile and one particular equalization (ANE off - Full cancellation) profile, respectively.

tion profile 4, β_1=β_2=0.95 and β_3=β_4=0 have been used for all frequencies. For the equalization profiles 2 and 3, the values of β_k depicted in Table 7.1 have been considered for all nodes. Note that these values of β_k have been calculated as $\boldsymbol{\beta_k}$=$10^{\frac{G}{20}}$ where G is the difference in dB between the SPL of the considered curve and the SPL of the noise evaluated both at the frequencies of interest. The SPL of the noise is measured in a previously stage. Since ANE cancels specific frequencies (the tonal components of the noise signal), $\boldsymbol{\beta}_k$ is calculated for the respective band that contains each frequency in order to obtain its attenuation. Regarding the step-size parameter and similarly to β_k, a value of μ_k may be defined for each frequency independently and calculated as depicted in (6.48) for each node.

The figures depicted in Figure 7.12 and Figure 7.13 show the SPL of the signals



(a) (b)

Figure 7.14: Sound pressure Level of the control signals power generated by the loudspeakers at (a) PAC 1 and (b) PAC 2 by using the full cancellation, the hearing threshold and the A-weighting hearing threshold profiles.

captured by the dummy head sensors with and without the ANE control system applying different equalization profiles. More specifically, the mean of the SPL values measured at the left sensor and the right sensor of the dummy head located close to the PAC 1 is considered. Idem for the dummy head at the PAC 2. In addition, we consider a particular time frame of the signals when convergence is achieved.

In Figures 7.12(a) and 7.12(b), the full cancellation profile is used at PAC 1 and PAC2, respectively. For both PACs, the block distributed ANE algorithm exhibits a good perfomance at the frequencies of interest providing an attenuation up to 40 dB on most frequencies. Same experiment is represented in Figures 7.12(c) and 7.12(d), but taking into account the hearing threshold profile in both PACs. Now, the performance of the algorithm at the PACs sensors is worst than the previous case providing an attenuation up to 20 dB at the same frequencies. However, from a perceptual point of view, no significant differences between equalization profiles 1 and 2 should be noted. Since human ear is less sensitive to low frequencies, as expected, the use of the A-weighting HT profile provides mainly lower attenuation at low frequencies, leaving some residual noise at 40 and 60 Hz, as shown in Figures 7.13(a) and 7.13(b). The robustness of the ANE system is demonstrated by applying the equalization profile 4, as it can be shown in 7.13(c) and 7.13(d). Note that the PAC 2 in 7.13(d) achieves the maximum noise reduction similar as obtained by the PAC 1 in 7.12(a) even if the ANE control in PAC 1 is not considered (see 7.13(c)). In this way, PAC systems collaborate allowing users to equalize the tonal noise as desired without interferences from the other PAC.

The array control effort (ACE) is defined as the energy that a loudspeaker array

requires to achieve the reproduced sound field in the control points [157]. In our case, we consider that the ACE is calculated as the power of the control signal $y(n)$ emitted by a loudspeaker. Figures 7.14 show the SPL values of the ACE measured at each PAC system by applying the equalization profiles 1,2 and 3. Particularly, these figures shown the mean of the contribution of the two loudspeakears at each PAC for a particular time frame when convergence is achieved. Note that when some PAC is inactive (ANE off), the ACE applied by the loudspeakers is null because there is no anti-noise signal to be generated. For that reason, that case is not included in Figures 7.14. It can be observed that at low frequencies (40 and 60 Hz) the HT profiles provides an ACE much lower than the obtained by using the full cancellation profile, reducing in almost 20 and 30 dB at each frequency when A-weighting HT is considered for both PAC systems.

Therefore, in some cases it may not be required to make efforts to try to cancel or equalize a tonal component of a noise because, considering how the human ear works, that frequency may not be perceived. Therefore, psychoacoustic analysis of the signals involved in the system can be also included in the ANE control with the aim to reduce ACE.

It is important to take into account that both the SPL values of the signals captured by the microphones and the SPL values of the signals generated by the loudspeakers have been calculated taking into account a previous calibration stage of the transducers.

7.6 Conclusions

In this chapter, a description of the design and implementation of a noise control application based on two distributed personal audio systems has been presented. Each system is composed of a car seat and a two-channel node equipped with two loudspeakers and two microphones. In this case, both nodes use the same CPU with Matlab to execute the audio processing and to simulate the communication model. These PAC systems have been mounted inside a sound-conditioned room. The aim of the application is to create a personal control zone close to the listener's ears by controlling an external disturbance noise. We want to ensure the acoustic comfort of the users allowing them to minimize the noise perceived as well as to equalize it according to a defined spectral shape. Depending of the type of noise (broadband or tonal noise), a different distributed application will be considered (ANC or narrowband ANE, respectively). In order to evaluate the performance of the distributed algorithms over the PAC systems, we have implemented a real-time adaptive noise controller based on the audio toolbox provided by Matlab®. It has been demostrated that the SFC applications suffer strong performance degradation when PAC systems are acoustically coupled. In those cases, a collaborative incremental strategy among

the SCNs has been used to ensure the stability of the systems. Experimental results show that, in acoustically coupled enviroments, the collaborative implementation avoids the unstability of the fully decentralized strategy achieving the convergence of the PAC systems. In addition, the listeners can use a web interface to control independently its control zone in real time. In the case of a multi-component tonal noise, this interface also allows users to select among several equalization presets or to adjust manually each frequency band without interfering with other PAC systems. In addition, equalization profiles based on hearing threshold curves present the advantage of saving effort while keeping the perceptual loudness of maximum cancellation profile, avoiding distortions and fatigue of the loudspeakers at low frequencies. However, subjective tests must be required to validate some assumptions detailed here.

Chapter 8

Conclusions

This chapter summarises the findings and contributions of this research work, revisiting the initial objectives given in the introductory chapter. At the beginning, the first section reviews the contents of this study, outlining the main conclusions that were extracted from each chapter. Next section presents different recommendations for future research. Aditionally, the final part contains a list of works published during the course of candidature for the PhD degree.

8.1 Main contributions

Novel audio signal processing applications often employ powerful centralized processing systems to control larger number of audio channels in order to improve audio quality or to create personal sound zones. Distributed signal processing techniques carried out in acoustic sensor networks provide a more flexible, more versatile and easier scalable solution in cases where the high computational demands of a centralized fashion become unapproachable. However, no study to date has made use of distributed sensor networks for multichannel sound field control applications. This thesis focuses mainly on the development of distributed control algorithms to implement multichannel SFC systems over networks composed of acoustic nodes. Concretely, we focus on the implementation of ANC and ANE applications considering LMS-based distributed algorithms devoted to the cancellation or the equalization of disturbances noises, respectively.

However, the performance of these applications may be highly influenced by the acoustic environment where the nodes are located. For this reason, the first contribution of this thesis lies in the attempt of determining the influence of the acoustic system on the behaviour of the ANC application. More specifically, Chapter 3 analyzed the acoustical coupling effects on control system stability over distributed networks

composed of acoustic nodes. In this regard, several experimental methods based on both the analysis of the acoustic system stability and the extraction of parameters derived from the study of the acoustic paths matrix, have been proposed in Appendix A. When collaborative strategies are considered, the problem of selecting which nodes are devoted to exchange information in acoustically coupled networks have been also analyzed in this Appendix. On the other hand, a generic formulation for the ANC problem over distributed networks is proposed in Chapter 3 with the aim to derive all distributed (non-collaborative and collaborative) ANC strategies.

In the cases when acoustic enviroment negatively affect to the control system stability, several non-collaborative and collaborative distributed techniques have been proposed throughout this thesis with the aim to ensure the convergence of the ANC systems. Chapter 4 presented several strategies of the decentralized ANC algorithm for distributed ASNs in which non-collaborative nodes aimed at estimating their local ANC solution in acoustically coupled environments. Initially, the stability conditions of an ANC system composed of non-collaborative acoustic nodes have been analysed. The behaviour of the distributed non-collaborative ANC algorithm in an isolated node may suffers degradation in terms of noise reduction if the same node executing the same algorithm is located in an acoustically coupled network. With the aim to estimate that degradation due to the acoustical coupling, an interference model has been also proposed. In addition, a strategy based on minimizing the effect of acoustic interference between nodes has been proposed and denoted as icNC-DMEFxLMS algorithm. The results show that the proposed icNC-DMEFxLMS algorithm ensures the distributed ANC stability in both uncoupled and coupled ASNs at the expense of degrading the system performance.

However, it is possible to stabilize the ANC system as well as to reach results equivalent to those of the centralized method with the use of distributed ANC system over a network of collaborative nodes. Collaboration between nodes implies that each node can share part of its information with the rest of nodes. This allows the local state calculated for each node to influence the calculation of the global state of the network, thus reducing the negative effects of non collaborative networks. Chapter 5 focused on the implementation of an ANC system over a network of distributed nodes acoustically coupled. To this end, several collaborative distributed algorithms executed in each node of the network considering possible practical constraints in real scenarios have been designed as follows:

- As the basis of the proposed strategies, we have considered the DMEFxLMS algorithm that allows collaboration between nodes following an incremental strategy.
- To avoid that control signals may increase unlimitedly, a control effort weighting has been introduced in the cost function of the distributed algorithm in each

acoustic node of a collaborative network. Thus, the stability of the distributed ANC system is ensured by achieving the same behaviour as its centralized version. In addition, different approaches of the proposed algorithm have also been presented for different scenarios showing their versatility. Simulation results showed that the proposed 1r re-scaling l-DMEFxLMS algorithm achieves a good control effort with a computational cost similar to the other methods but reducing significantly the communication demands of the ASN.

- In order to improve the convergence behaviour in transient state of the DMEFxLMS algorithm, the DFxAPL-I algorithm has been implemented along with an incremental collaborative network strategy. The results show that the proposed distributed algorithm, in addition to being scalable and versatile, provides higher convergence speed than the DMEFxLMS algorithm as well as the same behavior as its centralized version.

- With the aim to implement efficient real-time applications and due to the inherent block-operations of audio card, two collaborative distributed implementations of the FPBFxLMS algorithm for ANC systems have been presented considering incremental and diffusion learning, respectively. It has been demonstrated that implementations aspects such as the block size, the number of nodes and the network data transfer rate have to be considered in real ASNs. On the other hand, the diffusion learning allows each node to update the global state of the network through its local information and in turn, assuming some collaboration with its neighboring nodes. Thus, the diffusion FPBFxLMS algorithm obtains a good behavior in coupled networks since the information of the network state calculated in each node is diffused through all the nodes.

- To reduce the computational network requirements, an alternative strategy which brings together only the acoustically coupled nodes has been used and denoted as cl-DMEFxLMS algorithm. The rule to define the subsets of nodes is based on the collabarative methodology presented in Appendix A. Simulation results show that, if clustering is properly selected, the proposed strategy exhibits similar performance as the DMEFxLMS algorithm in terms of noise reduction and, depending on the nodes location, it may reduce significantly the computational and communication burden of the network.

- Finally, the DMEFxLMS algorithm considering the remote microphone technique (RM-DMEFxLMS) has been proposed with the aim to avoid the uncomfortable location of error microphones. Considering that the distributed RM technique achieves a similar performance at the virtual microphes as the DMEFxLMS algorithm at the physical error sensors, its use in distributed ANC headrest systems is justified. In addition, two variants called 1r-RM-DMEFxLMS algorithm and pnc-RM-DMEFxLMS algorithm have been also

proposed to reduce the computational and communication demands of the RM-DMEFxLMS algorithm in ASNs.

Chapter 6 considered a more general case in which the distributed SFC system is not only able to cancel a broadband noise but also to equalize both multi-tonal and broadband noises. To this end, both the narrowband and broadband DMEFxLMS ANE algorithms have been proposed to execute multichannel ANE applications over incremental ASNs. Both systems ensure acoustic comfort by creating independent equalization profiles for multiple users located in different areas inside an enclosure. In this way, each user can equalize a perceived multi-tonal noise according to a defined spectral profile. In addition, a distributed ANE algorithm by using the FeLMS structure is also proposed. This strategy, called broadband DMEFeLMS ANE algorithm, which can be especially useful when some part of the noise signal spectrum is aimed to be modified maintaining other parts intact.

Once the proposed distributed algorithms for SFC applications have been analysed, a real-time prototype that allows to execute both ANC and ANE applications using collaborative acoustic nodes has been developed in Chapter 7. The prototype is composed of two PAC systems, where each PAC consists of a car seat and a SCN which is equipped with two loudspeakers and two microphones. Both nodes use the same CPU with MATLAB® to execute the distributed audio processing and to simulate the communication model. Thanks to the collaboration among nodes, it is possible to minimize the acoustical interferences between both systems. Thus, for the ANC case, it is possible to create two quiet zones where the noise level is considerably reduced. In the case of the ANE application, two independent equalization zones can be created where each user can control the residual noise level according to several equalization presets. In this way, the acoustic comfort of the user is improved without the use of headphones. In addition, through the development of a web application, each user can independently control the configuration of each application (ANC or ANE) through a personal and independent interface.

Finally, in Appendix B an acoustic simulator has been implemented in MATLAB® to develop sound field control applications over acoustic sensor networks formed by acoustic nodes. This simulator has provided the framework for the implementation of any algorithm both centralized and distributed, both non-collaborative and collaborative, both in the time and the frequency domain, and both working sample-by-sample and stored in block of samples. To this end, the tool allows us to create nodes with an arbitrary number of microphones and loudspeakers as well as to execute the acoustic system sample-by-sample independently. In addition, it allows for modelling of both the behaviour of the communications system simulating the possible problems and the data acquisition performed in real time applications. In summary, this tool is intended to fill the gap between the initial mathematical idea of an algorithm and its final programming on a digital platform.

Therefore, the behavior of the proposed distributed algorithms that have been studied throughout the thesis have been validated through the results obtained in both the simulation tool and the real protoype.

8.2 Further work

The beginning of a line of research opens up different fronts that are impossible to address in a single PhD dissertation. This work is no exception, and along with the material presented in this thesis, there are other aspects which could serve as interesting directions for future research.

- Chapter 3 analyzed the stability of the acoustic system in order to identify the conditions under which nodes must collaborate in a distributed network in order to avoid the ANC system instability. In addition, Appendix A analyzed several methods to determine when acoustical interaction will negatively affect on the control system stability from a practical point of view. However, future investigations are necessary to theoretical validate the kinds of conclusions that can be drawn from both sections. In this way, a theoretical analysis about how changes in the entries of the collaboration matrix $\mathbf{A}$ affect on the system convergence is a challenge which must be investigated. Recents works [99, 100] may help to try to solve the optimization problem. From the results obtained in Appendix A, it seems that it may exist a relationship between frequency bands where broadband noise signals are stable and single frequencies that make the system stable using tonal noise signals. Idem for unstable cases. To this end and as a suggestion for future works, a distributed ANC system based on sub-band adaptive filtering structure where nodes exchange information only in acoustically coupled sub-bands (for broadband noises) or in specific frequencies (for tonal noises) should be considered. It must be also desirable to quantify how much the behavior of the system is degraded or improved by working in a non-collaborative way compared to the collaborative way. Moreover, given a particular acoustic environment and considering a specific SFC application, to develop techniques which allow to find the optimal node placement based on the level of acoustic coupling among the actuators and sensors of each node, may be desirable for future work.

- In Chapter 4 and Chapter 5, several distributed techniques to solve the ANC problem from both collaborative and non-collaborative points of view have been proposed. In future studies, it would be interesting to develop hybrid distirubted ANC algorithms capable of adapting their performance to both non-collaborative and collaborative distributed networks. Thus, it would be possible to switch between the two types of collaboration depending on whether

ASNs are acoustically coupled or not. Note that the idea of using heterogeneous networks is implicit in this proposal, since these networks are formed by acoustic nodes with different characteristics to those used in this thesis. Furthermore, it would be very relevant to explore the possibility of analysing the behavior of the proposed distributed algorithms over constrained networks (with special interest in asynchronous networks), e.g., nodes using different reference signals, synthesized tonal signals with different initial phases (for narrowband ANE systems), data loss or the inability to complete a communications cycle in sample time, among others.

- In Chapter 5 of this thesis, a distributed implementation of the FPBFxLMS algorithm considering a diffusion learning has been developed. Initially, a static combination of neighboring estimates through the use of a constant, α, has been considered. However, in some cases, a node may achieve better performance than its neighboring nodes. Therefore, the use of adaptive combination coefficients may allow the system to respond to the real time conditions in practical ASNs [158]. Moreover, the use of a specific α for each node could improve the system performance. Aditionally, it will be important that future research investigate the way of relating the parameter α and the level of acoustic coupling between nodes. In this regard, it has been demostrated that the performance of the control algorithm at each node of an ASN may vary depending on the relative physical location between the transducers used at each node. Therefore, an analysis of the influence of the transducers location over the behaviour of the proposed distributed algorithms in real enviroments is suggested for future works.

- Narrowband and broadband DMEFxLMS ANE approaches have been proposed in Chapter 6 for ASNs with the aim to create independent-zone equalization profiles. However, a subjective evaluation method to validate the performance of the ANE application should be considered in future works. It may be also included a psychoacoustic equalizer based on knowing how human ear works with the aim to reduce both the interferences outside the ANE control zone and loudspeakers' effort [159]. In addition, a robust ANE application should include a tracking system that allows the readjustment of the algorithm when some frequency of the synthesized reference signal does not match with the frequency of the noise to be controlled. Regarding the broadband approaches, optimised and automatic shaping filter design may improve the funcionality of these systems.

- In the real-time prototype designed in Chapter 7, a single central processing unit (CPU) performs the audio distributed processing as well as simulates the communications between PACs. However, each PAC should carry its own processing unit to individually capture, process and generate all the signals in-

volved on the control system. Therefore, future research efforts should be focused on the election of a suitable platform which allows a fully real-time implementation of SFC systems. Recently the emergence of Single-Board Computers (SBC) is enabling the implementation of many types of audio processing applications due to the increasing power and availability at low cost obtaining good results. These platforms are usually equipped with audio inputs, audio outputs, network cards that allow wireless communication and increasingly powerful processors. It is even possible to use, with certain graphics cards, parallel computing to increase the capacity of processing. In addition, these devices are small size and low power, allowing greater autonomy. For these reasons, this type of device may be a clear candidate to be used as the processing unit at each SCN. The election of the suitable platform involves tradeoffs between processing capacity, latency, communication capability, power consumption, ease of programming and price among others. Therefore, it will be important to further analyze the performance of the current embebbed plataforms used in audio processing, such as Raspberry Pi 3 (or its expected version 4) [160], BeagleRT [161] and BeagleBoard-X15 [162] among others, with the aim to validate if the real-time SFC system requirements are fulfilled.

8.3 List of publications

International Journal Papers

- **C. Antoñanzas**, M. Ferrer, M. de Diego, A. Gonzalez, "Control Effort Strategies for Acoustically Coupled Distributed Acoustic Nodes ". *Wireless Communications and Mobile Computing*, vol. 2017, pp. 1-15, 2017.
- **C. Antoñanzas**, M. Ferrer, M. de Diego, A. Gonzalez, "Blockwise Frequency Domain Active Noise Controller Over Distributed Networks ". *Applied Sciences. Special Issue about Audio Signal Processing*, vol. 6, no. 124, 2016.

Journal Papers in Preparation

- **C. Antoñanzas**, M. Ferrer, M. de Diego, A. Gonzalez, "Distributed active noise equalizer with different spectral profiles ".
- **C. Antoñanzas**, M. Ferrer, M. de Diego, A. Gonzalez, "Analysis of acoustically coupled decentralized networks on active noise control ".
- **C. Antoñanzas**, M. Ferrer, M. de Diego, A. Gonzalez, "Analysis of acoustical interaction effects on active noise control systems over distributed networks ".

- **C. Antoñanzas**, M. Ferrer, M. de Diego, A. Gonzalez, "Remote microphone technique for active noise control over distributed networks ".
- **C. Antoñanzas**, M. Ferrer, M. de Diego, A. Gonzalez, "Clustered active noise control system over distributed networks based on node-specific techniques ".

Peer-reviewed non-ISI Journals

- **C. Antoñanzas**, M. Ferrer, M. de Diego, A. Gonzalez, "A complete simulation tool for sound control applications over wireless acoustic sensor networks ". *Waves*, vol. 7, pp. 47-56, 2015.

International Journal Conferences

- **C. Antoñanzas**, M. Ferrer, M. de Diego, A. Gonzalez, "Personal active control systems over distributed networks ". *Proceedings of the 24nd International Congress on Sound and Vibration (ICSV24)*, London, UK, July 2017.
- **C. Antoñanzas**, M. Ferrer, M. de Diego, A. Gonzalez, "Collaborative method based on the acoustical interaction effects on active noise control systems over distributed networks ". *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal processing (ICASSP)*, New Orleans, USA, March 2017.
- **C. Antoñanzas**, M. Ferrer, M. de Diego, A. Gonzalez, "Affine-projection-like algorithm for active noise control over distributed networks ". *Proceedings of the IEEE 9th Sensor Array and Multichannel Signal Processing Workshop (SAM 2016)*, Rio de Janeiro, Brazil, July 2016.
- **C. Antoñanzas**, M. Ferrer, M. de Diego, A. Gonzalez, G. Piñero, "Diffusion algorithm for active noise control in distributed networks ". *Proceedings of the 24nd International Congress on Sound and Vibration (ICSV22)*, Firenze, Italy, July 2015.
- J. Lorente, **C. Antoñanzas**, M. Ferrer, A. Gonzalez, "Block-based distributed adaptive filter for active noise control in a collaborative network ". *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Nice, France, September 2015.

National Journal Conferences

- **C. Antoñanzas**, J. Estreder, M. Ferrer, A. Gonzalez, "Simulador de redes de sensores acústicos para aplicaciones de control de campo sonoro. ". *Proceed-*

ings of the 46th Congreso Español de Acústica (SEA-TECNIACUSTICA 2015), Valencia, Spain, October 2015.

- J. Estreder, **C. Antoñanzas**, G. Piñero, M. de Diego, "Gestor de aplicaciones de procesado de sonido sobre Android para la creación de redes de nodos acústicos con dispositivos comerciales ". *Proceedings of the 46th Congreso Español de Acústica (SEA-TECNIACUSTICA 2015)*, Valencia, Spain, October 2015.

Appendix A

Experimental methods to obtain the collaboration matrix

In this appendix, several practical methods are used to design their version of the collaboration matrix $\mathbf{A}$ according to their own criteria for measuring the level of acoustic coupling between nodes. Depending on the selected method, a certain condition related to the acoustic coupling will be provided and consequently, depending on that condition, the binary decision between to collaborate or not collaborate will be made. Initially, we consider that $\mathbf{A}$ is a identity matrix and then, each method will introduce ones at the corresponding entries of the matrix if necessary.

Before starting, it should be noted that it is possible to include the matrix $\mathbf{A}$ within the algorithm formulation. In this way, the global filter updating equation of the centralized MEFxLMS algorithm including a collaboration matrix, and denoted as cm-MEFxLMS algorithm, is given by

$$\mathbf{w}(n) = \mathbf{w}(n-1) \ - 2\mu \sum_{k=1}^{N} \tilde{\mathbf{u}}_k(n) e_k(n). \tag{A.1}$$

where

$$\tilde{\mathbf{u}}_k(n) = [\alpha_{1k}\mathbf{h}_{1k}^T\mathbf{X}^T(n) \quad \alpha_{2k}\mathbf{h}_{2k}^T\mathbf{X}^T(n) \ \ldots \ \alpha_{Nk}\mathbf{h}_{Nk}^T\mathbf{X}^T(n)]^T. \tag{A.2}$$

A more detailed description of this algorithm is presented in Section 5.6 of Chapter 5.

A.1 Methods to obtain the collaboration matrix

As introduced above, it is possible to obtain the theoretical collaboration matrices by previously analyzing the acoustic system derived from nodes location. To this

end, we propose several methods to shape the collaboration matrix based on both the previous analysis of the acoustic system stability and the extraction of parameters derived from the study of the acoustic paths matrix, respectively. At each method, we explain both how the acoustic parameter and the criterion for the design of the collaboration matrix are defined.

Methods based on the convergence analysis provides a stability condition based on the study of the eigenvalue distribution of a matrix related to the acoustic system. On the other hand, the methods based on acoustic system parameters, analyzes the degree of acoustic coupling between the nodes of a network based on its own criteria usually related to the measurement of an acoustic parameter. As a result of this analysis, both type of methods modify the entries of matrix $\mathbf{A}$ to design its version of the collaboration matrix that theoretically does not allow the acoustic system to make the ANC system unstable. As previously commented, matrix $\mathbf{A}$ is initially considered as a as a identity matrix where all elements under/above the main diagonal determine the cooperation relationship between nodes. The term cooperation or collaboration refers to the fact that in order to calculate its coefficients, a node can interchange information with other nodes of the network.

For the sake of clarity, the relation between the stability condition and collaboration among nodes is specified as

- If the stability condition between node j and node k is fullfilled, nodes j and k may work independently.
- If the stability condition between node j and node k is not fullfilled, node j must collaborate with node k.

Similarly, the criteria for designing the collaboration matrix $\mathbf{A}$ is given by:

- If node j collaborates with node k, α_{jk}=1.
- If node k collaborates with node j, α_{kj}=1.
- And finally, if both collaborate with each other, α_{jk}=α_{kj}=1.

A.2 Methods based on the stability analysis

Through the stability condition obtained from analyzing the eigenvalue distribution of a matrix related to the acoustic channels, it is possible to derive the desired collaboration matrix. However, depending on the nature of the disturbance signal (tonal or broadband noise), the convergence characteristics may vary. For broadband noise signals, a stability condition based on the analysis of the eigenvalue distribution of an

autocorrelation matrix composed of the filtered reference signals is considered. However, for tonal noise signals, we consider a stability condition based on the eigenvalue of the acoustic transfer matrix evaluated at the tonal frequency. As an exception, Method 2 uses a tonal noise as reference signal but considering the autocorrelation matrices.

The steps for designing the proposed collaboration matrix by the use of these methods are summarized as follows. As previously commented, the convergence characteristics of the cm-MEFxLMS algorithm considering $\mathbf{A}$ as a identity matrix is initially considered. If the result of the stability condition states that $\mathbf{A}$ makes the ANC system stable, all nodes of the N-node network may work independently. In the case of the stability condition is not fulfilled, the stability condition is again evaluated but by couple of nodes. And so on for each couple of nodes of the network until matrix $\mathbf{A}$ is completed.

Method 1: Eigenvalues analysis of autocorrelation matrix

As stated in [34], the convergence behavior of a time-domain decentralized ANC system using the FxLMS algorithm are affected by the eigenvalue distribution of $\mathbf{R}{=}E\{\tilde{\mathbf{U}}(n)\mathbf{U}^T(n)\}$. In this regard, the coefficients vector $\mathbf{w}(n)$ will converge on average to their asymptotic values, provided the real parts of all the eigenvalues of $\mathbf{R}$ are positive (assuming that the proper value of μ is used). A more detailed analysis of this is carried out in Section 4.3.1 of Chapter 4.

Let us consider λ_p as the p-th eigenvalue of $\mathbf{R}$ for $p{=}1,2,\ldots,LN$ and $\mathbf{R}_{[jk]}$ as a $[L2{\times}L2]$ submatrix of $\mathbf{R}$ arranged as

$$\mathbf{R}_{[jk]} = \begin{bmatrix} \mathbf{R}_{jj} & \mathbf{R}_{jk} \\ \mathbf{R}_{kj} & \mathbf{R}_{kk} \end{bmatrix}, \tag{A.3}$$

where

$$\mathbf{R}_{jk} = \begin{bmatrix} R_{jk}(0) & R_{jk}(1) & \cdots & R_{jk}(L-1) \\ R_{jk}(1) & R_{jk}(0) & \cdots & R_{jk}(L-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_{jk}(L-1) & R_{jk}(L-2) & \cdots & R_{jk}(0) \end{bmatrix}, \tag{A.4}$$

being $R_{jk}(l){=}R_{xx}(l){*}\mathbf{h}_{jj}(l){*}\hat{\mathbf{h}}_{kj}(l)$ with $R_{xx}(l){=}E\{x(n{+}l)\,x(n)\}$ and $\hat{\mathbf{h}}_{kj}(l) = \mathbf{h}_{kj}(-l)$. As stated before, this is widely explained in Section 4.3.1. In addition, λ_q is defined as the q-th eigenvalue of $\mathbf{R}_{[jk]}$ for $q{=}1,2,\ldots,L2$ and $j{\neq}k$. Consequently, collaboration condition will be evaluated as follows:

- *Collaboration condition*: All the nodes of a N-node ASN may work independently if $\mathbb{R}\mathrm{e}\{\lambda_p\}{>}0$ for all values of p, i.e., for all eigenvalues of $\mathbf{R}$. However,

if any of the real parts of λ_p is negative, the previous condition is not fulfilled. Then, collaboration condition is evaluated by pair of nodes. Considering nodes j and k, node j must collaborate with k if any $\mathbb{R}\mathrm{e}\{\lambda_q\}<0$. If this condition is fulfilled, then $\alpha_{jk}=\alpha_{kj}=1$. And so on for the rest of the nodes until matrix $\mathbf{A}$ is completed.

This method consider the use of a broadband noise as reference signal for the correct design of the collaboration matrix.

Method 2: Eigenvalues analysis of acoustic system matrix

Note that $\mathbf{R}$ depends on both the statistics of the reference signals and the acoustic paths. However, for well-known and statistically characterized reference signals, matrix $\mathbf{R}$ is only dependent on the acoustic system. As example, when the reference signal is a zero-mean Gaussian white noise with power σ^2, we get $R_{jk}(l)=\sigma^2(\mathbf{h}_{jj}(l) * \hat{\mathbf{h}}_{kj}(l))$. If we consider a tonal reference signal with amplitude $A\sqrt{2}$ and period T, $R_{jk}(l)=A^2 cos(\frac{2\pi l}{T}) * (\mathbf{h}_{jj}(l) * \hat{\mathbf{h}}_{kj}(l))$. Therefore, in such cases, it would only be necessary to analyse if any of the real part of the eigenvalues of the autocorrelation matrix of the acoustic channels, which we denoted as $\mathbf{R}_h$, is negative. If so, the system stability must be re-analyzed by pairs of nodes in order to design the required collaboration matrix, similar as the steps carried out by the method 1.

- *Collaboration condition*: All the nodes of a N-node ASN may work independently if $\mathbb{R}\mathrm{e}\{\lambda_p\}>0$ for all values of p, i.e., for all eigenvalues of $\mathbf{R}_h$. However, if any of the real parts of λ_p is negative, the previous condition is not fulfilled. Then, collaboration condition is evaluated by pair of nodes. Considering nodes j and k, node j must collaborate with k if $\mathbb{R}\mathrm{e}\{\lambda_q\}<0$. If this condition is fulfilled, then $\alpha_{jk}=\alpha_{kj}=1$. And so on for the rest of the nodes until matrix $\mathbf{A}$ is completed.

Note that this method consider the use of a tonal noise as reference signal for the correct design of the collaboration matrix.

Methods 3-4: Eigenvalues analysis of acoustic transfer matrix

An analysis of the convergence properties of the algorithm in the frequency domain could provide a better understanding of the physical problem. The eigenvalues analysis of the transfer matrix $(\mathbf{A}\circ\underline{\mathbf{H}})\underline{\mathbf{H}}^H$ evaluated at a single frequency ω_0 was presented in [34] with $\mathbf{A}$ defined as a identity matrix. The stability condition is based on the analysis of the eigenvalues of $(\mathbf{A}\circ\underline{\mathbf{H}})\underline{\mathbf{H}}^H$. In addition, it is possible to evaluate the

stability condition at each frequency bin (FFT) separately. Thus, if the stability condition is not fulfilled, two methods may be carried out by pair of nodes to obtain the collaboration matrix:

- *Method 3)* This method analyzes the real part of the eigenvalues of the [2×2] matrix $(\mathbf{A}_{[jk]} \circ \underline{\mathbf{H}}_{[jk]})\underline{\mathbf{H}}^H_{[jk]}$ at a single frequency where

$$\mathbf{A}_{[jk]} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \underline{\mathbf{H}}_{[jk]} = \begin{bmatrix} \underline{H}_{jj} & \underline{H}_{jk} \\ \underline{H}_{kj} & \underline{H}_{kk} \end{bmatrix}, \quad \underline{\mathbf{H}}^H_{[jk]} = \begin{bmatrix} \underline{H}^*_{jj} & \underline{H}^*_{kj} \\ \underline{H}^*_{jk} & \underline{H}^*_{kk} \end{bmatrix}. \quad \text{(A.5)}$$

  Collaboration condition: Considering the pair of nodes j and k and being $\boldsymbol{\lambda}_{\underline{\mathbf{H}}}$ the two eigenvalues of $(\mathbf{A}_{[jk]} \circ \underline{\mathbf{H}}_{[jk]})\underline{\mathbf{H}}^H_{[jk]}$, nodes j and k must collaborate if any of the real part of $\boldsymbol{\lambda}_{\underline{\mathbf{H}}}$ is negative, i.e., any $\mathbb{R}\mathrm{e}\{\boldsymbol{\lambda}_{\underline{\mathbf{H}}}\}<0$. In that case, $\alpha_{jk}=1$.

- *Method 4)* The previous study can be extended to each frequency bin, $\omega_m = 0, 1, 2, .., M/2$, where M is the length of the acoustic paths and it is considered as even number. The real part of the eigenvalues of $(\mathbf{A}_{[jk]} \circ \underline{\mathbf{H}}_{[jk]})\underline{\mathbf{H}}^H_{[jk]}$, is now analyzed at each frequency ω_m.

  Collaboration condition: Considering the pair of nodes j and k and being $\boldsymbol{\lambda}_{\underline{\mathbf{H}},m}$ the two eigenvalues of the matrix $(\mathbf{A}_{[jk]} \circ \underline{\mathbf{H}}_{[jk]})\underline{\mathbf{H}}^H_{[jk]}$ at ω_m, nodes j and k must collaborate if any of the real part of $\boldsymbol{\lambda}_{\underline{\mathbf{H}},m}$ is negative, i.e., any $\mathbb{R}\mathrm{e}\{\boldsymbol{\lambda}_{\underline{\mathbf{H}},m}\}<0$. In that case, $\alpha_{jk}=1$.

Note that methods 3 and 4 are only considered for reference signals composed of tonal components.

A.3 Methods based on acoustic parameters

In this section, several methods to design the collaboration matrix based on the extraction of parameters derived from the study of the acoustic paths matrix are proposed. For the last three methods, the inital design of $\mathbf{A}$ (identity matrix) is started and then, collaborations are added to the corresponding entries of $\mathbf{A}$ until the system becomes stable.

Method 5: Normalized cross-covariance coefficient

We define the normalized cross-covariance coefficient as

$$\rho_{jk} = \frac{\sum_{m=1}^{M} h_{jk,m} h_{kk,m}}{\sqrt{\sum_{m=1}^{M} h^2_{jk,m}} \sqrt{\sum_{m=1}^{M} h^2_{kk,m}}}. \quad \text{(A.6)}$$

where ρ_{jk} is the cross-covariance between the actuator of node j and the sensor of node k with respect to the actuator and sensor of node k. Values of ρ_{jk} close to 1 mean that acoustic path $\mathbf{h}_{jk}$ is very correlated to $\mathbf{h}_{kk}$.

Collaboration condition: It is considered that node j must collaborate with node k (α_{jk}=1) when $\rho_{jk} > 0.7$, i.e., $h_{jk,m}$ and $h_{kk,m}$ has a percent similarity of 70%. This value is selected by trial and error after testing in different scenarios.

Method 6: Equivalent acoustic channels delay

The equivalent delay of the acoustic channel $\mathbf{h}_{jk}$ is defined in [163] as

$$\Delta_{jk} = \frac{\sum_{m=1}^{M} m\, h_{jk}^2(m)}{\sum_{m=1}^{M} h_{jk}^2(m)}, \tag{A.7}$$

where Δ_{jk} returns the delay samples of the acoustic channel $\mathbf{h}_{jk}$. Since Δ_{jk} is an indicator of the physical location of the nodes, by comparing the values of Δ_{jk} for different acoustic channels, it is possible to estimate how much nodes may affect the behaviour of the rest.

Collaboration condition: We consider that node k must collaborate with node j (α_{kj}=1) if the difference between the equivalent delay of $\mathbf{h}_{kj}$ and the equivalent delay of $\mathbf{h}_{jj}$ is lower than 0 samples.

Method 7: Energy ratio

The Level of Interaction (LI) at the sensor of node k over node j is defined as the ratio between the energy of acoustic paths $\mathbf{h}_{kj}$ and $\mathbf{h}_{jj}$,

$$LI_{kj} = \frac{\sum_{m=1}^{M} h_{kj}^2(m)}{\sum_{m=1}^{M} h_{jj}^2(m)} = \frac{E_{\mathbf{h}_{kj}}}{E_{\mathbf{h}_{jj}}} \tag{A.8}$$

where $E_{\mathbf{h}_{kj}}$ is the energy of acoustic path $\mathbf{h}_{kj}$. Values of LI_{kj} close to 1 would indicate that node k and node j are acoustically coupled. However, values of close to 0 would mean that the level of acoustic coupling between the actuator of the node k and the sensor of the node j is not relevant (uncoupled). Finally, we have called the ANC system acoustically over-coupled for LI_{kj} much larger than 1.

Collaboration condition: If $E_{\mathbf{h}_{kj}} > E_{\mathbf{h}_{jj}}$, i.e., $LI_{kj} > 1$, node k must collaborate with node j (α_{kj}=1).

Note that for multi-frequency periodic noise signals, the energy ratio defined in (A.8) can be only evaluated at each frequency of interest.

Method 8: Accumulative Energy ratio

Taking into account the energy ratio presented in the previous method, in this case the accumulated energy ratio for each row of a matrix composed of elements defined in (A.8) is evaluated. If the rest of the nodes provide more energy than the node itself, then it collaborates with the one that provides more energy. If the rest continues to interfere with more energy, it collaborates with the next one that interferes more, until the interfering energy is less than that of the node itself. Consider the $[N \times N]$ matrix **LI** with non-diagonal entries LI_{jk} and diagonal entries equal to 1 in the j-th row, we denoted max_j^E as the largest entry in row j excluding the diagonal entry,

$$max_j^E = max(|LI_{j1}|, |LI_{j2}|, \ldots, |LI_{jk}| \ldots, |LI_{jN}|), \text{ for } k \neq j \; . \tag{A.9}$$

Collaboration condition: If **LI** is not diagonally dominant in the j-th row, i.e.,

$$1 - \sum_{\substack{k=1 \\ k \neq j}}^{N} |LI_{jk}| < 0, \tag{A.10}$$

the node k related to max_j^E forces a collaboration with the j-th node (α_{kj}=1). The condition (A.9) is again evaluated but the entry associated with the k-th node is not considered. As many collaborations as necessary are applied until the matrix is diagonally dominant in the j-th row. And so on for all the N rows of **LI**.

Note that for multi-frequency periodic noise signals, matrix **LI** can be evaluated at each frequency of interest or over a frequency range considering its frequency-averaged value as,

$$\widetilde{\mathbf{LI}} = \frac{1}{(f_h - f_l)} \int_{f_l}^{f_h} \mathbf{LI}(f) \cdot df \tag{A.11}$$

where f_h and f_l are the high and low frequencies of interest, respectively.

Method 9: Accumulative matrix transfer technique

Based on the Gershgorin circle theorem [164], the system stability can be also evaluated by examining the relative magnitude of the entries of the acoustic transfer matrix [34]. This method is based on these assumptions to design the collaboration matrix.

Consider the $[N \times N]$ matrix $\underline{\mathbf{HH}}^H$ with entries $\underline{\tilde{H}}_{jk}$ and being $\underline{\tilde{H}}_{jj}$ its diagonal entry in the j-th row, we denoted max_j^H as the largest entry in row j excluding the

diagonal entry,

$$max_j^H = max\{|\tilde{\underline{H}}_{j1}|, |\tilde{\underline{H}}_{j2}|, \ldots, |\tilde{\underline{H}}_{jk}|, \ldots, |\tilde{\underline{H}}_{jN}|\}, \text{ for } k \neq j \ . \tag{A.12}$$

where max returns the maximum elements of an array.

Collaboration condition: If $\underline{\mathbf{HH}}^H$ is not diagonally dominant in the j-th row, i.e.,

$$|\tilde{\underline{H}}_{jj}| - \sum_{\substack{k=1 \\ k \neq j}}^{N} |\tilde{\underline{H}}_{jk}| < 0, \tag{A.13}$$

the node k related to max_j^H forces a collaboration with the j-th node (α_{kj}=1). The condition (A.12) is again evaluated but the entry associated with the k-th node is not considered. As many collaborations as necessary are applied until the matrix is diagonally dominant in the j-th row. And so on for all the N rows of $\underline{\mathbf{HH}}^H$.

Note that this methods is only considered for reference signals composed of tonal components.

Method 10: Effective independence (EfI) method

In [108], the Effective independence (EfI) method is used to eliminate actuators at redundants positions which do not contribute to control certain area of interest. In our case, this method can be used to eliminate nodes from the collaboration matrix based on their contribution to the error signal measured at the sensor of the node under study. The Effective independence (EfI) method aims to minimize the covariant estimate by obtaining the maximum principal components of the Fisher information and performing an eigenvalue decomposition [165]. This method is similar to determine redundant elements of a matrix. To this end, it evaluates the linear independence of the entries of the acoustic transfer matrix $\underline{\mathbf{H}}$ [108]. When the range of a matrix is maximum, the contribution of a column or row in the matrix is determined by the mutual linear independence, i.e., no column or row is a linear combination of the remaining ones. Since $\underline{\mathbf{H}}$ is composed of N nodes (N actuators and N sensors), the maximum rank of $\underline{\mathbf{H}}$ is N. Acoustic channels with a high linear independency in $\underline{\mathbf{H}}$ would have a stronger influence on the control area. Therefore, the linear independency of this matrix may be related to the level of acoustic coupling among acoustic channels. Note that $\underline{\mathbf{H}}$ can be described by its singular value decomposition as

$$\underline{\mathbf{H}} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^H. \tag{A.14}$$

where $\mathbf{U}$ is the $[N \times N]$ matrix of eigenvectors of $\underline{\mathbf{HH}}^H$, $\mathbf{\Lambda}$ is the $[N \times N]$ matrix composed of the real singular values of $\underline{\mathbf{H}}$ and $\mathbf{V}$ is the $[N \times N]$ matrix of eigenvectors of $\underline{\mathbf{H}}^H\underline{\mathbf{H}}$ defined as

$$\mathbf{V} = [\mathbf{V}_1^H \ \mathbf{V}_2^H \cdots \ \mathbf{V}^z N], \tag{A.15}$$

being $\mathbf{V}_k$ the k-th right-singular vector of $\underline{\mathbf{H}}$ of size $[N\times 1]$, $\mathbf{V}_k=[V_{1k}\ V_{2k}\ \cdots\ V_{Nk}]^T$. We define the matrix $\mathbf{EfI}$ of size $[N\times N]$ as

$$\mathbf{EfI} = [\mathbf{EfI}_1\ \mathbf{EfI}_2 \cdots\ \mathbf{EfI}_N]. \tag{A.16}$$

The k-th vector column, $\mathbf{EfI}_k$, is calculated as

$$\mathbf{EfI}_k = diag(\mathbf{V}_k \circ \mathbf{V}_k^H) \tag{A.17}$$

where $\mathbf{EfI}_k=[EfI_{1k}\ EfI_{2k}\ldots\ EfI_{Nk}]^T$ with $0\leq EfI_{jk}\leq 1$. Values of EfI_{jk} close to 1 indicate greater linear independence. Since $\mathbf{EfI}$ is a left-stochastic matrix (its columns sums to 1), the EfI_{jk} value may be related to the actuator of the node j influencing the sensor of the node k in a percentage with respect to the loudspeaker-microphone pair of node k. Note that (A.17) is only evaluated for reference signals composed of a tonal component at the frequency ω_0. To determine (A.17) over a frequency range, the frequency-averaged $\mathbf{EfI_k}$ value, $\widetilde{\mathbf{EfI}}_k$, can be considered [166],

$$\widetilde{\mathbf{EfI}}_k = \frac{1}{(f_h - f_l)} \int_{f_l}^{f_h} \mathbf{EfI}_k(f) \cdot df \tag{A.18}$$

where f_h and f_l are the high and low frequencies of interest, respectively. Redefining (A.16) as the averaged-frequency matrix $\widetilde{\mathbf{EfI}} = [\widetilde{\mathbf{EfI}}_1\ \widetilde{\mathbf{EfI}}_2 \cdots\ \widetilde{\mathbf{EfI}}_N]$, we denote max_j^{EfI} as the largest entry in row j of $\widetilde{\mathbf{EfI}}$ excluding the diagonal entry,

$$max_j^{EfI}=max(|\widetilde{EfI}_{j1}|, |\widetilde{EfI}_{j2}|, \ldots, |\widetilde{EfI}_{jk}| \ldots, |\widetilde{EfI}_{jN}|), \text{ for } k \neq j\ . \tag{A.19}$$

Collaboration condition: Node k must collaborate with node j (α_{kj}=1) if $\widetilde{\mathbf{EfI}}$ is not diagonally dominant in the j-th row, i.e.,

$$|\widetilde{EfI}_{jj}| - \sum_{\substack{k=1 \\ k\neq j}}^{N} |\widetilde{EfI}_{jk}| \leqslant 0, \tag{A.20}$$

the node k related to max_j^{EfI} forces a collaboration with the j-th node. The condition (A.19) is again evaluated but the entry associated with the k-th node is not considered. As many collaborations as necessary are applied until the matrix is diagonally dominant in the j-th row. And so on for all the N rows of $\widetilde{\mathbf{EfI}}$.

Quick guideline

It can be observed that depending on the nature of the disturbance signal (tonal or broadband noise), some methods must be selected over others. Therefore, a summarized guidelines for obtaining the theoretical collaboration matrix according to the noise signal are explained as following:

1. For broadband noise signals:

   (a) To determine the stability condition from the eigenvalue distribution of $\mathbf{R}$. If the condition is not fulfilled (unstable system), to obtain the collaboration matrix from the same study but by pairs of nodes (Method 1).

   (b) To determine the stability condition from the eigenvalue distribution of $\mathbf{R}_H$. If the condition is not fulfilled (unstable system), to obtain the collaboration matrix from the same study but by pairs of nodes (Method 2).

   (c) To validate the system stability by using the collaboration matrices proposed in Methods 1 and 2 as well as in Methods from 5 to 10.

2. For narrowband noise signals:

   (a) To calculate the collaboration matrix from the following methods: energy ratio (Method 7), accumulative energy ratio (Method 8), Accumulative Gershgorin technique (Method 9) and effective independence (EfI) methods (Method 10).

   (b) To determine the stability condition from the eigenvalue distribution of $\mathbf{R}$. If the condition is not fulfilled (unstable system), to obtain the collaboration matrix from the same study but by pairs of nodes (Method 1).

   (c) To determine the stability condition from the eigenvalue distribution of $\mathbf{R}_H$. If the condition is not fulfilled (unstable system), to obtain the collaboration matrix from the same study but by pairs of nodes (Method 2).

   (d) To determine the stability condition from the eigenvalue analysis of acoustic transfer matrix. If the condition is not fulfilled (unstable system),

      i. To obtain the collaboration matrix from the same study but by pairs of nodes at a single frequency ω_0 (Method 3)

      ii. To obtain the collaboration matrix from the same study but by pairs of nodes extended to each frequency bin, $\omega_m = 0, 1, 2, .., M/2$. (Method 4)

   (e) To validate the system stability by using the collaboration matrices proposed in Methods from 3 to 10.

The proposed matrix is the one that implies the least collaboration of all the obtained collaboration matrices, that is to say, the matrix that presents the minimum number of them in matrix A.

A.4 Conclusions

This appendix presents several experimental collaborative criteria based on the previous analysis of the acoustic system in order to parameterize the degree of acoustic coupling between nodes. Based on this parameter, the collaboration between nodes is estimated before the ANC control.

Appendix B

Simulator of sound field controller over ASNs

B.1 Introduction

In this appendix, a software tool to simulate and model the behaviour of sound control systems is proposed. It includes the explanation of the main issues of developing a MATLAB® tool to this aim. The tool is able to simulate different network deployments and communication procedures, different acoustic systems and different algorithms for sound control. Therefore, first of all it is important to explain three different parts regarding ASNs (see Figure B.1).

- Algorithms. They define the processes to be performed on signals and the interaction between the system and the input-output devices (microphone-loudspeakers) as well as with other network elements.
- Communication system. It allows information exchanges among the nodes and can simulate real effects of communications as latencies, transmission errors, information loss, etc.
- Acoustic system. It models the transformation suffered by the signals from their reproduction at the loudspeakers till their recording at the microphones. It simulates the physical propagation and reflection of the sound by means of impulse responses measured at different points of the closed space.

Although these parts may share information, the simulation tool allows them to work independently. Another important feature of the software is that each node can use distributed signal processing if the communication network allows to it, which

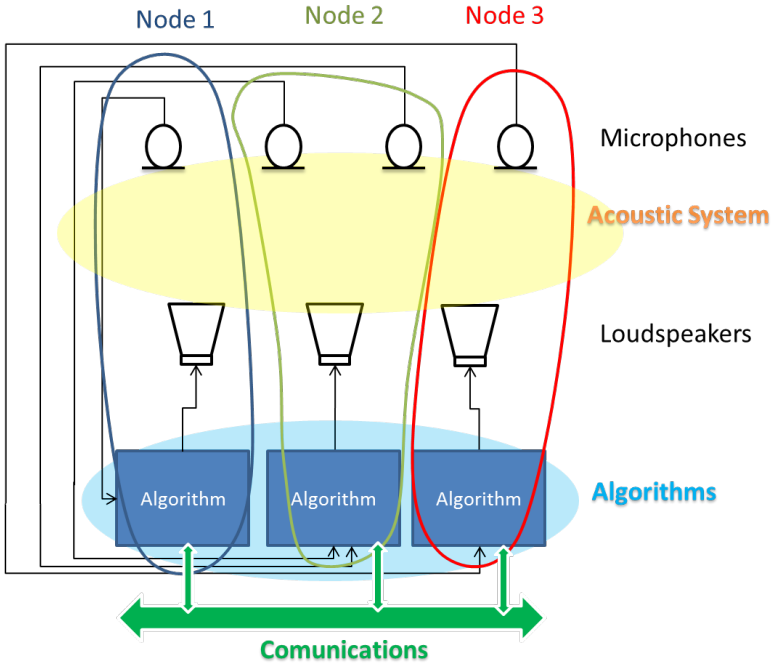


Figure B.1: Diagram of a sound control system with multi-channel nodes.

means that an output signal is obtained at each node as a result of processing both the local input signal and the information received from other nodes. Although each node processes the signals independently, each one is relevant to the working of the global system. The simulator presents other benefits like the storage, reproduction and analysis of all the audio signals or parameters in the simulation as well as tools for the definition, modification or study of the acoustic systems. The tool simulation has been developed to analyse the algorithms that define the sound field control applications in a real system without the need to program them on a DSP platform or to have an acoustic system in situ. Several types of communication errors between the network and the nodes and even model the data acquisition impairments in real-time applications can be simulated. In summary, this tool is intended to fill the gap between the initial mathematical idea of an algorithm and its final programming on a digital platform.

B.2 Description of the simulation tool

The simulator is based on the MATLAB® data type called *struct*, which is a structured array of data sets organized by groups called fields. Each field can contain any

type of data of any size. Structures can be built in two ways:

- Using the *struct* function (*structArray = struct('field1', val1, 'field2', val2, ...)*) where the arguments are field names and their corresponding values.
- Using an assignment statement that assigns data to individual fields (*structArray.field1='val1'; structArray.field2=val2; ...*).

Some of the most important features of this data type are:

- All structures in the array must have the same number of fields.
- All fields must have the same field names.
- Fields of the same name in different *struct* variables can contain different types or sizes of data.
- Besides, a *struct* provides hierarchical storage mechanisms to contain different types of data.
- Nested structures are allowed and vectors and matrices of structures can be created.
- It is possible to add a new field to a structure at any time.
- Moreover, the structures have optimized functions for specific operations. See MATLAB® documentation online for extensive information [167].

A brief description of the different steps to perform the simulation of an acoustic network model is defined as follows. We define acoustic network model as the model based on the configuration of all the parameters needed to simulate a sound field control application. The first step is to initialize the acoustic network through the configuration of all the system processes:

- *Acoustic system configuration*. This setting loads the file containing the acoustic channel impulse response between each loudspeaker and each microphone located in the enclosure. In this part, the number of loudspeakers and microphones that will manage each node is defined. It is possible to create as much multichannel nodes as number of loudspeakers and microphones.
- *Algorithms configuration*. The algorithm to be executed by each node is selected depending on the application to simulate. Besides, the different parameters needed for the operation of each algorithm are defined in such a way that several nodes can share the same algorithm with different configuration parameters.

- *Communication system configuration*. It defines the type of topology of the communication network and how the nodes are collaborating. For example, a node can interchange information with a small group of nodes or only with its neighbour nodes. On the other hand, the communication network can introduce latencies in the transmission of information to the nodes hence it is possible to simulate this delay in number of samples at each node.

- *Data acquisition system configuration*: The tool tries to simulate the data acquisition that would be carried out in real-time applications over asynchronous networks. Each node has its own clock and stores, process and sends data independently from the rest. Similar to a DSP performance, a node is in a permanent state of low consumption until an interruption occurs due to the arrival of data. At that time, the node performs the corresponding processing. Such processing ends with another interruption returning to the initial state of low consumption of the node. Thus, we define two interrupt flags: reading/writing (R/W) and processing time. R/W interruptions indicate the moment at which the node reads the new data and writes the previous data. Processing time indicates the time it takes to process a sample block. If the node can read new data (R/W interruption is active) but the node is still processing previous data (processing time interruption is active too), the new data will be lost. It also defines the sampling time as the minimum unit of time at which the simulator works. So, the acoustic system works sample-by-sample in the time domain (to approximate us as close as possible to the continuous domain) but the nodes can execute algorithms implemented with blocks of samples in the frequency domain too. Similarly, the communication network works sample-by-sample independently of the other processes. Summarizing, at the same time and regardless of the system unit of time, each node can work with a different size block, a different sampling frequency and a different processing time.

Once considered the configuration data explained above, the network structure is created and the acoustic network model is characterized. Notice that the network structure addresses the information but does not process it. Therefore, the structure must have particular fields containing the variables that will be used within the algorithms, the acoustic system and the communication system for data processing. Some of these fields are:

- *State*. Field containing the input and output variables of each process in the system.

- *Initial state*. Field containing the initial values of the variables in the previous field.

- *Historical*. Field to store the variables that want to analyse.

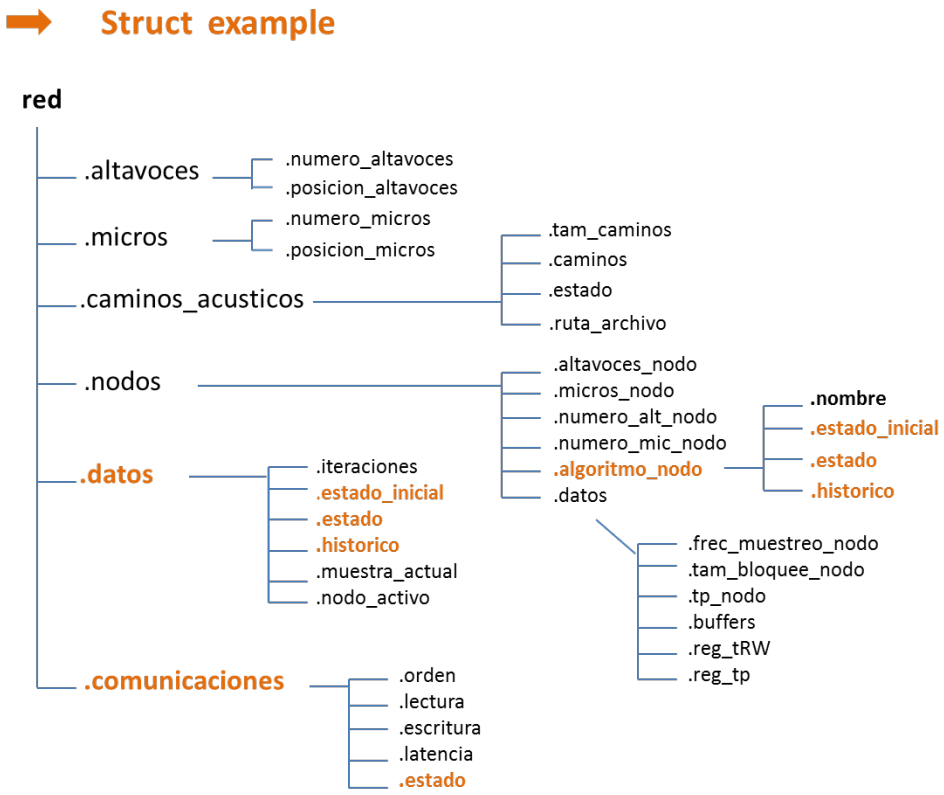


Figure B.2: Example of the type of structure used in the simulator.

A complete description of all the fields is out of the scope of this appendix, but an example of one of the data structure used in the tool is shown in Figure B.2.

Finally, Figure B.3 shows the steps to run a complete simulation. The tool runs the algorithms at each node and simulates if there are latencies in the information exchange among the nodes. Once the signals have been processed by the algorithms and have been sent to the loudspeakers, the acoustic system generates the signals captured at the microphones' locations. In this way, the tool can quickly perform different simulations for different applications by modifying the appropriate parameters in the initial configuration.

B.2.1 Simulation of the algorithms

The algorithms can work sample-by-sample or grouped in blocks of samples. A particular algorithm reads values in certain structure fields, executes the operations

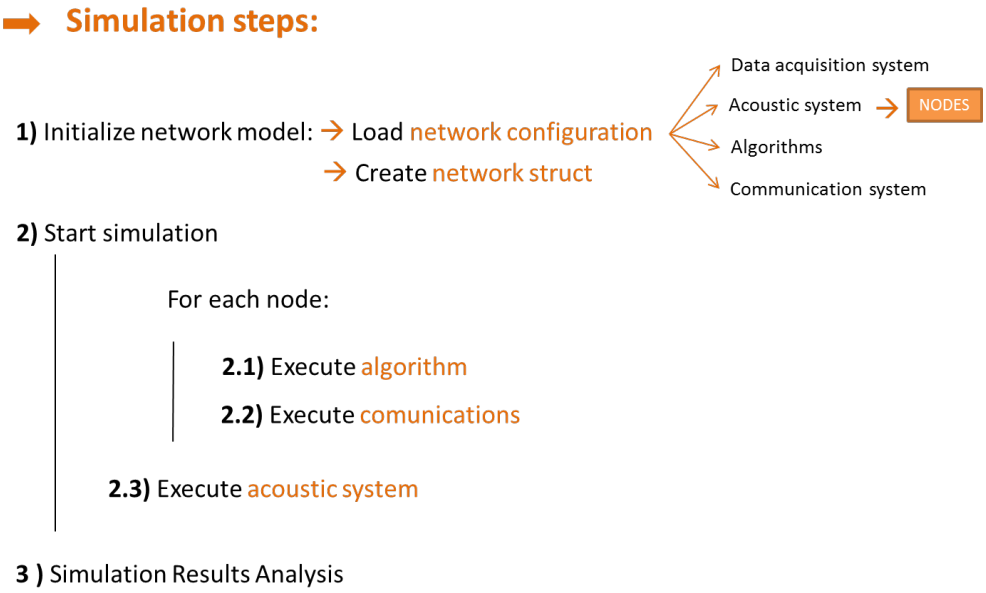


Figure B.3: Steps to perform the simulation of an acoustic network model.

and writes in certain structure fields. The nodes can execute any algorithm once it is adapted to the format defined by the structure.

The general definition of an algorithm consists of four parts. An example is shown in Figure B.4:

- *Declaration statement*. It states the function name, the needed input variables from the structure and the output variables that are going to be returned to the structure.
- *Reading*. It reads the field *State* of both the algorithm node and the communications system (if any information exchange between the nodes exists), obtaining the values of the variables used by the algorithm.
- *Processing*. In this section, the necessary operations for the proper performance of the algorithm are executed. Depending of the type, the algorithm can process one or several input variables (corresponding to the input signals or the signals captured by microphones or parameters sent by other nodes, etc.) and one or several output variables (corresponding to the signals reproduced by the speakers, parameters to send to other nodes, etc.).
- *Writing*. After processing and similarly to the reading process, the new values

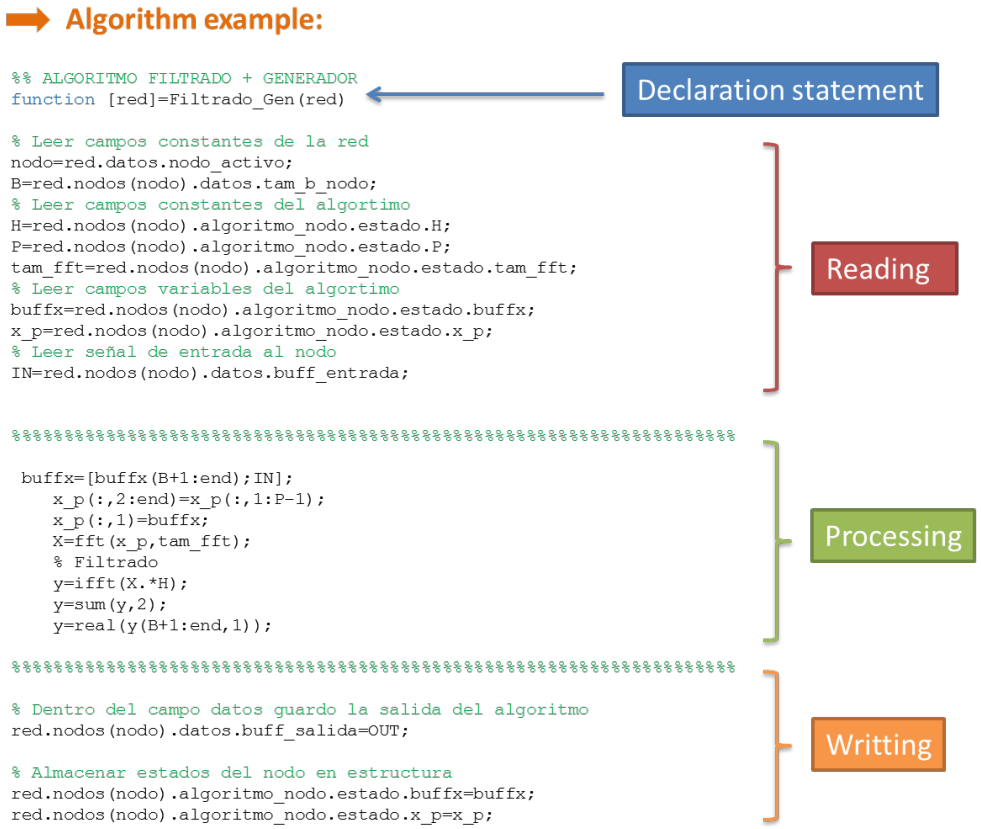


Figure B.4: Example of an algorithm function.

of the variables must have been stored in the state field of both the algorithm of each node and the communications network.

B.2.2 Simulation of the communication system

The communication system performs two types of operations in the simulator. On the one hand, it is possible to simulate latencies (or other effects) in the transmission of information to the nodes by delaying the input signals a certain number of samples at each node. On the other hand, if collaborative algorithms are used, the system allows to exchange information among the different nodes. In this case, the network topology explains how the nodes must collaborate stating for each node which nodes can read data from it and in which nodes it can write data. This information, along with the nodes that collaborate and their order in the network, must be introduced in

the configuration of the communications network. Depending on the topology, the network will be responsible for sending that information to the corresponding node as well as introducing errors or delays, if needed.

B.2.3 Simulation of the acoustic system

Acoustic systems are formed by the number of actuators (loudspeakers) and number of sensors (microphones) along with the impulse response of the acoustic paths between them, that is, the acoustic system defines an enclosure and the position where the loudspeakers and microphones are placed within it. The acoustic system must work sample-by-sample in the time domain in contrast to the algorithms that can be run sample-by-sample or in blocks of samples. However, the programming of the acoustic system is quite similar to the definition of the algorithms. It contains first the declaration statement of the function followed by a part where the necessary variables are read from the fields of the structure. Second, the processing where the signals reproduced by the loudspeakers are filtered by the acoustic paths. Finally, the signals captured by the microphones are stored in the structure. An example is shown in Figure B.5.

B.2.4 Analysis of the performance

The tool can analyse any existing signal or parameter in the system. Using a specific function, the tool stores in the structure only those signals captured by the microphones that have been previously selected as well as only those signals produced by the loudspeakers that have been previously chosen. This reduces the computational load and the stored memory in the cases of systems with a large number of microphones or speakers but that only need the signals captured or reproduced by some of them.

B.2.5 Limitations

At this moment the main limitation of the simulation tool is the lack of interaction with the user, but a graphical user interface is planned for the next phase of the project. From the point of view of efficient programming in MATLAB®, the tool tries to use multiple dimension arrays (vectors, matrices) but the use of loops is unavoidable. Therefore, when a large number of nodes are used, both runtime and memory storage become prohibitive. In addition, working with different time units increases the runtime. On the other hand, it would be desirable to simulate the main effects of analog-digital/digital-analog (AD/DA) converters in the audio input/output respectively, as for instance the saturation and finite quantization of the samples. Finally, the tool does not support dynamic or variable acoustic systems.

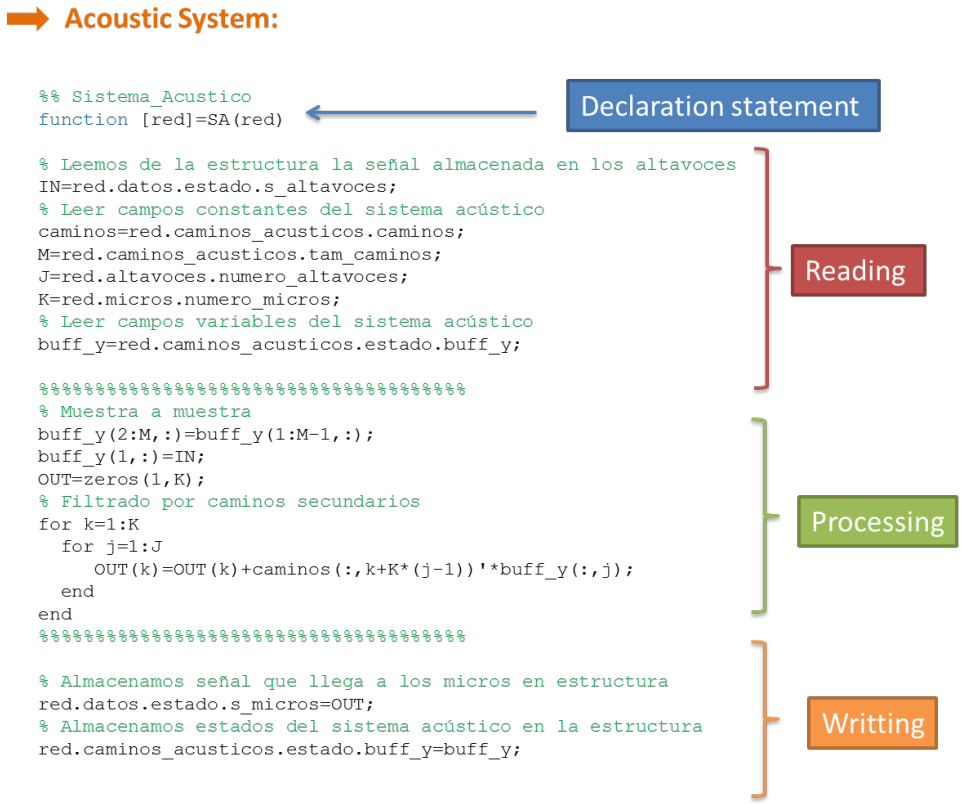


Figure B.5: Function that models the acoustic system.

B.3 Conclusions

In this appendix, a simulation tool that allows the development of sound field control applications over distributed networks has been presented. The main advantage of this tool is the simplicity and quickness to simulate different systems. The tool allows us to create nodes with an arbitrary number of microphones and loudspeakers and to execute any algorithm both in the time and the frequency domain, and both working sample-by-sample and grouped in block of samples. The simulator executes the acoustic system sample-by-sample independently and allows for modelling of both the behaviour of the communications system simulating the possible problems and the data acquisition performed in real time applications. Finally, the proposed tool allows to analyze any signal or parameter in the nodes, acoustic system or communication network, hence, at any point in the system.

Bibliography

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, 2002. 1.1, 2.1, 2.1

[2] A. Flammini, P. Ferrari, D. Marioli, E. Sisinni, and A. Taroni, "Wired and wireless sensor networks for industrial applications," *Microelectronics Journal*, vol. 40, no. 9, pp. 1322–1336, 2009. 1.1, 2.1, 2.1, 2.5

[3] S. Elliott, I. Stothers, and P. Nelson, "A multiple error LMS algorithm and its application to the active control of sound and vibration," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 10, pp. 1423–1434, 1987. 1.1, 2.4.2, 3.1.2, 5.1, 5.2.1, 5.2.2

[4] S. Elliott, *Signal processing for active control*. Academic press, London, UK, 2001. 1.1

[5] C. Lopes and A. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064–4077, 2007. 1.1, 2.1, 2.2, 5.1, 5.2.1

[6] ——, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, pp. 3122–3136, 2008. 1.1, 5.1, 5.2.1, 5.5.3

[7] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008. 2.1

[8] D. Puccinelli and M. Haenggi, "Wireless sensor networks: applications and challenges of ubiquitous sensing," *IEEE Circuits Syst. Mag.*, vol. 5, pp. 19–31, 2005. 2.1

[9] X. Du and H. Chen, "Security in wireless sensor networks," *IEEE Wireless Communications*, vol. 15, no. 4, pp. 60–66, aug 2008. 2.1

[10] M. A. Ameen, J. Liu, and K. Kwak, "Security and privacy issues in wireless sensor networks for healthcare applications," *Journal of Medical Systems*, vol. 36, no. 1, pp. 93–101, mar 2010. 2.1

[11] B. Harjito and S. Han, "Wireless multimedia sensor networks applications and security challenges," in *2010 International Conference on Broadband, Wireless Computing, Communication and Applications*, 2010, pp. 842–846. 2.1

[12] I. Akyildiz, T. Melodia, and K. Chowdury, "Wireless multimedia sensor networks: A survey," *IEEE Wireless Commun.*, vol. 14, no. 6, pp. 32–39, 2007. 2.1

[13] A. Bertrand, "Applications and trends in wireless acoustic sensor networks: A signal processing perspective," in *2011 18th IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*. IEEE, 2011, pp. 1–6. 2.1, 2.2

[14] Y. Guo and M. Hazas, "Acoustic source localization of everyday sounds using wireless sensor networks," in *Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing - Ubicomp*, 2010, pp. 411–412. 2.1

[15] K. Martinez, J. Hart, and R. Ong, "Environmental sensor networks," *Computer*, vol. 37, no. 8, pp. 50–56, aug 2004. 2.1

[16] J. Segura-Garcia, S. Felici-Castell, J. Perez-Solano, M. Cobos, and J. Navarro, "Low-cost alternatives for urban noise nuisance monitoring using wireless sensor networks," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 836–844, 2015. 2.1

[17] A. Bertrand and M. Moonen, "Robust distributed noise reduction in hearing aids with external acoustic sensor nodes," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, no. 1, pp. 1–14, 2009. 2.1

[18] H. Wang, "Wireless sensor networks for acoustic monitoring," Ph.D. dissertation, 2006. 2.1

[19] H. Kwon, V. Berisha, and A. Spanias, "Real-time sensing and acoustic scene characterization for security applications," in *2008 3rd International Symposium on Wireless Pervasive Computing*. IEEE, 2008, pp. 755–758. 2.1

[20] N. Bogdanovic, J. Plata-Chaves, and K. Berberidis, "Distributed incremental-based LMS for node-specific adaptive parameter estimation," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5382–5397, oct 2014. 2.1, 2.2

[21] A. Bertrand, "Signal processing algorithms for wireless acoustic sensor networks," Ph.D. dissertation, 2011. 2.1, 2.2

[22] 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. 2.1, 3.1.4, 3.2, 4.1, 5.1, 5.2, 5.2.1, 5.2.4

[23] G. Pinero, J. Estreder, F. Martinez-Zaldivar, M. Ferrer, and M. de Diego, "Sound-field reproduction system over a two-node acoustic network of mobile devices," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. Institute of Electrical and Electronics Engineers (IEEE), dec 2015. 2.1

[24] H. Kim, S. Lee, J.-W. Choi, H. Bae, J. Lee, J. Song, and I. Shin, "Mobile maestro: enabling immersive multi-speaker audio applications on commodity mobile devices," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2014, pp. 277–288. 2.1

[25] C. A. Kardous and P. Shaw, "Evaluation of smartphone sound measurement applications (apps) using external microphones: a follow-up study," *The Journal of the Acoustical Society of America*, vol. 140, no. 4, pp. EL327–EL333, 2016. 2.1

[26] J. C. Szurley, "Distributed signal processing algorithms for acoustic sensor networks," Ph.D. dissertation, 2015. 2.1

[27] A. Tanenbaum and D. Wetherall, *Computer Networks*. Pearson Education, Boston, Mass., US, 2012. 2.1

[28] T. Huynh and W.-J. Hwang, "Network lifetime maximization in wireless sensor networks with a path-constrained mobile sink," *International Journal of Distributed Sensor Networks*, vol. 11, no. 11, p. 679093, 2015. 2.1

[29] C. Tunca, S. Isik, M. Y. Donmez, and C. Ersoy, "Distributed mobile sink routing for wireless sensor networks: a survey," *IEEE Communications Surveys Tutorials*, vol. 16, no. 2, pp. 877–897, Second 2014. 2.1

[30] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, Feb 2009. 2.1

[31] M. Cobos, J. Perez-Solano, O. Belmonte, G. Ramos, and A. Torres, "Simultaneous ranging and self-positioning in unsynchronized wireless acoustic sensor networks," *IEEE Transactions on Signal Processing*, vol. 64, no. 22, pp. 5993–6004, nov 2016. 2.1

[32] C. Llerena-Aguilar, R. Gil-Pita, M. Rosa-Zurera, D. Ayllon, M. Utrilla-Manso, and F. Llerena, "Synchronization based on mixture alignment for sound source separation in wireless acoustic sensor networks," *Signal Processing*, vol. 118, pp. 177–187, 2016. 2.1

[33] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, 2013. 2.2, 5.1, 5.5.3

[34] S. Elliott and C. Boucher, "Interaction between multiple feedforward active control systems," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 521–530, 1994. 2.2, 3.1.3, 3.1.4, 4.1, 5.3, A.2, A.2, A.3

[35] J. Plata-Chaves, A. Bertrand, M. Moonen, S. Theodoridis, and A. M. Zoubir, "Heterogeneous and multitask wireless sensor networks —algorithms, applications, and challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 3, pp. 450–465, Apr. 2017. 2.2

[36] J. Plata-Chaves, A. Bertrand, and M. Moonen, "Incremental multiple error filtered-x lms for node-specific active noise control over wireless acoustic sensor networks," in *IEEE Sensor Array and Multichannel Signal Processing Workshop*. IEEE, 2016, pp. 1–5. 2.2, 5.1, 5.6, 5.6.1

[37] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks - part i: Sequential node updating," *IEEE Transactions on Signal Processing*, vol. 58, pp. 5277–5291, 2010. 2.2

[38] ——, "Distributed adaptive node-specific signal estimation in fully connected sensor networks - part ii: Simultaneous and asynchronous node updating," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5292–5306, Oct. 2010. 2.2

[39] I. D. Schizas, G. B. Giannakis, and Z. Q. Luo, "Distributed estimation using reduced-dimensionality sensor observations," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4284–4299, Aug. 2007. 2.2

[40] J. Fang and H. Li, "Optimal/near-optimal dimensionality reduction for distributed estimation in homogeneous and certain inhomogeneous scenarios," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4339–4353, Aug. 2010. 2.2

[41] B. Widrow and S. Stearns, *Adaptive signal processing*. Prentice Hall, Englewood Cliffs, NJ, US, 1985. 2.2, 2.3, 2.3.1, 5.3, 6.1.1

[42] P. M. Clarkson, *Optimal and adaptive signal processing*. Routledge, 2017. 2.3

[43] S. Haykin, *Adaptive filter theory*. Prentice-Hall, Upper Saddle River, NJ, US, 2002. 2.3, 2.3, 2.3.1, 2.4.1, 3.1.1, 3.1.4, 4.4.1, 5.3, 5.4

[44] B. Farhang-Boroujeny, *Adaptive filters: theory and applications*. John Wiley & Sons, Chicester, West Sussex, UK, 2013. 2.3, 5.5

[45] R. L. Plackett, "Some theorems in least squares," *Biometrika*, vol. 37, no. 1/2, pp. 149–157, 1950. 2.3

[46] R. Hastings-James and M. Sage, "Recursive generalised-least-squares procedure for online identification of process parameters," in *Proceedings of the Institution of Electrical Engineers*, vol. 116, no. 12. IET, 1969, pp. 2057–2062. 2.3

[47] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electronics and Communications in Japan (Part I: Communications)*, vol. 67, no. 5, pp. 19–27, 1984. 2.3.1, 2.3.3, 5.1, 5.4

[48] S. C. Douglas, "A family of normalized lms algorithms," *IEEE signal processing letters*, vol. 1, no. 3, pp. 49–51, 1994. 2.3.1

[49] E. Ferrara, "Fast implementations of lms adaptive filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 474–475, 1980. 2.3.1

[50] M. Ferrer, "Filtrado adaptativo multicanal para control local de campo sonoro basado en algoritmos de proyección afín," Ph.D. dissertation, Ph.D. dissertation, Universitat Politecnica de Valencia, Spain, 2008. 2.3.3

[51] T. Betlehem, W. Zhang, M. A. Poletti, and T. Abhayapala, "Personal sound zones: Delivering interface-free audio to multiple listeners," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 81–91, mar 2015. 2.4, 2.4.2, 4.3.2, 7.1

[52] D. Guicking, K. Karcher, and M. Rollwage, "Coherent active methods for applications in room acoustics," vol. 78, pp. 1426–1434, 1985. 2.4

[53] M. A. Poletti, "An assisted reverberation system for controlling apparent room absorption and volume," in *Audio Engineering Society Convention 101*, Nov 1996. 2.4

[54] P. Nelson and S. Elliott, *Active Control of Sound*. Academic Press, 1992, no. v.3. 2.4, 2.4.1

[55] M. A. Gerzon, "Ambisonics in multichannel broadcasting and video," *Journal of the Audio Engineering Society*, vol. 33, no. 11, pp. 859–871, 1985. 2.4

[56] A. J. Berkhout, D. de Vries, and P. Vogel, "Acoustic control by wave field synthesis," *The Journal of the Acoustical Society of America*, vol. 93, no. 5, pp. 2764–2778, 1993. 2.4

[57] S. Cohen and N. Weinstein, "Nonauditory effects of noise on behavior and health," *Journal of Social Issues*, vol. 37, no. 1, pp. 36–70, 1981. 2.4.1

[58] "Noise impacts on health," *Science for Environment Policy. European Commision*, vol. 47, Jan. 2015. 2.4.1

[59] "Future noise policy - european commission green paper," *European Commission*, Nov. 1996. 2.4.1

[60] P. Lueg, "Process of silencing sound oscillations," U.S. Patent 2.043.416, 1936. 2.4.1

[61] D. Guicking, "On the invention of active noise control by paul lueg," *The Journal of the Acoustical Society of America*, vol. 87, no. 5, pp. 2251–2254, 1990. 2.4.1

[62] H. Olson and E. May, "Electronic sound absorber," *The Journal of the Acoustical Society of America*, vol. 25, no. 6, pp. 1130–1136, nov 1953. 2.4.1

[63] W. B. Conover, "Fighting noise with noise," *Noise Control*, vol. 2, no. 2, pp. 78–92, 1956. 2.4.1

[64] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, J. E. Dong, and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692–1716, Dec 1975. 2.4.1, 2.4.2

[65] S. Elliott and P. Nelson, "Active noise control," *IEEE Signal Process. Mag.*, vol. 10, no. 4, pp. 12–35, 1993. 2.4.1, 2.4.1

[66] S. Elliott, P. Joseph, A. Bullmore, and P. Nelson, "Active cancellation at a point in a pure tone diffuse sound field," *Journal of Sound and Vibration*, vol. 120, no. 1, pp. 183 – 189, 1988. 2.4.1, 5.7

[67] P. Joseph, S. Elliott, and P. Nelson, "Near field zones of quiet," *Journal of Sound and Vibration*, vol. 172, no. 5, pp. 605–627, may 1994. 2.4.1, 5.7

[68] M. Pawełczyk, "Active noise control-a review of control-related problems," *Archives of Acoustics*, vol. 33, no. 4, pp. 509–520, 2008. 2.4.1

[69] S. M. Kuo, M. Tahernehadi, and W. Hao, "Convergence analysis of narrow-band active noise control system," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 2, pp. 220–223, Feb 1999. 2.4.1, 2.4.2

[70] D. Morgan, "An analysis of multiple correlation cancellation loops with a filter in the auxiliary path," vol. 28, no. 4, pp. 454–467. 2.4.1, 2.4.2

[71] J. Burgess, "Active adaptive sound control in a duct: A computer simulation," *The Journal of the Acoustical Society of America*, vol. 70, no. 3, pp. 715–726, sep 1981. 2.4.1

[72] L. de Oliveira, "Review and future perspectives on active sound quality control," in *Proceedings of the International Conference on Noise and Vibration Engineering–ISMA*, 2010. 2.4.2, 2.4.2

[73] M. J. Ji and S. M. Kuo, "An active harmonic noise equalizer," in *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 189–192 vol.1. 2.4.2

[74] S. M. Kuo and M. J. Ji, "Principle and application of adaptive noise equalizer," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 41, no. 7, pp. 471–474, Jul 1994. 2.4.2, 2.4.2

[75] ——, "Development and analysis of an adaptive noise equalizer," *IEEE Trans. on Speech and Audio Proces.*, vol. 3, no. 3, pp. 217–222, May 1995. 2.4.2, 2.4.2, 6.1.2

[76] J. Glover, "Adaptive noise cancelling applied to sinusoidal interferences," *IEEE Trans. on Acoustics Speech and Signal Proces.*, vol. 25, no. 6, pp. 484–491, December 1977. 2.4.2

[77] M. de Diego, A. Gonzalez, M. Ferrer, and G. Piñero, "Multichannel active noise control system for local spectral reshaping of multifrequency noise," *Journal of Sound and Vibration*, vol. 274, no. 1, pp. 249 – 271, 2004. 2.4.2, 7.1

[78] A. Gonzalez, M. de Diego, M. Ferrer, and G. Piñero, "Multichannel active noise equalization of interior noise," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 110–122, 2006. 2.4.2, 6.1.2, 6.1.3

[79] S. Kuo and D. Morgan, "Active noise control: a tutorial review," *Proceedings of the IEEE*, vol. 87, no. 6, pp. 943–975, jun 1999. 2.4.2

[80] L. E. Rees and S. J. Elliott, "Adaptive algorithms for active sound-profiling," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 2, pp. 711–719, March 2006. 2.4.2

[81] L. Wang and W. S. Gan, "Analysis of misequalization in a narrowband active noise equalizer system," vol. 311, no. 3, pp. 1438–1446. 2.4.2

[82] L. Wang and W.-S. Gan, "Convergence analysis of narrowband active noise equalizer system under imperfect secondary path estimation," *IEEE transactions on audio, speech, and language processing*, vol. 17, no. 4, pp. 566–571, 2009. 2.4.2

[83] J. Liu and X. Chen, "Adaptive compensation of misequalization in narrowband active noise equalizer systems," vol. 24, no. 12, pp. 2390–2399. 2.4.2

[84] J. A. Mosquera-SÃ¡nchez and L. P. R. de Oliveira, "A multi-harmonic amplitude and relative-phase controller for active sound quality control," vol. 45, no. 2, pp. 542–562. 2.4.2

[85] V. Patel, J. Cheer, and N. V. George, "Modified phase-scheduled-command FxLMS algorithm for active sound profiling," vol. 25, no. 9, pp. 1799–1808. 2.4.2

[86] S. M. Kuo, "Multiple-channel adaptive noise equalizers," in *Conference Record of The Twenty-Ninth Asilomar Conference on Signals, Systems and Computers*, vol. 2, Oct 1995, pp. 1250–1254 vol.2. 2.4.2

[87] W. F. Druyvesteyn and J. Garas, "Personal sound," *J. Audio Eng. Soc*, vol. 45, no. 9, pp. 685–701, 1997. 2.4.2, 7.1

[88] S. M. Kuo and J. Tsai, "Residual noise shaping technique for active noise control systems," *The Journal of the Acoustical Society of America*, vol. 95, no. 3, pp. 1665–1668, 1994. 2.4.2, 2.4.2, 6.2.2

[89] Y. Kajikawa, W.-S. Gan, and S. M. Kuo, "Recent advances on active noise control: open issues and innovative applications," *APSIPA Transactions on Signal and Information Processing*, vol. 1, 2012. 2.4.2

[90] E. Zwicker and H. Fastl, "Psychoacoustics: Facts and models. 1990." 2.4.2, 7.3

[91] A. Gonzalez, M. Ferrer, M. de Diego, G. Piñero, and J. Garcia-Bonito, "Sound quality of low-frequency and car engine noises after active noise control," *Journal of Sound and Vibration*, vol. 265, no. 3, pp. 663–679, aug 2003. 2.4.2, 7.1

[92] S. M. Kuo and Y. Yang, "Broadband adaptive noise equalizer," *IEEE Signal Processing Letters*, vol. 3, no. 8, pp. 234–235, Aug 1996. 2.4.2, 2.4.2, 6.2

[93] J. Feng and W.-S. Gan, "Broadband active noise compressor," *IEEE Signal Processing Letters*, vol. 5, no. 1, pp. 11–14, Jan 1998. 2.4.2

[94] S. M. Kuo, R. K. Yenduri, and A. Gupta, "Frequency-domain delayless active sound quality control algorithm," *Journal of Sound and Vibration*, vol. 318, no. 4, pp. 715–724, 2008. 2.4.2

[95] P. N. Samarasinghe, W. Zhang, and T. D. Abhayapala, "Recent advances in active noise control inside automobile cabins: Toward quieter cars," *IEEE Signal Processing Magazine*, vol. 33, no. 6, pp. 61–73, Nov 2016. 2.4.2

[96] Y. Xiao, L. Ma, K. Khorasani, and A. Ikuta, "A new narrowband active noise control system in the presence of sensor error," in *The 2004 47th Midwest Symposium on Circuits and Systems, 2004. MWSCAS '04.*, vol. 2, July 2004, pp. II–II. 2.4.2

[97] A. Berkhoff, "Optimum sensor-actuator distance for decentralized acoustic control," *Journal of the Acoustical Society of America*, vol. 110, pp. 260–266, 2001. 3.1.4, 4.1

[98] Y. Elgrichi and E. Zeheb, "Stability of multichannel sound control systems," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 144, no. 1, pp. 1–7, 1997. 3.1.4, 4.1

[99] G. Zhang, J. Tao, X. Qiu, and I. S. Burnett, "Decentralized two-channel active noise control for single frequency by shaping matrix eigenvalues," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2018. 3.1.4, 8.2

[100] F. An, Y. Cao, and B. Liu, "Optimized decentralized adaptive control of noise and vibration for periodic disturbances," *The Journal of the Acoustical Society of America*, vol. 144, pp. EL275–EL280, 2018. 3.1.4, 8.2

[101] S. Elliott, C. Boucher, and P. Nelson, "The behavior of a multiple channel active control system," *IEEE Transactions on Signal Processing*, vol. 40, no. 5, pp. 1041–1052, may 1992. 3.1.4, 3.1.4, 5.2.2

[102] B. Noble and J. W. Daniel, *Applied linear algebra*, E. C. Prentice-Hall, Ed., 1977. 3.1.4

[103] S. Elliott and K. Back, "Effort constraints in adaptive feedforward control," *IEEE Signal Processing Letters*, vol. 3, pp. 7–9, 1996. 3.2.1, 5.3

[104] X. Qiu and C. Hansen, "A study of time-domain FXLMS algorithms with control output constraint," *The Journal of the Acoustical Society of America*, vol. 109, no. 6, p. 2815, 2001. 3.2.1, 4.1, 4.4, 4.4, 5.3, 5.3.1, 5.3.2

[105] P. Welch, "The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, June 1967. 3.3

[106] Loudspeaker and microphone setup at the GTAC istening room, *http://www.gtac.upv.es/room.php*. 3.3.1

[107] P. Grosdidier and M. Morari, "Interaction measures for systems under decentralized control," *Automatica*, vol. 22, no. 3, pp. 309–319, 1986. 4.1

[108] W.-H. Cho, J.-G. Ih, and T. Toi, "Positioning actuators in efficient locations for rendering the desired sound field using inverse approach," *Journal of Sound and Vibration*, vol. 358, pp. 1–19, 2015. 4.1, A.3

[109] J. A. Mosquera-Sánchez, W. Desmet, and L. de Oliveira, "A multichannel amplitude and relative-phase controller for active sound quality control," *Mechanical Systems and Signal Processing*, vol. 88, pp. 145–165, may 2017. 4.1, 5.3

[110] J. Cordioli, C. Hansen, X. Li, and X. Qiu, "Numerical evaluation of a decentralised feedforward active control system for electrical transformer noise," *International J Acoust Vib*, vol. 7, no. 2, pp. 100–104, 2002, cited By 3. 4.1

[111] J. A. Mosquera-Sánchez, W. Desmet, and L. de Oliveira, "Multichannel feedforward control schemes with coupling compensation for active sound profiling," *Journal of Sound and Vibration*, vol. 396, pp. 1 – 29, 2017. 4.1

[112] F. Taringoo, J. Poshtan, and M. H. Kahaei, "Analysis of effort constraint algorithm in active noise control systems," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, pp. 1–10, 2006. 4.4, 5.3

[113] W. Kozacky and T. Ogunfunmi, "An active noise control algorithm with gain and power constraints on the adaptive filter," *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no. 1, p. 17, 2013. 4.4, 5.3, 5.3.1

[114] M. Z. A. Bhotto and A. Antoniou, "Affine-projection-like adaptive-filtering algorithms using gradient-based step size," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 61, no. 7, pp. 2048–2056, 2014. 5.1, 5.4

[115] G. Chen, T. Sone, and M. Abe, "Effects of multiple secondary paths on convergence properties in active noise control systems with lms algorithm," *Journal of Sound and Vibration*, vol. 195, pp. 217–228, 1996. 5.2.2

[116] B. Rafaely and S. Elliott, "A computationally efficient frequency-domain LMS algorithm with constraints on the adaptive filter," *IEEE Transactions on Signal Processing*, vol. 48, pp. 1649–1655, 2000. 5.3

[117] D. J. Rossetti, M. R. Jolly, and S. C. Southward, "Control effort weighting in feedforward adaptive control systems," *The Journal of the Acoustical Society of America*, vol. 99, no. 5, pp. 2955–2964, may 1996. 5.3

[118] M. Ferrer, M. De Diego, A. Gonzalez, and G. Pinero, "Efficient implementation of the affine projection algorithm for active noise control applications," in *Signal Processing Conference, 2004 12th European*. IEEE, 2004, pp. 929–932. 5.4

[119] M. Ferrer, A. Gonzalez, M. d. Diego, and G. Piñero, "Distributed affine projection algorithm over acoustically coupled sensor networks," vol. PP, no. 99, pp. 1–1. 5.4

[120] J. Lorente, M. Ferrer, M. de Diego, and A. Gonzalez, "GPU implementation of multichannel adaptive algorithms for local active noise control," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 1624–1635, 2014. 5.5

[121] J. P. Borrallo and M. G. Otero, "On the implementation of a partitioned block frequency domain adaptive filter (PBFDAF) for long acoustic echo cancellation," *Signal Processing*, vol. 27, pp. 301–315, 1992. 5.5

[122] J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Process. Mag.*, vol. 9, pp. 14–37, 1992. 5.5, 5.5.1

[123] D. Das, S. Kuo, and G. Panda, "New block filtered-x LMS algorithms for active noise control systems," *IET Signal Processing*, vol. 1, no. 2, pp. 73–81, jun 2007. 5.5.2

[124] F. S. Cattivelli and A. H. Sayed, "Diffusion lms strategies for distributed estimation," vol. 58, pp. 1035–1048, 2010. 5.5.3

[125] X. Zhao and A. Sayed, "Performance limits of lms-based adaptive networks," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 3768–3771. 5.5.3

[126] L. Kong, M. Xia, X. Liu, M. Wu, and X. Liu, "Data loss and reconstruction in sensor networks," in *2013 Proceedings IEEE INFOCOM*, April 2013, pp. 1654–1662. 5.6

[127] Y. Yu, F. Han, Y. Bao, and J. Ou, "A study on data loss compensation of wifi-based wireless sensor networks for structural health monitoring," *IEEE Sensors Journal*, vol. 16, no. 10, pp. 3811–3818, May 2016. 5.6

[128] B. Rafaely, S. Elliot, and J. Garcia-Bonito, "Broadband performance of an active headrest." *Journal of the Acoustical Society of America*, vol. 106(2), pp. 787–793, 1999. 5.7

[129] M. Pawelczyk, "Adaptive noise control algorithms for active headrest system," *Control Engineering Practice*, vol. 12, no. 9, pp. 1101–1112, sep 2004. 5.7

[130] J. Cheer and S. Elliott, "Design and implementation of a personal audio system in a car cabin." Acoustical Society of America (ASA), 2013. 5.7

[131] M. Misol, C. Bloch, H. Monner, and M. Sinapius, "Performance of active feedforward control systems in non-ideal, synthesized diffuse sound fields," *The Journal of the Acoustical Society of America*, vol. 135, no. 4, pp. 1887–1897, apr 2014. 5.7

[132] Y. Song, Y. Gong, and S. Kuo, "A robust hybrid feedback active noise cancellation headset," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 4, pp. 607–617, jul 2005. 5.7

[133] A. Roure and A. Albarrazin, "The remote microphone technique for active noise control," in *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, vol. 1999, 1999, pp. 1233–1244. 5.7

[134] W. Jung, S. J. Elliott, and J. Cheer, "Combining the remote microphone technique with head-tracking for local active sound control," *The Journal of the Acoustical Society of America*, vol. 142, no. 1, pp. 298–307, 2017. 5.7

[135] W. Jung, S. Elliott, and J. Cheer, "Estimation of the pressure at a listener's ears in an active headrest system using the remote microphone technique," *The Journal of the Acoustical Society of America*, vol. 143, no. 5, pp. 2858–2869, 2018. 5.7, 5.7.2, 5.7.2

[136] J. Buck, S. Jukkert, and D. Sachau, "Performance evaluation of an active headrest considering non-stationary broadband disturbances and head movement," *The Journal of the Acoustical Society of America*, vol. 143, no. 5, pp. 2571–2579, 2018. 5.7

[137] W. Jung, S. Elliott, and J. Cheer, "Local active control of road noise inside a vehicle," *Mechanical Systems and Signal Processing*, vol. 121, pp. 144–157, 2019. 5.7

[138] D. Moreau, B. Cazzolato, A. Zander, and C. Petersen, "A review of virtual sensing algorithms for active noise control," *Algorithms*, vol. 1, no. 2, pp. 69–99, 2008. 5.7

[139] S. Elliott and J. Cheer, "Modeling local active sound control with remote sensors in spatially random pressure fields," *The Journal of the Acoustical Society of America*, vol. 137, no. 4, pp. 1936–1946, apr 2015. 5.7.1

[140] E. A. P. Habets, "Room impulse response generator," Technische Universiteit Eindhoven, Tech. Rep. 1-17, 2006. 5.7.2

[141] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979. 5.7.2

[142] S. Kuo, A. Gupta, and S. Mallu, "Development of adaptive algorithm for active sound quality control," *Journal of Sound and Vibration*, vol. 299, no. 1-2, pp. 12–21, jan 2007. 6.2, 6.2.2

[143] S. Kuo and M. Ji, "Passband disturbance reduction in adaptive narrowband noise control systems," *IEEE Transactions on Circuits System I*, vol. 46, pp. 220–223, 1999. 6.2.2

[144] M. de Diego, A. Gonzalez, and C. Garcia, "On the performance of a local active noise control system," in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*. Institute of Electrical and Electronics Engineers (IEEE), 1999. 7.1

[145] A. Gonzalez, M. Ferrer, M. de Diego, and G. Pinero, "Subjective considerations in multchannel active noise control equalization of repetitive noise," in *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, vol. 2002, no. 7. Institute of Noise Control Engineering, 2002, pp. 387–400. 7.1

[146] "Apart model sdq5p," *http://www.apart-audio.com/*. 7.2

[147] "Behringer model ecm8000," *https://www.behringer.com*. 7.2

[148] "Jbl model lsr305," *www.jblpro.com/www/products/recording-broadcast/3-series/lsr305*. 7.2

[149] "Neumann ku 100 dummy head," *http://www.neumann.com/*. 7.2

[150] "Rme octamic ii," *https://www.rme-audio.de/en/products/*. 7.2

[151] "Motu 16 avb," *http://motu.com/products/avb/16a*. 7.2

[152] "Mathworks. audio system toolbox." *https://es.mathworks.com/products/audio-system.html*. 7.3

[153] "Steinberg, asio." *http://www.steinberg.net/nc/en/company/developers/sdk-download-portal.html*. 7.3

[154] "Node.js," *https://nodejs.org/en/*. 7.3

[155] "ISO 226: 2003(E). Acoustics –Normal equal-loudness-level contours," International Organization for Standardization, Geneva, Switzerland, Tech. Rep., 2003. 7.3

[156] R. Burdisso, "Causality analysis of feedforward-controlled systems with broadband inputs," *The Journal of the Acoustical Society of America*, vol. 94, pp. 234–242, 1993. 7.4

[157] S. J. Elliott, J. Cheer, H. Murfet, and K. R. Holland, "Minimally radiating sources for personal audio," *The Journal of the Acoustical Society of America*, vol. 128, no. 4, pp. 1721–1728, 2010. 7.5.2

[158] N.Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4795–4810, 2010. 8.2

[159] J. Estreder, G. Piñero, F. Aguirre, M. de Diego, and A. González, "On perceptual audio equalization for multiple users in presence of ambient noise," in *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, IEEE, Ed., 2018, pp. 445–449. 8.2

[160] "Raspberry pi," *https://www.raspberrypi.org/*. 8.2

[161] A. McPherson and V. Zappi, "An environment for submillisecond-latency audio and sensor processing on beaglebone black," in *Audio Engineering Society 138th Convention*, 2015. 8.2

[162] "Beagleboard-x15," *http://elinux.org/Beagleboard:BeagleBoard-X15*, 2016. 8.2

[163] I. T. Ardekani and W. H. Abdulla, "On the convergence of real-time active noise control systems," vol. 91, pp. 1262–1274, 2011. A.3

[164] D. G. Feingold and R. S. Varga, "Block diagonally dominant matrices and generalizations of the gerschgorin circle theorem." *Pacific J. Math.*, vol. 12, no. 4, pp. 1241–1250, 1962. A.3

[165] D. C. Kammer, "Sensor placement for on-orbit modal identification and correlation of large space structures," *Journal of Guidance, Control, and Dynamics*, vol. 14, pp. 251–259, 1992. A.3

[166] B.-K. Kim and J.-G. Ih, "On the reconstruction of the vibro-acoustic field over the surface enclosing an interior space using the boundary element method," *The Journal of the Acoustical Society of America*, vol. 100, no. 5, pp. 3003–3016, 1996. A.3

[167] "Matlab online documentation," *https://es.mathworks.com/help/*. B.2